

CSS fundamentals

Katarzyna Wójcik (UP)
Ján Skalka (UKF)
Lucia Benková (UKF)
Jan Turčínek (MENDELU)
Iwona Polak (US)
Zenón José Hernández-Figueroa (ULPGC)
José Daniel González-Domínguez (ULPGC)
Juan Carlos Rodríguez-del-Pino (ULPGC)
Jan Francisti (UKF)



CSS Fundamentals

Published on

November 2021

Authors

Katarzyna Wójcik | Pedagogical University of Cracow, Poland

Ján Skalka | Constantine the Philosopher University in Nitra, Slovakia

Lucia Benková | Constantine the Philosopher University in Nitra, Slovakia

Jan Turčínek | Mendel University in Brno, Czech Republic

Iwona Polak | Pedagogical University of Cracow, Poland

Zenón José Hernández-Figueroa | University of Las Palmas de Gran Canaria, Spain

José Daniel González-Domínguez | University of Las Palmas de Gran Canaria, Spain

Juan Carlos Rodríguez-del-Pino | University of Las Palmas de Gran Canaria, Spain

Jan Francisti | Constantine the Philosopher University in Nitra, Slovakia

Reviewers

Martin Drlík | Constantine the Philosopher University in Nitra, Slovakia

Cyril Klimeš | Mendel University in Brno, Czech Republic

Piet Kommers | Helix5, Netherland

Eugenia Smyrnova-Trybulská | University of Silesia in Katowice, Poland

Peter Švec | Teacher.sk, Slovakia

Graphics

Lubomír Benko | Constantine the Philosopher University in Nitra, Slovakia

David Sabol | Constantine the Philosopher University in Nitra, Slovakia

Erasmus+ FITPED

Work-Based Learning in Future IT Professionals Education

Project 2018-1-SK01-KA203-046382

Co-funded by the
Erasmus+ Programme
of the European Union



The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.



Licence (licence type: Attribution-Non-commercial-No Derivative Works) and may be used by third parties as long as licensing conditions are observed. Any materials published under the terms of a CC Licence are clearly identified as such.

All trademarks and brand names mentioned in this publication and all trademarks and brand names mentioned that may be the intellectual property of third parties are unconditionally subject to the provisions contained within the relevant law governing trademarks and other related signs. The mere mention of a trademark or brand name does not imply that such a trademark or brand name is not protected by the rights of third parties.

© 2021 Constantine the Philosopher University in Nitra

ISBN 978-80-558-1787-3

Table of Contents

1 Introduction to CSS.....	5
1.1 Introduction to CSS.....	6
2 External Style Sheet	8
2.1 External style sheet	9
3 Font and Text Formatting	11
3.1 Font and text formatting	12
4 Background Property.....	19
4.1 Background property	20
5 Internal Style Sheet, Inline Style.....	27
5.1 Internal style sheet, inline style.....	28
6 Priorities	32
6.1 Priorities	33
7 Class, Identifier	36
7.1 Class, identifier	37
8 DOM and Selectors.....	40
8.1 DOM and selectors	41
9 Pseudo-class.....	47
9.1 Pseudo-class.....	48
10 Pseudo-element.....	55
10.1 Pseudo-element.....	56
11 Box Model, Image and Float	60
11.1 Box model	61
11.2 Box model (programs)	70
11.3 Float.....	80
11.4 Float (programs).....	83
11.5 Image.....	88
11.6 Image (programs).....	93
11.7 Font-face rule	96
11.8 Font-face rule (program).....	99
12 List, Links and Navs.....	104
12.1 Links and navigation	105
12.2 Links (programs)	108
12.3 Lists	113

12.4 Lists (programs)	116
13 Property Display.....	120
13.1 Display - flex.....	121
13.2 Display - flex (programs).....	129
13.3 Display and visibility	142
13.4 Display and visibility (programs)	145
14 Property Position	150
14.1 Position	151
14.2 Position (programs)	154
15 Tables	166
15.1 Tables	167
15.2 Tables (programs).....	168
16 Media Queries	175
16.1 Media queries	176
16.2 Media queries (programs)	182
17 Validation, Units, Selectors Hierarchy.....	190
17.1 Units.....	191
17.2 Units (programs).....	197
17.3 Validators	202
17.4 Validators (programs)	204
17.5 Selector priorities	205
17.6 Selector priorities (programs)	209
17.7 Enjoy CSS (programs)	214

Introduction to CSS

Chapter **1**

1.1 Introduction to CSS

1.1.1

HTML is a markup language for preparing webpage structure and content. To describe the presentation layer of a webpage you need to use CSS. It helps in defining fonts, colours, margins, position and many more.

CSS standard is being developed by World Wide Web Consortium (W3C). You will find its official specification on W3C webpage: www.w3.org/Style/CSS/

1.1.2

CSS is for:

- preparing a webpage structure
- providing a webpage content
- describing a webpage look

1.1.3

CSS is an acronym of words: ***Cascading Style Sheets***. This name actually defines what these style sheets do. CSS formatting can be placed in:

- external file – it's called an external style sheet,
- the header of the webpage – it's called an internal style sheet,
- the element itself (body of webpage) – it's called inline style.

When there is a conflict of formatting between any of the above elements, the priority has style definition in the element itself (inline style), then in the header of the webpage (internal style sheet) and then in an external file (external style sheet). That is why the styles are called "***cascading***".

There are few versions of the CSS standard. The current course will discuss the CSS3 version.

1.1.4

Order the following parts of CSS styling in terms of hierarchy. Start with the one that has the highest priority.

1._____

2._____

3._____

- inline style
- external style sheet
- internal style sheet

External Style Sheet

Chapter 2

2.1 External style sheet

2.1.1

External style sheet is placed in **<head>** section of HTML document. The external style covers the entire content of the page and can also be used on subpages.

To add external style sheet use **<link>** tag with the following syntax: **<link href="URL" rel="stylesheet">**, where URL is a relative or absolute address of the file containing style sheet.

Look at this example:

```
<html>
  <head>
    <link href="style.css" rel="stylesheet" >
  </head>
  <body>
    <div>Lorem ipsum dolor sit</div>
  </body>
</html>
```

2.1.2

The imported style sheet is downloaded from a separate file located at the address indicated. Like the external style sheet, the imported one covers the entire content of the page on which it was used.

To import style sheet use the command **@import url("URL")** inside a **<style>** tag, where URL is a relative or absolute address of the file containing style sheet.

Look at this example:

```
<html>
  <head>
    <style >
      @import url("style.css");
    </style>
  </head>
  <body>
    <div>Lorem ipsum dolor sit</div>
  </body>
</html>
```

 **2.1.3**

An external style sheet is a regular text file. It is important so that this file has a **.css** extension, e.g. **style.css**.

This is an example content of a style sheet file:

```
/* this is a comment */
div {
    font-size: 12pt;
    background-color: blue;
}
```

Every single declaration of property and its value has to end with a semicolon ";".

 **2.1.4**

Which command is used for importing a style sheet?

- @import
- @stylesheet
- @style
- @css
- @link

 **2.1.5**

Fill the code so that there is a proper style sheet file embedded in the head of the HTML document.

```
<html>
    <head>
        <____ href="style.____" ____="stylesheet" >
    </head>
    <body>
        <div>Lorem ipsum dolor sit</div>
    </body>
</html>
```

Font and Text Formatting

Chapter 3

3.1 Font and text formatting

3.1.1

One of the most important features offered by cascading style sheets is the ability to format the appearance of the text and font used to compose the page.

CSS allows you to specify the following text attributes:

- *alignment,*
- *decoration,*
- *direction,*
- *indent,*
- *line height,*
- *spacing,*
- *transformation,*
- *white space.*

You can align text both ***horizontally*** and ***vertically***.

Horizontal text alignment is made by ***text-align*** property. The following values are allowed: ***left, right, center, justify*** and ***inherit***. The default value is: ***left***.

Vertical text alignment is made by the ***vertical-align*** property. The following values are allowed: ***baseline, sub, super, bottom, text-bottom, middle, top, text-top*** and ***inherit***. You can also enter a *numerical value* or a *percentage*; for both negative values are allowed. The default value is: ***baseline***.

Look at this example:

```
div {
    text-align: justify;
    vertical-align: sub;
}
```

The text in this **<div>** will be horizontally aligned both to left and right (***text-align: justify;***) and vertically aligned with the subscript baseline of the parent (***vertical-align: sub;***).

3.1.2

To make a simple text decoration use a ***text-decoration*** property. The following values are allowed: ***none, underline, overline, line-through, blink*** and ***inherit***. The default value is: ***none***.

To make a simple text transformation use a **text-transform** property. The following values are allowed:

- **none** – no changes,
- **capitalize** – the first letter of a word is a capital letter,
- **uppercase** – all letters are capital letters,
- **lowercase** – all letters are lowercase
- **inherit**.

The default value is: **none**.

If an element has **inherit** keyword for some property, the computed value of this property is taken from a parent element.

Without any styling the text will look as follows:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nibh augue, suscipit a, scelerisque sed, lacinia in, mi. Cras vel lorem. Etiam pellentesque aliquet tellus. Phasellus pharetra nulla ac diam.

When you embed the following style:

```
div {
    text-decoration: overline;
    text-transform: capitalize;
}
```

the text in this **div** will be overlined (**text-decoration: overline;**) and the first letter of every word will be a capital letter (**text-transform: capitalize;**):

*Lore*m* Ipsum Dolor Sit Am*e*t, Con*s*ectet*u*r*
*A*d*ipiscing E*l*it. P*ro*i*n* N*ib*h A*ug*ue, S*us*c*ip*it A*,**
*S*cl*er*is*que S*ed*, L*aci*n*ia* I*n*, M*i*. C*ra*s V*el* L*o*rem.*
*E*t*iam P*el*l*en*tes*que* A*li*quet T*ell*us. P*ha*re*tr*a N*ull*a A*c* D*iam*.*

3.1.3

To change the letter or word spacing use respectively a **letter-spacing** or **word-spacing** property. The following values are allowed: **normal** and **inherit**. You can also enter a *numerical value* (in **px**, **pt**, **cm**, **em**); negative values are allowed. The default value is: **normal**.

To change height between lines use a **line-height** property. The following values are allowed:

- **normal** – depends on the user agent,
- **number** – a multiple of the font height,
- **length** – specified value,
- **percentage** – percent of the height of the font,
- **inherit**.

The default value is: **normal**.

Without any styling the text will look as follows:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nibh augue, suscipit a, scelerisque sed, lacinia in, mi. Cras vel lorem. Etiam pellentesque aliquet tellus. Phasellus pharetra nulla ac diam.

When you embed the following style:

```
div {
    letter-spacing: -1px;
    word-spacing: 0.5em;
    line-height: 150%;
}
```

the text in this **div** will have narrower spacing between letters (**letter-spacing: -1px;**), wider spacing between words (**word-spacing: 0.5em;**) and bigger space between lines (**line-height: 150%;**):

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nibh augue, suscipit a, scelerisque sed, lacinia in, mi. Cras vel lorem. Etiam pellentesque aliquet tellus. Phasellus pharetra nulla ac diam.

3.1.4

Which property defines horizontal text alignment?

3.1.5

Which values are possible for **text-decoration** property?

- none
- underline

- line-through
- blink
- inherit
- capitalize
- lowercase
- super
- center

3.1.6

CSS allows you to specify the following font attributes:

- *family*,
- *size*,
- *style*,
- *variant*,
- *weight*.

The size of the font can be changed by the **font-size** property. The following values are allowed: **xx-small**, **x-small**, **small**, **medium**, **large**, **x-large**, **xx-large**, **smaller**, **larger** and **inherit**. You can also enter a *numerical value* or a *percentage*. The default value is: **medium**.

The weight of the font can be changed by the **font-weight** property. The following values are allowed: **normal**, **bold**, **100**, **200**, **300**, **400**, **500**, **600**, **700**, **800**, **900**, **lighter** and **bolder**. You can also enter a *numerical value* or a *percentage*. The default value is: **normal**.

Without any styling the text will look as follows:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nibh augue, suscipit a, scelerisque sed, lacinia in, mi. Cras vel lorem. Etiam pellentesque aliquet tellus. Phasellus pharetra nulla ac diam.

When you embed the following style:

```
div {
    font-size: 16pt;
    font-weight: 600;
}
```

the font in this <div> will have the size of 16 points (**font-size: 16pt;**) and will be bold (**font-weight: 600;**):

**Lorem ipsum dolor sit amet,
 consectetur adipiscing elit. Proin
 nibh augue, suscipit a, scelerisque
 sed, lacinia in, mi. Cras vel lorem.
 Etiam pellentesque aliquet tellus.
 Phasellus pharetra nulla ac diam.**

3.1.7

To change font family use a **font-family** property. The following values are allowed:

- **font name** – the proper name of the font, e.g. **Arial**; font family names that contain space should be surrounded by quote marks, e.g. "**Times New Roman**",
- **serif** – any serif font,
- **sans-serif** – any sans-serif font,
- **monospace** – any font with a fixed character width,
- **fantasy** – any decorative font,
- **cursive** – any font with partially or completely connected characters; it makes an impression of handwriting,
- **inherit**.

The default value is inherited from the user agent.

The style of the font can be changed by the **font-style** property. The following values are allowed: **normal**, **italic**, **oblique** and **inherit**. The default value is: **normal**.

A variant of the font can be changed by the **font-variant** property. Values allowed are e.g.: **normal**, **small-caps** and **inherit**. The default value is: **normal**.

Without any styling the text will look as follows:

**Lorem ipsum dolor sit amet, consectetur adipiscing
 elit. Proin nibh augue, suscipit a, scelerisque sed,
 lacinia in, mi. Cras vel lorem. Etiam pellentesque
 aliquet tellus. Phasellus pharetra nulla ac diam.**

When you embed the following style:

```
div {
    font-family: sans-serif;
    font-style: italic;
    font-variant: small-caps;
}
```

the font in this **div** will be sans-serif one (**font-family: sans-serif;**) and written in italic (**font-style: italic;**). The text will be written in small capitals (**font-variant: small-caps;**), that is the letters will have the form of uppercase, but the size of lowercase letters:

*LOREM IPSUM DOLOR SIT AMET, CONSETETUR
ADIPISCING ELIT. PROIN NIBH AUGUE, SUSCIPIT
A, SCELERISQUE SED, LACINIA IN, MI. CRAS VEL
LOREM. ETIAM PELLENTESQUE ALIQUET
TELLUS. PHASELLUS PHARETRA NULLA AC
DIAM.*

3.1.8

You can collectively change several properties of a font in one property declaration. To do this use **font** property with a list of values:

font: [font-size] [font-family] [font-style] [font-variant] [font-weight] [font-stretch] [/line-height]

The first two, i.e. **[font-size]** and **[font-family]** are obligatory, whereas all other are optional.

Also, the order of values is important:

- **[font-size]** must be after **[font-style]**, **[font-variant]** and **[font-weight]**,
- **[line-height]** can occur only with **[font-size]** and the syntax is: **[font-size]/[line-height]**,
- **[font-family]** must be the last value.

When you embed the following style:

```
div {
    font: bold 14px/2 "Courier New";
}
```

the font in this **div** will be bold and changed to **Courier New**. Also, the font will have a size of 14 px and the height of the line will be doubled. See the example below:

```
    Lorem ipsum dolor sit amet, consectetur  
    adipiscing elit. Proin nibh augue,  
    suscipit a, scelerisque sed, lacinia in,  
    mi. Cras vel lorem. Etiam pellentesque  
    aliquet tellus. Phasellus pharetra nulla  
    ac diam.
```

 3.1.9

It is possible to change font style, weight, size and family in one property declaration.

- True
- False

 3.1.10

What is the correct order of individual fragments in **font** property:

1 _____

2 _____

3 _____

4 _____

- [font-variant]
- [line-height]
- [font-family]
- [font-size]

Background Property

Chapter **4**

4.1 Background property

4.1.1

With CSS, you can specify the appearance of the background and colour for selected elements. Depending on your needs, you can define a background that is a colour value or an image.

To change the colour of the element's text and text decorations, use **color** property. It can have the following values:

- *colour name*, e.g. **red**, **orange**, **blue**, **green**,
- *colour value* in hexadecimal preceded with the number sign (hash), where first two digits specify the value of the red component, next two digits – green and the last two – blue, e.g. **#ff0000** (for red), **#a31507**, **#1123fb**; you can add two more digits to specify the alpha channel for transparency, e.g. **#51a63960**,
- *colour value* in integer (0-255) or in percentage (0-100%), according to the format *rgb(RED, GREEN, BLUE)*, e.g. **rgb(5, 250, 13)**; you can also add an alpha channel, which can be a number between 0 and 1 or a percentage: *rgb(RED, GREEN, BLUE, ALPHA)*, e.g. **rgb(50%, 2%, 90%, 0.3)**,
- **inherit** – the computed value of this property is taken from a parent element.

The default value depends on the settings of the user agent.

The text below:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nibh augue, suscipit a, scelerisque sed, lacinia in, mi. Cras vel lorem. Etiam pellentesque aliquet tellus. Phasellus pharetra nulla ac diam.

with the following style:

```
div {
    color: maroon;
}
```

will look like this:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nibh augue, suscipit a, scelerisque sed, lacinia in, mi. Cras vel lorem. Etiam pellentesque aliquet tellus. Phasellus pharetra nulla ac diam.

 4.1.2

Which declarations are correct for specifying a colour?

R – red, G – green, B – blue, A – alpha

- color: #RRGGBB;
- color: rgba(R, G, B, A);
- color: inherit;
- colour: #RRGGBBAA;
- colour: rgb(R, G, B);
- colour: inherit;
- text: rgb(R, G, B);
- text: blue;
- color: #BBGGRR;

 4.1.3

To change the colour of the background, use **background-color** property. It can have the same values that were listed for the **color** property and also: **transparent**. The **transparent** value is the default value for the **background-color** property.

The text below:

Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Proin nibh augue, suscipit a, scelerisque sed,
lacinia in, mi. Cras vel lorem. Etiam pellentesque
aliquet tellus. Phasellus pharetra nulla ac diam.

with the following style:

```
div {
    background-color: #fafad2;
}
```

will look like this:

Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Proin nibh augue, suscipit a, scelerisque sed,
lacinia in, mi. Cras vel lorem. Etiam pellentesque
aliquet tellus. Phasellus pharetra nulla ac diam.

 4.1.4

What is the default value for the *background-color* property?

 4.1.5

It is also possible to set an image as a background. To do instead of **background-color** you need to use **background-image** property. This property can have the following values:

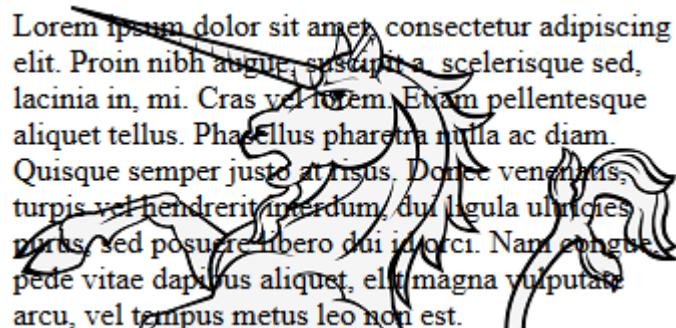
- **url("URL")** – an absolute or relative URL of an image,
- **none**,
- **inherit**.

The default value is: **none**.

Compare CSS code:

```
div {
    background-image: url("unicorn.png");
}
```

with this screenshot:



~~Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nibh augue, suscipit a scelerisque sed, lacinia in, mi. Cras vel lorem. Etiam pellentesque aliquet tellus. Phasellus pharetra nulla ac diam. Quisque semper justo at risus. Donec venenatis, turpis vel hendrerit interdum, dui ligula ultricies purus, sed posuere libero dui in orci. Nam congue pede vitae dapibus aliquet, elit magna vulputate arcu, vel tempus metus leo non est.~~

Since CSS3 the **background-image** property supports multiple background images. Look at this example:

```
div {
    background-image: url("rainbow.svg"), url("unicorn.png");
}
```

and a corresponding screenshot:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nibh angue, suscipit ac scelerisque sed, lacinia in, mi. Cras vel lorem. Etiam pellentesque aliquet tellus. Phasellus pharetra nulla ac diam. Quisque semper justo at risus. Donec venenatis, turpis vel hendrerit interdum, dui ligula ultricies purus, sed posuere libero dui id orci. Nam congue pede vitae dapibus aliquet, elit magna vulputate arcu, vel tempus metus leo non est.

4.1.6

It is allowed to put only one image in a *background-image* property.

- False
- True

4.1.7

A graphic background provides many options. You can specify, how it should be repeated or positioned and how it should behave when a user scrolls the webpage. For those purposes use properties: **background-repeat**, **background-position** and **background-attachment**.

A **background-attachment** property can have the following values:

- **fixed** – the background image is immobilized while scrolling,
- **scroll** – the background image scrolls with the text,
- **inherit**.

The default value is: **scroll**.

A **background-position** property can have the following values:

- **left, right, top, bottom** – set the background image at the left/right/top/bottom edge,
- **center** – sets the background image in the centre,
- *length* – an absolute value that determines the position of the background,
- *percentage* – a relative value that determines the position of the background,
- **inherit**.

The default value is: **0% 0%**.

A **background-repeat** property can have the following values:

- **repeat-x** – horizontal background duplication,
- **repeat-y** – vertical background duplication,
- **repeat** – horizontal and vertical background duplication,
- **no-repeat** – no background duplication,
- **space** – the background image is duplicated without clipping,
- **round** – the repeated images are stretched until there is no room,
- **inherit**.

The default value is: **repeat**.

Here is a CSS code with some of the above properties:

```
div {
    background-image: url("rainbow.svg");
    background-position: bottom;
    background-repeat: repeat-x;
}
```

Here is the corresponding screenshot:

Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Proin nibh augue, suscipit a, scelerisque sed,
lacinia in, mi. Cras vel lorem. Etiam pellentesque
aliquet tellus. Phasellus pharetra nulla ac diam.
Quisque semper justo at risus. Donec venenatis,
turpis vel hendrerit interdum, dui ligula ultricies
purus, sed posuere libero dui id orci. Nam congue,
pede vitae dapibus aliquet, elit magna vulputate
arcu, vel tempus metus leo non est.



Background-images are set to bottom edge (**background-position: bottom;**) and they are duplicated only horizontally (**background-repeat: repeat-x;**).

4.1.8

Complete the following CSS code so that:

- text is **green**,
- the background is graphical and uploaded from image **cat.png**, which is not duplicated,
- the background image is in the centre and it is immobilized while scrolling.

```
div {
    background-image: ____;
    background-attachment: ____;
    ____: center;
```

```
background-repeat: ____;
____: green;
}
```

4.1.9

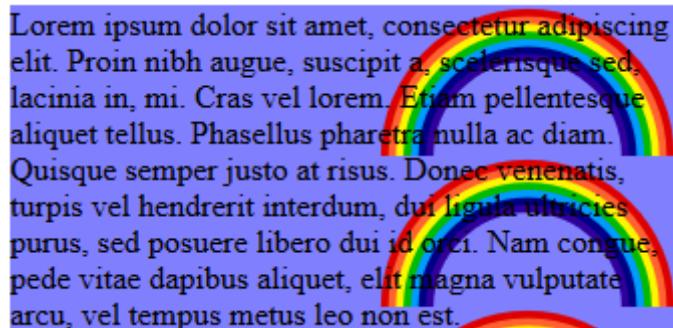
You can collectively change several properties of a background in one property declaration. To do this use **background** property with a list of values:

background: [background-color] [background-image] [background-position]
[background-repeat] [background-attachment]

The default value of this property depends on the default value of each sub-value.

Compare the CSS code and the corresponding screenshot below:

```
div {
    background: rgb(128, 128, 255) url("rainbow.svg") right
top repeat-y scroll;
}
```



The background is light blue (`rgb(128, 128, 255)`). There is also a background image (`url("rainbow.svg")`) which is positioned to the right top corner (`right top`) and is duplicated only vertically (`repeat-y`). The image will scroll with the text (`scroll`).

4.1.10

Connect *background-repeat* property values with their descriptions:

space _____

no-repeat _____

repeat _____

round ____

repeat-y ____

inherit ____

repeat-x ____

- no background duplication
- horizontal background duplication
- horizontal and vertical background duplication
- the computed value is taken from a parent element
- the background image is duplicated without clipping
- vertical background duplication
- the repeated images are stretched until there is no room

Internal Style Sheet, Inline Style

Chapter 5

5.1 Internal style sheet, inline style

5.1.1

Apart from using an external style sheet in a separate file, the styles can be embedded in the same file as HTML code. This can be done in two ways:

- as an internal style sheet,
- as an inline style.

External style sheets, internal style sheets and inline styles can all appear on the same page.

5.1.2

The internal style sheet is placed in the page header. By doing this, it can cover the entire document. The same CSS code which would appear in an external file is placed in the page header inside a `<style>` tag. Locate `<style>` tag and a CSS inside it in the following code:

```
<html>
  <head>
    <style >
      div {
        font-size: 12pt;
        background-color: blue;
      }
    </style>
  </head>
  <body>
    <div>Lorem ipsum dolor sit amet</div>
  </body>
</html>
```

Font and background styling will affect all **divs** that are present on this page.

5.1.3

Internal style sheet is placed inside a tag:

- `<style>`
- `<stylesheet>`

- <stylesheet type="text/css">
- <css type="text">

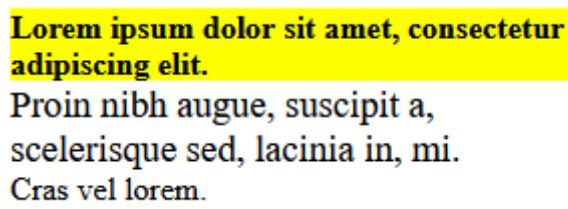
5.1.4

The inline style is placed inside the tag of the formatted element. The style works only in the element in which it is located. Inline style is placed as a value of tag's **style=""** attribute, e.g. <div style="color: blue;">.

Look at this example:

```
<html>
  <head>
    <title>Lorem ipsum</title>
  </head>
  <body>
    <div style="background-color: yellow; font-weight: bold;">Lorem ipsum dolor sit amet, consectetur adipiscing elit.</div>
    <div style="font-size: 14pt;">Proin nibh augue, suscipit a, scelerisque sed, lacinia in, mi.</div>
    <div>Cras vel lorem.</div>
  </body>
</html>
```

Here is a corresponding screenshot:



**Lorem ipsum dolor sit amet, consectetur
adipiscing elit.**

Proin nibh augue, suscipit a,
scelerisque sed, lacinia in, mi.
Cras vel lorem.

In the first **div** font is bold and the background is yellow (**style="background-color: yellow; font-weight: bold;"**). In the second **div** the font size is larger: 14 pt instead of the default 12pt (**style="font-size: 14pt;"**). The last has no additional styling.

5.1.5

How to apply an inline style for an element?

- as a value of element's **style=""** attribute

- as an element's style value
- as a <style> tag inside the element

5.1.6

Complete the following code so that the **div** has an inline style that changes font style to **oblique**.

```
<html>
    <head>
        <title>Lorem ipsum</title>
    </head>
    <body>
        <div _____="_____>Lorem ipsum dolor sit amet</div>
    </body>
</html>
```

5.1.7

Order the following parts of CSS styling in terms of hierarchy. Start with the one that has the lowest priority.

1 _____

2 _____

3 _____

- inline style
- external style sheet
- internal style sheet

5.1.8

Complete the following code so that there is a proper declaration of an internal style sheet and it affects all **div**'s on this page.

```
<html>
    <head>
        <_____ >
            _____ {
                font-family: serif;
```

```
        text-transform: uppercase;
    }
</style>
</head>
<body>
    <div>Lorem ipsum dolor sit amet</div>
</body>
</html>
```

Priorities

Chapter 6

6.1 Priorities

6.1.1

CSS is an acronym of words:

6.1.2

As you saw CSS formatting can be placed in:

- external style sheet,
- internal style sheet,
- inline style.

When there is a conflict of formatting between any of the above elements, the priority has style definition in the element itself (inline style), then in the header of the webpage (internal style sheet) and then in an external file (external style sheet). That is why the styles are called "cascading".

This is the content of **style.css** file:

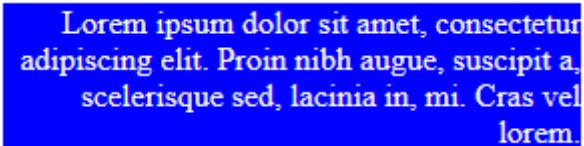
```
div {
    background-color: green;
    color: white;
    text-align: center;
}
```

This is HTML code with an external style sheet, internal style sheet and inline style:

```
<html>
    <head>
        <link href="style.css" rel="stylesheet"
type="text/css">
        <style >
            div {
                background-color: yellow;
                text-align: right;
            }
        </style>
    </head>
    <body>
        <div style="background-color: blue;">Lorem ipsum dolor
sit amet, consectetur adipiscing elit. Proin nibh augue,
```

```
suscipit a, scelerisque sed, lacinia in, mi. Cras vel
lorem.</div>
</body>
</html>
```

This is a corresponding screenshot:



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin nibh augue, suscipit a, scelerisque sed, lacinia in, mi. Cras vel lorem.

The font colour is calculated from an external style sheet (**color: white;**), because there is no **color** property in other places. The alignment of the text is taken from the internal style sheet (**text-align: right;**), even though there is also the same property in the external style sheet (**text-align: center;**). The background colour values from external (**background-color: green;**) and internal (**background-color: yellow;**) style sheets were overwritten by inline style (**background-color: blue;**).

Very good and detailed information about which CSS rule is applied you can find on MDN web docs from Mozilla:

https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS/Cascade_and_inheritance

6.1.3

Which CSS code can be easily reused on another page?

- external style sheet
- internal style sheet
- inline style

6.1.4

Which CSS code can cover the entire page?

- external style sheet
- internal style sheet
- inline style

 6.1.5

Complete the following code so that:

- in both **divs**, there is a sans-serif font,
- in both **divs** text is transformed to lowercase,
- only in the first **div**'s line-height is set to 150%.

This is the **style.css** file:

```
div      {  
    _____  
    line-height: 75%;  
    text-transform: uppercase;  
}
```

This is the **page.html** file:

```
<html>  
    <head>  
        <link href="_____" rel="stylesheet" type="text/css">  
        <style type="text/css">  
            div      {  
                line-height: 125%;  
                _____  
            }  
        </style>  
    </head>  
    <body>  
        <div style="_____">Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.</div>  
        <div>Proin nibh augue, suscipit a, scelerisque sed,  
lacinia in, mi.</div>  
    </body>  
</html>
```

Class, Identifier

Chapter 7

7.1 Class, identifier

7.1.1

The class allows to vary the formatting of selected elements on the page depending on the class attribute placed in the tag. The class can be used repeatedly. Class's name in style sheet begins with a period. The class name cannot include space.

Here are examples of two classes:

```
.someClass {
    font-weight: bold;
}
.anotherClass {
    color: blue;
}
```

A tag to use a given class must have a **class=""** attribute defined with the class name (without a period). A tag can use more than one class. In such a case, classes' names have to be space-separated.

Here are examples of tags using defined above classes:

```
<div class="someClass anotherClass">Lorem ipsum dolor sit
amet, consectetur adipiscing elit.</div>
<div class="anotherClass">Proin nibh augue, suscipit a,
scelerisque sed, lacinia in, mi.</div>
<div class="someClass">Cras vel lorem.</div>
```

The result is as follows:

**Lorem ipsum dolor sit amet, consectetur
adipiscing elit.
 Proin nibh augue, suscipit a, scelerisque sed,
lacinia in, mi.
 Cras vel lorem.**

The first **div** is both blue and bold because it uses both classes. The second **div** is only blue and the last one – only bold, as both of them uses only one class.

7.1.2

The identifier allows you to assign formatting to the selected tag having a unique id. This enables differentiation of the ways of presenting elements on the page. The identifier's name in the style sheet begins with a number sign (hash). An id has to be unique on the whole page.

Here is an example of an identifier:

```
#fancy {
    text-decoration: overline;
}
```

A tag to use a given identifier must have an **id=""** attribute defined with the id name (without a number sign).

Here is an example of a tag using defined above identifier:

```
<div id="fancy">Lorem ipsum dolor sit amet, consectetur
adipiscing elit.</div>
```

The result is as follows:

 Lorem ipsum dolor sit amet, consectetur
 adipiscing elit.

A tag can use both identifier and class at the same time:

```
<div id="fancy" class="anotherClass">Lorem ipsum dolor sit
amet, consectetur adipiscing elit.</div>
```

 Lorem ipsum dolor sit amet, consectetur
 adipiscing elit.

7.1.3

Which symbol corresponds with which selector type?

: _____

- pseudo-class
- identifier
- class

 7.1.4

If a tag uses an identifier (*id*), the use of a class (*class*) is prohibited.

- False
- True

 7.1.5

Selectors can be freely grouped if certain values are to be valid for each of them. The individual selectors included in the group are separated by commas. This allows you to simplify the structure of the sheet.

If this CSS is used:

```
h1, h2, h3 {  
    color: brown;  
    text-decoration: underline;  
}
```

headers of all three levels (**h1**, **h2**, **h3**) will be brown and underlined (**color: brown**; **text-decoration: underline**).

DOM and Selectors

Chapter 8

8.1 DOM and selectors

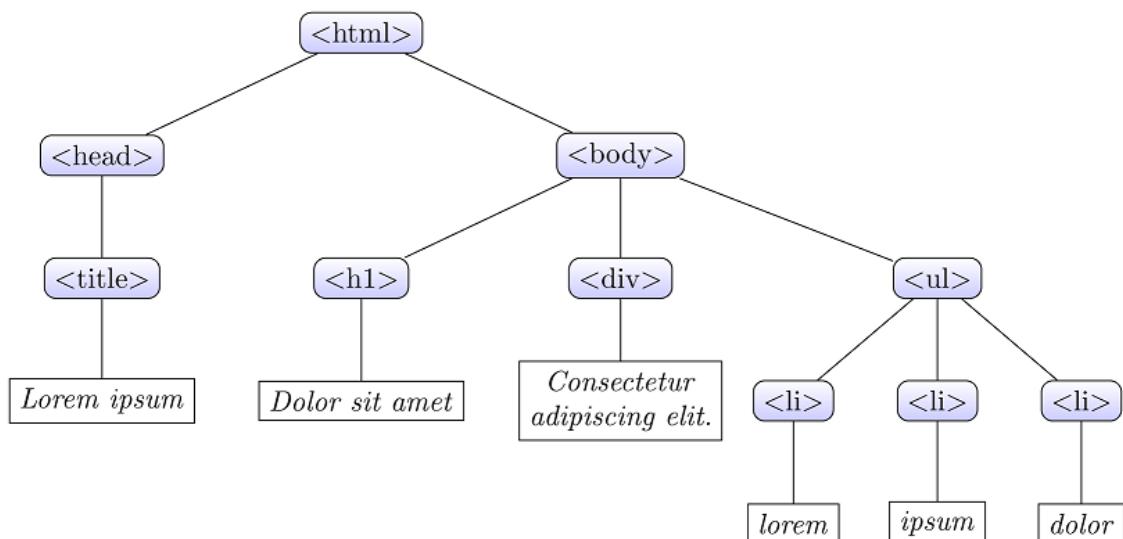
8.1.1

Tags of an HTML page are represented in memory as Document Object Model (DOM). The document is represented with a logical tree. This makes it easier to connect HTML pages to scripts like JavaScript. DOM methods allow changing the document's style, content and even structure.

Look at the sample HTML code:

```
<html>
  <head>
    <title>Lorem ipsum </title>
  </head>
  <body>
    <h1>Dolor sit amet</h1>
    <div>Consectetur adipiscing elit.</div>
    <ul>
      <li>lorem</li>
      <li>ipsum</li>
      <li>dolor</li>
    </ul>
  </body>
</html>
```

The code is represented as the following DOM tree:



 8.1.2

What does DOM stand for?

 8.1.3

The universal selector allows you to set the formatting for all elements of the page. The universal selector is represented by an asterisk.

This is an example of the universal selector:

```
*      {
    color: olive;
}
```

After applying the above CSS to the page with the following content inside the **<body>** tag:

```
<h1>Lorem ipsum</h1>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing
elit.</div>
<ul>
    <li>lorem</li>
    <li>ipsum</li>
    <li>dolor</li>
</ul>
```

all elements will have olive font colour:

Lorem ipsum

Lorem ipsum dolor sit amet, consectetur
adipiscing elit.

- lorem
- ipsum
- dolor

 8.1.4

The descendant combinator allows you to set the formatting for an element occurring inside another tag. It is typically represented by a single space:

```
div span {
    color: teal;
}
```

In the example formatting is applied only to the first and second , because they are descendants of a div:

```
<div>
    <h1><span>Lorem ipsum</span></h1>
    <span>dolor sit amet</span>
</div>
<span>consectetur adipiscing elit</span>
```

Lorem ipsum

dolor sit amet
consectetur adipiscing elit

The child combinator also allows the formatting of an element occurring inside another element. But the second element has to be an immediate descendant of the outside tag. It is represented by a greater-than sign. The following CSS code:

```
div > span {
    color: teal;
}
```

applied to the same HTML code as previously, will produce results as follows:

Lorem ipsum

dolor sit amet
consectetur adipiscing elit

Spot the difference in header formatting. A tag inside a <h1> tag is not an immediate descendant of a <div> tag, that's why its font colour doesn't change.

8.1.5

The sibling combinator allows the formatting of an element directly adjacent to another element. Both tags must have a common parent. It is represented by a plus sign. The following CSS code:

```
div + span {
    color: teal;
}
```

applied to the same HTML code as previously

```
<div>
    <h1><span>Lorem ipsum</span></h1>
    <span>dolor sit amet</span>
</div>
<span>consectetur adipiscing elit</span>
```

will produce results as follows:

Lorem ipsum

dolor sit amet
consectetur adipiscing elit

Now only the last `` is formatted, because it is directly adjacent to `<div>` element.

8.1.6

The attribute selector refers to the selected tag having a specific attribute. Its syntax consists of square brackets which contain an attribute name:

```
p[lang] {
    color: red;
}
```

For the below HTML code:

```
<p>Lorem ipsum</p>
<p lang="pl">dolor sit amet</p>
<p>consectetur adipiscing elit</p>
```

it will produce the following result:

Lorem ipsum
dolor sit amet
consectetur adipiscing elit

The second paragraph is coloured red because it has the **lang** attribute (**< p lang="pl">**).

It is also possible to:

- select all elements with the attribute with a specific value (**[attribute=value]** and **[attribute~=value]**),
- select all elements with the attribute starting with a specific value (**[attribute^=value]**),
- select all elements with the attribute ending with a specific value (**[attribute\$=value]**),
- select all elements with the attribute including a specific value (**[attribute*=value]**).

8.1.7

Selectors can be freely grouped if certain values are to be valid for each of them. Thanks to the grouped selectors, the layout of the sheet can be simplified. The individual selectors included in the group are separated by commas:

```
h1, h2, h3 {
    color: fuchsia;
}
```

Corresponding HTML code:

```
<h1>Lorem ipsum</h1>
<div>Lorem ipsum dolor sit amet, consectetur adipiscing
elit.</div>
<h2>Dolor sit amet</h2>
<ul>
    <li>lorem</li>
    <li>ipsum</li>
    <li>dolor</li>
</ul>
```

And the resulting screenshot:

Lore ipsum

 Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Dolor sit amet

- lorem
- ipsum
- dolor

Only headers (**h1**, **h2** in this example) are coloured in fuchsia.

8.1.8

Select an example that will specify the formatting of a `` tag only if it is an immediate descendant of the `<div>` tag:

- div img
- div > img
- div + img
- div, img

8.1.9

Select an example that will specify the formatting of a `<p>` tag only if it is directly adjacent of the `<h1>` tag:

- h1 p
- h1 > p
- h1 + p
- h1, p

8.1.10

It is not possible to assign a specific feature to more than one selector at a time.

- False
- True

Pseudo-class

Chapter **9**

9.1 Pseudo-class

9.1.1

The CSS specification provides several strictly defined classes that allow the formatting of some elements of the document.

The general syntax is as follows:

```
selector:pseudo-class {
    property: value;
}
```

The most popular pseudo-classes are pseudo-classes of links:

- **:link** applies to the formatting of the link intact,
- **:visited** applies to the formatting of a link that has already been visited; it allows to distinguish it from other links on the page,
- **:hover** applies to the formatting of the element over which the mouse cursor is currently positioned,
- **:active** applies to the formatting of the active element on the page.

To work properly above mentioned pseudo-classes need to be put in the LVHA-order: **:link**, **:visited**, **:hover**, **:active**.

Look at the following CSS code:

```
a:link      {
    color: green;
}
a:visited    {
    color: gray;
}
a:hover     {
    font-weight: bold;
}
a:active    {
    font-style: italic;
}
```

This code will produce the following behaviour:

I am a normal hyperlink

I am a visited hyperlink

I am a hovered hyperlink



I am a normal text

The first link is intact so it is green (**a:link { color: green; }**). The second link was visited so its colour changed to grey (**a:visited { color: gray; }**). The third link is pointed with the mouse cursor that's why it is bold (**a:hover { font-weight: bold; }**).

📝 9.1.2

Put links' pseudo-classes in the recommended order:

- 1 _____
- 2 _____
- 3 _____
- 4 _____

- :hover
- :link
- :active
- :visited

📖 9.1.3

The **:focus** pseudo-class allows to highlight the element that has received focus, i.e. when the user clicks on an element or selects it with the “tab” key.

Here is the exemplary CSS code:

```
textarea:focus {
    background-color: silver;
}
```

In the below screenshot there are two **<textarea>** elements:

```
    Lorem ipsum dolor sit amet,
    consectetur adipiscing elit.|
```

```
    Proin nibh augue, suscipit a,
    scelerisque sed, lacinia in, mi.
```

The first element received focus and it has a light grey background.

9.1.4

The **:first-child** pseudo-class is responsible for formatting the first child of the parent element. The formatting of the other children is not changed.

The **:last-child** pseudo-class is responsible for formatting the last child of the parent element. The formatting of the other children is not changed.

CSS:

```
p:first-child {
    background-color: black;
    color: white;
}
p:last-child {
    font-style: italic;
    text-decoration: underline;
}
```

HTML:

```
<div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit.</p>
<p>Proin nibh augue, suscipit a, scelerisque sed, lacinia in,
mi.</p>
<p>Cras vel lorem.</p>
<p>Etiam pellentesque aliquet tellus.</p>
</div>
```

The result is as on the screenshot:

**Lorem ipsum dolor sit amet, consectetur
 adipiscing elit.**

**Proin nibh augue, suscipit a, scelerisque sed,
 lacinia in, mi.**

Cras vel lorem.

Etiam pellentesque aliquet tellus.

The first paragraph has a different background and font colour. The last paragraph has a different style and decoration. The second and third paragraphs' formatting is intact.

9.1.5

The **:nth-child()** pseudo-class is responsible for formatting elements based on their position in a group of children. The formatting of the other children is not changed.

This pseudo-class has a single argument from one of the following options:

- **odd** – elements whose numeric position is odd (i.e. 1, 3, 5, ...),
- **even** – elements whose numeric position is even (i.e. 2, 4, 6, ...),
- functional notation: **An+B**, where **A** and **B** are integers – elements whose numeric position is calculated for zero and positive integer values of **n**.

The index of the first element is 1.

CSS:

```
p:nth-child(3n+1) {
    background-color: navy;
    color: white;
}
```

HTML:

```
<div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit.</p>
<p>Proin nibh augue, suscipit a, scelerisque sed, lacinia in,
mi.</p>
```

```
<p>Cras vel lorem.</p>
<p>Etiam pellentesque aliquet tellus.</p>
<p>Phasellus pharetra nulla ac diam.</p>
<p>Quisque semper justo at risus.</p>
<p>Donec venenatis, turpis vel hendrerit interdum, dui ligula
ultricies purus, sed posuere libero dui id orci.</p>
</div>
```

Screenshot:

Lore ipsum dolor sit amet, consectetur
adipiscing elit.

Proin nibh augue, suscipit a, scelerisque sed,
lacinia in, mi.

Cras vel lorem.

Etiam pellentesque aliquet tellus.

Phasellus pharetra nulla ac diam.

Quisque semper justo at risus.

Donec venenatis, turpis vel hendrerit
interdum, dui ligula ultricies purus, sed
posuere libero dui id orci.

Only the first ($3*0+1$), fourth ($3*1+1$) and seventh ($3*2+1$) paragraphs' formatting were changed.

9.1.6

Match the following pseudo-classes with a functional notation that will give the same result:

:nth-child(even) _____
 :nth-child(odd) _____
 :first-child _____

- :nth-child(1n+2)
- :nth-child(2n+1)
- :nth-child(2n)
- :nth-child(2)
- :nth-child(1)
- :nth-child(1n)

- :nth-child(2n+2)

9.1.7

The **:checked** pseudo-class is responsible for formatting form elements that are checked or selected. It can be associated with **<checkbox>**, **<radio>** and **<option>** elements. To format also the label use **+ label** pseudo-class.

See the CSS example below:

```
input:checked + label {
    color: purple;
    text-decoration: underline;
}
```

Here are five elements **<input type="checkbox">**:

- lorem
- ipsum
- dolor
- sit
- amet

The first and third one is checked and their formatting is changed: the font colour is purple and the text is underlined.

There are many more interesting pseudo-classes. A full list of them is available on:

<https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes>

9.1.8

Complete CSS in the following code so that:

- an element that received focus has a green font,
- an element that is selected has a blue font.

```
<html>
    <head>
        <_____ type="text/css">
```

```
input:_____ {
    color: green;
}
input:_____ + _____ {
    color: blue;
}
</style>
</head>
<body>
<form>
    <input value="lorem"><br>
    <input value="ipsum"><br>

    <input type="radio" name="dolor" id="sit">
    <label for="sit">sit</label>
    <input type="radio" name="dolor" id="amet">
    <label for="amet">amet</label>
</form>
</body>
</html>
```

Pseudo-element

Chapter 10

10.1 Pseudo-element

10.1.1

Another group of selectors are CSS pseudo-elements, which allow assigning subsequent styles to elements of an HTML page.

The **::first-letter** pseudo-element allows formatting the first letter in the text block.

The **::first-line** pseudo-element, unlike the predecessor, includes formatting of the entire first line, not just the first letter.

Look at these CSS styles:

```
div::first-letter {
    color: green;
    font-size: 200%;
}

div::first-line {
    color: blue;
    font-variant: small-caps;
}
```

In the HTML document, there are three **divs**. This is how it will look like:

LOREM IPSUM DOLOR SIT AMET,
consectetur adipiscing elit. Proin nibh
augue, suscipit a, scelerisque sed, lacinia in,
mi.
CRAS VEL LOREM. ETIAM PELLentesque
aliquet tellus. Phasellus pharetra nulla ac
diam. Quisque semper justo at risus. Donec
venenatis, turpis vel hendrerit interdum, dui
ligula ultricies purus, sed posuere libero dui
id orci.
NAM CONGUE, PEDE VITAE DAPIBUS
aliquet, elit magna vulputate arcu, vel
tempus metus leo non est. Etiam sit amet
lectus quis est congue mollis.

Every **div** is beginning with a big green letter (**color: green; font-size: 200%;**) and the rest of their first lines are written in blue capitalised letters (**color: blue; font-variant: small-caps;**).

 **10.1.2**

Pseudo-elements **::before** and **::after** are used to place text respectively before and after the formatted element of the page. To add something you should use the **content** property. By default those pseudo-elements are inline.

See the document's complete code below including the internal style sheet:

```
<html>
  <head>
    <style type="text/css">
      p::before {
        content: "Desmond. The Moon Bear.";
        color: blue;
        font-weight: bold;
      }
      p::after {
        content: "How did I get here?";
        color: green;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit.</p>
  </body>
</html>
```

The result is as on the screenshot below:

Desmond. The Moon Bear. Lorem ipsum
dolor sit amet, consectetur adipiscing
elit. **How did I get here?**

The paragraph is preceded by the blue and bold text and it is followed by a green and bold text. Those texts appear only in the internal style sheet in the document's header.

 **10.1.3**

The **::first-letter** and **::after** selectors are called:

- pseudo-elements
- pseudo-classes
- sibling selectors
- descendant selectors

10.1.4

Which pseudo-element adds a styling only for the first letter of the text?

- ::first-letter
- :first-letter
- first-letter

10.1.5

Complete the code below so that:

- before the text the word “Hello!” appears,
- the first line of the text has a lime background.

Here is a helpful visualization:

Hello! Lorem ipsum dolor sit amet,
consectetur adipiscing elit.

```
<html>
  <head>
    <style type="text/css">
      div_____ { _____ "Hello!";
        background-color: yellow;
      }
      div_____ {
        background-color: lime;
      }
    </style>
  </head>
  <body>
    <_____>Lorem ipsum dolor sit amet, consectetur  
adipiscing elit.</div>
  </body>
</html>
```

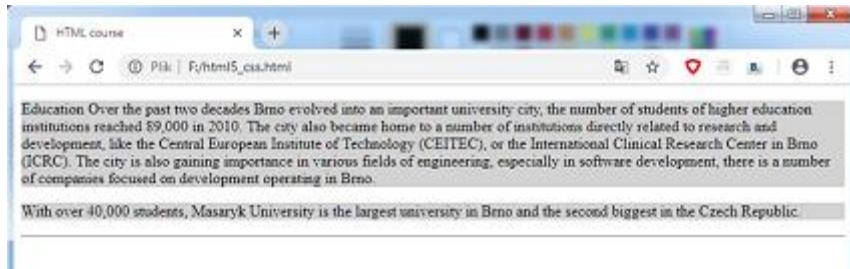

Box Model, Image and Float

Chapter 11

11.1 Box model

11.1.1

Each element inserted into a website naturally occupies a certain area.



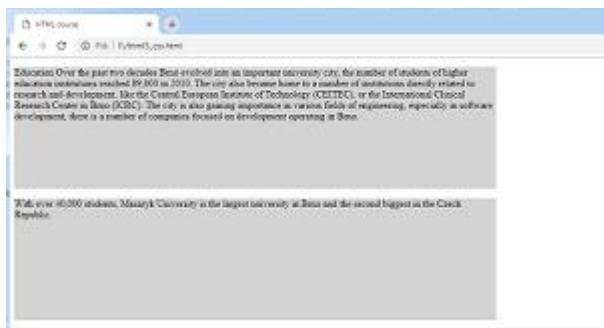
The most common properties that change the size of the area used by page elements are width and height.

In the CSS worksheet:

```
p{width:800px; height:200px; background-color:lightgrey}
```

In the HTML file:

```
<p>Education Over the past two decades Brno evolved into an important university city,  
...</p>  
<p>With over 40,000 students, Masaryk University is the largest university in Brno and the second biggest in the Czech Republic.</p>
```



With the width of the mobile device less than 800px, the value set in this way will insert a horizontal scroll bar. Such a page is not responsive.

 **11.1.2**

Complete the fragment of the style code so that the `<section>` tags on the page have a width of 400px and a height of 300px.

```
section {_____ : 400px; _____ : 300px}
```

 **11.1.3**

Specifying height or width in pixels may cause the browser to insert additional scrollbars if its dimensions are smaller than those declared in the elements.

The width and height values can also be specified in %. It's a very convenient way. It allows you to change the size of the element whenever the parent element changes (eg the browser window). Specifying values with % works well in the design of responsive sites.

```
div {width:50%; height:auto;}
```

In the case of such given dimensions, the problem may be too large a container size (for large resolutions). You can then apply properties that limit the size of the element:

- *max-width*,
- *min-width*,
- *max-height*,
- *min-height*.

```
article {width:50%; max-width:560px; min-width:300px;}
```

This definition of the `<article>` dimensions will cause that its width will not be less than 300px and will not exceed 560px, and for resolutions between 300px and 560px, it will occupy 50% of the width of the parent element.

 **11.1.4**

Each of the block elements, inserted on the page, can have its own border. Properties associated with the element border allow you to change the line width, color and style. You can also define a border with a pattern inserted from a file or border only selected edges of an element, etc. There are many different properties associated with a border.

Below we present only selected elements:

- *border-width* - defines the width of the border line,

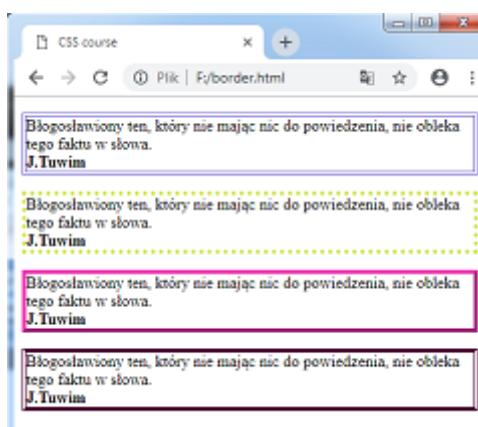
- *border-color*- indicates the color of the border,
- *border-style* - allows you to choose the type of line (*solid, double, dotted, groove* and so on),
- *border-radius*- allows you to round the vertices of the border.

In the CSS file, the first paragraph has formatting:

```
p {border-width:4px; border-style:double; border-
color:#6a5acd; }
```

In the HTML file:

```
<p>Błogosławiony ten, który nie mając nic do powiedzenia, nie
obleka tego faktu w słowa.<br>
<b>J.Tuwim</b></p>
```



Translation of Julian Tuwim's maxim:

"Blessed is he who, having nothing to say, does not get this fact into words."

11.1.5

To create a double line border for the page element, you should use the property:

**Write only the CSS property, without its value.*

11.1.6

If the border of the element is constant on each side, then you can use the collective property of the border, giving in the proper order the values:

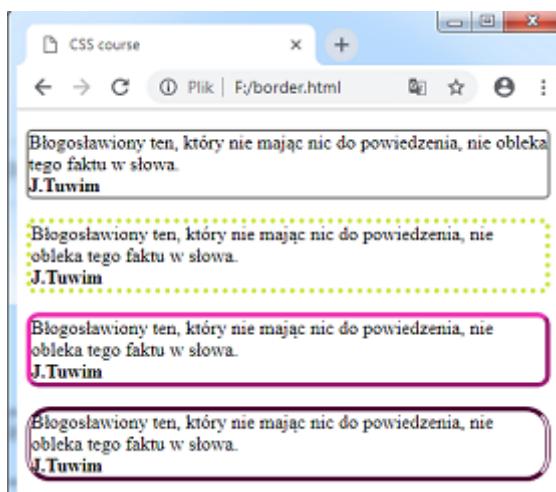
border: width style color;

```
article {border: 2px solid grey; border-radius:6px;}
```

In the above example, <article> will be bordered with a single line, 2px thick, in grey color.

The width of the border is most convenient to give in pixels, the color - as before, the style of the border can take one of many values: solid, dotted, double, outset, etc.

In the presented example, the border-radius property is also used. It allows to round the corners of the border. The value given here defines the diameter of the circle whose fragment is an element of the border. The higher the value, the more rounded the item.



11.1.7

The width property applies only to the area that the element occupies, e.g. text on the page. The border is placed outside the content of the box element. The distance of the box content from the border can be changed with the padding property, which determines the inner margin of the element.

```
article {width:300px; border: 4px solid red; padding:20px;}
```

Padding: 20px means setting all four (from each side) inner margins for the <article> tag to 20px.

Of course, you can also change only the selected margin:

- *padding-top*
- *padding-right*

- *padding-bottom*
- *padding-left*

By setting the width of the inner margin you can use absolute units, for example, px (pixels) or relative units, e.g. percentages. The value given in percent is calculated relative to the width of the parent element, e.g. 2% of the width of the browser window. The value given in % will change as the parent's width changes.

11.1.8

On the outside of the border of the element lies the area of the outer margin, defined by the margin property.

By setting the outer margins of the element, you can also use the properties:

- *margin-top*
- *margin-right*
- *margin-bottom*
- *margin-left*

Example:

```
img {
  margin:100px;
  padding:50px;
  background-color:grey;
  border:solid 10px #b3ccff;
  width:400px;
}
```

- *margin: 100px;* means setting each of the four outer margins to 100px,
- *padding: 50px;* sets each internal margin to 50px,
- *border: solid 10px blue;* creates a blue solid line border with a width of 10px.



The total width of the element thus defined is then:

$$2 * 100px + 2 * 50px + 2 * 10px + 400px = 720px.$$

It is necessary to know the total width to properly position elements on the page.

11.1.9

Match the given values.

Rounding the corners of the border _____

Style of the drawn line _____

External bottom margin _____

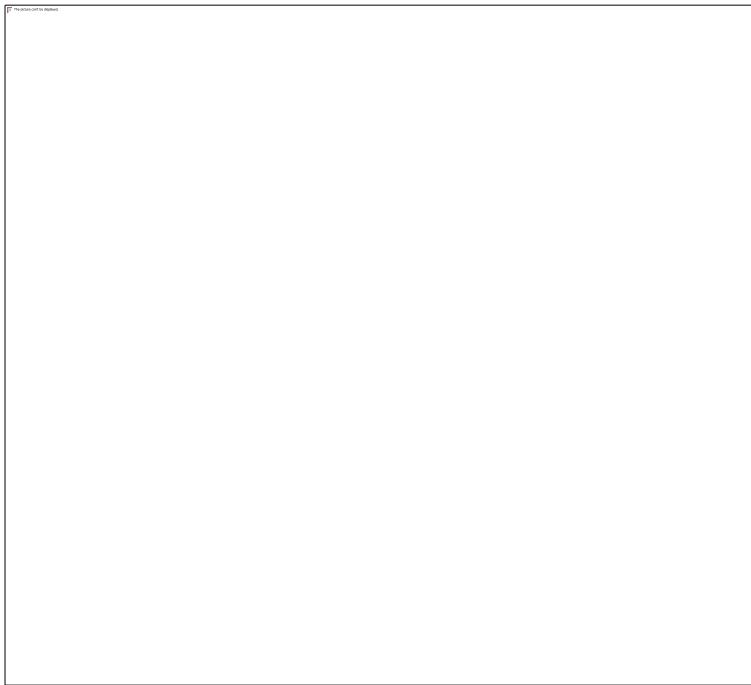
The internal margin of the element _____

The color of the border line _____

- margin-bottom
- border-color
- margin-top
- padding
- border-radius
- border-width
- border-style

11.1.10

The standard property layout that affects the width of the element is shown in the figure below:



Outer margins, borders, inner margins, the width of the element determined the total width of the element. When preparing the website, we try to calculate the total width of the element, taking into account the widths of the listed properties.

11.1.11

```
aside {
  padding-left:2%;
  padding-top:5%;
  padding-right:4%;
  width:50%;
  border:3% dotted green;
}
```

The total width of the element defined above is:

- 62%
- 59%
- 67%
- 50%

11.1.12

The box-sizing property allows the border and padding (internal margin) to be included in the given element width.

Inserting the *box-sizing: border-box* property causes the width to specify the width of the element along with the inner margin and border.

```
section {
    box-sizing: border-box;
    width:300px;
    padding:20px;
    border-width:10px;
}
```

The total width of the section defined in this way is 300px.

This property is inherited.

```
body {box-sizing: border-box;}
```

If this property is set for the **<body>** section, the child elements inherit this dimensioning.

This property was introduced in CSS3 and is very popular among website designers.

11.1.13

```
aside {
    padding-left:2%;
    padding-top:5%;
    padding-right: 4%;
    width:50%;
    border:3% dotted green;
    box-sizing:border-box;
}
```

The actual width of the element defined above is:

- 50%
- 62%
- 59%
- 67%

 11.1.14

If you try to add too much water to a small glass it will spill out. The same applies to the elements of the page, which have a too low height.

If the data planned in the block element occupy a larger area than the block dimensions predict, then they naturally "spill" out of the reserved area.

The overflow property can be used to determine the display of this excess.

- **overflow: scroll** - inserts vertical scrolling bar into the element,
- **overflow: auto** - a vertical scroll bar will be inserted only if the data does not fit in the element,
- **overflow: hidden** - hides elements that won't fit in.

```
article{ width:40%;height:70%; overflow:auto; }
```

In the `<article>` tag, the scroll bar will be inserted only when needed, ie when the data does not fit in the declared area.

 11.1.15

There are several ways to set elements in the centre of the screen in CSS. The most popular are:

- `margin:auto`,
- `display:flex`; (*this solution will be discussed later*).

In order for the element to be centred horizontally, it should be the block element, have a defined width, and the margins should be calculated using the auto value.

```
div {width:40%; margin:auto; }
img {display:block; width:20%; margin:auto; }
a {display:block; width:100px; margin:auto; }
```

The `` and `<a>` tags are linear elements. To set them in the centre of the page, they must be given a block (`display: block`) character. The `<div>` tag is a block element, so all you need to do is define the width and the automatic margins.

 11.1.16

Complete the code fragment so that the `<section>` tag is displayed in the middle of the page and the scroll bar automatically appears when the overflow occurs.

```
section {
    width: 400px;
    margin: ____;
    ____: auto;
}
```

11.2 Box model (programs)

11.2.1 Width

Set the width of the `<article>` tag to 400px.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML </title>
        <style>
            article {background-color:lightgreen;}
            h5 {text-align:right;}
            footer {font-size:0.6em;color:blue;}
        </style>
    </head>
    <body>
        <header>
            <h1> Page title </h1>
            <nav id="top">
                <a href="#"> Link1 </a>|
                <a href="#"> Link2 </a>|
                <a href="#"> Link3 </a>|
                <a href="#"> Link4 </a>
            </nav>
        </header>
        <article>
            <h4> How do I log out of the network? Coach wanted.</h4>
            <p> I'm looking for someone to teach me how to rest. I'm programming in several languages, but I have a problem with turning off my computer and logging out of the network. I'm waiting for applications from people with a lot of pedagogical experience. Generally I am looking for a coach.
            </p>
            <h5> <b>Max</b></h5>
        </article>
    </body>
</html>
```

```

        </article>
        <footer>
            <p>The copyrights are held by Jan Kowalski.
            This website reflects the author's views only. <br>
<a href="#top">TOP </a>           </p>
        </footer>
    </body>
</html>

```

11.2.2 Min-width

Set the width of the `<article>` tag to not less than 200px.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML </title>
        <style>
            article {background-color:lightgreen;      }
            h5 {text-align:right;}
            footer {font-size:0.6em;color:blue;}
        </style>
    </head>
    <body>
        <header>
            <h1> Page title </h1>
            <nav id="top">
                <a href="#"> Link1 </a>|
                <a href="#"> Link2 </a>|
                <a href="#"> Link3 </a>|
                <a href="#"> Link4 </a>
            </nav>
        </header>
        <article>
            <h4> How do I log out of the network? Coach
wanted.</h4>
            <p> I'm looking for someone to teach me how to rest.
I'm programming in several languages, but I have a problem
with turning off my computer and logging out of the network.
I'm waiting for applications from people with a lot of
pedagogical experience. Generally I am looking for a couch.
            </p>
            <h5> <b>Max</b></h5>

```

```
</article>
<footer>
  <p>The copyrights are held by Jan Kowalski.
  This website reflects the author's views only. <br>
<a href="#top">TOP </a>           </p>
  </footer>
</body>
</html>
```

11.2.3 Border

For all `<p>` tags, enter a 2px wide border.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>HTML</title>
    <style>
      p {background-color:lightgreen; border-style: solid;
}
      h4 {text-align:center;  }
    </style>
  </head>
  <body>
    <h4>Who said that? </h4>
    <p>
      I'm not sleeping at all. I'm just checking my eyelids'
light permeability.
    </p>
    <p>
      It's not a mess. It's just another way of putting
things in order.
    </p>
    <p>
      I'd like something different, with something more.
    </p>
  </body>
</html>
```

11.2.4 Border 2

For all `<p>` tags, enter a 3px wide border, double line, in green colour. Use the grouping formatting method.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            p {background-color:lightgreen;      }
            h4 {text-align:center;    }

        </style>
    </head>
    <body>
        <h4>Who said that? </h4>
        <p>
            I'm not sleeping at all. I'm just checking my eyelids' light permeability.
        </p>
        <p>
            It's not a mess. It's just another way of putting things in order.
        </p>
        <p>
            I'd like something different, with something more.
        </p>
    </body>
</html>
```

11.2.5 Padding

For all `<p>` tags, enter a left internal margin of 20px width.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            p {background-color:lightgreen; border: 3px double green;  }
```

```

        h4 {text-align:center;  }

    </style>
</head>
<body>
<h4>Who said that? </h4>
<p>
    I'm not sleeping at all. I'm just checking my eyelids'
light permeability.
</p>
<p>
    It's not a mess. It's just another way of putting
things in order.
</p>
<p>
    I'd like something different, with something more.
</p>

</body>
</html>

```

■ 11.2.6 Padding 2

For all `<p>` tags, enter internal margins of 20px width (on each side). Use the margin shorthand property with one value.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            p {background-color:lightgreen; border: 3px double
green; }
            h4 {text-align:center;  }

        </style>
    </head>
    <body>
        <h4>Who said that? </h4>
        <p>
            I'm not sleeping at all. I'm just checking my eyelids'
light permeability.
        </p>

```

```
<p>
  It's not a mess. It's just another way of putting
things in order.
</p>
<p>
  I'd like something different, with something more.
</p>

</body>
</html>
```

💻 11.2.7 Margin

For all `<p>`tags, enter external margins of 50px width (on each side). Use the margin shorthand property with one value.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>HTML</title>
    <style>
      p {background-color:lightgreen; border: 3px double
green; padding:20px; }
      h4 {text-align:center; }

    </style>
  </head>
  <body>
    <h4>Who said that? </h4>
    <p>
      I'm not sleeping at all. I'm just checking my eyelids'
light permeability.
    </p>
    <p>
      It's not a mess. It's just another way of putting
things in order.
    </p>
    <p>
      I'd like something different, with something more.
    </p>

  </body>
</html>
```

11.2.8 Margin-auto

Center the `<p>` markers on the page using the left and right margin settings with `auto` value.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            p {background-color:lightgreen; border:3px double green; width:70%; padding:20px; }
            h4 {text-align:center; }

        </style>
    </head>
    <body>
<h4>Who said that? </h4>
<p>
    I'm not sleeping at all. I'm just checking my eyelids' light permeability.
</p>
<p>
    It's not a mess. It's just another way of putting things in order.
</p>
<p>
    I'd like something different, with something more.
</p>

</body>
</html>
```

11.2.9 Height

Set the width of the article tag to 40% and the height to 200px. Note that at certain resolutions the text "spills" from the tag.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
```

```

        <style>
            article {background-color:lightgreen; border:3px
double green; padding:20px; margin-left:auto; margin-
right:auto;}
            h4 {text-align:center;  }

        </style>
    </head>
    <body>

        <article>
            <h4>Bootstrap </h4>
            <p>Create responsive mobile projects on the Internet
with the world's most popular front-end component library.
Bootstrap is an open source toolkit for programming in HTML,
CSS and JS. Quickly prototype your ideas or create an entire
application with our Sass variables and mixes, responsive grid
system, powerful components and powerful jQuery-based
plugins.<br>
            <a
                href="https://getbootstrap.com/">https://getbootstrap.com/</a>
            <br>
            Other popular frameworks: Bulma, Materialize, Semantic UI,
Pure CSS, Fundacja Zurb.
        </p>
        </article>
    </body>
</html>

```

11.2.10 Overflow

Use the overflow property to enter a scroll bar into the `<article>` tag.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            article {background-color:lightgreen; border:3px
double green; width:40%; height:200px; padding:20px; margin-
left:auto; margin-right:auto;}
            h4 {text-align:center;  }

```

```

        </style>
    </head>
    <body>

        <article>
            <h4>Bootstrap </h4>
            <p>Create responsive mobile projects on the Internet with the world's most popular front-end component library. Bootstrap is an open source toolkit for programming in HTML, CSS and JS. Quickly prototype your ideas or create an entire application with our Sass variables and mixes, responsive grid system, powerful components and powerful jQuery-based plugins.<br>
                <a href="https://getbootstrap.com/">https://getbootstrap.com/</a>
                <br>
                Other popular frameworks: Bulma, Materialize, Semantic UI, Pure CSS, Fundacja Zurb.
            </p>
        </article>
    </body>
</html>

```

11.2.11 Box-sizing

The `<article>` tags in the example below occupy a different area despite the same width and height values.

Use the `box-sizing` property to change the calculation of the element `width` and cause both `<article>` tags to occupy the same area.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>

            article#a1 {background-color:lightgreen;
width:40%; height:200px; padding:20px; overflow:scroll;}
            article#a2 {background-color:lightblue;
width:40%; height:200px; padding:50px; overflow:scroll;}
            h4 {text-align:center; }

        </style>
    </head>
    <body>
        <article>
            <h4>Article 1</h4>
        </article>
        <article>
            <h4>Article 2</h4>
        </article>
    </body>
</html>

```

```
</style>
</head>
<body>

    <article id="a1">
        <h4>Bootstrap </h4>
        <p>Create responsive mobile projects on the Internet with the world's most popular front-end component library. Bootstrap is an open source toolkit for programming in HTML, CSS and JS. Quickly prototype your ideas or create an entire application with our Sass variables and mixes, responsive grid system, powerful components and powerful jQuery-based plugins.<br>
            <a href="https://getbootstrap.com/">https://getbootstrap.com/</a>
            <br>
            Other popular frameworks: Bulma, Materialize, Semantic UI, Pure CSS, Fundacja Zurb.
        </p>
        </article>
        <article id="a2">
            <h4>Bootstrap </h4>
            <p>Create responsive mobile projects on the Internet with the world's most popular front-end component library. Bootstrap is an open source toolkit for programming in HTML, CSS and JS. Quickly prototype your ideas or create an entire application with our Sass variables and mixes, responsive grid system, powerful components and powerful jQuery-based plugins.<br>
            <a href="https://getbootstrap.com/">https://getbootstrap.com/</a>
            <br>
            Other popular frameworks: Bulma, Materialize, Semantic UI, Pure CSS, Fundacja Zurb.
        </p>
        </article>
    </body>
</html>
```

11.3 Float

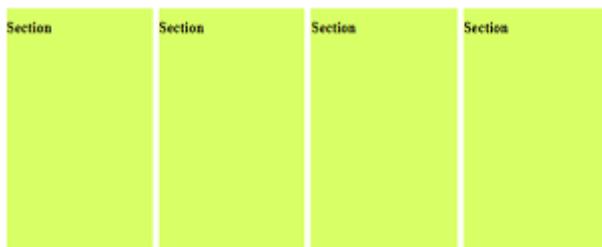
11.3.1

All block elements are normally placed on the page vertically one by one. You can use one of several formatting methods to arrange them horizontally (side by side).

The easiest way is to use the *float* property. It can take one of two values: *left* or *right*.

```
section {float:left; width:24%; margin:0.5%; }
```

Each of the four sections in the project below is placed on the side next to each other and fills the entire width of the page ($4 * 24\% + 8 * 0.5\% = 100\%$).



The *float* property causes that the element is aligned to the *left* (or *right*) side of the page/of the parent. The tags inserted below it are also affected - they stick to it on the *left/right*.

11.3.2

What values can take *float* property?

- left
- right
- center
- justify

11.3.3

The *float* property is very often used to surround the image with text.

```
img {float:right; width:40%; margin:2%;}
```

Iceland

Iceland is a Nordic island country in the North Atlantic, with a population of 348,380 and an area of 103,000 km² (40,000 sq mi), making it the most sparsely populated country in Europe.^[3] The capital and largest city is Reykjavík, with Reykjanes and the surrounding areas in the southwest of the country being home to over two-thirds of the population. Iceland is volcanically and geologically active. The interior consists of a plateau characterised by sand and lava fields, mountains, and glaciers, and many glacial rivers flow to the sea through the lowlands. Iceland is warmed by the Gulf Stream and has a temperate climate, despite a high latitude just outside the Arctic Circle. Its high latitude and marine influence keep summers chilly, with most of the archipelago having a tundra climate.



<https://en.wikipedia.org/wiki/Iceland>

In this case, remember that in the HTML code, the image was placed before the text that surrounds it.

```

<section>
<h2>Iceland</h2>
<p>Iceland is a Nordic island country....</p>
</section>
```

11.3.4

In order for an image that has a *float* property to be surrounded by text, the HTML code should be inserted:

- first the image, then the text.
- first the text, then the picture.

11.3.5

At different browser window widths, the position of elements with *float* property on the page changes dynamically.

For a given tag, you can turn off the permission to surround other elements.

This property can take one of several values:

- *left*,
- *right*,
- *both*.

Not turning off the *float* property might cause the effect as shown in the figure below.



After applying the `clear` property for the `<h2>` tag (the city names are saved in it), these tags will not be placed on objects with the `float: left;` property.

```
h2{clear:left}
```



11.3.6

Complete the code to end the `float: right;` property for the tag.

```
article { ____: ____;
```

11.3.7

The following example defines classes `/f` for images aligned to the left margin and `/r` for right-aligned ones.

```
img.lf {float:left; width:30%; margin:2%;}
img.rt {float:right; width:30%; margin:2%;}
p{text-align:justify;}
h2{text-align:center;}
```



Iceland is a Nordic island country in the North Atlantic, with a population of 341,580 and an area of 103,000 km² (40,000 sq mi), making it the most sparsely populated country in Europe.^[1] The capital and largest city is Reykjavík, with Reykjanes and the surrounding areas in the southwest of the country being home to over three-quarters of the population.

<https://en.wikipedia.org/wiki/Iceland>

```


<section>
  <h2>Iceland</h2>
  <p>Iceland is a Nordic island country ....</p>
</section>
```

11.3.8

Based on the *float* property, you can create a page template. The following code shows one of the many options for arranging elements on the page.

```
header {height:100px; margin:0.5%;}
nav {height:30px; margin:0.5%;}
section {float:left; margin:0.5%;width:32.3%; height:400px;}
footer {height:50px; margin:0.5%;clear:both;}
```



11.4 Float (programs)

11.4.1 Float right

Format the *<article>* tags so that they are aligned to the right side and occupy 30% of the width of the parent element. Use the *float* property.

```
<!DOCTYPE html>
<html lang="en">
  <head>
```

```

<meta charset="UTF-8">
<title>HTML</title>
<style>
    article {background-color:lightgreen;
padding:30px; border: 1px solid green; box-sizing:border-box;
}
</style>
</head>
<body>
<article>
<h4>Bootstrap </h4>
<p>Create responsive mobile projects on the Internet  

with the world's most popular library of front-end components.  

Bootstrap is an open source toolkit for programming in HTML,  

CSS and JS. Quickly prototype your ideas or create an entire  

application with our Sass variables and mixes, responsive grid  

system, powerful components and powerful jQuery-based  

plugins.<br>
<a href="https://getbootstrap.com/">Bootstrap</a>

</article>
<article>
<h4>React.js </h4>
<p>React.js (other names used: React, ReactJS) -  

JavaScript programming language library, which is used to  

create graphical interfaces of web applications. It was  

created by Jordan Walke, a Facebook programmer, and inspired  

by the PHP language extension - XHP.<br>

<a
href="https://pl.wikipedia.org/wiki/React.js">React.js</a>
</p>
</article>
<article>
<h4>Laravel </h4>
<p>Laravel is an open-source framework for creating web  

applications. With a very elegant and expressive syntax.  

Laravel provides the user with typical features such as:  

authentication, sessions or routing. It is also based on MVC  

model (Model-View-Control). It is used for programming in PHP  

language.<br>

<a
href="https://pl.wikipedia.org/wiki/React.js">React.js</a>

```

```

    </p>
</article>
</body>
</html>

```

11.4.2 Clear

Format the **<figure>** tags so that they are aligned to the left and the text surrounds them on the right. Format **<h2>** headers so that they are not surrounded anymore. Use the float and clear properties.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<base href="https://priscilla.fitped.eu/images/">
<style>
p{text-align:justify;}
figure {width:15%; margin:10px; padding:5px; border:solid 1px black; margin-top:0px; }
figcaption { margin:0px; padding:10px; background-color:lightgrey; text-align:center; }
img {max-width:100%; margin:0px; }
h2{margin:2%;}
</style>
</head>

<body>
<h2> Dubrownik</h2>
<figure>

<figcaption>
Dubrownik
</figcaption>
</figure>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent tempus dui a nibh tempor tempus. Donec vestibulum posuere congue. Quisque pellentesque euismod mollis. Maecenas iaculis risus feugiat sodales volutpat. Nunc pretium venenatis ipsum, eget iaculis tellus feugiat at. Nam arcu enim, volutpat eu nisi id, ullamcorper pharetra felis. Aenean ullamcorper ullamcorper pellentesque. Aliquam bibendum massa ultrices nibh

```

```
gravida ornare. Fusce vehicula mauris ut nibh tempor viverra  
tincidunt vitae nisl.  
</p>  
<br>  
<h2> Korcula</h2>  
<figure>  
  
<figcaption>  
Korcula  
</figcaption>  
</figure>  
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Praesent tempus dui a nibh tempor tempus. Donec vestibulum  
posuere congue. Quisque pellentesque euismod mollis. Maecenas  
iaculis risus feugiat sodales volutpat. Nunc pretium venenatis  
ipsum, eget iaculis tellus feugiat at. Nam arcu enim, volutpat  
eu nisi id, ullamcorper pharetra felis. Aenean ullamcorper  
ullamcorper pellentesque. Aliquam bibendum massa ultrices nibh  
gravida ornare. Fusce vehicula mauris ut nibh tempor viverra  
tincidunt vitae nisl.  
</p>  
<br>  
<h2> Zadar</h2>  
<figure>  
  
<figcaption>  
Zadar  
</figcaption>  
</figure>  
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Praesent tempus dui a nibh tempor tempus. Donec vestibulum  
posuere congue. Quisque pellentesque euismod mollis. Maecenas  
iaculis risus feugiat sodales volutpat. Nunc pretium venenatis  
ipsum, eget iaculis tellus feugiat at. Nam arcu enim, volutpat  
eu nisi id, ullamcorper pharetra felis. Aenean ullamcorper  
ullamcorper pellentesque. Aliquam bibendum massa ultrices nibh  
gravida ornare. Fusce vehicula mauris ut nibh tempor viverra  
tincidunt vitae nisl.  
</p>  
</body>  
</html>
```

11.4.3 Img - classes

Define the *img.lf* class so that the images are aligned to the left and surrounded on the right.

Define the *img.rt* class so that the images are aligned to the right.

Use the *float* property.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<base href="https://priscilla.fitped.eu/images/">
<style>
img{ width:30%; max-width:100%;}

</style>
</head>

<body>
<figure></figure>
<figure></figure>

</body>

</html>
```

11.4.4 Clear - both

In the following section of the page, format title tags **<h2>** so that they never surround elements from either side.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<base href="https://priscilla.fitped.eu/images/">
<style>
img{ width:30%; max-width:100%;}
```

```



h2{ text-align:center;}
</style>
</head>
<body>


## Islandia </h2> <figure></figure> <figure></figure> Description of the trip</p> Norwegia </h2> <figure></figure> <figure></figure> Description of the trip</p> </body> </html>


```

11.5 Image

11.5.1

The width and height properties are used to change the dimensions of the image placed on the page.

The width and height of images can be determined in absolute units, e.g. in px or in relative units, e.g. in percent relative to the parent element, i.e. the container.

```

![bird](1.jpg)

```

To preserve the proportions of the inserted image, one of the properties should be given the auto value.

A big problem with such given units (absolute) is the lack of responsiveness of the image.

If the image is wider than the browser window, a horizontal scroll bar will appear.

11.5.2

The width of the inserted image can be given in percent units. This value determines what percentage of the container will be occupied by the image in relation to the entire width of the container.

```

```

Such width of the image will cause the image to fill the entire available area of the container. If the area to be filled is larger than the width of the image saved in the file - the image will be stretched and will lose focus.

The most convenient way to define the size of the image in this situation will be to use the max-width property.

```

```

Such a record will cause the image to not exceed the width of the file. If the container is smaller than the width given in the file - the image will use 100% width, otherwise, it will have the width as in the file.

11.5.3

In order to keep the proper size of the image (in relation to its actual size and image proportions), especially while changing the browser's window you should add CSS definition:

- max-width:100%; height:auto;
- max:100%; height:auto;
- width:100%; height:100%;
- width:auto; height:100%;

11.5.4

Interesting effects in formatting images can be obtained by using the border-radius property.

```



```



The border-radius property allows you to round the corners of the graphic.

Its value can be given in pixels, percentages and other available units.

The value of 50% causes that the inserted picture has the shape of an ellipse.

11.5.5

To round the corners of the inserted image, use the CSS property:

11.5.6

Images inserted into websites can have a shadow added. The box-shadow property has several arguments.

box-shadow: right bottom blur color

When defining a shadow for a block element, specify the shadow displacement direction, e.g. right and down, shadow blur radius and color.



```


```

11.5.7

Adding a shadow to the image requires the use of a CSS property:

 11.5.8

In CSS, there is no property that is used to centre the image. To place the image horizontally in the middle of the page, two parameters should be defined: give it the property of a block object (*display: block*) and enter automatic margins for this object (*margin: auto*).

```
img {
    display: block;
    margin-left: auto;
    margin-right: auto;
    width: 50%;
}
```



Another way to centre the image is the new *display: flex* property that you can read about in the next lesson.

 11.5.9

Enter the graphics formatting elements so that it is placed in the centre of the page.

```
img {
    ____: block;
    margin-left: ____;
    margin-right: ____;
    width: 50%;
}
```

 11.5.10

Complete the code so that the image *mountain.jpg*.

- had a width not greater than that saved in the file,

- had rounded corners - 10px;
- with a 1% wide shadow, in grey colour.

```
<img _____="mountain.jpg" _____="mountain"
style="max-width:_____; _____:10px; _____: 5px 5px 10px
grey; ">
```

11.5.11

The CSS3 specification has a new property that allows you to adjust the size of the inserted object (images or video files) to the changing area of the container (ie the parent element, sometimes it is the browser window area).

The ***object-fit*** property allows one of the following values to be used:

- *fill* - default value, if the container area is too small or too big it will perfectly fit the image to fill all available space;
- *contain* - the image is scaled to fill the required area horizontally and keep the proportions - it can be reduced or increased;
- *cover* - cuts off part of the image to fill a too small container and keep the image's height.
- *scale-down* - scales an image to fill the desired area horizontally, but only "down" or only decreases.

This property is relatively new, before using it one should check the support of individual browsers.

This property fits well in the design of responsive websites.

11.5.12

Preparation of a simple gallery based on *float* ownership.

Here the picture fills the entire container. The ***<figure>*** tag is a responsive container (20% wide), has *float* property (elements will wrap when the width of the browser window is changed), *margins* and *borders* are defined. The signature below the image - ***<figcaption>*** has margins and *background*.

CSS

```
figure {
width:20%;
margin:10px;
padding:5px;
```

```

float:left;
border:solid 1px black;
}
figcaption {
margin:0px;
padding:10px;
background-color:lightgrey;
}
img {
max-width:100%;
margin:0px;
}

```

HTML

```

<figure>

<figcaption>
1. Mogielica zimą :)
</figcaption>
</figure>

```



11.6 Image (programs)

11.6.1 Img

Change the width of the placed image. Set the width property to 40%, but not more than 600px;

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<base href="https://priscilla.fitped.eu/images/">
<style>
img{ }

```

```

h2{ text-align:center; }
</style>
</head>
<body>
<h2>Iceland </h2>
<figure></figure>
<figure></figure>

</body>
</html>

```

11.6.2 Img - border-radius

Enter the property that will allow you to round the corners of the image. Set the value to 50%.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<base href="https://priscilla.fitped.eu/images/">
<style>
img{ width:40%; max-width:600px; }
h2{ text-align:center; }
</style>
</head>
<body>
<h2>Iceland </h2>
<figure></figure>
<figure></figure>

</body>
</html>

```

11.6.3 Img - shadow

Define the shadows for the pictures. Set the shadow shift by 3px to the right and 3px down and the color to light grey.

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<meta charset="UTF-8">
<title>Title of the document</title>
  <base href="https://priscilla.fitped.eu/images/">
<style>
img{ width:40%; max-width:600px; border-radius:50%; }
h2{ text-align:center; }
</style>
</head>
<body>
<h2>Iceland </h2>
<figure></figure>
<figure></figure>

</body>
</html>

```

11.6.4 Img - figcaption

Set the background colour of the image caption to light green and its width to 40%.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
  <base href="https://priscilla.fitped.eu/images/">
<title>Title of the document</title>
<style>
img{ width:40%; max-width:600px; border-radius:50%;box-shadow: 3px 3px lightgrey; }
h2{ text-align:center; }
figcaption{ };
</style>
</head>
<body>
<h2>Islandia </h2>
<figure> <figcaption>
Picture 1</figcaption> </figure>
<figure> <figcaption>
Picture 2</figcaption> </figure>

</body>
</html>

```

11.6.5 Img - border

Format the `<figure>` tag so that it occupies 30% of the page width and has 5px internal margins (collective property) and a 1px border, continuous line, in black (use the `border: width type color;` property).

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<base href="https://priscilla.fitped.eu/images/">
<style>

h2{ text-align:center; }

figure { margin:10px; }
figcaption {margin:0px; padding:10px; background-color:lightgrey; text-align:center;}
img {max-width:100%; }

</style>
</head>
<body>
<h2>Iceland </h2>
<figure> <figcaption> Pix 1</figcaption> </figure>
<figure> <figcaption> Pix 2</figcaption> </figure>

</body>
</html>
```

11.7 Font-face rule

11.7.1

Typography (printing letters and other writing marks) is considered one of the most important elements taken into account when designing websites. When designing a website, we usually use standard fonts installed on the user's computer.

If the browser does not find the fonts we suggest, it displays the text using the default font.

More and more websites have their own imported fonts.

```
@font-face {font-family: 'font-fitped';
src: url('path/to/font.woff');}
```

- font-family: 'font-fitped' specifies the name of the new font that will be used in CSS;
- src indicates the path to the font file and its name,

The defined font can be used to format the text on the page, e.g.

```
p, article {font-family: font-fitped;}
```

11.7.2

The **@font-face** rule allows you to insert your own fonts saved in separate files and placed on the server or on the website.

The most popular file formats with fonts are:

- **WOFF (The Web Open Font Format)** is a font format, developed in 2009, recommended by W3C for use on websites,
- **TTF (TrueType Fonts)** is a standard developed by Apple and Microsoft in the '80,
- **OTF (Open Type Font)** is a format of scalable computer fonts, it is a registered trademark of Microsoft,
- **WOFF 2.0 (The Web Open Font Format)** successor of WOFF, provides better compression (it has more and more support for browsers).

Currently, most browsers support WOFF and OTF formats. Files with fonts in such formats should therefore be attached to pages.

Because the @font-face command is a relatively new element in page design, you should closely follow changes in W3C recommendations and browser rendering of pages.

11.7.3

Fill in the code to attach the wonderful.woff font (placed in the current directory) to the page and format the **<h2>** header:

```
_____ {
_____ : _____;
src: url(wonderful.woff')
}
```

```
h2 { ____ : wonderful }
```

11.7.4

Taking into account the fact that the support of popular font formats by browsers is not full, adding new fonts to the site, it is worth placing files in two formats: WOFF and OTF:

```
@font-face {
    font-family: 'MyFonts';
    src: url('fontname.otf'), url('fontname.woff'),
url('fontname.ttf');
    font-style: normal;
    font-weight: normal;
}
```

Many websites on the web provide one format conversion service to another.

New fonts can be stored on your own server or you can use font hosting on an external server (but this requires more knowledge).

In the network you can find many websites offering fonts for websites, for example:

<https://fonts.google.com/>

<https://fonts.adobe.com/>

<https://www.fontsquirrel.com/>

When using them, always pay attention to the terms of the license.

11.7.5

Select the font format recommended by W3C.

- woff
- ttf
- otf

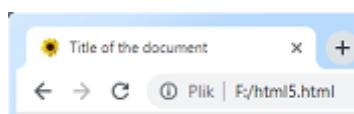
 **11.7.6**

Most pages put a website logo icon on the tab in the browser, in front of the page title.

```
<link rel="icon" href="logo.gif" type="image/gif"
sizes="16x16">
```

Typically, this icon is 16x16 px, but different devices can display this icon in different sizes.

```
<link rel="icon" href="logo.gif" type="image/gif" sizes="16x16
32x32">
```

 **11.7.7**

Complete the code in such a way that the logo.png image is placed on the page tab.

```
<link rel="____" ____="logo.png" type="image/gif"
sizes="16x16 32x32">
```

11.8 Font-face rule (program)

 **11.8.1 Font-face (link)**

A file with a font called "Open Sans" was attached to the page. Define a class for the paragraph named "opens" and format the font to Open Sans. Use the class you created in the first paragraph.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>FITPED</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <link
      href="https://fonts.googleapis.com/css?family=Open+Sans|Source
      +Sans+Pro&display=swap" rel="stylesheet">
```

```
<style>

</style>
</head>
<body>
    <section>
        <h2>Title</h2>
<p >Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent tempus dui a nibh tempor tempus. Donec vestibulum posuere congue. Quisque pellentesque euismod mollis. Maecenas iaculis risus feugiat sodales volutpat. Nunc pretium venenatis ipsum, eget iaculis tellus feugiat at. Nam arcu enim, volutpat eu nisi id, ullamcorper pharetra felis. Aenean ullamcorper ullamcorper pellentesque. Aliquam bibendum massa ultrices nibh gravida ornare. Fusce vehicula mauris ut nibh tempor viverra tincidunt vitae nisl. Nunc turpis tortor, facilisis at pretium sit amet, lobortis ac ex. Aenean turpis metus, euismod in augue id, rutrum porttitor quam. In velit neque, pharetra vehicula libero sit amet, tempus vehicula risus. Cras in pharetra purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec vulputate ante a turpis molestie auctor. Interdum et malesuada fames ac ante ipsum primis in faucibus.
</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent tempus dui a nibh tempor tempus. Donec vestibulum posuere congue. Quisque pellentesque euismod mollis. Maecenas iaculis risus feugiat sodales volutpat. Nunc pretium venenatis ipsum, eget iaculis tellus feugiat at. Nam arcu enim, volutpat eu nisi id, ullamcorper pharetra felis. Aenean ullamcorper ullamcorper pellentesque. Aliquam bibendum massa ultrices nibh gravida ornare. Fusce vehicula mauris ut nibh tempor viverra tincidunt vitae nisl. Nunc turpis tortor, facilisis at pretium sit amet, lobortis ac ex. Aenean turpis metus, euismod in augue id, rutrum porttitor quam. In velit neque, pharetra vehicula libero sit amet, tempus vehicula risus. Cras in pharetra purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec vulputate ante a turpis molestie auctor. Interdum et malesuada fames ac ante ipsum primis in faucibus.
</p>

    </section>
</body>
```

```
</html>
```

■ 11.8.2 Font-face (@import)

A font called "IBM Plex Serif" was imported into the style sheet. Define a class associated with a particular tag named "ibm_p", which will allow you to format the first tag `<p>` and `<div>` with the attached font. Use the defined class in the tags.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>FITPED</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1">
        <style>
            @import
url('https://fonts.googleapis.com/css?family=IBM+Plex+Serif|Li
bre+Franklin|Source+Sans+Pro&display=swap');

        </style>
    </head>
    <body>
        <section>
            <h2>Title</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent tempus dui a nibh tempor tempus. Donec vestibulum posuere congue. Quisque pellentesque euismod mollis. Maecenas iaculis risus feugiat sodales volutpat. Nunc pretium venenatis ipsum, eget iaculis tellus feugiat at. </p>
<div>Nam arcu enim, volutpat eu nisi id, ullamcorper pharetra felis. Aenean ullamcorper ullamcorper pellentesque. Aliquam bibendum massa ultrices nibh gravida ornare. Fusce vehicula mauris ut nibh tempor viverra tincidunt vitae nisl. Nunc turpis tortor, facilisis at pretium sit amet, lobortis ac ex. Aenean turpis metus, euismod in augue id, rutrum porttitor quam. In velit neque, pharetra vehicula libero sit amet, tempus vehicula risus. Cras in pharetra purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec vulputate ante a turpis molestie auctor. Interdum et malesuada fames ac ante ipsum primis in faucibus.
</div>
```

```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent tempus dui a nibh tempor tempus. Donec vestibulum
posuere congue. Quisque pellentesque euismod mollis. Maecenas
iaculis risus feugiat sodales volutpat. Nunc pretium venenatis
ipsum, eget iaculis tellus feugiat at. Nam arcu enim, volutpat
eu nisi id, ullamcorper pharetra felis. Aenean ullamcorper
ullamcorper pellentesque. Aliquam bibendum massa ultrices nibh
gravida ornare. Fusce vehicula mauris ut nibh tempor viverra
tincidunt vitae nisl. Nunc turpis tortor, facilisis at pretium
sit amet, lobortis ac ex. Aenean turpis metus, euismod in
augue id, rutrum porttitor quam. In velit neque, pharetra
vehicula libero sit amet, tempus vehicula risus. Cras in
pharetra purus. Class aptent taciti sociosqu ad litora
torquent per conubia nostra, per inceptos himenaeos. Donec
vulputate ante a turpis molestie auctor. Interdum et malesuada
fames ac ante ipsum primis in faucibus.
</p>
        </section>
    </body>
</html>

```

11.8.3 Font-face - file

Using the **@font-face** rule, attach a file with the font named "Poppins-Light.ttf" to the page, name the font "new1". Format the paragraphs only.

```

<html lang="en">
    <head>
        <title>FITPED</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1">
        <base href="https://priscilla.fitped.eu/images/">
        <style>

        </style>
    </head>
    <body>
        <section>
            <h2>Title</h2>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Praesent tempus dui a nibh tempor tempus. Donec vestibulum
posuere congue. Quisque pellentesque euismod mollis. Maecenas

```

```
iaculis risus feugiat sodales volutpat. Nunc pretium venenatis ipsum, eget iaculis tellus feugiat at. </p>
<div>Nam arcu enim, volutpat eu nisi id, ullamcorper pharetra felis. Aenean ullamcorper ullamcorper pellentesque. Aliquam bibendum massa ultrices nibh gravida ornare. Fusce vehicula mauris ut nibh tempor viverra tincidunt vitae nisl. Nunc turpis tortor, facilisis at pretium sit amet, lobortis ac ex. Aenean turpis metus, euismod in augue id, rutrum porttitor quam. In velit neque, pharetra vehicula libero sit amet, tempus vehicula risus. Cras in pharetra purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec vulputate ante a turpis molestie auctor. Interdum et malesuada fames ac ante ipsum primis in faucibus.
</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent tempus dui a nibh tempor tempus. Donec vestibulum posuere congue. Quisque pellentesque euismod mollis. Maecenas iaculis risus feugiat sodales volutpat. Nunc pretium venenatis ipsum, eget iaculis tellus feugiat at. Nam arcu enim, volutpat eu nisi id, ullamcorper pharetra felis. Aenean ullamcorper ullamcorper pellentesque. Aliquam bibendum massa ultrices nibh gravida ornare. Fusce vehicula mauris ut nibh tempor viverra tincidunt vitae nisl. Nunc turpis tortor, facilisis at pretium sit amet, lobortis ac ex. Aenean turpis metus, euismod in augue id, rutrum porttitor quam. In velit neque, pharetra vehicula libero sit amet, tempus vehicula risus. Cras in pharetra purus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Donec vulputate ante a turpis molestie auctor. Interdum et malesuada fames ac ante ipsum primis in faucibus.
</p>
      </section>
    </body>
</html>
```

List, Links and Navs

Chapter **12**

12.1 Links and navigation

12.1.1

Links are one of the most important elements of the website. It is difficult to imagine a website without a well-formatted and clear menu.

The link can be inserted in the inline style:

```
<a href="https://www.fitped.eu/" style="color:darkgreen;
background-color:grey;">
Work-Based Learning in Future IT Professionals Education
</a>
<a href="https://www.ukf.sk/" style="color:darkgreen;
background-color:grey;">
Constantine the Philosopher University in Nitra
</a>
```

The text of the links above will be dark green. Both links will be placed on the side next to each other. This formatting applies to the overall presentation of the link.

Due to the dynamic nature of these elements, these links are formatted differently than other page elements.

12.1.2

Links are unusual elements of the site. Depending on the behavior of the user on the site they can change their appearance. The user can see a regular link placed on the page, and one that was already visited by the user, indicated by the mouse cursor or the one that we clicked (active).

Each of these types of links may look different and require different formatting.

Many links in the network have standard formatting: the usual link is usually blue and underlined, the link is visited purple, and the active - red.

12.1.3

Formatting links, depending on their status, can be defined in the external style or embedded in the <head> section.

This formatting is defined using the pseudo-class selectors:

- *a:link*- regular link,
- *a:visited*- user visited link,
- *a:hover*- the link indicated by the mouse,
- *a:active*-active link (clicked but not yet executed/redirected).

```
a:link, a:visited {color:white; background-color:darkgrey ;
text-decoration: none;}
a:hover {color:darkgrey; background-color: white; text-decoration: none; }
```

The example above uses inverse colors for the font and background in regular and visited links and in the links indicated by the mouse. This type of formatting is very often used in creating a menu on the site. The link pointed to by the mouse has changed colors in relation to the other links.

12.1.4

By default, the links that we insert on the pages are underlined. Provide a property that allows you to turn off link underlining.

**Write only the name of the property without its value.*

12.1.5

When entering your own formatting for individual link states, you should pay attention to the order of the definitions you enter.

CSS has certain priorities in the interpretation of individual definitions.

The order in which the link pseudo-classes are defined must be:

- *a:link*
- *a:visited*
- *a:hover*
- *a:active*

Of course, the user can only format selected "states" of links, but can not change the order.

To remember the order presented, one can imagine the length of time in which we are watching the link in a given state. The usual link may never be visited and it will

be the longest viewed on the site. An active link is seen as the shortest, most visited link - we usually see longer than indicated by the mouse.

12.1.6

Fill in the given code so that the regular links put on the page are "blue" and visited - "purple".

```
_____ {color: _____}  
_____ {color: _____}
```

12.1.7

In what order should the formatting of links in styles be placed?

1 _____

2 _____

3 _____

4 _____

- a:visit
- a:href
- a:active
- a:visited
- a:link
- a:hover

12.1.8

```
a {  
    display:inline-block;  
    font-size:18px;  
    border-radius:5px;  
    margin:1px;  
    padding:10px;  
    width:120px;  
    text-align:center;  
    box-shadow: 3px 3px lightgrey;  
}
```

```
a:link, a:visited {
    color:#FFE8DC;
    background-color: #9D3802;
    text-decoration: none;
}
a:hover {
    color:#9D3802;
    background-color:#FFE8DC;
    text-decoration: none;
}
```

In the above example, many formatting elements were set together for all link states. An important element in this formatting is the `display: inline-block;` property. This command gives a block character to links, thanks to which they can have a fixed width, margins, etc., and allows you to arrange menu items in a row.



If we replace this command with a `display: block;` the menu items will be arranged in a column.



12.2 Links (programs)

12.2.1 Links - underline

Disable the underlining of the given links. Use the internal CSS.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>

            </style>
    </head>
    <body>
        <nav>
```

```

<a href="page1.html">Link1 </a>
<a href="page2.html">Link2 </a>
<a href="page3.html">Link3 </a>
</nav>
</body>
</html>

```

12.2.2 Links - formating

Format the font size of all link types. Set it to 1.5em. Also, disable the underlining of links. Use internal CSS.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>

            </style>
    </head>
    <body>
        <nav>
            <a href="page1.html">Link1 </a>
            <a href="page2.html">Link2 </a>
            <a href="page3.html">Link3 </a>
        </nav>
    </body>
</html>

```

12.2.3 Links - formating 1 (hover)

Format the link indicated with the mouse so that the font colour of the link is black and the background is pink.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            a {text-decoration:none;font-size:1.5em;}

```

```

        </style>
    </head>
    <body>
        <nav>
            <a href="nav.html">Link1 </a>
            <a href="nav.html">Link2 </a>
            <a href="nav.html">Link3 </a>
        </nav>
    </body>
</html>
```

12.2.4 Links - formating 2 (hover inverse)

Check the background and text color of all links. Format the link indicated with the mouse so that the background color and the text color are switched.

```

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            a {font-size:1.5em; text-decoration:none;
color:blue; background-color:white; }

        </style>
    </head>
    <body>
        <nav>
            <a href="page1.html">Link1 </a>
            <a href="page2.html">Link2 </a>
            <a href="page3.html">Link3 </a>
        </nav>
    </body>
</html>
```

12.2.5 Links - formating 3 (display)

The *display:block* property allows you to change the linear character of the link to a block and give it a width. Define the link width to 15% and apply alignment to the right side (*float* command).

```

<html lang="en">
    <head>
```

```

<meta charset="UTF-8">
<title>HTML</title>
<style>
body{ background-color:lightgrey;}
a {font-size:1.5em; text-decoration:none;
color:blue; background-color:white; display: block; }
a:hover {color:white; background-color:blue;}
</style>
</head>
<body>
<nav>
<a href="page1.html">Link1 </a>
<a href="page2.html">Link2 </a>
<a href="page3.html">Link3 </a>
</nav>
</body>
</html>

```

12.2.6 Links - formating 4

Align the text in the link to the centre of the box and add a 5px corner rounding.

```

<html lang="en">
<head>
<meta charset="UTF-8">
<title>HTML</title>
<style>
body{ background-color:lightgrey;}
a {font-size:1.5em; text-decoration:none;
color:blue; background-color:white; display: block;
width:15%; float:right; }
a:hover {color:white; background-color:blue;}
</style>
</head>
<body>
<nav>
<a href="page1.html">Link1 </a>
<a href="page2.html">Link2 </a>
<a href="page3.html">Link3 </a>
</nav>
</body>
</html>

```

12.2.7 Links - formating 5

In the `<head>` style section, set the link formatting in the correct order.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>HTML</title>
    <style>
        a {font-family:batang;text-decoration:none;}
        //a:visited {color:#00ff00;}
        //a:link {color:#0000ff;}
        //a:active {color:#ffff00;}
        //a:hover {color:#ff0000;}
    </style>
    </head>
    <body>
        <aside>
            <a href="https://www.nytimes.com/"> The New York Times
        </a> <br>
            <a href="https://www.smashingmagazine.com/"> Smashing
            magazine </a><br>
            <a href="https://www.oscars.org/"> Academy of Motion
            Picture Arts and Sciences </a> <br>
        </aside>
    </body>
</html>
```

12.2.8 Link visited

Change the text color of the visited links to green (use color number #00ff00).

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <title>html</title>
    <style>
        a {font-family:batang;text-decoration:none; color:red;}
    </style>
    </head>
```

```

<body>
  <aside>
    <a href="https://www.nytimes.com/"> The New York Times
  </a> <br>
    <a href="https://www.smashingmagazine.com/"> Smashing
  magazine </a><br>
    <a href="https://www.oscars.org/"> Academy of Motion
  Picture Arts and Sciences </a> <br>
  </aside>
</body>
</html>

```

12.3 Lists

12.3.1

Formatting of the text placed in numbered lists, bulleted lists or definition lists is the same as formatting text in a paragraph. The lists text can have its own *background, color, font size, shadow* etc.

The *list-style-type* property can change the bullet or numbering character.

For the **** list, you can choose a numbering method from a dozen or so values, e.g.

- none (disables/removes the mark)
- decimal-leading-zero (01., 02., 03., ... 10.,11., ...)
- lower-alpha (a., b., c., ...)
- lower-roman (i., ii., iii., ...)
- upper-latin (A., B., C., ...)

Greek, Arabic, Japanese letters and many more are also available.

```

ol {
  list-style-type: alpha;
  font-size: 1em;
  color:purple;
}

```

 **12.3.2**

The same property `list-style-type` changes the bullet character in `` lists. The bullet character can only take one of three values:

- *circle*,
- *disc*,
- *square*.

For each of the lists, you can disable the bullet character:

```
ol {list-style-type:none;}
```

Bulleted and numbered lists are very often used to design the navigation menu on pages, due to their structure. In this case, it is mandatory to disable the list mark.

```
ol li a {list-style-type:none;}
```

 **12.3.3**

When preparing a menu for a website, we often put navigation links in the lists.

What value should be entered for the `/list-style-type` property to remove the bullets?

 **12.3.4**

The bullet character can be changed not only with the help of the `list-style-type` property.

Each list can start with any character saved in the graphic file. The `list-style-image` property places any graphic character at the top of the list instead of the standard circle or square.

```
list-style-image: url('plik.gif');
```

-  Monday
-  Tuesday
-  Wednesday
-  Thursday

Make sure that the picture is not too large and does not interfere with the reception of the text.

The position of the image and graphic mark can be changed using `list-style-position`.

```
list-style-position: inside;
```

By default, the image is placed outside the list and so its default value is list-style-position: outside;.

12.3.5

Complete the fragment of the list formatting code , so that the bullet sign is an image ok.png, the list had internal margins of 30px, and the elements had a light grey background.

```
ul {_____ : _____ (ok.png) ; _____ : 30px ; }
ul li {_____ : lightgrey}
```

12.3.6

To help you read and format longer lists, you can use the nth-child pseudo-class selector.

```
ol {width:15%; background-color:#FEFE34; padding:30px ; }
li:nth-child(2n) {background-color:#FCCB1D ; }
li:nth-child(2n+1) {background-color:#FC9803 ; }
```



In the or lists, you can format the elements that you can specify using the formula.

You can format the indicated children-elements, e.g.

- nth-child(5) - only the fifth element of the list,
- nth-child(2n+1) - all odd elements of the list,
- nth-child(2n) lub nth-child(even) - all even elements of the list,
- nth-child(3n+1) - every third element starting from the first.

In a similar way, you can independently invent the principle of formatting child elements of any list.

 12.3.7

Complete the code so that every third element of the list (starting from the second) had a red font.

```
li: _____ {color:red;}
```

12.4 Lists (programs)

 12.4.1 None

Delete the bulletin symbol in the bulletin list. Use internal CSS.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>

            </style>
    </head>
    <body>
<h2>Shopping list</h2>
<ul>
<li>orange juice</li>
<li>3 bananas</li>
<li>5 apples</li>
<li>2 lemons</li>
</ul>
</body>
</html>
```

 12.4.2 List style

In the specified numbered list, change the numbering character to upper-latin. Use internal CSS.

```
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
```

```

        </style>
    </head>
    <body>
<h4>Spring:</h4>
<ol>
<li>March</li>
<li>April</li>
<li>May</li>
</ol>
</body>
</html>

```

12.4.3 Picture as a bulletin symbol

In the list of bullet points, set the image 'ok.png' as a bulletin symbol. Use internal CSS.

```

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <base href="https://priscilla.fitped.eu/images/">
    <style>

        </style>
    </head>
    <body>
<h4> New Year's resolutions:</h4>
<ul>
<li>Smile more often!</li>
<li>To find more time for friends!</li>
<li>Make one dream come true. (monthly)!</li>
<li> Buy a new car! </li>
</ul>
</body>
</html>

```

12.4.4 N-th list element

For even-numbered list elements, change the background color to light blue. Use the nth-child selector for list items (*li:nth-child(rule)*). The rule should specify the elements that we want to format, e.g. $3n$, $2n + 1$, $2n + 3$.

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>HTML</title>
    <style>

      </style>
  </head>
  <body>

<h4>New Year's resolutions:</h4>
<ul>
<li>Smile more often!
<li>To find more time for friends!
<li>Make one dream come true (monthly) !
<li>Buy a new car! </li>
</ul>
</body>
</html>
```

12.4.5 Formating

In the list provided, change the font typeface to Arial and the text color to purple.

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>HTML</title>
    <style>
      li:nth-child(2n) {background-color:lightblue; }

      </style>
  </head>
  <body>

<h4>New Year's resolutions:</h4>
<ul>
<li>Smile more often!
<li>To find more time for friends!
<li>Make one dream come true. (monthly) !
<li>Buy a new car! </li>
</ul>
```

```
</body>  
</html>
```

Property Display

Chapter 13

13.1 Display - flex

13.1.1

Display property *flex* is the latest and revolutionary approach to defining multi-column page templates. Giving the parent a *display:flex* property opens up a whole range of new possibilities in CSS.

The most important benefits:

- centring of page elements horizontally and vertically,
- setting equal spacing between columns, which are responsive,
- many ways to wrap items in a container,
- elastic changes in the height of elements, including stretching of elements vertically.

To start working with *display:flex*, you must place a container in the HTML file, and in it, elements that will be formatted.

```
<main>
<article>Text 1 </article>
<article>Text 2 </article>
<article>Text 3 </article>
<article>Text 4 </article>
<article>Text 5 </article>
<article>Text 6 </article>
</main>
```

In the CSS file:

```
main {display:flex; background-color: #133863; color:#204E83}
main > article { margin: 10px; padding: 10px; font-size: 20px;
background-color: #E1ECF9; }
```



13.1.2

Many commands on how to arrange items in a *flex* container can be defined as container properties.

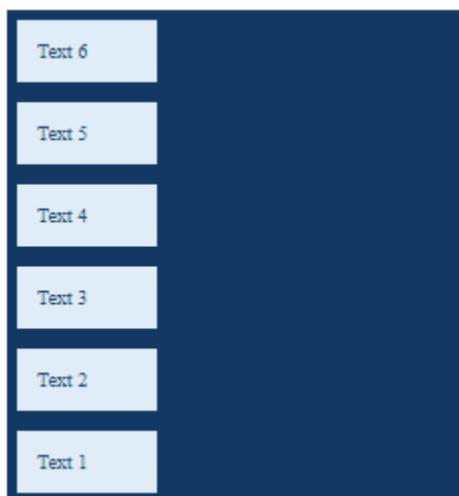
As a standard, elements in the *display:flex* containers are arranged in one line.

The *flex-direction* property allows you to switch to:

- *column* - elements arranged in a column,
- *column-reverse* - elements arranged in a column, in reverse order, the first element at the end,
- *row* - elements arranged in a row (default value),
- *row-reverse* - elements arranged in a row, in reverse order, the first element at the end.

In the CSS file:

```
main {display:flex; flex-direction:column-reverse; background-color: #133863; color:#204E83;}
main > article { margin: 10px; padding: 20px; font-size: 20px; background-color: #E1ECF9; width:100px;}
```



13.1.3

A container with flex elements placed in a row or column does not wrap when changing the size of the window.

To change this, the container should have a *flex-wrap* property.

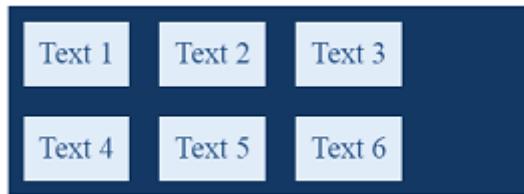
The same page viewed on a mobile device may have a different column or row layout.

Flex-wrap can take one of the values *wrap*, *nowrap* or *wrap-reverse*.

In the CSS file:

```
main {display:flex; flex-wrap:wrap; background-color: #133863; color:#204E83;}
```

```
main > article { margin: 10px; padding: 10px; font-size: 20px;
background-color: #E1ECF9; }
```



When changing the window size, these elements can be placed in one column or in several.

What is the maximum number of columns for this container?

13.1.4

Complete the code so that the elements placed in the flex container can wrap when you change the width of the browser window.

```
div {display: _____; _____ :wrap; }
```

13.1.5

A very big problem for the website designers was to centre or evenly distribute several elements on the page.

The *justify-content* property allows horizontal alignment of elements inside the container.

With the help of space-around, the elements can be evenly distributed horizontally so that they have the same margins (right and left), regardless of the size of the window.

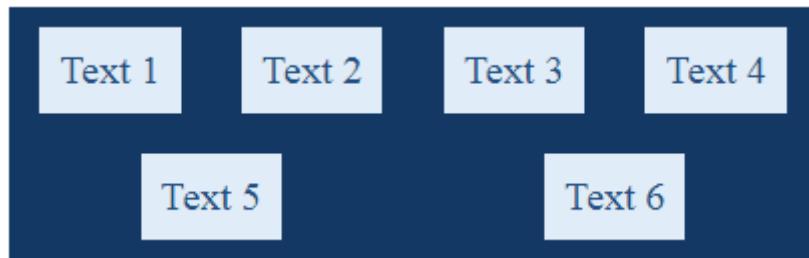
In the CSS file:

```
main {display:flex; justify-content: space-around; background-
color: #133863; color:#204E83; }

main > article { margin: 10px; padding: 10px; font-size: 20px;
background-color: #E1ECF9; }
```



```
main {display:flex; flex-wrap: wrap; justify-content: space-around; background-color: #133863; color:#204E83;}
```



13.1.6

The arrangement of elements in an external container can be created using *justify-content: space-between*. In this command, the first and last elements are aligned to the left and right of the container, and the remaining part of the space is divided evenly between the elements.

```
main {display:flex; justify-content: space-between;
background-color: #133863; color:#204E83;}
main > article { margin: 10px; padding: 10px; font-size: 20px;
background-color: #E1ECF9;}
```



13.1.7

Complete the code so that the elements placed in the flex container can be evenly distributed on the page, and the outlines aligned to the left and right margins.

```
div {display: ____; ____:space-around; }
```

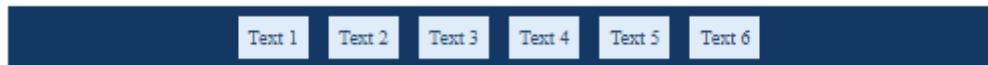
13.1.8

A big problem in the design of the pages was the exact centring of the element on the page.

An element placed in a flexible container can be centred using *justify-content: center*.

In the CSS file:

```
main {display:flex; justify-content:center; background-color:#133863; color:#204E83}
main > article { margin: 10px; padding: 10px; font-size: 20px; background-color: #E1ECF9; }
```



Several properties have been defined for a container with the *display: flex* property. They concern the entire container.

To format the horizontal arrangement of elements, you can use:

- *flex-direction* - arranges elements in a row or column,
- *flex-wrap* - allows you to wrap (or not) elements,
- *justify-content* - aligns the elements in the container horizontally.

13.1.9

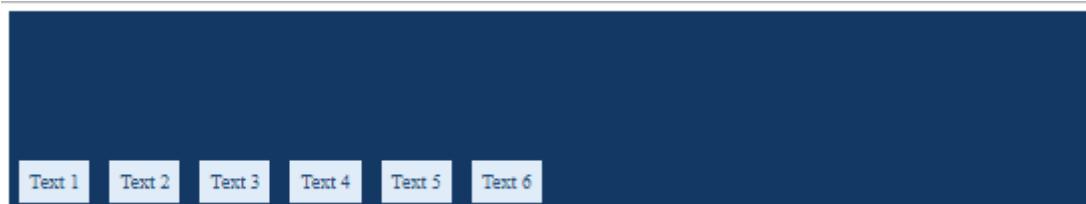
To format the vertical arrangement of elements, you can use:

- *align-items* - aligns the elements vertically or extends vertically,
- *align-content* - specifies the vertical alignment of items when the container has wrapping properties.

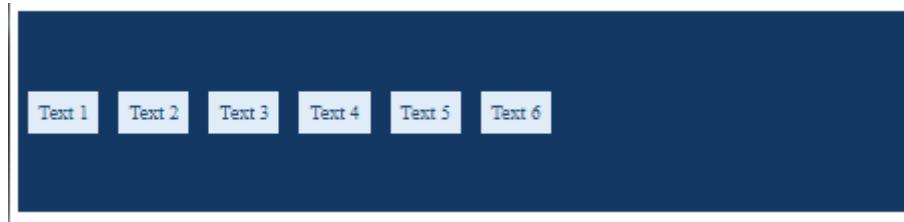
The *align-items* property allows you to align elements to the top of the container, to the bottom or to the inside.

In the CSS file:

```
main {display:flex; align-items: flex-end; background-color:#133863; color:#204E83}
main > article { margin: 10px; padding: 10px; font-size: 20px; background-color: #E1ECF9; }
```



```
main {display:flex; align-items: center; background-color:#133863; color:#204E83}
main > article { margin: 10px; padding: 10px; font-size: 20px; background-color: #E1ECF9; }
```



13.1.10

Complete the code so that the element placed in the flex container is centred horizontally and vertically.

```
div {____:flex; ____:center; ____: center;}
```

13.1.11

Property: *align-items: stretch;* allows you to give the same height to all elements placed in the container with the *flex* property.

```
main {display:flex; align-items: stretch; background-color: #133863; color:#204E83;}
main > article { margin: 10px; padding: 10px; font-size: 20px; background-color: #E1ECF9;}
```



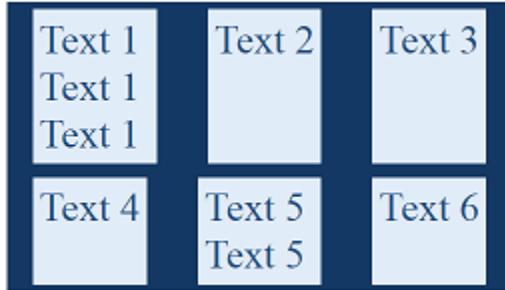
13.1.12

The *align-content* property is used when the *display: flex* container has wrapping properties, that is *flex-wrap: wrap*;

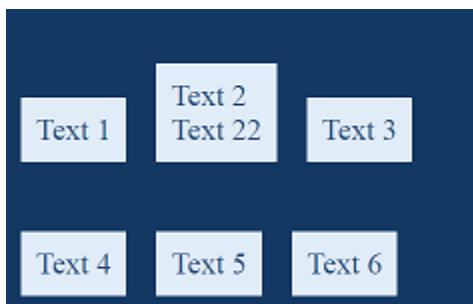
How can wrapped elements be vertical aligned?

```
main {
display:flex;
flex-wrap:wrap;
justify-content: space-around;
align-content:stretch;
background-color: #133863; color:#204E83;}
```

```
main > article {
margin: 10px;
padding: 10px;
font-size: 20px;
background-color: #E1ECF9;}
```



```
main {
display:flex;
align-items: flex-end;
flex-wrap:wrap;
background-color: #133863;
color:#204E83;
height:200px;}
main > article {
margin: 10px;
padding: 10px;
font-size: 20px;
background-color: #E1ECF9;}
```



13.1.13

If the container has a `display: flex` and `flex-wrap: wrap;` property, to define the vertical alignment of the wrapped elements, use the property:

- `align-content`

- align-items
- justify-content
- flex-vertical

13.1.14

The properties described in previous lessons applied to all child elements in the flex container. However, individual elements can be individually formatted.

Property *flex-grow*- allows you to specify which of the elements and how many times will be enlarged when the size of the browser window changes.

Eg. the third element will increase 4 times faster than the other elements:

```
<main>
  <article style="flex-grow: 1">Text 1</div>
  <article style="flex-grow: 1">Text 2</div>
  <article style="flex-grow: 4">Text 3</div>
</main>
```

Flex-shrink- determines which of the elements and how quickly it will decrease.

```
<main>
  <article style="flex-shrink: 6">Text 1</div>
  <article style="flex-shrink: 1">Text 2</div>
  <article style="flex-shrink: 1">Text 3</div>
</main>
```

Here, the first element will decrease the fastest.

The actions of these properties are best seen when the size of the browser window changes.

13.1.15

In addition to the *flex-grow* and *flex-shrink* properties, the specification specifies several other options for formatting individual elements in the flex container.

- *order*- determines the order in which the element is displayed (it may be different from the one saved in the file),
- *flex-basis*- allows you to set the initial size for the element,
- *align-self*- allows a different way of aligning a given element, in relation to the others (it overwrites the values for align-items).

Due to the limited volume of the course, it is worth checking the operation of the listed properties yourself.

13.2 Display - flex (programs)

13.2.1 Flex container

Give the property of the *flex* container for the `<main>` tag so that its children can be properly formatted.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS course</title>
    <style>
        main { justify-content:center; align-items:stretch; background-color: #133863; color:#204E83}
        main > article { margin:10px; padding:10px; font-size:20px; background-color:#E1ECF9;}
    </style>
</head>
<body>
<main>
<article>Text 1 <br>Text 1 <br>Text 1<br> Text 1<br>
</article>
<article>Text 2 <br>Text 2 </article>
<article>Text 3 </article>
<article>Text 4 </article>
<article>Text 5 </article>
<article>Text 6 </article>
</main>
</body>
</html>
```

13.2.2 Flex - direction

For the `<main>` tag, enter the formatting so that his children are arranged in a column. Use *flex-direction*.

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS course</title>
    <style>
        main {display:flex; background-color: #133863;
color:#204E83;}
        main > article { margin:10px; padding:10px; font-
size:20px; background-color:#E1ECF9; }
    </style>
</head>
<body>
<main>
<article>Text 1 </article>
<article>Text 2 </article>
<article>Text 3 </article>
<article>Text 4 </article>
<article>Text 5 </article>
<article>Text 6 </article>
</main>
</body>
</html>

```

13.2.3 Flex - justify-content

For the `<main>` tag, which has the *flex* property, enter the formatting so that its children are placed horizontally in the middle of the page. Use *justify-content*.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CSS course</title>
    <style>
        main {display:flex; background-color: #133863;
color:#204E83;}
        main > article { margin:10px; padding:10px; font-
size:20px; background-color:#E1ECF9; width: 10%; }
    </style>
</head>
<body>
<main>
<article>Text 1 </article>
<article>Text 2 </article>

```

```

<article>Text 3 </article>
<article>Text 4 </article>
<article>Text 5 </article>
<article>Text 6 </article>
</main>
</body>
</html>

```

13.2.4 Flex - justify-content - img

Centre the horizontal image on the page. Use the *justify-content* property.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
    <base href="https://priscilla.fitped.eu/images/">
<style>
body{background-color: #6A98C8; }

</style>
</head>
<body>
<figure>

</figure>
</body>
</html>

```

13.2.5 Flex - centre horizontally and vertically

Align the picture to the centre horizontally and vertically. Use *align-items* property.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">

```

```
<base href="https://priscilla.fitped.eu/images/">
<style>
body{background-color: #6A98C8;}
figure{height:100vh;display:flex; justify-content:center;}
</style>
</head>
<body>
<figure>

</figure>
</body>
</html>
```

13.2.6 Flex - top right

Align the `` list items to the top right corner of the `<nav>` area. Use the *justify-content* and *align-items* property.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            nav {width:100vw; height:10vh; background-color:lightgrey;}
            ul{list-style-type:none; display:flex; justify-content:flex-end; align-items:center}
            li{margin:1px;}
            a {text-decoration:none; background-color:lightgreen; text-align:center; display:block; width:70px; padding:5px;}
            a:hover {background-color: lightgrey;}
        </style>
    </head>
    <body>
        <nav>
            <ul>
                <li> <a href="file1.html"> File 1</a> </li>
                <li> <a href="file2.html"> File 2</a> </li>
                <li> <a href="file3.html"> File 3</a> </li>
            </ul>
        </nav>
    </body>
</html>
```

13.2.7 Flex - wrap

Enter the property that will wrap the `` list items. Use *flex-wrap*.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
    <style>
        nav {width:100vw; background-color:lightgrey;}
        ul{list-style-type:none;display:flex; justify-content:flex-end; align-items:flex-start; }
        li{margin:1px;}
        a {text-decoration:none;background-color:lightgreen;
text-align:center; display:block; width:120px; padding:5px;}
        a:hover {background-color: lightgrey;}
    </style>
    </head>
    <body>
        <nav>
            <ul>
                <li> <a href="file1.html"> File 1</a> </li>
                <li> <a href="file2.html"> File 2</a> </li>
                <li> <a href="file3.html"> File 3</a> </li>
                <li> <a href="file4.html"> File 4</a> </li>
                <li> <a href="file5.html"> File 5</a> </li>
                <li> <a href="file6.html"> File 6</a> </li>
            </ul>
        </nav>
    </body>
</html>
```

13.2.8 Flex - space-around

For the `` tag, which has the *flex* property, enter the formatting so that its children have the same external margins horizontally. Use the appropriate value for *justify-content*.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
```

```

<style>

    nav {background-color:lightgrey;}
    ul{margin:0px;padding:0px;list-style-type:none;
display:flex;      }
    li{margin:1px;background-color:lightgreen;  }
        a {text-decoration:none; text-align:center;
display:block; width:70px; padding:5px; }
        a:hover {background-color: lightgrey; }

</style>
</head>
<body>
<nav>
<ul>
    <li> <a href="file1.html"> File 1</a> </li>
    <li> <a href="file2.html"> File 2</a> </li>
    <li> <a href="file3.html"> File 3</a> </li>
    <li> <a href="file4.html"> File 4</a> </li>
    <li> <a href="file5.html"> File 5</a> </li>
    <li> <a href="file6.html"> File 6</a> </li>
</ul>
</nav>
</body>
</html>

```

13.2.9 Flex - space between

For the `` tag, which has the `flex` property, enter the formatting so that its children are equally spaced horizontally and the first and last elements of the list are aligned to the appropriate margins (left and right). Use *justify-content*.

```

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
    <style>
        nav {background-color:lightgrey;}
        ul{margin:0px;padding:0px;list-style-type:none;
display:flex;      }
        li{margin:1px;background-color:lightgreen;  }
            a {text-decoration:none; text-align:center;
display:block; width:70px; padding:5px; }
            a:hover {background-color: lightgrey; }

        </style>

```

```

</head>
<body>
<nav>
<ul>
    <li> <a href="file1.html"> File 1</a> </li>
    <li> <a href="file2.html"> File 2</a> </li>
    <li> <a href="file3.html"> File 3</a> </li>
    <li> <a href="file4.html"> File 4</a> </li>
    <li> <a href="file5.html"> File 5</a> </li>
    <li> <a href="file6.html"> File 6</a> </li>
</ul>
</nav>
</body>
</html>

```

13.2.10 Flex - space-between and wrap

For a defined `` list, enter the wrapping of its elements. Use the `flex-wrap` property.

Check the operation of the command by changing the width of the browser window.

```

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>

            nav {background-color:lightgrey;}
            ul{margin:0px;padding:0px;list-style-type:none;
display:flex; justify-content:space-between;    }
            li{margin:1px;background-color:lightgreen; }
            a {text-decoration:none; text-align:center;
display:block; width:120px; padding:5px;}
            a:hover {background-color: lightgrey; }

        </style>
    </head>
    <body>
        <nav>
            <ul>
                <li> <a href="file1.html"> File 1</a> </li>
                <li> <a href="file2.html"> File 2</a> </li>
                <li> <a href="file3.html"> File 3</a> </li>
                <li> <a href="file4.html"> File 4</a> </li>

```

```

<li> <a href="file5.html"> File 5</a> </li>
<li> <a href="file6.html"> File 6</a> </li>
</ul>
</nav>
</body>
</html>

```

13.2.11 Align-items:flex-end (columns)

Format the `<article>`tags so that they are aligned to the bottom edge of the container. Use `align-items` property.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
body{box-sizing:border-box;}
section {min-height:40vh;background-color: #6A98C8;
display:flex;justify-content:space-between; }
article{background-color: white;width:33.33%; border:1px
solid black; padding:5px; }
</style>
</head>
<body>
<section>
<article>
<h2>Iceland 1 </h2>
<p>Iceland lies on the border between the Arctic Ocean and the
Atlantic Ocean. The main island is located south of the Arctic
Circle.</p>
</article>
<article>
<h2>Iceland 2 </h2>
<p>In geological terms, Iceland is the youngest area on the
European continent. Located on the 'hot spot' of the Mid-
Atlantic Ridge, it has many, including active, volcanoes,
including Hekla, Katla, Askja, Grimsvötn, Hvannadalshnúkur
(the highest peak of the country). Volcanic activity is also
evidenced by numerous hot springs and geysers (the word itself
is of Icelandic origin).</p>

```

```
</article>
<article>
<h2>Iceland 3 </h2>
<p>In numerous bays and fjords there are smaller or larger settlements. The main town of the country is its capital Reykjavík, with 119.1 thousand inhabitants, i.e. almost 2/5 of the country's population. </p>
</article>
</section>
</body>
</html>
```

13.2.12 Align-items:center (columns)

Format the `<article>`tags so that they are aligned vertically to the centre of the container. Use `align-items` property.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body{box-sizing:border-box;}
section {min-height:40vh;background-color: #6A98C8;
display:flex;justify-content:space-between;}
article{background-color: white;width:33.33%; border:1px solid black; padding:5px;}
</style>
</head>
<body>
<section>
<article>
<h2>Iceland 1 </h2>
<p>Iceland lies on the border between the Arctic Ocean and the Atlantic Ocean. The main island is located south of the Arctic Circle.</p>
</article>
<article>
<h2>Iceland 2 </h2>
<p>In geological terms, Iceland is the youngest area on the European continent. Located on the 'hot spot' of the Mid-
```

```

Atlantic Ridge, it has many, including active, volcanoes,
including Hekla, Katla, Askja, Grimsvötn, Hvannadalshnúkur
(the highest peak of the country). Volcanic activity is also
evidenced by numerous hot springs and geysers (the word itself
is of Icelandic origin).</p>
</article>
<article>
<h2>Iceland 3 </h2>
<p>In numerous bays and fjords there are smaller or larger
settlements. The main town of the country is its capital
Reykjavík, with 119.1 thousand inhabitants, i.e. almost 2/5 of
the country's population. </p>
</article>
</section></body>
</html>

```

13.2.13 Align-items:stretch (columns)

Check the effect of the *align-items: stretch* statement. Apply it to *<article>* tags. The columns should be stretched to the entire height of the container (here *<section>* tag).

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
body{box-sizing:border-box;}
section {min-height:40vh;background-color: #6A98C8;
display:flex;justify-content:space-between; }
article{background-color: white;width:33.33%; border:1px
solid black; padding:5px;margin:5px;}
</style>
</head>
<body>
<section>
<article>
<h2>Iceland 1 </h2>
<p>Iceland lies on the border between the Arctic Ocean and the
Atlantic Ocean. The main island is located south of the Arctic
Circle.</p>

```

```

</article>
<article>
<h2>Iceland 2 </h2>
<p>In geological terms, Iceland is the youngest area on the European continent. Located on the 'hot spot' of the Mid-Atlantic Ridge, it has many, including active, volcanoes, including Hekla, Katla, Askja, Grimsvötn, Hvannadalshnúkur (the highest peak of the country). Volcanic activity is also evidenced by numerous hot springs and geysers (the word itself is of Icelandic origin).</p>
</article>
<article>
<h2>Iceland 3 </h2>
<p>In numerous bays and fjords there are smaller or larger settlements. The main town of the country is its capital Reykjavík, with 119.1 thousand inhabitants, i.e. almost 2/5 of the country's population. </p>
</article>
</section>
</body>
</html>

```

13.2.14 Align-self

Use the *align-self* property (*flex-start*) so that the first and third columns are aligned to the top edge of the container and the second column - to the bottom edge. Complete the formatting of the prepared classes and invoke the appropriate classes in *<article>* tags.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body{box-sizing:border-box;}
section {min-height:40vh;background-color: #6A98C8;
display:flex;justify-content:space-between;}
article{background-color: white;width:33.33%; border:1px solid black; padding:5px; margin:5px;}
article.top{ }
article.bot{ }

```

```

</style>
</head>
<body>
<section>
<article>
<h2>Iceland 1 </h2>
<p>Iceland lies on the border between the Arctic Ocean and the Atlantic Ocean. The main island is located south of the Arctic Circle.</p>
</article>
<article>
<h2>Iceland 2 </h2>
<p>In geological terms, Iceland is the youngest area on the European continent. Located on the 'hot spot' of the Mid-Atlantic Ridge, it has many, including active, volcanoes, including Hekla, Katla, Askja, Grimsvötn, Hvannadalshnúkur (the highest peak of the country). Volcanic activity is also evidenced by numerous hot springs and geysers (the word itself is of Icelandic origin).</p>
</article>
<article>
<h2>Iceland 3 </h2>
<p>In numerous bays and fjords there are smaller or larger settlements. The main town of the country is its capital Reykjavík, with 119.1 thousand inhabitants, i.e. almost 2/5 of the country's population. </p>
</article>
</section></body>
</html>

```

13.2.15 Flex-shrink

The *flex-shrink* property determines how quickly an item should decrease when changing the display area dimensions. A value of 1 means that the element shrinks as the others, 0 - shrinks the least of the remaining elements.

Define the formatting in *shr* class and set the *flex-shrink* property to 0.5 and see how the item changes when changing the viewer size. Call the class in the last *<article>* tag.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>

```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body{box-sizing:border-box;}
section {min-height:40vh;background-color: #6A98C8;
display:flex;justify-content:space-between;}
article{background-color: white; border:1px solid black;
padding:5px;margin:5px;}
.shr {  }

</style>
</head>
<body>
<section>
<article>
<h2>Iceland 1 </h2>
<p>Iceland lies on the border between the Arctic Ocean and the Atlantic Ocean. The main island is located south of the Arctic Circle.</p>
</article>
<article>
<h2>Iceland 2 </h2>
<p>In geological terms, Iceland is the youngest area on the European continent. Located on the 'hot spot' of the Mid-Atlantic Ridge, it has many, including active, volcanoes, including Hekla, Katla, Askja, Grimsvötn, Hvannadalshnúkur (the highest peak of the country). Volcanic activity is also evidenced by numerous hot springs and geysers (the word itself is of Icelandic origin).</p>
</article>
<article>
<h2>Iceland 3 </h2>
<p>In numerous bays and fjords there are smaller or larger settlements. The main town of the country is its capital Reykjavík, with 119.1 thousand inhabitants, i.e. almost 2/5 of the country's population. </p>
</article>
</section>
</body>
</html>
```

13.3 Display and visibility

13.3.1

Many tags in HTML are *linear* by nature, meaning that they are not forced to go to a new line on the page, usually it is not possible to give them width and height. Such tags include, among others: `<a>`, ``, ``, `<cite>`, ``, `<button>`, `<input>`, `<select>` and many others.

These tags can not contain block markers.

However there are many **block tags** in HTML. The data contained in them always starts with a new line. These tags can be given width and height. Block tags include: `<p>`, `<section>`, `<article>`, `<div>`, `<figure>`, `<h1>`, `` and many more.

The CSS specification allows you to change the nature of the displayed element.

```
a {display:block; width:50px;}
```

The above sentence will cause the links to be displayed in the new lines (like block markers) and will have a width of 50px.

```
figure {display:inline-block; width:200px; height:400px;}
```

The `<figure>` tag is a block tag. When preparing a gallery, it's a good idea to put the images side by side in a row. Using the `display: inline;` property will cause the elements to be arranged in a row, but they will lose the ability to change dimensions. The `display: inline-block;` combines two in one. The elements are blocky, but they are placed in a row.

13.3.2

For the `` tag to change from linear to block (starting with a new line), the display property should be:

- block
- inline
- block-inline
- none

13.3.3

The `display: none;` property also lets you hide items. It is used on pages to create animated elements.

Very often, with the help of this property, a dropdown menu is created, the selected elements of which are being hidden until shown. The HTML menu structure is usually based on lists.

```
<ul>
<li>Link1
<ul>
<li><a href="s11.html">Link11</a></li>
<li><a href="s12.html">Link12</a></li>
<li><a href="s13.html">Link13</a></li>
</ul>
</li>
<li>Link2
<ul>...
```

In CSS, you just have to hide or show the appropriate elements when the user indicates the item with the mouse.

```
ul li ul {display : none;}
ul li:hover ul {display : block;}
```

The "`ul li ul`" indicates the unordered list `ul` (last) that is the child of a `li` (middle) element that is a child of a `ul` (first) element. This `ul` list will not be visible, because of `display: none;` property.

If the element `li` of the external list is indicated by a mouse, then the internal list has the property `display: block;` and then it is visible.

It's good to know that the descendants (child elements) of the `display: none;` element will **also** be invisible.

13.3.4

The `Display` property is also used to create games. Many animations in `JavaScript` require the use of this property.

```
<div onmouseover="f1()" onmouseout="f2()" > Show article</div>
<article id="art1" >
"Friends are like silent angels who raise us when our wings
forget how to fly."
```

```

<b>Antoine de Saint-Exupéry</b>
</article>
<script>
function f1() {
document.getElementById("art1").style.display = "block";
}
function f2() {
document.getElementById("art1").style.display = "none";
}
</script>

```

A mouse pointer to the text in the `<div>` tag calls the function `f1()`, which causes the tag with `id = "art1"` to be displayed. If the mouse cursor is outside the `<div>` text, the function `f2()` will be called, which will hide the contents of the tag with `id = "art1"`.

13.3.5

Complete the code so that all links inside the `<nav>` tag can be 100px wide and 10px internal margins.

```
nav a {____:block; ____:100px; ____:10px; }
```

13.3.6

The alternative for `display: none;` is `visibility: hidden;`

`Visibility` property, just like the `display` property, allows you to hide an element, but leaves an empty space in the document, fitting the size of the hidden element. The element occupies a place in the document, but it is not visible.

Otherwise, the child elements of the container are displayed, which has the property `visibility: hidden;`

```
img#new {visibility:hidden;}
```

If the descendant has the property `visibility: visible;` it is displayed on the page, despite `hiding` the ancestor.

13.4 Display and visibility (programs)

■ 13.4.1 Display block

Format the `<a>` tag to become a block tag (*display:block*). Give it a width of 80 px and insert a 5px wide inner margin (padding) on each side. Use a collective padding notation.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>FITPED</title>
<style>
a { background-color: lightgrey; border:1px solid black;
width:100px; }
</style>
</head>
<body>
<a href="#">Link 11</a>
<a href="#">Link 12</a>
<a href="#">Link 13</a>
</body>
</html>
```

■ 13.4.2 Display inline-block

Format the `<a>` tag so that it becomes a linear and block tag, i.e. so that it can be given width and height and at the same time placed like a line marker (*display:inline-block*).

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>FITPED</title>
<style>
a { background-color: lightgrey; border:1px solid
black; width:100px; padding:5px; }
</style>
</head>
<body>
<a href="#">Link 11</a>
<a href="#">Link 12</a>
```

```
<a href="#">Link 13</a>
</body>
</html>
```

13.4.3 Display - none

Hide the page header (`<header>`) in this section of the page. Use the *display* property.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<style>
header,nav,section,footer {background-color:#cccd9ff; border:1px solid black; box-sizing:border-box;}
header {height:20vh; }
nav {height:10vh; }
section { width:100%; height:60vh; }
footer {height:10vh; clear:both; }
</style>
</head>
<body>
<header>
<h1>Page title 1</h1>
<h2>Page title 2</h2>
</header>
<nav> <a href="#">Link 11</a>
<a href="#">Link 12</a>
<a href="#">Link 13</a>
</nav>
<section><h3>Section</h3>
</section>
<footer>
  Footer
</footer>
</body>
</html>
```

13.4.4 Display - hover

Complete the given section of the page so that the `<nav>` tag will only be displayed if `<header>` is pointed with the mouse, otherwise, it will not be displayed.

Use `display: block` and `display: none`.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<style>
header,nav {background-color:#ccd9ff; border:1px solid black; box-sizing:border-box;}
nav { }
header:hover nav { height:5vh; }
</style>
</head>
<body>
<header>
<h1>Point me with the mouse!</h1>
<nav>
<a href="#">Link 11</a>
<a href="#">Link 12</a>
<a href="#">Link 13</a>
</nav>
</header>
</body>
</html>
```

13.4.5 Display - figure

Format the `<figure>` tag so that the images are arranged in a row. Use the `display` property.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <base href="https://priscilla.fitped.eu/images/">
    </head>
    <style>
```

```

        figure { width:100px; }
        img {width:100px;}
        figcaption {background-color:lightgreen;}
        </style>
    </head>
    <body>
<h2> Holiday </h2>
<figure>

<figcaption>
Photo 1. Boats
</figcaption>
</figure>
<figure>

<figcaption>
Photo 2. Sea
</figcaption>
</figure>
<figure>

<figcaption>
Photo 3. Boats
</figcaption>
</figure>
<figure>

<figcaption>
Photo 4. Sea
</figcaption>
</figure>
    </body>
</html>

```

13.4.6 Visibility hidden

Complete the formatting for **<header>** and **<section>** tags so that they are hidden, but the area they occupy remains visible. Use *visibility:hidden* property.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>

```

```
<style>
body {background-color:lightgreen;}
nav,footer {background-color:#cccd9ff; border:1px solid black; box-sizing:border-box;}
header, section {background-color:lightgrey;}
nav {height:10vh; }
section { width:100%; height:60vh; }
footer {height:10vh; }
</style>
</head>
<body>
<header>
<h1>Page title 1</h1>
<h2>Page title 2</h2>
</header>
<nav> <a href="#">Link 11</a>
<a href="#">Link 12</a>
<a href="#">Link 13</a>
</nav>
<section><h3>Section</h3>
</section>
<footer>
  Footer
</footer>
</body>
</html>
```

Property Position

Chapter 14

14.1 Position

14.1.1

A very interesting way of placing elements on the page gives you a property *position*. This property has nothing to do with the positioning of pages in search engines, although the name itself can be misleading.

The *position* property allows you to place any element anywhere on the page. The effects obtained thanks to this property are surprising. Elements can be arranged like layers and cover each other. You can change the order of the layers you have arranged. Elements can slide along with the scroll bar of the browser window or be attached to a specific place. It is a very useful property.

Position can take one of the following values:

- *static*,
- *relative*,
- *absolute*,
- *fixed*,
- *sticky*.

To define the positioning of an element:

1. determine the type of positioning,
2. determine the position of the element using two values: *top*, *right*, *bottom* or *left*.

Due to the different behavior of elements, while changing the browser's window, it is worth checking the operation of these elements online.

14.1.2

The *position* property disturbs the linear order of placing elements on the page.

An element positioned with *fixed* property is "glued" to a specific place in the browser window. The element stays in the same place all the time, even when the page is scrolling.

With the help of this property, a menu is often designed on a page that is "glued" most often to the top edge of the browser window.

Position: fixed; is also used to insert advertisements, banners and important information on www.

The position of the element must be determined in relation to the two adjacent edges of the window: *top, left, right, bottom*.

```
nav {position: fixed; top:0px; left:0px;}
```

The top of the `<nav>` element will be placed in the top left corner of the browser window. It will remain visible even when the page is scrolled.

14.1.3

Position of an object defined by `display: flex`, sets the element relative to:

- browser windows
- container
- parent element

14.1.4

An **absolute positioned element** must be placed inside another element having `position` property.

The position of this element is determined relative to the parent element (container).

```
footer {position: fixed; bottom:10px; right:10px;}
footer div{position:absolute; bottom:10px; right:10px;}
```

Absolute positioning allows you to set an element anywhere in the container. Items placed below it will be moved to fill the space left by him.

Absolute positioning of an element thus ***influences other elements***.

14.1.5

Select the appropriate terms for the absolute positioned element.

- The element is positioned relative to the parent element.
- The element can affect the position of other elements.
- It must be placed in the relative position element.
- It must be placed in a sticky positioned element.
- The element does not affect the position of other elements.
- The element is always positioned relative to the browser window.

 **14.1.6**

The *position: relative;* property sets the new position of the element relative to its normal place, in the exact order specified in the *.html file.

The relatively positioned element ***does not affect*** other elements inserted after it. The space it occupies becomes empty - as if it was still reserved for this element.

Relative and absolute positioning values are often mistaken. The main difference in their behavior is the parent element, of which the elements are positioned:

- *position: absolute* - positioned relative to the container;
- *position: relative* - positioned relative to its normal position.

 **14.1.7**

Select the appropriate terms for the relatively positioned element.

- The element is positioned relative to the parent element.
- The element can affect the position of other elements.
- The element does not affect the position of other elements.
- It must be placed in an absolute positioned element.
- The element is always positioned relative to the browser window.
- The element is positioned relative to its position resulting from the normal course of the document.

 **14.1.8**

Position *property: sticky;* benefits from relative and fixed positioning features.

The element positioned in this way is placed in its normal place until the user scrolls the page to the fixed position. Then this element "sticks" to a new place for some time. Repeated rewinding can bring it back to the previous position.

```
div {position: sticky; top: 0px; background-color: grey;}
```

On many pages, you can see menus, ads or dialogue boxes that change their location as the user starts scrolling the page.

This property is not yet fully supported by the latest browsers.

 **14.1.9**

Elements formatted with the *position* property can overlap and cover. The *z-index* property can change the order in which element layers are arranged.

```
nav {position:fixed; right:0px; top:0px; z-index:4;}
```

The *z-index* property can take integer values (eg: 2, 0, 3, 7, etc.).

The higher the *z-index* value for an element, the higher it is placed in the page layer, i.e. more visible.

This property can be used only for elements that have set position property.

14.1.10

Which of the defined elements will be placed in the highest (most visible) layer:

- div { position: absolute; left: 0px; top: 0px; z-index: -1; }
- section { position: absolute; left: 0px; top: 0px; z-index: 0; }
- aside { position: absolute; left: 0px; top: 0px; z-index: 2; }

14.2 Position (programs)

14.2.1 Fixed

Using the *position:fixed* property, stick the menu (*<nav>* tag) to the top right of the page.

Use the scroll bar to see the effect of this property.

```
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
    <style>
        nav {}
        ul{list-style-type:none; }
        li {float:right;}
        a {text-decoration:none;background-color:lightgreen;
text-align:center; display:block; width:70px; padding:5px;}
        a:hover {background-color: lightgrey;}
        section{clear:right;}
    </style>
```

```

</head>
<body>
<nav>
<ul>
<li> <a href="file1.html"> File 1</a> </li>
<li> <a href="file2.html"> File 2</a> </li>
<li> <a href="file3.html"> File 3</a> </li>
</ul>
</nav>
<section>
<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing
    elit. Vivamus rhoncus odio enim, eget pretium tellus gravida
    sed. Nulla egestas velit lectus, vitae consequat augue posuere
    ac. Suspendisse a metus pharetra, consequat mauris ac,
    facilisis ex. Vestibulum porttitor quam eu auctor laoreet.
    Cras ut malesuada felis, eu elementum enim. Ut leo nulla,
    mollis congue vehicula id, molestie eu neque. Etiam ac varius
    augue, in auctor lectus. Vivamus dapibus risus vel neque
    finibus, quis tincidunt enim varius. Phasellus quis placerat
    urna. Integer vel ornare dolor. Aenean a massa sed odio
    vestibulum iaculis. Nullam finibus volutpat metus, a ultricies
    sem dignissim vel.
</p>
<p>Ut in imperdiet neque. Ut scelerisque lacinia nisl sit amet
ullamcorper. Donec auctor vitae nunc quis elementum. Morbi et
posuere tellus. Donec eu consectetur odio, vel consectetur
lectus. Fusce a quam vitae sapien varius molestie ac pretium
orci. Donec ut ipsum at enim ornare faucibus sed vitae arcu.
Vivamus vel orci luctus, mattis augue vitae, placerat mauris.
Vivamus porttitor convallis orci.
</p>  </section>
</body>
</html>

```

14.2.2 Fixed (logo)

Use the *position:fixed* property to stick the tag *<figure>* to the top left-hand corner of the page.

Use the vertical scroll bar to see the effect.

```

<!DOCTYPE html>
<html lang="en">

```

```

<head>
    <meta charset="UTF-8">
    <title>HTML</title>
    <base href="https://priscilla.fitped.eu/images/">
    <style>
        nav {position:fixed; top:0;right:0;}
        a {text-decoration:none;background-color:lightgreen;
text-align:center; display:block; width:70px; padding:5px;}
        a:hover {background-color: lightgrey;}
        ul{list-style-type:none; }
        li {float:right;}
        section{clear:right; margin:50px;}
        figure{width:50px;}
    </style>
</head>
<body>
<nav>
<ul>
    <li> <a href="file1.html"> File 1</a> </li>
    <li> <a href="file2.html"> File 2</a> </li>
    <li> <a href="file3.html"> File 3</a> </li>
</ul>
</nav>
<figure>

</figure>
<section>
<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing
    elit. Vivamus rhoncus odio enim, eget pretium tellus gravida
    sed. Nulla egestas velit lectus, vitae consequat augue posuere
    ac. Suspendisse a metus pharetra, consequat mauris ac,
    facilisis ex. Vestibulum porttitor quam eu auctor laoreet.
    Cras ut malesuada felis, eu elementum enim. Ut leo nulla,
    mollis congue vehicula id, molestie eu neque. Etiam ac varius
    augue, in auctor lectus. Vivamus dapibus risus vel neque
    finibus, quis tincidunt enim varius. Phasellus quis placerat
    urna. Integer vel ornare dolor. Aenean a massa sed odio
    vestibulum iaculis. Nullam finibus volutpat metus, a ultricies
    sem dignissim vel.
</p>
<p>Ut in imperdiet neque. Ut scelerisque lacinia nisl sit amet
    ullamcorper. Donec auctor vitae nunc quis elementum. Morbi et
    posuere tellus. Donec eu consectetur odio, vel consectetur

```

```

lectus. Fusce a quam vitae sapien varius molestie ac pretium
orci. Donec ut ipsum at enim ornare faucibus sed vitae arcu.
Vivamus vel orci luctus, mattis augue vitae, placerat mauris.
Vivamus porttitor convallis orci.
</p>  </section>
</body>
</html>

```

14.2.3 Fixed (cookie)

Stick the information about the use of cookies on the website (in the `<article>` tag) to the left bottom of the page.

Use the vertical scroll bar to check the effect.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <base href="https://priscilla.fitped.eu/images/">
        <style>
            nav {position:fixed; top:0;right:0;}
            a {text-decoration:none;background-color:lightgreen;
text-align:center; display:block; width:70px; padding:5px;}
            a:hover {background-color: lightgrey;}
            ul{list-style-type:none; }
            li {float:right;}
            section{clear:right;}
            figure{width:100px; position:fixed; left:0px; top:0px;
margin:10px;}
            article{background-color: lightgrey; }
        </style>
    </head>
    <body>
        <nav>
            <ul>
                <li> <a href="file1.html"> File 1</a> </li>
                <li> <a href="file2.html"> File 2</a> </li>
                <li> <a href="file3.html"> File 3</a> </li>
            </ul>
        </nav>
        <figure>
            

```

```

</figure>
<section>
<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing
    elit. Vivamus rhoncus odio enim, eget pretium tellus gravida
    sed. Nulla egestas velit lectus, vitae consequat augue posuere
    ac. Suspendisse a metus pharetra, consequat mauris ac,
    facilisis ex. Vestibulum porttitor quam eu auctor laoreet.
    Cras ut malesuada felis, eu elementum enim. Ut leo nulla,
    mollis congue vehicula id, molestie eu neque. Etiam ac varius
    augue, in auctor lectus. Vivamus dapibus risus vel neque
    finibus, quis tincidunt enim varius. Phasellus quis placerat
    urna. Integer vel ornare dolor. Aenean a massa sed odio
    vestibulum iaculis. Nullam finibus volutpat metus, a ultricies
    sem dignissim vel.
</p>
<p>Ut in imperdiet neque. Ut scelerisque lacinia nisl sit amet
ullamcorper. Donec auctor vitae nunc quis elementum. Morbi et
posuere tellus. Donec eu consectetur odio, vel consectetur
lectus. Fusce a quam vitae sapien varius molestie ac pretium
orci. Donec ut ipsum at enim ornare faucibus sed vitae arcu.
Vivamus vel orci luctus, mattis augue vitae, placerat mauris.
Vivamus porttitor convallis orci.
</p>  </section>
<article>This website uses cookies to analyze traffic and
measure the effectiveness of advertisements.
You can find more information about how we use cookies here.
<button>OK</button>
</article>
</body>
</html>

```

14.2.4 Fixed (format)

For the `<section>` tag, enter the upper margin of 70px so that the logo and menu do not cover the text.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
    <base href="https://priscilla.fitped.eu/images/">
    <style>

```

```

        nav {position:fixed; top:0;right:0;}
        a {text-decoration:none;background-color:lightgreen;
text-align:center; display:block; width:70px; padding:5px;}
        a:hover {background-color: lightgrey;}
        ul{list-style-type:none; }
        li {float:right;}
        section{clear:right;}
        figure{width:100px; position:fixed; left:0px; top:0px;
margin:10px;}
        article{background-color: lightgrey; position: fixed;
bottom:0px; left:0px;}
    </style>
</head>
<body>
<nav>
<ul>
    <li> <a href="file1.html"> File 1</a> </li>
    <li> <a href="file2.html"> File 2</a> </li>
    <li> <a href="file3.html"> File 3</a> </li>
</ul>
</nav>
<figure>

</figure>
<section>
<p>
    Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Vivamus rhoncus odio enim, eget pretium tellus gravida
sed. Nulla egestas velit lectus, vitae consequat augue posuere
ac. Suspendisse a metus pharetra, consequat mauris ac,
facilisis ex. Vestibulum porttitor quam eu auctor laoreet.
Cras ut malesuada felis, eu elementum enim. Ut leo nulla,
mollis congue vehicula id, molestie eu neque. Etiam ac varius
augue, in auctor lectus. Vivamus dapibus risus vel neque
finibus, quis tincidunt enim varius. Phasellus quis placerat
urna. Integer vel ornare dolor. Aenean a massa sed odio
vestibulum iaculis. Nullam finibus volutpat metus, a ultricies
sem dignissim vel.
</p>
<p>Ut in imperdiet neque. Ut scelerisque lacinia nisl sit amet
ullamcorper. Donec auctor vitae nunc quis elementum. Morbi et
posuere tellus. Donec eu consectetur odio, vel consectetur
lectus. Fusce a quam vitae sapien varius molestie ac pretium
orci. Donec ut ipsum at enim ornare faucibus sed vitae arcu.

```

```
Vivamus vel orci luctus, mattis augue vitae, placerat mauris.
Vivamus porttitor convallis orci.
</p> </section>
<article>This website uses cookies to analyze traffic and
measure the effectiveness of advertisements.
You can find more information about how we use cookies here.
<button>OK</button>
</article>
</body>
</html>
```

14.2.5 Relative

Use the *position:relative* property to move the "green rectangle" (*<article id="a2">*) to the left side so that it touches the corners of the other rectangles. The relative property changes the position of the element, relative to its resting position.

The units vw viewport's width and vh viewport's height calculate the size of the item in relation to the browser window.

```
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            section {position: relative;}
            #a1{background-color: red; width: 30vw; height: 20vh; }
            #a2{background-
color:green; width: 30vw; height: 20vh; }
            #a3{background-
color:blue; width: 30vw; height: 20vh; }
        </style>
    </head>
    <body>
<section>
    <article id="a1"> 1</article>
    <article id="a2"> 2</article>
    <article id="a3"> 3</article>
</section>
</body>
</html>
```

14.2.6 Relative (top)

Use the *position:relative* property to move the "green rectangle" (*<article id="a2">*) so that its left upper corner is at the point of intersection of diagonals of the "red rectangle". The relative property changes the position of the element, relative to its resting position. Moving the element upwards requires the use of negative values, e.g.: *top:-20vh*.

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>HTML</title>
    <style>
      section {position:relative;}
      #a1{background-color:red;width:20vw;height:20vh;}
      #a2{background-
color:green;width:20vw;height:20vh;}
      #a3{background-
color:blue;width:20vw;height:20vh;}
    </style>
  </head>
  <body>
<section>
  <article id="a1"> 1</article>
  <article id="a2"> 2</article>
  <article id="a3"> 3</article>
</section>
</body>
</html>
```

14.2.7 Relative 3

Use the *position:relative* property to move the "blue rectangle" (*<article id="a2">*) so that its left upper corner is at the point of intersection of diagonals of the "green rectangle". The relative property changes the position of the element, relative to its resting position.

Moving the element upwards requires the use of negative values, e.g.: *top:-20vh*.

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>HTML</title>
    <style>
      section {position:relative;}
```

```

        #a1{background-color:red; width:20vw; height:20vh; }
        #a2{background-
color:green; width:20vw; height:20vh; position:relative; left:10vw
; top:-10vh; }
        #a3{background-
color:blue; width:20vw; height:20vh; }
    </style>
</head>
<body>
<section>
    <article id="a1"> 1</article>
    <article id="a2"> 2</article>
    <article id="a3"> 3</article>
</section>
</body>
</html>

```

14.2.8 Z-index

Arrange the coloured rectangles so that red is on top, green in the middle, and blue at the lowest level. The *z-index* property can only be used for items that have *position* property.

Use a *z-index* property of 1,2,3.

```

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            section {position:relative;}
            #a1{background-color:red; width:20vw;
height:20vh; position:relative;}
            #a2 {background-color:green; width:20vw; height:20vh;
position:relative; left:10vw; top:-10vh;}
            #a3{background-
color:blue; width:20vw; height:20vh; position:relative; left:20vw;
top:-20vh; }
        </style>
    </head>
    <body>
<section>
    <article id="a1"> 1</article>
    <article id="a2"> 2</article>

```

```

<article id="a3"> 3</article>
</section>
</body>
</html>

```

14.2.9 Absolute

Use the *absolute* property to move the green rectangle so that its left upper point is aligned with the right lower point of the red rectangle.

```

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            section {position:relative;}
            #a1{background-color:red; width:20vw; height:20vh;}
            #a2{background-color:green; width:20vw;
height:20vh;}
            #a3{background-color:blue; width:20vw;
height:20vh; position:absolute; left:40vw; top:40vh;}
        </style>
    </head>
    <body>
        <section>
            <article id="a1"> 1</article>
            <article id="a2"> 2</article>
            <article id="a3"> 3</article>
        </section>
    </body>
</html>

```

14.2.10 Absolute - middle

Use the *position* property to place the red rectangle in the middle of the page. Use units vw and vh.

```

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            body{margin:0px; padding:0px;}

```

```

        #a1{background-color:red; width:20vw;
height:20vh; }
    </style>
</head>
<body>
<article id="a1"> 1</article>
</body>
</html>

```

14.2.11 Sticky

Use the *sticky* property to make the menu (nav) "sticky" to its top edge while scrolling vertically. Use the scroll bar to see how the command works.

```

<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            body{ background-color:lightgrey;}
            a {font-size:5vw; text-decoration:none;
color:blue; background-color:white; display: block;
width:15%; float:right; text-align:center; border-radius:5px;}
            a:hover {color:white; background-color:blue;}
            h1{padding:40px;}
            nav{ }
        </style>
    </head>
    <body>
        <h1>Lorem ipsum dolor sit amet</h1>
        <nav>
            <a href="page1.html">Link1 </a>
            <a href="page2.html">Link2 </a>
            <a href="page3.html">Link3 </a>
        </nav>
        <article><h2>Lorem ipsum </h2>
            <p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Vivamus rhoncus odio enim, eget pretium tellus gravida
sed. Nulla egestas velit lectus, vitae consequat augue posuere
ac. Suspendisse a metus pharetra, consequat mauris ac,
facilisis ex. Vestibulum porttitor quam eu auctor laoreet.
Cras ut malesuada felis, eu elementum enim. Ut leo nulla,
mollis congue vehicula id, molestie eu neque. Etiam ac varius
augue, in auctor lectus. Vivamus dapibus risus vel neque

```

```
finibus, quis tincidunt enim varius. Phasellus quis placerat urna. Integer vel ornare dolor. Aenean a massa sed odio vestibulum iaculis. Nullam finibus volutpat metus, a ultricies sem dignissim vel.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus rhoncus odio enim, eget pretium tellus gravida sed. Nulla egestas velit lectus, vitae consequat augue posuere ac. Suspendisse a metus pharetra, consequat mauris ac, facilisis ex. Vestibulum porttitor quam eu auctor laoreet. Cras ut malesuada felis, eu elementum enim. Ut leo nulla, mollis congue vehicula id, molestie eu neque. Etiam ac varius augue, in auctor lectus. Vivamus dapibus risus vel neque finibus, quis tincidunt enim varius. Phasellus quis placerat urna. Integer vel ornare dolor. Aenean a massa sed odio vestibulum iaculis. Nullam finibus volutpat metus, a ultricies sem dignissim vel.</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus rhoncus odio enim, eget pretium tellus gravida sed. Nulla egestas velit lectus, vitae consequat augue posuere ac. Suspendisse a metus pharetra, consequat mauris ac, facilisis ex. Vestibulum porttitor quam eu auctor laoreet. Cras ut malesuada felis, eu elementum enim. Ut leo nulla, mollis congue vehicula id, molestie eu neque. Etiam ac varius augue, in auctor lectus. Vivamus dapibus risus vel neque finibus, quis tincidunt enim varius. Phasellus quis placerat urna. Integer vel ornare dolor. Aenean a massa sed odio vestibulum iaculis. Nullam finibus volutpat metus, a ultricies sem dignissim vel.</p>
</article>
</body>
</html>
```

Tables

Chapter 15

15.1 Tables

15.1.1

Formatting **tables** is not difficult, as it includes properties already used in other tasks.

When designing tables, one must remember about the border. No border property - no frame of the table.

```
table {border:1px dashed lime;}
```

The border so defined will put only one frame around the whole table, but not for cells.

To also enclose individual cells (**<td>** and **<th>**), add them to the selector.

```
table, td, th {border:1px solid lime;}
```

The border can only apply to selected edges:

```
td, th {border-bottom: 1px double grey;}
```

15.1.2

Complete the code to format only the outer edges of the table.

```
_____ {_____ : 2px dotted grey;
font-size:2vw;
background-color: skyblue;
}
```

15.1.3

The **background-color** property defines the background color for the entire table or its elements.

```
table {background-color: orange;}
```

When formatting the table, you can use the child's *n-th* selector.

```
tr:nth-child(2n+1) {background-color:beige;}
```

If the table is too wide and there is a concern that at some resolutions there will be a problem displaying it, you can enter a horizontal scroll bar: *overflow-x: auto*.

To make it easier to read entire rows in a table, you can change their formatting with the `hover` pseudo-class.

```
tr:hover {color:white; background-color:grey;}
```

15.1.4

Text alignment in table cells can be changed horizontally and vertically.

```
td {text-align:center, vertical-align:middle;}
```

The text, in such formatted cells, will be placed horizontally and vertically in the centre.

The `vertical-align` property can take one of the possible values: top, bottom, middle, etc.

When formatting data in the table, you can also use `padding` and `margin`.

15.1.5

To place text in a cell centered vertically the `vertical-align` property should be:

- middle
- center
- medium
- midst

15.2 Tables (programs)

15.2.1 Table - outer border

For the table below, insert the outer border of the entire table with a width of 1px, solid line, in red colour.

```
<!DOCTYPE HTML>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
```

```

        </style>
    </head>
<body>
<section>
<h2> List of tour participants</h2>
<table>
<tr> <th>Surname </th> <th>First name </th>
</tr>
<tr> <td>Nowak </td> <td> Anna</td>
</tr>
<tr> <td>Kowalski </td> <td>Jan </td>
</tr>
<tr> <td> Leszczyńska</td> <td>Maria </td>
</tr>
<tr> <td>Zawiślak </td> <td>Dariusz </td>
</tr>
</table>
</section>
</body>
</html>

```

15.2.2 Cell border

For all cell types in the table below (`<td>` and `<th>`), insert a 1px wide border, dotted line, in green.

Use a collective `border` property and selector grouping (`selector1, selector2 {formatting;}`).

```

<!DOCTYPE HTML>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            table {border: 1px solid red; }

        </style>
    </head>
<body>
<section>
<h2> List of tour participants</h2>
<table>
<tr> <th>Surname </th> <th>First name </th>

```

```

</tr>
<tr> <td>Nowak </td> <td> Anna</td>
</tr>
<tr> <td>Kowalski </td> <td>Jan </td>
</tr>
<tr> <td> Leszczyńska</td> <td>Maria </td>
</tr>
<tr> <td>Zawiślak </td> <td>Dariusz </td>
</tr>
</table>
</section>
</body>
</html>

```

15.2.3 nth child row

Use the *nth* child's selector to change the background of even rows (*nth-child(2n)*) to grey and odd rows (*nth-child(2n+1)*) to light grey.

```

<!DOCTYPE HTML>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            table {border: 1px solid red;}
            td, th {border: 1px dotted green; }

        </style>
    </head>
<body>
<section>
<h2> List of tour participants</h2>
<table>
<tr> <th>Surname </th> <th>First name </th>
</tr>
<tr> <td>Nowak </td> <td> Anna</td>
</tr>
<tr> <td>Kowalski </td> <td>Jan </td>
</tr>
<tr> <td> Leszczyńska</td> <td>Maria </td>
</tr>
<tr> <td>Zawiślak </td> <td>Dariusz </td>
</tr>

```

```
</table>

</section>
</body>
</html>
```

15.2.4 Row - hover

For the table below, use the formatting to change the background color of the entire row to light grey after pointing the mouse at the cell. Use the *hover* property.

```
<!DOCTYPE HTML>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            table {border: 1px solid red;}
            td, th {border: 1px dotted green;}
```



```
            </style>
        </head>
    <body>
        <section>
            <h2> List of tour participants</h2>
            <table>
                <tr> <th>Surname </th> <th>First name </th>
                </tr>
                <tr> <td>Nowak </td> <td> Anna</td>
                </tr>
                <tr> <td>Kowalski </td> <td>Jan </td>
                </tr>
                <tr> <td> Leszczynska</td> <td>Maria </td>
                </tr>
                <tr> <td> Zawislak </td> <td>Dariusz </td>
                </tr>
            </table>

        </section>
    </body>
</html>
```

15.2.5 Table and cell dimensions

Change the width of the table to 40% of the container and the height of the header cells to 50px.

```
<!DOCTYPE HTML>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            table {border: 1px solid red;}
            td, th {border: 1px dotted green;}
            tr:hover {background-color:lightgrey;}
            th{background-color:grey;}
        </style>
    </head>
    <body>
        <section>
            <h2> List of tour participants</h2>
            <table>
                <tr> <th>Surname </th> <th>First name </th>
                </tr>
                <tr> <td>Nowak </td> <td> Anna</td>
                </tr>
                <tr> <td>Kowalski </td> <td>Jan </td>
                </tr>
                <tr> <td> Leszczynska</td> <td>Maria </td>
                </tr>
                <tr> <td> Zawislak </td> <td>Dariusz </td>
                </tr>
            </table>

        </section>
    </body>
</html>
```

15.2.6 Table in the middle of the page

Place the table in the middle of the page (vertical and horizontal). Use the *justify-content* property for the *<section>* tag.

```
<!DOCTYPE HTML>
<html lang="en">
```

```

<head>
    <meta charset="UTF-8">
    <title>HTML</title>
    <style>
        section{ }
        table {border: 1px solid red; width:40%; }
        td, th {border: 1px dotted green; }
        tr:hover {background-color:lightgrey; }
        th{background-color:grey; height:50px; }
    </style>
</head>
<body>
<section>
<table>
<caption>List of tour participants</caption>
<tr> <th>Surname </th> <th>First name </th>
</tr>
<tr> <td>Nowak </td> <td> Anna</td>
</tr>
<tr> <td>Kowalski </td> <td>Jan </td>
</tr>
<tr> <td> Leszczynska</td> <td>Maria </td>
</tr>
<tr> <td>Zawislak </td> <td>Dariusz </td>
</tr>
</table>
</section>
</body>
</html>

```

15.2.7 Cell formating

Center horizontally all `<td>` cells in the table and insert 5px wide internal margins (padding) on each side.

```

<!DOCTYPE HTML>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>HTML</title>
        <style>
            section{display:flex; justify-content:center; }
            table {border: 1px solid red; width:40%; }
            td, th {border: 1px dotted green; }
        </style>
    </head>
    <body>
        <table>
            <caption>List of tour participants</caption>
            <tr> <th>Surname </th> <th>First name </th>
            </tr>
            <tr> <td>Nowak </td> <td> Anna</td>
            </tr>
            <tr> <td>Kowalski </td> <td>Jan </td>
            </tr>
            <tr> <td> Leszczynska</td> <td>Maria </td>
            </tr>
            <tr> <td>Zawislak </td> <td>Dariusz </td>
            </tr>
        </table>
    </body>
</html>

```

```
        tr:hover {background-color:lightgrey; }
        th{background-color:grey; height:50px; }
        td{   }
    </style>
</head>
<body>
<section>
<table>
<caption>List of tour participants</caption>
<tr> <th>Surname </th> <th>First name </th>
</tr>
<tr> <td>Nowak </td> <td>Anna</td>
</tr>
<tr> <td>Kowalski </td> <td>Jan </td>
</tr>
<tr> <td>Leszczynska</td> <td>Maria </td>
</tr>
<tr> <td>Zawislak </td> <td>Dariusz </td>
</tr>
</table>
</section>
</body>
</html>
```

Media Queries

Chapter **16**

16.1 Media queries

16.1.1

As more and more people use the Internet on mobile devices, changes in the way they design websites are a must.

In April 2015, the most popular Google search engine announced changes in the page positioning algorithm. It was decided that if the search is carried out on a mobile device, then pages that are not adapted to mobile devices will fall in the positioning ranking.

This decision motivated the website owners to adapt them to the correct display on mobile devices.

Responsive Web Design (RWD) sites, ie those that adapt themselves to the resolution of the device on which they are displayed, are therefore a standard recommended by W3C (the organization responsible for the development of HTML and CSS).

Preparing a good responsive website requires time and knowledge.

A site recognized by mobile users as user-friendly should meet several conditions:

- should only be scrolled vertically,
- the menu should contain buttons that are easy to point your finger on touch screens,
- links and buttons of forms should also be large enough,
- the font should be properly scaled,
- graphics and video should load quickly.

16.1.2

What does the RWD abbreviation mean for website design?

- Responsive Web Design
- Reactive Web Developer
- Responsive Web Developer

 **16.1.3**

When starting a responsive website design, you should not define large fixed-width elements. It will be very difficult to place two large elements with a width, eg 300px side by side, with small screen resolutions.

When determining the size, it is best to use relative units: %, rem, em, vw, vh etc. Large page elements should be flexible.

The external container, covering all elements of the page, should not have a fixed width. By setting its width to 100% or 100vw, it always fills the full area of the browser. When determining the size of child elements, remember that the percentage unit determines the size of the element relative to the parent element, and vw (viewport width) relative to the display area.

```
.container {width:100vh; box-sizing:border-box; margin:0px;}
```

 **16.1.4**

It is most convenient to design a website by planning the location of elements for devices with small resolutions (about 300-400px), and then gradually expanding it taking into account the bigger devices. This design method is called "mobile first".

The template for ***low-resolution*** devices should be built on ***one column***.

The same method of setting the elements is most often not convenient for devices with high resolutions (eg 1800px).

The use of media queries will make it very easy to customize the template.

In order to check whether the prepared project meets the basic requirements for ***responsive websites***, it is worth using the tools available on the network.

The new validator project, prepared by W3C, also checks the mobile use of websites:

<https://validator.w3.org/unicorn/>

Google also allows you to test the mobile use of pages:

<https://search.google.com/test/mobile-friendly>

 **16.1.5**

What does the term "mobile first" mean?

- The website for mobile devices is more important than the website for desktop devices.
- The same page is correctly displayed on devices with high and low resolution.
- The preparation of the responsive website design starts with the devices with the smallest resolutions, gradually adapting the design to ever higher resolutions.

16.1.6

The optimization of text, graphics and multimedia will be crucial to better website availability.

- In order for the text placed on the page to be easy to read, the line should not contain more than 70-80 characters (that is, about 8-10 words). The font should be defined using a relative unit.
- When preparing the menu on the page, it is necessary to ensure that the elements selected using touching devices are large enough. The minimum link size should be about 40px x 40px.
- Form buttons and other links available on the page should be large enough and placed not too close to each other.
- Graphics and videos placed on websites should be optimized, so they could be loaded as fast as possible. In the network, you can find pages that will help in better compression of selected formats. When preparing a responsive website, just scaling the graphics is not enough - a large file will still be loaded, only displayed on a smaller scale. It is worth using a new **<picture>** tag here, which will load the image already scaled and adapted to the media query.
- Flash art elements, quite popular until recently, should not be used on websites. Flash is not supported by mobile browsers.

16.1.7

By default, the browser scales the page so that it is fully displayed on the device's screen. This reduces the number of page elements.

The first step in preparing a responsive page is to put a **<meta>** tag in the page header, which will determine the page display on the full width of the device, in the initial scale of 1.0, in a different way than automatically.

```
<head>
...
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
```

```
...
</head>
```

This tag will cause the page content to fill the entire browser window.

When designing a website, you can use media queries.

You can put them in the `<link>` tag or directly in the CSS stylesheets.

16.1.8

Complete the fragments of the `<meta>` tag, which is mandatory when designing responsive pages:

```
<meta name="____" content="width=____, initial-scale=1.0">
```

16.1.9

The basis in the preparation of the responsive page is ***media inquiries***, ie the ***@media*** rule.

In general, media queries have the following format:

```
@media mediatype and|only|not (expression) {
formating;
...
}
```

- ***mediatype*** - is one of the types of media that a browser can recognize. It can take the following values: ***screen***, ***print***, ***speech*** (visually impaired screen readers that read the page aloud) and all (defines all types of devices),
- ***expression*** - this is to specify additional device attributes.

For screens with a minimum resolution of 960px, elements with the `.art_top` class will occupy 30vw, will be aligned to the left and surrounded by other elements placed on the page.

```
@media screen and (min-width:960px) {
.art_top {width:30vw; float:left;}
}
```

You can specify additional device properties for the selected media type using the ***and***, ***not***, ***only*** conjunctions.

The most common questions are:

- maximum window width - *max-width*,
- the minimum width of the window - *min-width*,
- the horizontal or vertical orientation of the window - *orientation: landscape* (or *portrait*).

16.1.10

Examples of media queries.

```
@media screen and (min-width:576px) {formating...}
```

Formatting will apply to screen devices that have a window width of 576px or larger.

```
@media speech {formating...}
```

Formatting will apply to programs that read the user's screen (for the visually impaired).

```
@media screen and (min-width:920px and max-width:1200px)  
{formating...}
```

Formatting applies to devices with a browser window resolution from 920px to 1200px.

```
@media print {formating...}
```

Formatting prepared for pages sent for printing.

16.1.11

Define a query that will select mobile devices with a resolution below 576px:

```
@_____ screen and (_____:575px) {formating}
```

16.1.12

Define a question that will determine the formatting for the printers:

```
@media _____ {formating...}
```

16.1.13

When designing a responsive page, it is worth defining the boundary points for which the formatting or layout of the elements will be changed. How to do it?

Currently, with such a huge number of different mobile devices, there are no set limits at which individual types of devices end or start. Increasingly, cell phones have a higher resolution than tablets, etc. It is worth following statistics or using the experience of others.

The popular **Bootstrap framework**, which is used to design responsive pages, offers the following breakdown:

- <576px - devices whose width is less than 576px,
- > = 576px - devices whose width is greater than or equal to 576px,
- > = 768px - devices whose width is greater than or equal to 768px,
- > = 992px - devices whose width is greater than or equal to 992px,
- > = 1200px - devices whose width is greater than or equal to 1200px.

The website designer can define the boundary points of media queries, adapted to ones own project.

16.1.14

Media requests can also be placed in the `<link>` tag, which appends a file with a CSS style to the page.

In the `<head>` section, you can insert several `<link>` tags, which, depending on the media types recognized by the browser, will apply the appropriate style sheet. The general syntax of such elements is as follows:

```
<link rel="stylesheet"
      href="style.css" media="mediatype and|not|only
      (expressions)" >
```

- *mediatype* - determines the media type for the attached stylesheet,
- *and/not/only* - it is possible to choose logical values that can be used to add details about selected media defined in (expressions).

```
<link rel="stylesheet" href="style1_screen.css" media="screen
      and (min-width: 920px)">
<link rel="stylesheet" href="style2_screen.css" media="screen
      and (max-width: 600px)">
<link rel="stylesheet" href="print.css" media="print">
```

 **16.1.15**

Complete the code so that for devices with a minimum width of 860px there was *style1.css* file attached.

```
<link rel="stylesheet" href="_____"
      _____="screen and
      (_____ : 860px)">
```

 **16.1.16**

In a CSS file, usually, there are many rules that describe how to format elements for specific resolutions.

```
/* for small resolutions, each of the three columns fills 100%
of the width of the browser */
.kol1, .kol2, .kol3 {width: 100%;}
/* for larger screens, the first column occupies 100%, the
remaining 50% */
@media only screen and (min-width: 620px) {
.kol1{width: 100%;}
.kol2 {width: 50%; float:left;}
.kol3 {width: 50%;float:left;}
}
/* for resolutions above 820px - the data is arranged in three
columns */
@media only screen and (min-width: 820px) {
.kol1 {width: 20%;float:left}
.kol2 {width: 50; float:left;}
.kol3 {width: 30%;float:left;}
}
```

16.2 Media queries (programs)

 **16.2.1 @media - max-width**

Insert a media query so that for devices with a maximum width of 500px, the background of the *<div>* tag is red and the *floating* was disabled.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
```

```

<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
  div {background-color:black; color:white;
margin:2px;padding:40px; float:left;}

</style>
</head>
<body>
<section >
  <div>TEXT 1</div>
  <div>TEXT 2</div>
  <div>TEXT 3</div>
  <div>TEXT 4</div>
</section>
</body>
</html>

```

16.2.2 @media - menu

Insert a media query so that for devices with a maximum resolution of 600px, the background of `<nav>` tag is red and the navigation links are placed vertically. Use the `flex-direction` property.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
  nav {display: flex; background-color: black;}
  nav a {color: white; padding: 15px; text-decoration:
none;text-align: center;}
  nav a:hover {background-color:grey; }

</style>
</head>
<body>
<nav >
  <a href="#">Link 1</a>
  <a href="#">Link 2</a>
  <a href="#">Link 3</a>

```

```

<a href="#">Link 4</a>
</nav>
</body>
</html>

```

16.2.3 @media - print

For the following page, place a media query for the printout of the page and turn off the *floating*. Use the device type: *print*.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
  div { margin:2px;padding:40px;background-
color:grey;float:left; }

</style>
</head>
<body>
  <div>TEXT 1</div>
  <div>TEXT 2</div>
  <div>TEXT 3</div>
  <div>TEXT 4</div>
</body>
</html>

```

16.2.4 @media display:none

For the following page place a media query that, for devices with a maximum width of 550px, will turn off the image display. Use the *display* property.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">

```

```

<base href="https://priscilla.fitped.eu/images/">
<style>
  body{background-color: lightyellow;}
  nav {display: flex;flex-wrap:wrap; background-color: lightgrey;}
  nav a {color: white; padding: 15px; text-decoration: none;text-align: center;}
  nav a:hover {background-color:grey;}
  img {width:50px; }

</style>
</head>
<body>
<nav >

  <a href="#">Link 1</a>
  <a href="#">Link 2</a>
  <a href="#">Link 3</a>
  <a href="#">Link 4</a>
</nav>
</body>
</html>

```

16.2.5 @media - orientation

For the following page add a media query that will place menu items in the column for vertically oriented devices. Use the *flex-direction* property.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
  nav {display: flex; background-color: black;}
  nav a {color: white; padding: 15px; text-decoration: none;text-align: center;}
  nav a:hover {background-color:grey;}

</style>
</head>
<body>

```

```

<nav >
  <a href="#">Link 1</a>
  <a href="#">Link 2</a>
  <a href="#">Link 3</a>
  <a href="#">Link 4</a>
</nav>
</body>
</html>

```

16.2.6 @media - min-width

For the following page add a media query that will set the width of div tags to 50% for devices from 576px to 768px and change the background color to green. Use only *screen* and *min-width* properties in the query.

Insert the query in the appropriate place of the style sheet.

You will see the effect of the query by changing the width of the browser window.

```

<!DOCTYPE html>
<html lang="pl">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
  div {box-sizing:border-box; border:1px solid black;
padding:60px; float:left; }
  @media screen and (max-width:576px) { div {background-color:
red; width:100%;} }
  @media screen and (min-width:768px) { div {background-color:
blue; width:25%;} }
</style>
</head>
<body>
<section >
  <div>TEXT 1</div>
  <div>TEXT 2</div>
  <div>TEXT 3</div>
  <div>TEXT 4</div>
</section>
</body>
</html>

```

16.2.7 @media - background

Insert the media query so that for devices with a maximum width of 576px, the page background is white.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
  body{background-color: #6A98C8;}
  body{background-color: white;}
</style>
</head>
<body>

</body>
</html>
```

16.2.8 @media - page layout

For the following page add a media query, which for devices with a maximum width of 650px will set `<section>` tags one under another (vertically) and give them a width of 100%.

Use only `screen` and `max-width` properties in the query.

You will see the effect of the query by changing the width of the browser window.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<style>
  header,nav,section,footer {background-color:#cccd9ff; border:1px solid black; box-sizing:border-box;}
  header {height:20vh; }
  nav {height:10vh; }
  section {float:left; width:33.33%; height:60vh; }
  footer {height:10vh; clear:both; }
```

```

</style>
</head>
<body>
<header>
Header
</header>
<nav> Nav</nav>
<section>Section
</section>
<section>Section
</section>
<section>Section
</section>
<footer>
Footer
</footer>
</body>
</html>

```

16.2.9 @media - link

Insert in the `<head>` section the links that will include a file with the style: "portrait.css" for horizontal devices. When changing the ratio it should change the background color of the page (external style formatting).

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>Graphics - HTML course</title>
        <base href="https://priscilla.fitped.eu/images/">
        <link rel="stylesheet" media="(orientation:
landscape)" href="landscape.css">
            <!-- the attached styles change the background
color of the page depending on the page layout (horizontal or
vertical)-->
    </head>
    <body>
        <article>
            Lorem Ipsum is simply dummy text of the printing and
typesetting industry. Lorem Ipsum has been the industry's
standard dummy text ever since the 1500s, when an unknown
printer took a galley of type and scrambled it to make a type

```

specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

```
</article>
</body>
</html>
```

Validation, Units, Selectors Hierarchy

Chapter 17

17.1 Units

17.1.1

One of the most difficult things in web design is to define the right size of elements (text, graphics, blocks, etc.). What the designer has to face is the fact that the same page will be viewed on devices of different sizes. The designed elements must therefore be ***flexible***.

Defining units is required by elements such as:

- *width*,
- *height*,
- *margin*,
- *padding*,
- *font-size*
- and many others.

Until recently, the basic unit in determining the size was ***pixel (px)***. However, it is considered as an absolute unit, i.e. one whose size will be similar on all devices. Using px is convenient when we know the resolution of the device on which the page is displayed.

W3C encourages the use of the unit ***em***. It is calculated relative to the size of the element in the parent.

The W3C specification specifies the font size of header tags in ***em***.

```
h1{font-size:2em; }
h2{font-size:1.5em; }
```

We assume that in this example the parent of ***<h1>*** and ***<h2>*** is ***<body>*** element. Since the default font size in browsers is 16px, then:

1em=16px.

In the above example, $2em = 2 * 16px$, therefore ***<h1>*** will have a font size of *32px*, ***<h2>*** respectively - *24px*.

17.1.2

To understand the unit well, it is worth following the example below.

In CSS:

```
body {font-size:14px;}
section, article, div {font-size:1.2em}
```

In HTML:

```
<body>
<section >
Test <!-- This font will be the size of: 14 * 1.2 = 16.8px -->
<article>
  Test <!-- This font will be the size of: 16.8 * 1.2 =
20.16px -->
<div>
  Test <!-- This font will be the size of: 20.16 * 1.2 =
24.192px -->
</div>
</article>
</section>
</body>
```

17.1.3

```
<style>
html {font-size:10px;}
div {font-size:1.4em;}
p {font-size:1.2em;}
</style>
<body>
<div>
<p>Text</p>
</div>
</body>
```

What size will be the font in the paragraph in the given part of the page?

- 16.8px
- 12px
- 14px
- None of the above answers is true.

17.1.4

The unit ***em*** is a relative unit. When using this unit, always pay attention to the formatting of the parent elements. Sometimes it can be embarrassing.

An alternative to ***em*** unit is ***rem***.

The rem unit is calculated in relation to the value set for the whole page (by the parent). For rem, it is not important to format the elements defined for the parent. The element dimension is always set relative to the ***root (root em)***.

```
html {font-size:20px;}
article {font-size:2rem;}
p {font-size:1rem;}
h1 {font-size:1.5rem;}
```

In the example above, the paragraph inside the **<article>** tag will have a *20px* font (relative to **<html>** instead of **<article>**).

These units (***em*, *rem***) are relative, but they change their size only when the value of the parent element (parent or HTML document in general) changes.

17.1.5

```
<style>
html {font-size:10px;}
div {font-size:1.4rem;}
p {font-size:1.2rem;}
</style>
<body>
<div>
<p>Text</p>
</div>
</body>
```

What size will be the font in the paragraph in the given part of the page?

- 12px
- 14px
- 16.8px
- None of the above answers is true.

17.1.6

The often-used way of entering values is percent (%). It is a relative unit. The percentage size is always counted relative to the parent element.

```
section {width:400px; }
```

```
section article {width:25%; float:left; box-sizing:border-box;}
```

The width of the `<article>` tag defined in this way is 100px.

17.1.7

The relative units ***vw (viewport width)*** and ***vh (viewport height)*** determine the size/width of the element in relation to the size of the browser window.

1vw is 1% of the width of the browser window (viewport),

1vh is 1% of the browser window's height (viewport).

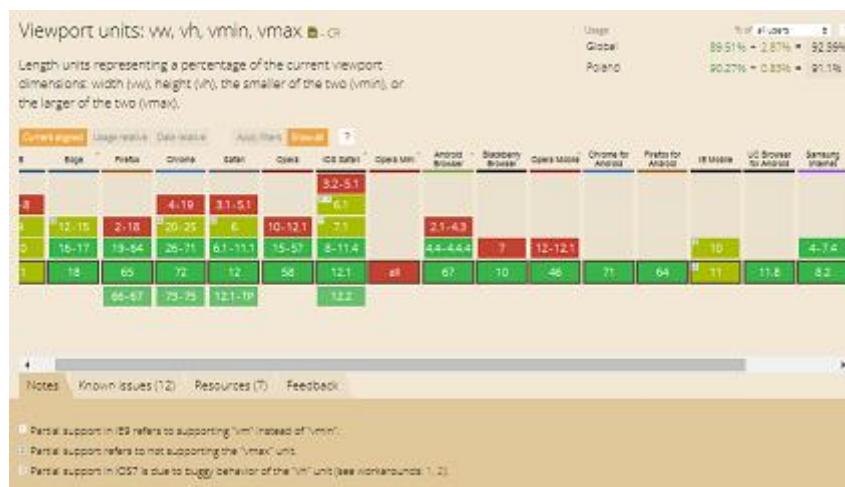
```
article {width:40vw; }
p {font-size:2vw; }
```

If the width of the viewport is 1000px, then the size of the font $2\text{vw} = 20\text{px}$, when we display the page in a browser with a width of 400px, then $2\text{vw} = 8\text{px}$.

The width of the element determined by ***vw*** depends on the width of the browser's workspace.

When designing responsive websites, pay attention to the very fast growing popularity of these units.

These units are new and not all browsers interpret them correctly. You can follow their support, for example at caniuse.com.



 17.1.8

```
aside {width:50vw;
height:50vh;
box-sizing:border-box;
background-color:purple;}
```

The `<aside>` tag will take up half of the browser workspace horizontally and half vertically. It does not matter if the page is displayed in landscape or portrait orientation. The `<aside>` element defined in this way will be automatically adjusted.

```
div {width:100vw;
height:100vh;
box-sizing:border-box;
background-color: grey;}
```

This `<div>` will fill 100% of the workspace horizontally and vertically.

 17.1.9

```
<style>
div {width:800px;}
p {width:25vw;, font-size:1vw;}
</style>
<body>
<div>
<p>Text</p>
</div>
</body>
```

What width will have `<p>` if the browser workspace is 960px?

- 240px
- 200px
- None of the answers is correct.

 17.1.10

Because the pages are displayed on devices with horizontal orientation (eg monitors) or vertical (eg often telephones), W3C has introduced a new unit that allows to determine which

of the dimensions of the device is larger and depending on this, choose the size of the elements.

`1vmin = smaller of the numbers: 1vw or 1vh`

`1vmax = the larger of the numbers: 1vw or 1vh`

```
.box1 {width:100vmin; height:100vmin;}
```

If the horizontal value is smaller, the square will fill the entire horizontal width. If the height is smaller, the square will fill the entire height of the window. A defined class - `box1` for any block tag, will place a square element on the page that always fills at least one of the horizontal or vertical dimensions.

```
.box2 {width:100vmax; height:100vmax;}
```

With such a defined class, the scroll bar will appear when the page has horizontal or vertical orientation?

17.1.11

```
<style>
div {width:800px; height:400px}
p {width:25vmin; font-size:1vw;}
</style>
...
<body>
<div>
<p>Text</p>
</div>
</body>
```

What width will have `<p>` if the browser workspace is 960px (width) at 600px (height)?

- 150px

- 240px
- 100px
- None of the answers is correct.

17.1.12

Match the elements together.

p {font-size: 1rem} ____

p {font-size: 10%} ____

p {font-size: 10vw} ____

p {font-size: 10vw} ____

- sets the font size relative to the font in the entire document
- sets the font size in percentage of the parent font
- sets the size of the font relative to the width of the browser window area
- sets the font size relative to the height of the browser window area
- sets the font size relative to the browser window area

17.2 Units (programs)

17.2.1 Unit em

Assuming the font size of the whole page is 16px, set the text size in the `<section>` tag to double the size. Save its value using the unit `em`.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>FITPED</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <style>
      body{font-size:16px;}
      section{}
    </style>
  </head>
  <body>
```

```

<p>TEXT- font-size: 16px
</p>
<section>TEXT- font-size: ?em
</section>
</body>
</html>

```

17.2.2 Unit em-px

Assuming the font size of the whole page is 20px, set the text size in the `<h2>` tag to 30px. Save its value using the unit `em`.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>FITPED</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <style>
      body{font-size:20px ; }
      h2{ }
    </style>
  </head>
  <body>
    <p>TEXT
    </p>
    <section>
      <h2>TITLE TEXT
      </h2>
    </section>
  </body>
</html>

```

17.2.3 Unit rem

Assuming the font size of the whole page is 20px, set the text size in the `<article>` tag to make it half the size (10px). Save its value using the `rem` unit.

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>FITPED</title>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width,
initial-scale=1">
<style>
html{font-size:20px;}
section{font-size:1.5em;}
article {}
</style>
</head>
<body>
    <p>TEXT
    </p>
    <section>
        <h2>TITLE TEXT
        </h2>
        <article>
            TITLE ARTICLE
        </article>
    </section>
</body>
</html>

```

■ 17.2.4 Unit vw and vh

Using *vw* and *vh* units, format the *<article>* tag so that its width and height cover the entire browser area (100%). Use *vw* and *vh* units.

```

!DOCTYPE html>
<html lang="en">
    <head>
        <title>FITPED</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1">
        <style>
            section{background-color:pink;font-size:1.5em;  }
        </style>
    </head>
    <body>
        <p>TEXT
        </p>
        <section>
            <h2>TITLE TEXT
            </h2>

```

```

        </section>
    </body>
</html>
```

17.2.5 Unit vmin

Using *vmin* or *vmax* units, format the *<section>* tag so that it creates a square on the page and does not show the scroll bars.

If the orientation of the page is vertical, the square should fill the entire width of the browser, if horizontal - the opposite.

```

<!DOCTYPE html>
<html lang="en">
    <head>
        <title>FITPED</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width,
initial-scale=1">
        <style>
            body{margin:0px;}
            section{background-color:pink; }
        </style>
    </head>
    <body>
        <section>
        </section>
    </body>
</html>
```

17.2.6 Unit - percent

For a *<article>* tag with *a1* identifier, set the width and height to 50% of the *parent* element. Use the *width* and *height* properties.

Compare both regions of a *<article>* tags.

Note the units used to format *<article>* tag with *a2*. Here 50vw and 50vh refer to the size of the browser window.

```

<!DOCTYPE html>
<html lang="en">
    <head>
```

```
<meta charset="UTF-8">
<title>HTML</title>
<style>
article, body {margin:0px;padding:0px;}
article#a1 { background-color:lightgreen;}
article#a2 {width:50vw; height:50vh;background-
color:lightblue; }
</style>
</head>
<body>
<article id="a1">


Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.


</article>
<article id="a2">


Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.


</article>
</body>
</html>
```

17.3 Validators

17.3.1

The browser's task is to interpret and display the tags stored in the file. If the page contains errors, the browser interprets them according to its algorithm. According to the assumptions of W3C projects, it should still correctly interpret obsolete tags.

So how do you check if the prepared page is correct? How do you find an error when the page is displayed differently than expected?

There are many websites on the Internet that verify the correctness of the code from a different angle. One of the most important is the site prepared by W3C:

<https://validator.w3.org/>

On this page, you can check the syntax of any website.

The validator will indicate lines containing errors, e.g.

- no required markers, attributes,
- typos in the name of tags attribute or used values,
- tags, value attributes that do not match the declared version.

The validator, however, does not check the semantics of the code.

On the validator's page, you can enter the HTML code in three ways:

- indicate the website address in the network (a page published on the Internet),
- upload the file from the user's local disk,
- paste the page code into the prepared dialogue box.

17.3.2

In the same way as HTML code you can verify files containing CSS code.

W3C provides a page to verify the CSS code:

<https://jigsaw.w3.org/css-validator/>

Similarly to an HTML file, the page indicates syntax errors.

The use of validators has many advantages. On the one hand, it helps to improve the code of the page, and on the other hand teaches the proper use of tags, attributes or properties in CSS.

17.3.3

The HTML validator improves:

- tag syntax
- correct placement of text on the page
- spelling - according to the language declared in <html>

17.3.4

From 2019, one more W3C website is available, containing a unified validator:

<https://validator.w3.org/unicorn/>

Unified validator. HTML, CSS, Links & Mobile

This page allows you to check the page using a standard test suite. You can check only CSS sheets or configure the set of tasks. Similarly to previous validators, you can enter the code in three ways: URI address, file uploading or pasting the code.

In principle, this page was supposed to check HTML and CSS code, the correctness of links and adaptation of pages to mobile devices. Unfortunately, HTML and CSS are constantly changing so not all functionalities of the new validator are working properly.

Links to many tools that validate pages can be found here:

<https://www.w3.org/developers/tools/>

17.3.5

Google also provides many websites that support web developers.

Noteworthy is the website that tests accessibility for mobile devices.

<https://search.google.com/test/mobile-friendly>

The most common errors indicated by this type of validator are:

- too small font makes it difficult to read the text,
- the visible area does not adjust to the screen width of the device,
- clickable elements are too close to each other.

17.4 Validators (programs)

17.4.1 Correct errors 1

Correct the errors on the site. Close the tags and enter the required attributes.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<base href="https://priscilla.fitped.eu/images/">
<style>
body{background-color:lightgrey;}
section{border:1px solid red; width:25%; padding:5px;}
    img{width:50%;}
</head>
<body>
<h2>Months</h2>
<section>

    <ul>
        <li> January </li>
        <li> February </li>
        <li> March </li>
        <li> April
        <li> May </li>
        <li> June </li>
    </ul>
</section>
</body>
</html>
```

17.4.2 Correct errors 2

Correct the errors on the site.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED </title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
    <base href="https://priscilla.fitped.eu/images/">
<style>
body{background-color:lightgrey;}
section{border:1px solid red; width:25%; padding:5px;}
    img{width:50%;}
</style>
</head>
<body>
<h2>Months</h2>
<section>

    <ul>
        <li> January </li>
        <li> February </li>
        <li> March </li>
        <li> April </li>
        <li> May </li>
        <li> June </li>
    </ul>
</section>
</html>

```

17.5 Selector priorities

17.5.1

In web design, it often happens that the property of one of the elements is formatted in several rules.

What color will be the text of the paragraph, which is a child of `<aside>`?

```

<style>
p.new {color:red;}
aside p {color:blue;}
p#info {color:grey;}

```

```
</style>
...
<aside>
<p id="info" class="new" style="color:green"> TEXT</p>
</aside>
```

The CSS specification clearly and specifically defines the priorities of individual ***selectors***.

Here, the paragraph TEXT will be displayed in green, due to the detail of the selector.

17.5.2

At the beginning of the course, one of the most important CSS rules - a style cascade was presented.

The cascade means that the closer the style is to the tag, the more important it is. Inline-style formatting has a higher priority than formatting in the embedded style in the ***<head>*** section and then the formatting saved in the external file. This rule, however, is very general, it applies only to the property defined with the same selector, but in different styles.

If the same property has been formatted in the same style several times, the definition which was later will be applied (it overrides earlier formatting).

```
<style>
article {text-align:center;}
</style>
...
<article style="text-align:right;">
TEXT
</article>
```

In the above example, TEXT will be right-aligned because the inline style has a higher priority than the style embedded in the ***<head>*** section.

17.5.3

What will be the alignment of the TEXT in the ***<article>*** tag?

```
<style>
article {text-align:center; }
```

```

article{background-color:grey;}
article {text-align:right;}
</style>
...
<article>
TEXT
</article>
```

- left
- right
- center
- justify

17.5.4

The identifier is more important than the class selector. This is because it is more unique. An ID with a given name can be used in the document only once, the same class name can be used multiple times.

The more identifiers the selector has, the more detailed it is.

```

<style>
article#chapter p#info {color: red }
p#identyfikator {color: green }
p {color: black }
<style>
...
<article id="chapter">
<p id="info">TEXT </p>
</article>
```

TEXT will be written in red because the first selector is the most detailed.

17.5.5

```

<style>
p#info {color: green;}
p.test {color: blue;}
p {color: black;}
<style>
...
<p id="info" class="test"> TEXT </p>
```

In what color will "TEXT" be displayed after interpreting the above code snippet?

- green
- blue
- black

17.5.6

If the considered selectors do not contain any identifiers or contain the same number of IDs, the number of classes, pseudo-classes and attributes used in the rule is decisive for validity.

```
<style>
  footer.root1 p{ color: green; }
  p { color: purple; }
</style>
...
<footer class="root1">
  <p> TEXT (color:green)</p>
</footer>
```

If any property is described in several classes that have the same priority, the browser will perform the formatting of the last one added.

```
<style>
  h2.cent {text-align:center;}
  h2.le {text-align:left;}
  h2.just {text-align:justify;}
</style>
...
<h2 class= "le just cent">
  Title <!--text-align:center-->
</h2>
```

17.5.7

```
<style>
  p.cl1t {text-align:center;}
  p.cl2 {text-align:left;}
  p.cl3 {text-align:right;}
</style>
...
<p class= "cl2 cl3 cl1">
```

```
TEXT
</p>
```

After interpreting the above code, TEXT will be aligned to the:

- center
- left
- right

17.5.8

If the number of identifiers, classes, pseudo-classes and attributes is the same in the selector, the number of elements or pseudo-elements decides on their *validity*.

```
<style>
div p { color:red; font-size: 1.5em; }
p { color:blue; font-size: 3em; }
</style>
...
<div>
<p>TEXT </p>
</div>
```

TEXT in the paragraph will be written in red because the first selector was specified more precisely (it considered every *p* tag that is a child of *div*).

17.6 Selector priorities (programs)

17.6.1 Descendant selector

Define a selector that will allow you to format (text in green) all *<h2>* tags except the first one (containing text: Title 0). Use the descendant selector.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
    body{background-color:lightgrey;}
```

```

    section{border:1px solid red; width:25%; padding:5px; }
        article {border:1px solid black; }
            {color:green; }
</style>
</head>
<body>
<h2>Title 0</h2>
<section>
<h2>Title 1</h2>
<article >
<h2>Title 2 </h2>
</article>
<article >
<h2>Title 3 </h2>
</article>
<article >
<h2>Title 4 </h2>
</article>
</section>
</body>
</html>

```

17.6.2 Child selector

Define a selector that allows you to format (pink text) only the second tag `<h2>` (containing text: Title 1). Use the child selector.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
    body{background-color:lightgrey; }
    section{border:1px solid red; width:25%; padding:5px; }
        article {border:1px solid black; }
            {color:pink; }
</style>
</head>
<body>
<h2>Title 0</h2>
<section>

```

```

<h2>Title 1</h2>
<article>
<h2>Title 2 </h2>
</article>
<article>
<h2>Title 3 </h2>
</article>
<article>
<h2>Title 4 </h2>
</article>
</section>
</body>
</html>

```

17.6.3 Adjacent sibling selector

Define a selector that allows you to format (orange text) only the `<h2>` tags placed in the last two `<article>` (Title 3, Title 4). Use the adjacent sibling selector ("+") and descendant selector.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
  body{background-color:lightgrey;}
  section{border:1px solid red; width:25%; padding:5px;}
  article {border:1px solid black;}
    {color:orange;}
</style>
</head>
<body>
<h2>Title 0</h2>
<section>
<h2>Title 1</h2>
<article>
<h2>Title 2 </h2>
<p></p>
</article>
<article>
<h2>Title 3 </h2>

```

```

<p> </p>
</article>
<article >
<h2>Title 4 </h2>
<p> </p>
</article>
</section>
</body>
</html>

```

17.6.4 Nth-child selector

Define the selector of the nth-child for `<i>` in a letter numbered list so that the even elements (February, April, June) have a white background.

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">
<style>
    body{background-color:lightgrey;}
    section{border:1px solid red; width:25%; padding:5px;}
        {background-color:white;}
</style>
</head>
<body>
<h2>Months</h2>
<section>
    <ul>
        <li> January </li>
        <li> February </li>
        <li> March </li>
        <li> April </li>
        <li> May</li>
        <li> June </li>
    </ul>
</section>
</body>
</html>

```

17.6.5 Nth-child selector 2

Insert a formatting rule for the nth-child (for </i>) so that the white background has every third element of the list, starting with the first one (January, April).

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>FITPED</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
    body{background-color:lightgrey;}
    section{border:1px solid red; width:25%; padding:5px;}
        {background-color:white;}
</style>
</head>
<body>
<h2>Months</h2>
<section>
    <ul>
        <li> January </li>
        <li> February </li>
        <li> March </li>
        <li> April </li>
        <li> May</li>
        <li> June </li>
    </ul>
</section>
</body>
</html>
```

17.6.6 Descendant and pseudo-class

Using the descendant and pseudo-class link selector, format only the menu links. Use the formatting saved in style (red background and black text links, indicated by the mouse vice versa). The other links on the page should keep the formatting as initially placed in style.

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
```

```

<title>html</title>
<style>
    a {font-family:batang;text-decoration:none; color:green;}
    a:link { background-color: yellow;}
    a:visited {background-color: lightgreen;}
    a:hover {background-color: lightgrey;}
    ul {list-style-type:none;}
        {background-color:red; color:black; display:block;
padding:5px; width:70px; margin:1px; text-align:center;
border-radius:3px;}
        {background-color:black; color:red; display:block;
padding:5px; width:70px; margin:1px; text-align:center;
border-radius:3px;}
</style>
</head>
<body>
<nav>
<ul>
    <li> <a href="file1.html"> file 1</a> </li>
    <li> <a href="file2.html"> file 2</a> </li>
    <li> <a href="file3.html"> file 3</a> </li>
</ul>
</nav>
<aside>
<a href="https://www.nytimes.com/"> The New York Times </a>
<br>
<a href="https://www.smashingmagazine.com/"> Smashing
magazine </a><br>
<a href="https://www.oscars.org/"> Academy of Motion Picture
Arts and Sciences </a> <br>
</aside>
</body>
</html>

```

17.7 Enjoy CSS (programs)

17.7.1 CSS transformations

It's time to be independent. We can't teach you everything. You are prepared to learn and experiment on your own.

The properties used in the file below were not discussed in our course. However, we are confident that you will manage the job perfectly.

- Rotate the first picture left by 20 degrees.
- Tilt the second image horizontally, to the left by 10 degrees.
- The third picture tilts down 20 degrees vertically.
- Scale the fourth image so that it is twice as small horizontally and twice as big vertically as the original.

```
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<base href="https://priscilla.fitped.eu/images/">
<style>
  img.rot{transform:rotate(30deg) ;}
  img.skex{transform:skewX(-40deg) ;}
  img.skey{transform:skewY(40deg) ;}
  img.ska{transform:scale(0.5,0.5) ;}
  figure {width:200px; border:1px solid black;float:left;}
</style>
</head>
<body>
<figure>

</figure>
<figure>

</figure>
<figure>

</figure>
<figure>

</figure>
</body>
</html>
```

17.7.2 CSS animations

Time for being independent!

The properties used in the file below were not discussed in our course. However, we are confident that you will be able to handle the task perfectly.

- Change the direction of rotation of the image.

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title of the document</title>
<base href="https://priscilla.fitped.eu/images/">
<style>
    figure {width:150px; border:1px solid black;float:left;}
    img { transform:rotate(0deg);
        animation-name: example;
        animation-duration: 6s;
        animation-iteration-count: infinite;
    }
    @keyframes example {
        0%   {transform:rotate(30deg) ;}
        10%  {transform:rotate(60deg) ;}
        20%  {transform:rotate(120deg) ;}
        40%  {transform:rotate(180deg) ;}
        60%  {transform:rotate(240deg) ;}
        80%  {transform:rotate(300deg) ;}
        100% {transform:rotate(360deg) ;}
    }
</style>
</head>
<body>
<figure>

</figure>
</body>
</html>
```



PRISCILLA



priscilla.fitped.eu