

0. Shortcuts writing HTML/CSS in VS code

Using Emmet Abbreviation: пишем структурата на html кода със CSS селектори

ul->li*3>a

```
ul->li*3>a
<body>
<ul->
  <li><a href=""></a></li>
  <li><a href=""></a></li>
  <li><a href=""></a></li>
</ul->
```

header.site-header+main>article>header

```
<header class="site-header"></header>
<main>
  <article>
    <header></header>
  </article>
</main>
```

emmet за писане на CSS-a

m0

margin: 0;

fwb

font-weight: bold;

1. HTML & CSS Intro

1.1. HTML Intro

В програмата Visual Studio Code: Записваме файла като .html

След това пишем html:5 + Enter и ни се генерира тялото.

Става и с удивителен знак също + Enter/Tab.

<https://webplatform.github.io/>

1.1.1. Basic structure

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
```

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

Head

The <head> contains markup not visible to the user. But helps the browser to render correctly the HTML document.

Metadata definitions

```
<meta charset="UTF-8">
```

Styles declarations

```
<link rel="stylesheet" type="text/css" href="site.css">
```

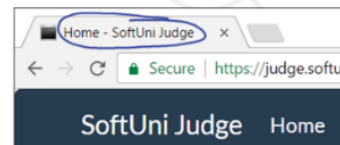
Scripts declarations

```
<script src="main.js"></script>
```

title

The <title> tag – HTML document title

```
<head>
  <title>Home - ... </title>
</head>
```



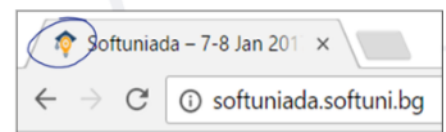
favicon

<https://favicon.io/favicon-converter/>

<https://favicon.io/favicon-generator/>

Favicon – a site icon

```
<link href="/favicon.ico"
rel="shortcut icon" type="image/x-icon"/>
```



Installation

First, use the download button to download the files listed below. Place the files in the root directory of your website.

- android-chrome-192x192.png
- android-chrome-512x512.png
- apple-touch-icon.png
- favicon-16x16.png
- favicon-32x32.png
- favicon.ico
- site.webmanifest

Next, copy the following link tags and paste them into the `head` of your HTML.

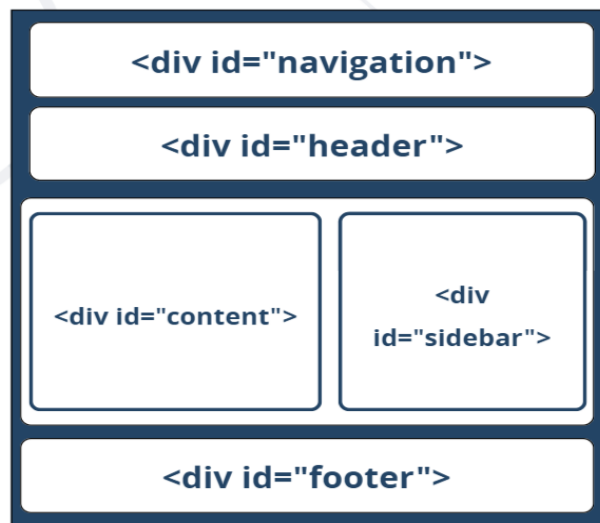
```
<link rel="apple-touch-icon" sizes="180x180" href="/apple-touch-icon.png">
<link rel="icon" type="image/png" sizes="32x32" href="/favicon-32x32.png">
<link rel="icon" type="image/png" sizes="16x16" href="/favicon-16x16.png">
<link rel="manifest" href="/site.webmanifest">
```

The HTML 4 & Old way

The "HTML4 & Old" Way

Using **<div>**s with **IDs** (the IDs are needed for styling)

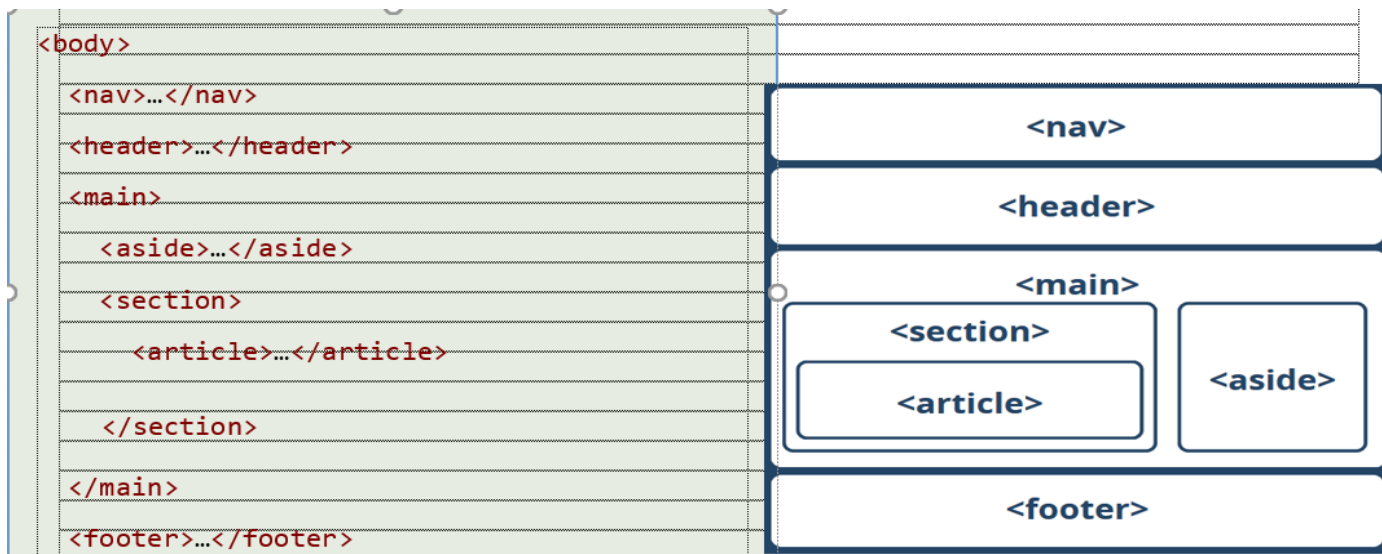
```
<html>
  <head> ... </head>
  <body>
    <div id="navigation">...</div>
    <div id="header">...</div>
    <div id="sidebar">...</div>
    <div id="content">...</div>
    <div id="footer">...</div>
  </body>
</html>
```



HTML5 way

Добавени са вече семантични тагове!

```
<nav>
<header>
<main>
<section>
<article>
<aside>
<footer>
```



`<aside>` tag

- използва се в новинарски сайтове и като реклама отстрани на съдържанието
`<aside>` – defines a sidebar (left/right navigation)

Semantic structure of tags

HTML document structure:

`<!DOCTYPE>`, `<html>`, `<head>`, `<body>` COPY

Semantic Structure Tags

`<nav>`, `<header>`, `<main>`, `<footer>`


Lists

``

``

`<ol reversed>`

<code></code>	
<code>Unordered list item 1</code>	
<code>Unordered list item 2</code>	
<code></code>	
<code></code>	
<code>Ordered list item 1</code>	
<code>Ordered list item 2</code>	
<code></code>	



HTML

←
→
🔧
🔄
🏠
ℹ️

- Unordered list item 1
- Unordered list item 2

1. Ordered list item 1
2. Ordered list item 2

```

<ol type="I">
  <li>
    HTML5
    <ol type="a">
      <li>markup language</li>
      <li>used for structuring and presenting content on the World Wide Web</li>
    </ol>
  </li>

  <li>
    HTML Versions
    <ol type="1" reversed>
      <li>цифри reversed</li>
      <li>
        HTML versions timeline
        <ul type="circle">
          <li>HTML draft version timeline</li>
          <li>XHTML versions</li>
        </ul>
      </li>
      <li>Markup</li>
    </ol>
  </li>

  <li>
    HTML5 Semantic Tags
    <ul type="disc">
      <li>article</li>
      <li>aside</li>
      <li>details</li>
      <li>figcaption</li>
      <li>figure</li>
      <li>footer</li>
      <li>header</li>
      <li>main</li>
    </ul>
  </li>
</ol>

```

```

        <li>mark</li>
        <li>nav</li>
        <li>section</li>
        <li>summary</li>
        <li>time</li>
    </ul>
</li>
</ol>

```

Definition list

```

<dl>
    <dt>Definition title</dt>
    <dd>Definition description</dd>
    <dt>Definition title</dt>
    <dd>Definition description</dd>
</dl>

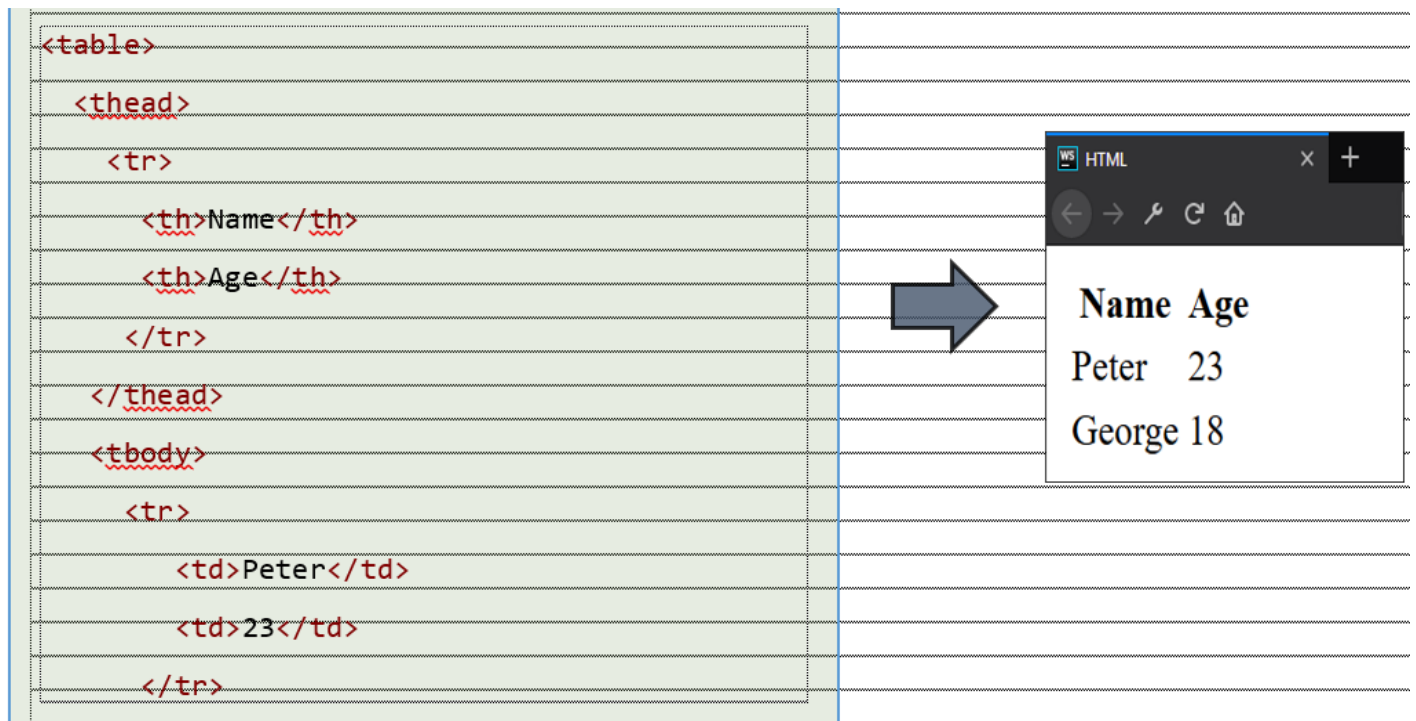
```

HTML Tables

Table header - **<thead>** group header content in an HTML table. Holds **<tr>** (table row) with **<th>** (table header) header cells

Table body - **<tbody>** group. Holds **<tr>** (table row) with **<td>** (table data) cells

Table footer - **<tfoot>** group footer content in an HTML table. Holds **<tr>** (table row) with **<td>** (table data) cells



```

<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Age</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Peter</td>
      <td>23</td>
    </tr>
  </tbody>
</table>

```

Name	Age
Peter	23
George	18

```

<table>
  <thead>
    <tr>

```

```

        <th>Name</th>
        <th>Mark</th>
    </tr>
</thead>

<!-- TODO: <tbody> -->

<tfoot>
    <tr>
        <td>Average</td>
        <td>4.12</td>
    </tr>
</tfoot>

</table>

```

Merge rows and columns

```

<td colspan="2">Sum: $180</td>
<td rowspan="2">Average: $150</td>

```



```

<table>
    <thead>
        <tr>
            <th colspan="4">Web Fundamentals</th>
        </tr>
    </thead>

    <tbody>
        <tr>
            <td class="bold">&#8470;</td>
            <!-- TODO: put the rest <td> here ... -->
        </tr>

        <!-- TODO: put the rest <tr> with <td> here ... -->
    </tbody>

    <tfoot>
        <tr><th colspan="4">Average score from 05 participants: 500</th></tr>
    </tfoot>

```

Web fundamentals exam result			
№	First name	Last name	Score
01	Gosho	Goshev	500
02	Tosho	Toshev	500
03	Gencho	Genchev	500
04	Draga	Draganova	500
05	Gosho	Goshev	500
Average score from 05 participants:500			

Section Element – section tag

```
<section>
  <h2>Heading</h2>
  
</section>
```

Division Element – div tag

Блоков/block елемент, който разделя

```
<div>
  <h3>This is Heading</h3>
  <p>This is paragraph</p>
  <p>This is paragraph</p>
</div>
```

Span element tag

Inline елемент

```
<p>I like:
  <span>C#, Java and JavaScript</span>
</p>
```

```
<div>This is a <span>span element</span> within a div element</div>
```

Линк <a> tag

```
<body>
  <a href="https://softuni.bg">SoftUni link</a>
</body>
```

Снимка – img tag

```

```


Figure + Figcaption tags

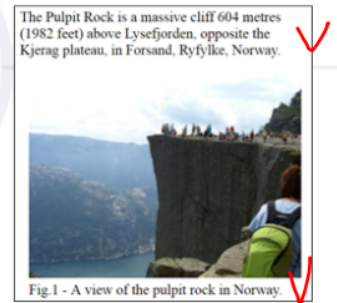
<figure> – element to mark up a photo in a document:

```
<p>The Pulpit Rock ...</p>
<figure>
  
</figure>
```



<figcaption> – a caption for a figure element

```
<p>The Pulpit Rock ...</p> ✓
<figure> ✓
  
  ✓ <figcaption>Fig.1 - A view ...</figcaption>
</figure>
```



Forms / Формуляри tag:

The action attribute is **used to specify where we want to send the form data when the form is submitted**. So the value of the action is the page that will process the form.

Пример 1

```
<body>
  <form action= "/orders/add" method="POST">
    <label for="fullName">Full name</label>
    <input id="fullName" type="text" name="fullName" /><br />

    <label for="language">Language</label>
    <select id="language">
      <!-- TODO: -->
      <option> tags </option></select>

    <label for="knowledge">Basic Level</label>
    <input id="knowledge" type="checkbox" name="knowledge" value="yes" /><br />

    <button type="submit">Submit</button>
  </form>
</body>
```

Пример 2 – всички видове

```
<form action="/orders/add" method="POST">
  <fieldset>    ограда в правоъгълник
    <legend>Customer Details</legend>
```

Autofocus при зареждане на страница там отива курсора първо – там е кликнато все едно!

```
<label for="fname">First Name <input type="text" id="fname" autofocus  
placeholder="George"></label><br>  
<label for="lname">Last Name <input type="text" id="lname"  
placeholder="Devis"></label><br>
```

Радиобутони:

```
<div>  
  <label for="male">Male <input type="radio" id="male" name="sex"></label>  
  <label for="female">Female <input type="radio" id="female" name="sex"></label>  
  <label for="other">Other <input type="radio" id="other" name="sex"></label>  
</div><br>  
  
<div>  
  <label>Preferred contact method:</label>  
  <label for="ml">Email <input type="radio" id="ml"  
name="preferred_contact_method"></label>  
  <label for="phn">Phone <input type="radio" id="phn"  
name="preferred_contact_method"></label>  
</div><br>  
  
  <label for="mail">E-mail <input type="email" id="mail"  
placeholder="your@email.com"></label><br>  
  <br>  
  <label for="phone">Phone <input type="tel" id="phone"  
placeholder="+359885236225"></label><br>  
  <br>
```

```
<div>  
  <label>Under 20</label>  
  <input type="radio" name="age">  
  
  <label>20 or more</label>  
  <input type="radio" name="age">  
</div>
```

Dropdown

```
<label for="town">Town  
  <select id="town" name="select_town">  
    <option value="varna" name="select_town">Varna</option>  
    <option value="plovdiv" name="select_town">Plovdiv</option>  
    <option value="burgas" name="select_town">Burgas</option>  
    <option value="sofia" name="select_town">Sofia</option>  
    <option value="pleven" name="select_town">Pleven</option>  
  </select>  
</label><br>  
  
<br>
```

```
    <label for="age">Age <input type="number" id="age" required min="18" max="70"
empty=""></label><br>
    <br>
```

HTML Data List element

```
    <label for="lan">Programming language
    <input list="languages" name="browser" id="lan">
    <datalist id="languages">
        <option value="Java"></option>
        <option value="PHP"></option>
        <option value="C#"></option>
        <option value="JavaScript"></option>
        <option value="Python"></option>
        <option value="Ruby"></option>
        <option value="Other"></option>
    </datalist>
    </label><br>

    <br>
    <label for="txtarea">
        <textarea id="txtarea" cols="20" rows="5" placeholder="More
information..."></textarea>
    </label><br>
```

Checkbox

```
    <br>
    <label for="terms"><input type="checkbox" id="terms">I agree the Terms and Conditions
</label><br>

    <br>
    <button type="submit">Send</button>
</fieldset>
</form>
```

To make a checkbox checked by default, you give it the **checked** attribute

```
<div>
    <input type="checkbox" id="scales" name="scales" checked>
    <label for="scales">Scales</label>
</div>
```

checkboxes allow you to turn single values on and allow multiple values to be selected.

Value - a string representing the value of the checkbox. This is not displayed on the client-side, but on the server. This is the value given to the data submitted with the checkbox's name. Take the following example:

```
<div>
    <input type="checkbox" id="coding" name="interest" value="coding" />
    <label for="coding">Coding</label>
</div>
<div>
    <input type="checkbox" id="music" name="interest" value="music" />
```

```
<label for="music">Music</label>
</div>
```

Details + Summary tags

Магията се случва от тага, а не чрез JS!!!

<details> – tag specifies additional details

```
<details>
  <p>More info about the details.</p>
</details>
```

<summary> – a visible heading for the **<details>**

```
<details>
  <summary>Some details</summary>
  <p>More info about the details.</p>
</details>
```

▼ Details

More info about the details.

► Some details

Time tag

```
<p>Open from <time>10:00</time> to <time>21:00</time> every weekday.</p>
<p>I have a date on <time datetime="2008-02-14 20:00">Valentines day</time>.</p>
```

The **datetime** attribute of this element is used to translate the time into a machine-readable format so that browsers can offer to add date reminders through the user's calendar, and search engines can produce smarter search results.

Address tag

<address> – contact information for site author / owner

```
<address>
  Street Address: Karlstraße 120<br/>
  Country Name: Germany<br/>
  Tel: +49 1234 5678<br/>
  Fax: +49 1234 5679
</address>
```

*Street Address: Karlstraße 120
Country Name: Germany
Tel: +49 1234 5678
Fax: +49 1234 5679*

Multimedia Context tags

<audio>, <video>

```
<audio controls autoplay>
  <source src="horse.mp3" type="audio/mpeg" />
  Your browser does not support the audio tag.
</audio>
```

```
<video controls="controls">
  <source src="shuttle.mp4" type="video/mp4" />
  Your browser does not support the HTML5 video.
</video>
```

Lorem

Когато искаме да добави текст от думи:
Lorem50 и се генерират 50 думи (с Toggle Wrap ги Wrap-ваме)

 tag

A new line in HTML

Character entities show special characters in HTML

Result	Entity Name	Entity Number
	 	
<	<	<
>	>	>
&	&	&
"	"	"
'	'	'
¢	¢	¢
£	£	£
¥	¥	¥
€	€	€
©	©	©
®	®	®

Incorrect HTML tag:

<some-Text-example>

Correct HTML tag:

<some-Text-example>

№ №

<footer id="footer">Svilen Velikov & SoftUni © 2022</footer>

Text formatting

 - bold има семантична стойност
 - bold
 - italic

Quotes – цитат

```
<cite></cite>
```

Multiple paragraph quotes – параграф цитат

```
<blockquote>  
  <p>Smart quote</p>  
</blockquote>
```

Siding термин

Когато 2 тага са на едно и също ниво, се нарича siding.

1.1.2. Tag attributes

Attributes provide additional information about HTML elements.

- Tags elements can have attributes
- Attributes provide additional information about an element
- Attributes are always specified in the start tag
- Attributes come in name/value pairs like: name="value"
- Always Quote Attribute Values. Attribute values should always be enclosed in quotes.
- Double style quotes are the most common, but single style quotes are also allowed.
- Be careful when combining single and double quotes, make sure you use only one type

Атрибути

```
<p id="myId"></p>
```

```
<div class="divElement" id="mainContainer">
```

In this case, the attributes will not affect the content of the div.
</div>

Атрибута class може да има много класове в себе си – разделени със space:

```
<button class="button fill hover">Learn More</button>
```

1.1.3. Metadata sections

The <meta> tag provides additional information about the HTML document.

<head> TAG

The <head> element is a container for all the head elements. Elements inside <head> can include scripts, instruct the browser where to find style sheets, provide meta information, and more.

The following tags can be added to the head section:

```
<title>,
```

```
<style>,  
<meta>,  
<link>,  
<script>,  
<noscript> - обикновено съобщение – примерно че този сайт не поддържа JS  
  <head>  
    <title>HTML Document title</title>  
  </head>
```

В по-старите версии на html

```
<!-- Define keywords for search engines: -->  
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">  
  
<!--Define a description of your web page:-->  
<meta name="description" content="Free Web tutorials on HTML and CSS">  
  
<!--Define the author of a page:-->  
<meta name="author" content="Hege Refsnes">  
  
<!--Refresh document every 30 seconds:-->  
<meta http-equiv="refresh" content="30">
```

`<link>`

The `<link>` tag defines the relationship between a document and an external resource.
The `<link>` tag is most used to link to style sheets.

```
<head>  
  <link rel="stylesheet" type="text/css" href="mystyle.css">  
</head>
```

`<style>`

The `<style>` tag is used to define style information for an HTML document.

Inside the `<style>` element you specify how HTML elements should render in a browser:

```
<head>  
  <style type="text/css">  
    p { color: #369; }  
  </style>  
</head>
```

1.4. Indentation and code formatting

We will be using tabs that are 4 spaces long for indentation.

1.2. CSS Intro

CSS stands for **Cascading Style Sheets**.

Styles define the visual presentation of HTML elements.

1.2.1. CSS syntax

- Every CSS document is a collection of CSS rules.
- CSS rule has two main parts - Selector and one or more declarations
- Each declaration consists of a property and a value.

CSS Rule

CSS rule has two main parts:

1. Selector
2. One or more declarations

```
[selector] {  
    [declaration]  
    [declaration]  
}
```

CSS Selector

The selector is an identifier of the HTML element or the group of HTML elements you want to style.

CSS Declaration

Declarations end with a semicolon, and declaration groups are surrounded by curly brackets.

```
body {  
    font: 16px/1.5 Verdana, sans-serif;  
    color: #333;  
}
```

1.2.2. THREE WAYS TO INSERT CSS

Цвят и размер на текста

I) Inline style – за конкретен елемент

An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly! **Умерено/пестеливо**

To use inline styles you use the style attribute in the relevant tag. The style attribute can contain any CSS property. The example shows how to change the color and the left margin of a paragraph:

```
<p style="color: red; font-size: 22px;">
```

II) Internal style – в head-а на някоя страница

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section of an HTML page

```
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>CSS Demo</title>  
    <style>  
        p {  
            color: yellow;  
        }
```



```
        font-size: 24px;
    }
</style>
</head>
<body> </ body >
```

III) External style – в отделен .css файл / за няколко страници наведнъж

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the tag. The tag goes inside the head section:

В частта Head:

```
<link rel="stylesheet" href="style.css">
```

Във файлът style.css пишем единствено и само:

```
p {
    color: yellowgreen;
    font-size: 24px;
}

h1 {
    color: red;
    font-size: 30px;
}
```

```
html {
    font-size: 16px;
}
```

По специфичното винаги бие по-общото.

Ctrl + F5 - hard re-load of the web page

1.2.3. CASCADE ORDER

External <link>

In the <head>

Inline style attribute

Using !important

По специфичното винаги бие по-общото.

Последната изредена CSS настройка печели. И тъй като ги чете отгоре надолу в style.css, то започваме от общите настройки за група елементи, и след това конкретна допълнителна настройка за някой/всеки от елементите!

1.2.4. Selector priority specificity

Когато се случи да сме написали html tag, който има и id атрибут, и class атрибут с 3 класа, то три пъти десет е по-малко от един път 100. Т.е. id атрибута има по-голяма тежест.

Когато обаче имаме 11 класа в class атрибута, и ги задаваме в style.css, ако зададем #header, то 110 е повече от 100 и тежест ще имат тези 11 класа каквито визуални CSS настройки сме им казали пред CSS чрез id атрибут.

Дано никога да не ни се наложи да използваме това.

Затова id селекторите е за предпочитане да не ги използваме за CSS – да не се чудим след това как да ги override-нем.

0	0	0	0
inline styles	# of id selectors	# of class selectors	# of element selectors
<code>p { color: #FFF; }</code>			0, 0, 0, 1
<code>.intro { color: #345678; }</code>			0, 0, 1, 0
<code>#header { color: #000; }</code>			0, 1, 0, 0
<code><p style="color: #000;">Text</p></code>			1, 0, 0, 0
<code>p { color: #000 !important; }</code>			God-mode

1.2.5. Selectors CSS

Primary Selectors

Select by Tag (p, h1, li, ul, ol...)

- selects all specified tags

Using the HTML tag names as selectors will apply styles to all tags in the document.

```
<span>Here's a span with some text.</span>
<p>Here's a p with some text.</p>
<span>Here's a span with more text.</span>
```

Във файлът style.css:

```
span {
  background: DodgerBlue;
  color: #ffffff;
}
```

text-decoration: none; - ако има ефекти върху текста, то ги премахни. Премахва и подчертаването.

Select by ID (#id)

- selects a unique element by ID (да го избягваме / опитът показва че не добре – по-горе обясних в Selector priority specificity)

The id selector uses the id attribute of the HTML element, and is defined with a "#".

Using the id selector will give you the exact element you are referring to.

Пример 1

В страницата, която искаме:

```
- <p id="special-quote">
```

Във файлът style.css:

```
#special-quote {  
    background-color: hotpink;  
}
```

Пример 2

```
<span id="top">Here's a span with some text.</span>
```

```
<span>Here's another.</span>
```

```
span#top {  
    background: DodgerBlue;  
}
```

Select by Class (.class)

- selects a group of elements with the specified class

The class selector uses the HTML class attribute, and is defined with a "." (dot)

The class selector allows you to set a particular style for many HTML elements with the same class.

Пример 1

В страницата, която искаме:

```
- <p class="quoted-text">
```

Във файлът style.css:

```
.quoted-text {  
font-style: italic;  
font-size: 18px;  
color: gray  
}
```

Пример 2:

You can also specify that only one HTML tag should be affected by a class.

```
<span class="sky">Here's a span with some text.</span>
```

```
<span>
```

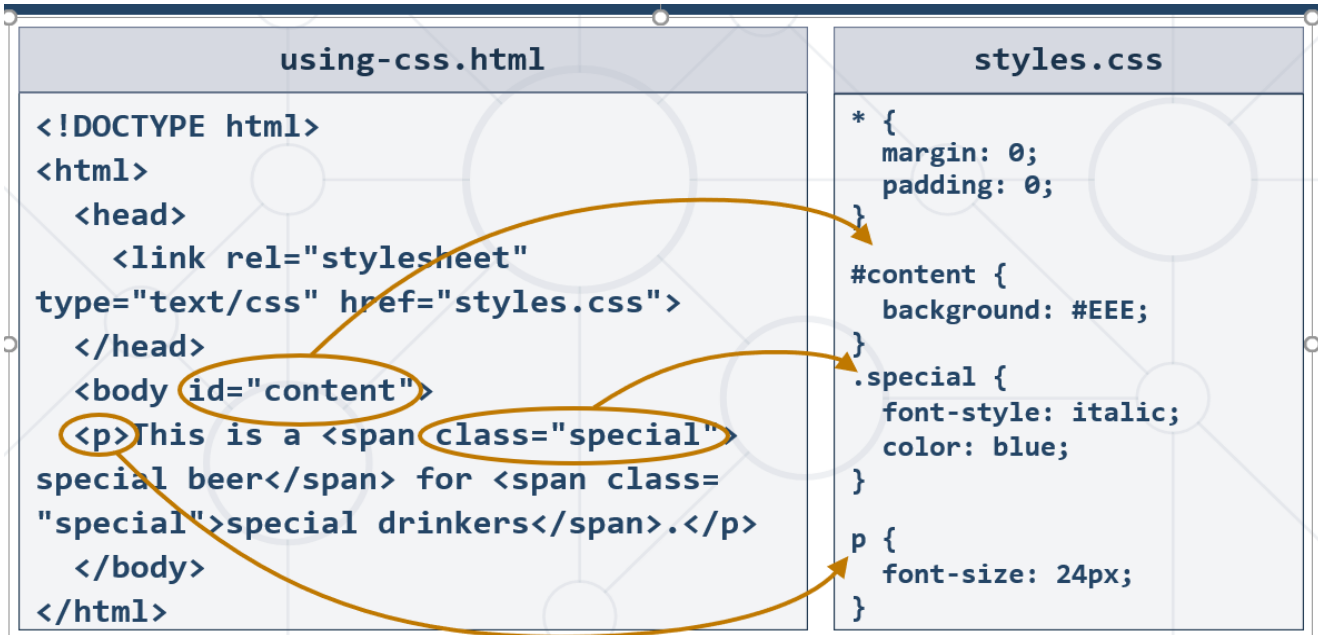
```
    Another <span class="code">&lt;span&gt;</span>.
```

```
</span>
```

```
span.sky {  
    background: DodgerBlue;  
}
```

```
.code {  
  font-family: Consolas;  
}
```

- * - selects everything – почти не се използва



Когато променяме кодът в брауъра (View page source), то промяната се помни докато не сме дали Refresh

Nested Selectors

CSS relies heavily on specificity and style overwriting.

Its in the name!

Cascading Style Sheets

Descendant

Tara `<div>` с клас `items`, всички подтагове `<a>` – хваща директни и по-надълбоко елементи `<a>` също.

```
<div class="items">
  <a href="#">Item1</a>
  <a href="#">Item2</a>
  <a href="#">Item3</a>
  <ul>
    <li><a href="#">Item4</a></li>
    <li><a href="#">Item5</a></li>
    <li><a href="#">Item6</a></li>
  </ul>
</div>
```

COPY



```
div.items a {
  color: green;
  font-weight: bold;
}
```

Adjacent Sibling

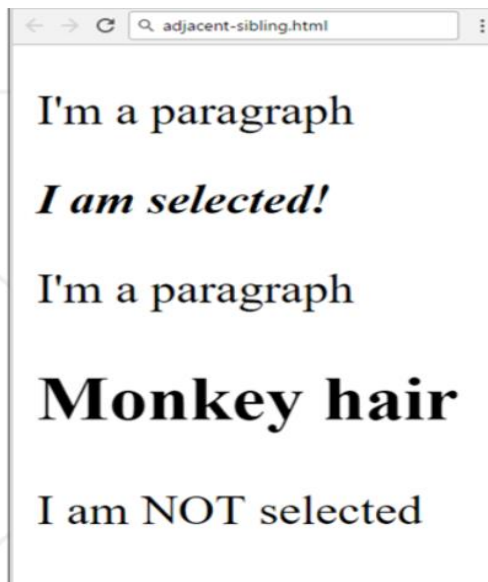
Един след друг <p> в първия <div>

p + p означава дай ми параграфи, **който следва веднага предишен параграф**. Т.е. между предишния параграф и настоящия да няма друг html елемент.

И след това до края – не хваща друг параграф в случая. А Monkey hair не е bold-нат, а просто изглежда по-голям.

```
<div>
  <p>I'm a paragraph</p>
  <p>I am selected!</p>
</div>
<div>
  <p>I'm a paragraph</p>
  <h2>Monkey hair</h2>
  <p>I am NOT selected</p>
</div>
```

COPY




```
p + p {
  font-style: italic;
  font-weight: bold;
}
```

Direct Child

Директен наследник само вземаме със знака > (по-голямо).

```
<div>
  <span>Span #1, in the div.
    <span>Span #2, in the span that's...</span>
  </span>
</div>
<span>Span #3, not in the div at all.</span>
```

```
div > span {
  background: DodgerBlue;
}
span { background: #fff; }
```



Span #1, in the div. Span #2,
in the span that's...
Span #3, not in the div at all.

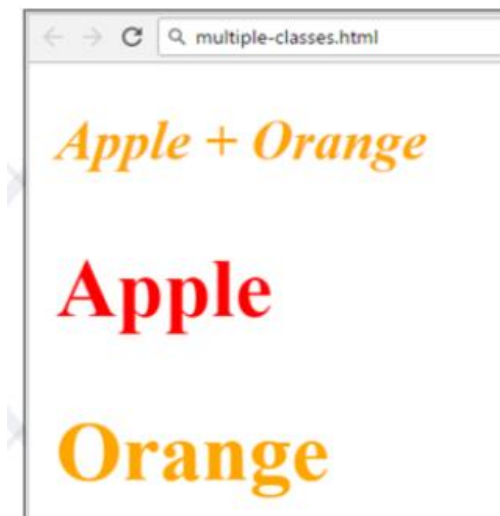
Multiple Classes

Последната изредена CSS настройка печели

```
<h2 class="apple orange small">Apple + Orange</h2> изреждаме класовете тук
<h2 class="apple">Apple</h2>
<h2 class="orange">Orange</h2>
```

Пример за class ="apple orange small"

```
.apple {
  color: red;    първо стани червен
}
.orange {
  color: orange; след това се промени на оранжев
}
.small {
  font-size: 16px; след това си смени фонт-а на 16px
}
.apple.orange {
  font-style: italic;    накрая стани Italic
}
```



Attribute Selectors

```
<ul>
  <li><a href="#">Home</a></li>
  <li><a href="#">Products</a></li>
  <li><a href="#" title="menu">Menu</a></li>
</ul>
```

```
a[title="menu"] {           в tag a с атрибут title да е равен на "menu"
  text-decoration: none;
  color: #962103;
  font-size: 22px;
}
```



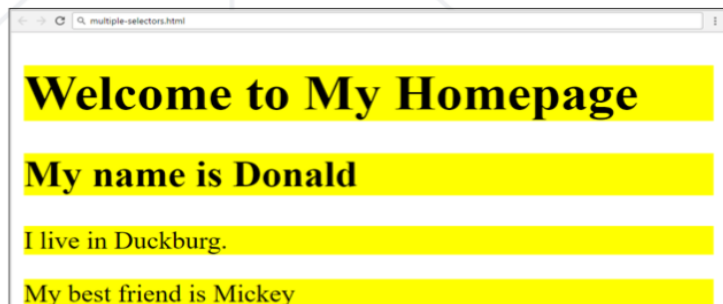
```
a[title] {                 в tag a с атрибут title без значение каква е стойността на title
  color:blue;
}
```

Multiple Selectors

Изреждаме със запетая в css за кои tag елементи да важи

```
<h1>Welcome...</h1>  
<h2>My name is...</h2>  
<p>I live in Duckburg.</p>  
<p>My best friend is...</p>
```

```
h1, h2, p {  
  background: yellow;  
}
```



Combining Multiple Selectors

Как можем да го селектираме по най-подробния начин.

```
<h1 id="header" class="intro">HTML and CSS</h1>
```

```
h1#header.intro {  
  text-decoration: underline;  
  color: #C00;  
}
```



Pseudo Selectors

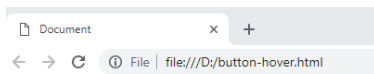
General syntax

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

:hover

```
<button>Select</button>
```

```
button:hover {  
  background: blue;  
  color: white;  
}
```

Select

:visited

```
<a href="#">Go to Google</a>
```



Go to Google

```
a:visited {  
  color: orange;  
}
```



Go to Google



:active

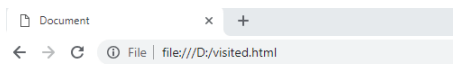
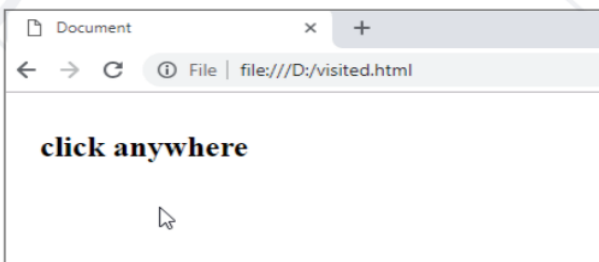
Когато сме клиkali и едновременно задържали бутон или друг html елемент

```
<h2>click anywhere</h2>
```



click anywhere

```
html:active {  
  background: grey;  
}
```

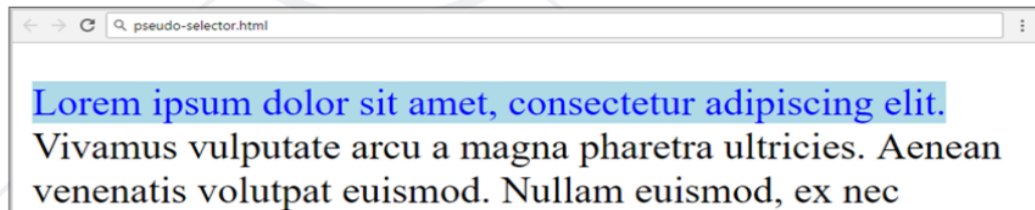


click anywhere

:first-line

```
<p>Lorem ipsum dolor  
sit amet, consectetur  
adipiscing elit.</p>
```

```
p:first-line {  
  color:blue;  
  background-color: lightblue;  
}
```



:before and :after

Начин за добавяне на елементи през CSS-a

```
<h1>CSS Pseudo Selectors</h1>
```



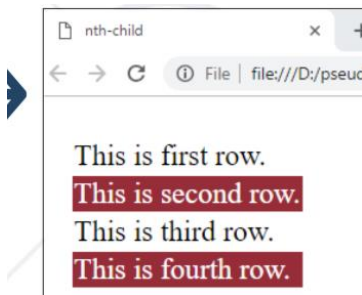
```
h1:before { /* "« " */  
  content: "\00AB\0020"  
}  
h1:after { /* " »" */  
  content: "\0020\00BB"  
}
```



:nth-child(n)

```
<table>  
  <tr><td>This is first row.</td></tr>  
  <tr><td>This is second row.</td></tr>  
  <tr><td>This is third row.</td></tr>  
  <tr><td>This is fourth row.</td></tr>  
</table>
```

```
tr:nth-child(2n) {  всеки втори параграф, като започва от втория  
  background: #95313b;  
  color: #fff;  
}
```



```
tr:nth-child(1) {
    background: #95313b;
    color: #fff;
}
tr:nth-child(3) {
    background: #95313b;
    color: #fff;
}
```

само първият параграф боядисай

само третия параграф боядисай

```
main section:nth-child(2) article{
    background: orange;
    color: white;
    width: 29%;
    display: inline-block;
    padding: 12px;
}
```

1.2.6. Box Model

Margin - Area **outside** the border. It is transparent by default

Padding - Area **inside** the border, around the content. It is transparent by default

Можем да задаваме margins и padding и само от една страна

element.style {

margin-left: 10px само отляво

margin-top: 10px само отгоре

margin: 0 без пиксел можеме

margin: 12px 19px първата цифра отгоре и отдолу едновременно, втората цифра отляво и

отдясно едновременно

margin: 4px 8px 12px 20px четирите посоки една след друга (горе-дясно-долу-ляво или top-right-

bottom-left)

```
element.style {
    padding: 10px 15px;
    padding-top: 10px;
    padding-right: 15px;
    padding-bottom: 10px;
    padding-left: 15px;
    margin: 0 0 100px 100px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 100px;
    margin-left: 100px;
}
```

border-radius: 7px закръгляне по ръбовете

```
    cursor: pointer като отидем върху бутона, да става ръчичка курсора
}
```

Borders - Border that goes around the content
Content - Where text and images appear

Може да записваме по 2 различни начина едно и също:

```
border-width: 1px;
border-style: solid;
border-color: yellow;
```

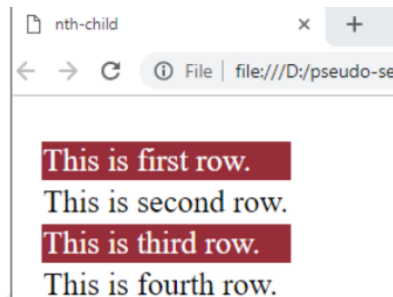
```
border: 1px solid yellow;
```

:nth-child(n+1)

Боядисай всички нечетни параграфи

```
<table>
  <tr><td>This is first row.</td></tr>
  <tr><td>This is second row.</td></tr>
  <tr><td>This is third row.</td></tr>
  <tr><td>This is fourth row.</td></tr>
</table>
```

```
tr:nth-child(2n+1) {
  background: #95313b;
  color: #fff;
}
```



Можем и така, с odd:

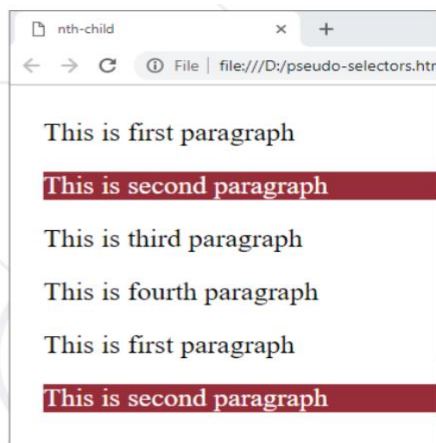
```
table tr:nth-child(odd) td {
  background: rgb(241, 241, 241);
}
```

:nth-of-type(n)

Тук работим с директни елементи, без child – хвани от първите последователни на едно ниво параграфи втория.

```
<p>This is first paragraph</p>
<p>This is second paragraph</p>
<p>This is third paragraph</p>
<p>This is fourth paragraph</p>
<div>
  <p>This is first paragraph</p>
  <p>This is second paragraph</p>
</div>
```

```
p:nth-of-type(2) {
  background: #95313b;
  color: #fff;
}
```

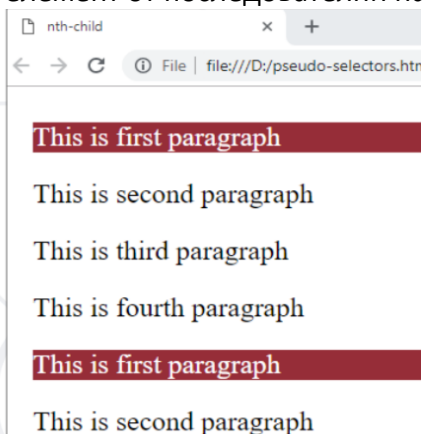


:first-of-type

Като предходното, но винаги взема първия елемент от последователни на едно ниво параграфи.

```
<p>This is first paragraph</p>
<p>This is second paragraph</p>
<p>This is third paragraph</p>
<p>This is fourth paragraph</p>
<div>
  <p>This is first paragraph</p>
  <p>This is second paragraph</p>
</div>
```

```
p:first-of-type {
  background: #95313b;
  color: #fff;
}
```

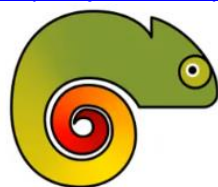


1.2.7. CSS tricks

<https://css-tricks.com/pseudo-class-selectors/>

Just Color Picker for Windows – хваща цвета

<https://just-color-picker.en.softonic.com/>

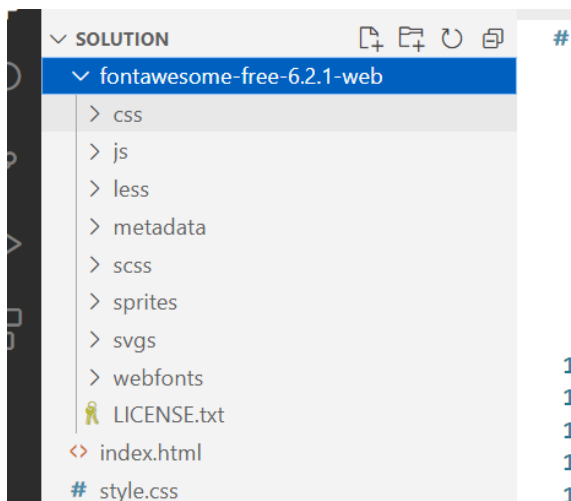


<https://fontawesome.com/> - добавяне на иконки на footer-а примерно

Регистрираме се,

I. Вкарваме fontawesome в нашия проект

A. Можем да си го свалим безплатно локално



В <head> на html кода слагаме следния link

```
<link rel="stylesheet" href="./fontawesome-free-6.2.1-web/css/all.css">
```

В. Или да ползваме Kit, който ни предоставя линк, който можем да вкараме в html/css кода.

В <head> на html кода слагаме следния скрипт

```
<script src="https://kit.fontawesome.com/aadeb913df.js" crossorigin="anonymous"></script>
```

От известно време е трудно да вкараме през CDN от сайта css нещата директно в нашия typography.css файл.

Затова за import в CSS, то варианта е следния:

Сваляме си съгласно точка А. fontawesome-a, и след това пишем само в нашия css файл:

```
@import url("./fontawesome-free-6.2.1-web/css/all.css");
```

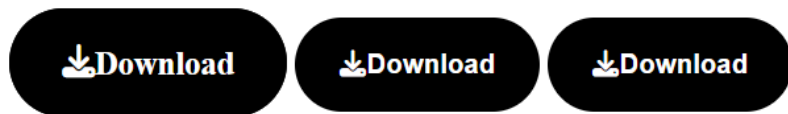
Така judge ще ни го признае за верно.

II. Харесваме си от сайта съответната иконка

Вземаме html кода за иконката и го пействаме където искаме

```
<i class="fa-solid fa-download"></i>
```

```
<a class="button fill icon" href="#"><i class="fa-solid fa-download"></i>Download</a>
```



1.2.8. Block Elements

1) Block Items – на нов ред:

```
<ul>
  <li><a href="about.html">Home</a></li>
  <li><a href="about.html">Items</a></li>
  <li><a href="about.html">About</a></li>
  <li><a href="about.html">Contacts</a></li>
</ul>
```

```
<p>
<h1>
<div>
```

2) Inline items – на същия ред:

```
<span>
```

3) Inline-Block items:

```
.navigation li {
  display: inline-block;
}
```

Rectangles arranged one after another - Just like words in a sentence

В CSS файла можем да пре-дефинираме елементи – да станат от Inline на Block

Във файла style.css:

```
.slogan {
  font-weight: bold;
  display: block;
}
```

```
.navigation li {
  display: inline;
}
```

1.2.9. Hover

Подчертава линка `Home`

```
.navigation a{
  text-decoration: none;
  color: gray;
}

.navigation a:hover {
  text-decoration: underline;
}
```

```
.navigation a{
  text-decoration: none;
  color: white;
  display: inline-block;
  background-color: red;
  padding: 5px 10px;
  margin: 3px;
  border-radius: 10px;
  border: 1px solid red;
}
```

```

}

.navigation a:hover {
  background-color: transparent;
  color: red;
  transform: scale(1,1);
}

```

hovering

- когато курсора на мишката минава над бутона

```

.nav-link:hover {
  color: #d70026;
}

```

1.2.10. Link styles

```

a:link      { color: #369; } /* unvisited link */
a:visited   { color: #147; } /* visited link */
a:hover     { color: #58B; } /* mouse over link */
a:active    { color: #C00; } /* selected link */

```

1.2.11. Fonts

In CSS, there are two types of font family names:

- generic family** - a group of font families with a similar look
- font family - a **specific font family**

Във файла style.css:

```

body {
  font-family: sans-serif;
}

```

Слагане на шрифт от Google - <https://fonts.google.com/> чрез CDN(Content Delivery Network)

Линкът се генерира от google fonts.

Има опция от google fonts да си вземем/заредим само определени символи/група от символи(цифрите примерно), а не целия шрифт.



В страницата, която искаме:

```
<link href="https://fonts.googleapis.com/css2?family=Lato:wght@100&display=swap" rel="stylesheet">
```

Във файлът style.css:

```
@import url('https://fonts.googleapis.com/css2?family=PT+Sans&family=PT+Serif&display=swap');  
body {  
    font-family: 'Lato', sans-serif;  
}
```

```
p {  
    font-family: "Times New Roman", Times, serif;  
}
```

The font-family property should hold several font names as a "fallback" system.

End with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available

Serif шрифтове са тези с ченгелчетата.



Sans-serif – без ченгелчета.

Generic family	Font family	Description
Serif (не се използва за сайтове!!!)	Times New Roman Georgia	Have small lines at the ends on some characters
Sans-serif (за уеб стават)	Arial Verdana	Do not have the lines at the ends of characters
Monospace (всяка буква заема едно и също място)	Courier New Lucida Console	All characters have the same width

1.3 Landing page – a demo project

index.html

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

```
<head>
```

```
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

<title>KidsCorney</title>
<script src="https://kit.fontawesome.com/aadeb913df.js" crossorigin="anonymous"></script>
<link rel="stylesheet" href="style.css">
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
</head>

<body>
  <header>
    <section class="navigation">
      <p>kids<span><i class="fas fa-futbol"></i>rner</span></p>
      <nav>
        <ul>
          <li><a href="#">ABOUT US</a></li>
          <li><a href="#">OFFERINGS</a></li>
          <li><a href="#">CAMPS</a></li>
          <li><a href="#">BLOG</a></li>
          <li><a href="#">CONTACT US</a></li>
          <li>
            <ul>
              <li><a href="#"><i class="fa-brands fa-facebook"></i></a></li>
              <li><a href="#"><i class="fa-brands fa-twitter"></i></a></li>
              <li><a href="#"><i class="fa-brands fa-instagram"></i></a></li>
            </ul>
          </li>
        </ul>
      </nav>
    </section>

    <section class="site-header">
      <h1>a comfort corner for you child</h1>
      <button>LEARN MORE</button>
    </section>
  </header>

  <main>
    <section class="welcome">
      <h2>Welcome to Kids Corner</h2>
      <p>The Heart's Kids Club provides lots of fun activities with an educational twist. Kids can join for free and will receive a goody bag and a membership card on induction. Each workshop will be interactive, educational, and will introduce children to important concepts that they will really enjoy. Children must be accompanied by a responsible adult at all times and workshop places are available on a first come, first served basis.</p>
      <p>If you are parent looking for a place where your child will grow the best, look further, talk to us now.</p>
    </section>
  </main>

```

```

<section class="offers">
  <h2>Our offerings</h2>
  <ul>
    <li><a href="">
      
      <h3>School Pickups</h3>
      <p>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</p>
    </a></li>
    <li><a href="">
      
      <h3>Extensive Library</h3>
      <p>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</p>
    </a></li>
    <li><a href="">
      
      <h3>Outdoor Trips</h3>
      <p>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</p>
    </a></li>
    <li><a href="">
      
      <h3>Healthy Food</h3>
      <p>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</p>
    </a></li>
    <li><a href="">
      
      <h3>First Aid Services</h3>
      <p>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</p>
    </a></li>
    <li><a href="">
      
      <h3>Games and Activities</h3>
      <p>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</p>
    </a></li>
  </ul>
</section>
<section class="camps">
  <h2>Upcoming CAMPS</h2>
  <p>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</p>
  <ul>
    <li>
      
      <section class="info">
        <div>

```

```

        <h3>Soccer Camp</h3>
        <p>15h February 2019</p>
    </div>
    <div>
        $35
    </div>
</section>
<section class="more-info">
    <p>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</p>
    <button>ENROLL TODAY</button>
</section>
</li>
<li>
    
    <section class="info">
        <div>
            <h3>Soccer Camp</h3>
            <p>15h February 2019</p>
        </div>
        <div>
            $35
        </div>
    </section>
    <section class="more-info">
        <p>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</p>
        <button>ENROLL TODAY</button>
    </section>
</li>
<li>
    
    <section class="info">
        <div>
            <h3>Soccer Camp</h3>
            <p>15h February 2019</p>
        </div>
        <div>
            $35
        </div>
    </section>
    <section class="more-info">
        <p>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</p>
        <button>ENROLL TODAY</button>
    </section>
</li>
</ul>
</section>
</main>

```

```

<footer>
  <p>CopyRight 2014. All Rights Reserved.</p>
  <ul>
    <li><a href="#"><i class="fa-brands fa-facebook"></i></a></li>
    <li><a href="#"><i class="fa-brands fa-twitter"></i></a></li>
    <li><a href="#"><i class="fa-brands fa-instagram"></i></a></li>
  </ul>
</footer>

</body>

</html>

```

style.css

```

@import url('https://fonts.googleapis.com/css2?family=PT+Sans&family=PT+Serif&display=swap');

body {
  font-family: 'PT Sans', sans-serif;
  font-size: 16px;
  max-width: 1300px;
  margin: 0 auto;
}

.navigation {
  padding: 0 12px;
}

.navigation p {
  display: inline-block;
  font-size: 22px;
  width: 730px;
}

.navigation p span {
  color: #50b9e1;
}

nav {
  display: inline-block;
}

li {
  display: inline-block;
}

li:not(:last-child){
  padding-right: 10px;
}

```

```

a {
    text-decoration: none;
    color: black;
}

nav i {
    border: 1px solid black;
    border-radius: 50%;
    padding: 5px;
}

body>header>section.navigation>nav>ul>li:nth-child(5) {
    padding-right: 5 px;
}

ul {
    padding: 0cm;
}

a:hover {
    color: #50b9e1;
    cursor: pointer;
}

/*site-header*/
.site-header {
    background: linear-gradient(rgba(0, 0, 0, 0.3), rgba(0, 0, 0, 0.6)), /*red green blue
alpha(transparent)*/
    url("../Images/kids_main.jpg"), no-repeat, center;
    /*ако е по-голямо пространството, да не се повтаря снимката*/
    background-size: cover;
}

.site-header h1 {
    font-size: 66px;
    color: white;
    width: 38%;
    margin: 0;
    padding: 200px 0 80px 100px;
}

.site-header button {
    font-size: 26px;
    background: transparent;
    border: 2px solid white;
    color: white;
    padding: 10px 15px;
    margin: 0 0 100px 100px;
}

```

```
.site-header button:hover {
    background: #50b9e1;
    border: 2px solid #50b9e1;
    cursor: pointer;
}

/*welcome*/
.welcome {
    color: #5a7c8a;
    width: 60%;
    margin: 0 auto;
    /*Центрирай секцията по средата*/
    padding: 20px 0;
}

h2 {
    text-align: center;
    font-family: "PT Serif", serif;
    font-size: 32px;
    font-style: italic;
    letter-spacing: 2px;
    /*разстоянието между самите букви да бъде 2 пиксела*/
}

.welcome p {
    text-align: center;
}

/*offers*/
.offers {
    display: inline-block;
    color: white;
    font-family: "PT Serif", serif;
    background: #50b9e1;
    padding: 20px 30px;
}

.offers img {
    width: 70px;
}

.offers li {
    width: 28%;
    text-align: center;
    padding: 20px 30px;
}

.offers a {
    color: white;
```

```

}

/*camps*/
.camps {
    color: #5a7c8a;
    padding: 20px 0;
}

body>main>section.camps>p {
    width: 60%;
    text-align: center;
    margin: 0 auto;
    padding-bottom: 30px;
}

.camps img {
    width: 400px;
    height: 250px;
}

.camps li {
    width: 400px;
}

.camps li:not(:last-child) {
    /*на всички li-та с изключение на последното li*/
    padding-right: 42px;
}

.info div {
    display: inline-block;
}

.info div p .inf div h3 {
    margin: 0;
    padding: 0;
}

.info div:nth-child(1) {
    width: 330px;
    /*отблъскваме десния елемент, като правим левия с по-голям div*/
}

.info div:nth-child(2) {
    background: #50b9e1;
    padding: 10px;
    color: white;
    font-size: 22px;
}

```



```
.info {
    border-bottom: 2px solid #5a7c8a;
    padding-bottom: 10px;
}

.more-info button{
    border: 2px solid #50b9e1;
    color: #50b9e1;
    background: transparent;
    padding: 5px 10px;
    font-size: 18px;
}

.more-info button:hover {
    background: #50b9e1;
    color: white;
    cursor: pointer;
}

/*footer*/
footer {
    background: #50b9e1;
    color: #5a7c8a;
    padding: 0 40px 0 80px; /*избутва навътре съдържанието отляво и отдясно*/
}

footer p {
    display: inline-block; /*и на двете слагаме inline block, за да се получи на един ред*/
    width: 88%; /*можем да избутваме и с процент*/
}

footer ul {
    display: inline-block; /*и на двете слагаме inline block, за да се получи на един ред*/
}

footer ul a {
    color: #5a7c8a;
}

footer ul a i {
    border: 1px solid #5a7c8a;
    border-radius: 50%;
    padding: 5px;
}
```

1.4. Демо HTML с използване на JavaScript

```
<body>
  <label for="userName">Enter userName</label>
  <input type="text" id="userName" name="userName">

  <label for="password">Enter Password</label>
  <input type="text" id="password" name="password">

  <input type="submit" name="submit" value="submit" onclick="login()">
</body>
<script>
  function login() {
    let correctUser = "Todor";
    let correctPassword = "123456";
    let userName = document.getElementById("userName").value;
    let password = document.getElementById("password").value;
    let isValid = false;
    if (userName === correctUser && password === correctPassword) {
      isValid = true;
    }
    if (isValid) {
      window.alert("The user is correct"); // pop up window
    } else {
      window.alert("The user and/or password is incorrect"); // pop up window
    }
  }
</script>
```

2. HTML5 Semantics Structure

2.1. SEMANTIC HTML

What is semantic HTML?

Semantic HTML or semantic markup is HTML that introduces meaning to the web page rather than just presentation. For example,

 a tag indicates that the enclosed text is a paragraph. This is both semantic and presentational, because people know what paragraphs are and browsers know how to display them.

Why you should care about semantics

The benefit of writing semantic HTML stems from what should be the driving goal of any web page — the desire to communicate

By adding semantic tags to your document, you provide additional information about that document, which aids in communication.

Specifically, semantic tags make it clear to the browser what the meaning of a page and its content is. That clarity is also communicated with search engines, ensuring that the right pages are delivered for the right queries.

Use semantic tags correctly

When you want to use semantic tags to convey meaning rather than for presentation purposes, you need to be careful that you don't use them incorrectly simply for their common display properties.

2.2. Tags

<header>

Represents introductory content, typically a group of introductory or navigational aids. It may contain some heading elements but also other elements like a logo, a search form, an author name, and so on.

<nav>

Represents a section of a page whose purpose is to provide navigation links, either within the current document or to other documents. Common examples of navigation sections are menus, tables of contents, and indexes.

<main>

Represents the dominant content of the 'body' of a document. The main content area consists of content that is directly related to or expands upon the central topic of a document, or the central functionality of an application.

<aside>

Represents a portion of a document whose content is only indirectly related to the document's main content. Asides are frequently presented as sidebars or call-out boxes.

<footer>

Represents a footer for its nearest sectioning content or sectioning root element. A footer typically contains information about the author of the section, copyright data or links to related documents.

<article>

Represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable (e.g., in syndication). Examples include: a forum post, a magazine or newspaper article, or a blog entry.

<figcaption>

Represents a caption or legend for the rest of the contents its parent 'figure' element, if any. – за снимка/фигура примерно

<figure>

Represents self-contained content, frequently with a caption 'figcaption' in it, and is typically referenced as a single unit.

<mark>

Represents text which is marked or highlighted for reference or notation purposes, due to the marked passage's relevance or importance in the enclosing context.

<section>

Represents a standalone section — which doesn't have a more specific semantic element to represent it — contained within an HTML document. Typically, but not always, sections have a heading. Section has articles.

<time>

Represents a specific period in time. It may include the datetime attribute to translate dates into machine-readable format, allowing for better search engine results or custom features such as reminders.

<details>

Отваря по-малката част от текста

Creates a disclosure widget in which information is visible only when the widget is toggled into an "open" state. A summary or label can be provided using the 'summary' element.

<summary>

Отваря по-голямата част от текста

specifies a summary, caption, or legend for a 'details' element's disclosure box. Clicking the 'summary' element toggles the state of the parent 'details' element open and closed.

самозатравящ се таг

В HTML5 можем и да не използваме самозатварящ таг, т.е. можем да не затваряме тага.

```

```

```

```

```

```

2.3. More metadata

Microdata

WHATWG HTML specification used to nest metadata within existing content on web pages.

Search engines, web crawlers, and browsers can extract and process Microdata from a web page and use it to provide a richer browsing experience for users.

Search engines benefit greatly from direct access to this structured data because it allows them to understand the information on web pages and provide more relevant results to users.

Schema.org

<https://schema.org/>

Schema.org is a collaborative, community activity with a mission to create, maintain, and promote schemas for structured data on the Internet, on web pages, in email messages, and beyond.

Example

```
<section>
```

```
    Hello, my name is Konstantin Dankov,
```

I am a trainer at SoftUni.
My friends call me Johnny.
You can visit my homepage at <https://dankov.me>.
I live at 123123, Kamelia str, Sofia.

</section>

Несемантичност: - нямаме <p> и <h1>
<https://schema.org/Person>

```
<section itemscope itemtype="http://schema.org/Person">
  Hello, my name is <span itemprop="name">Konstantin Dankov</span>
  I am a <span itemprop="jobTitle">trainer</span> at <span
itemprop="affiliation">SoftUni</span>.
  My friends call me <span itemprop="additionalName">Koko</span>.
  You can visit my homepage at <a href="https://dankov.me" itemprop="url">https://dankov.me</a>.
  <section itemprop="address" itemscope itemtype="http://schema.org/PostalAddress">
    I live at
      <span itemprop="streetAddress">123123</span>,
      <span itemprop="addressLocality">Kamelia str</span>,
      <span itemprop="addressRegion">Sofia</span>.
  </section>
</section>
```

2.4. Html entities

https://www.w3schools.com/html/html_entities.asp
<https://developer.mozilla.org/en-US/docs/Glossary/Entity>

<p>© Copyright 2008-2023</p>

<http://jekyllrb.com/>

3. CSS and Typography

3.1. What is typography?

What is typography?

Техниката за подреждане на текст

- Typography is the art and technique of arranging type to make written language legible, readable, and appealing when displayed.
- Typography is the visual component of the written word
- Style or appearance of text
- The art of working with text

https://en.wikipedia.org/wiki/The_Elements_of_Typographic_Style

<http://webtypography.net/>

Typography and the web

For too long typographic style and its accompanying attention to detail have been overlooked by website designers, particularly in body copy. In years gone by this could have been put down to the technology, but now the web has caught up. The advent of much improved browsers, text rendering and high resolution screens, combine to negate technology as an excuse.

The Elements of Typographic Style Applied to the Web Introduction

Choose a comfortable measure

The measure is the number of characters in single line of a column of text. HTML doesn't have a concept of columns per se, instead text is held within boxes. In CSS the width of a box is set using the width property with any unit of length

Anything from 45 to 75 characters is widely regarded as a satisfactory length of line for a single-column page set in a serifed text face in a text size. **The 66-character line (counting both letters and spaces) is widely regarded as ideal.** For multiple column work, a better average is 40 to 50 characters.

Choose a basic leading

Leading (pronounced "ledging" олово) is so called because, in mechanical presses, strips of lead are placed between lines of type to space the lines apart. Leading is achieved in css through the **line-height property** – разстояние между редовете.

```
body {  
    font-size: 18px;  
    padding: 50px 100px;  
    line-height: 1.5;  
    font-family: Arial, Helvetica, sans-serif;  
}
```

3.2. Fonts and Font families

Font family

In typography, a font family (also known as typeface) is a set of one or more fonts each composed of glyphs that share common design features. Each font of a typeface has a specific weight, style, condensation, width, slant, italicization, ornamentation, and designer or foundry.

Font

A computer font (or font) is implemented as a digital data file containing a set of graphically related glyphs, characters, or symbols such as dingbats. Although the term font first referred to a set of movable metal type pieces in one style and size, since the 1990s it is generally used to refer to a set of digital shapes in a single style, scalable to different sizes.

Generic font families

- **Serif** – с ченгелчета в края на буквата
- **sans-serif** – за web страници подходящи
- **monospace** – еднаков разстояние ширина всяка буква
- **cursive** - ръкописен
- **fantasy** – по-странни шрифтове

Web fonts

A technique to refer to and automatically download remote fonts was first specified in the CSS2 specification, which introduced the `@font-face` construct. At the time, fetching font files from the web was controversial because fonts meant to be used only for certain web pages could also be downloaded and installed in breach of the font license

Web fonts - EOT

Microsoft first added support for downloadable EOT fonts in Internet Explorer 4 in 1997.

EOT custom font файлове, които се качват. Добри, но не се е приела тази технология.

Web fonts – WOFF (Web Open Font Format)

In 2010, the WOFF compression method for TrueType and OpenType fonts was submitted to W3C by the Mozilla Foundation, Opera Software and Microsoft, and browsers have since added support.

Специален софтуер, който не пречи да крадем и дисплейваме различни видове шрифтове – платени, а и вече и безплатни.

Трябва да се внимава – ако шрифта е 5Mb, то не е добре заради размера.

Слагане на шрифт от Google - <https://fonts.google.com/>

Виж в 1.2.11 Fonts за повече инфо.

<https://fonts.adobe.com/> - тук са платени

3.3. CSS Units

CSS Units and Values

- Numeric values
- Percentages
- Colors
- Coordinate positions
- Functions

Numeric values – absolute units

- px: Pixels
- mm, cm, in: Millimeters, centimeters, or inches.
- pt, pc: Points (1/72 of an inch) or 1 picas(12 points.)

Numeric values – relative units

- em: 1em is the same 100% as the font-size of the current element
- rem: The rem (root em)
- ex, ch: Respectively these are the height of a lower case x, and the width of the number 0
- pt, pc: Points (1 point = 1/72 of an inch) or picas (1 pica = 12 points)
- vw, vh: Relative to the Viewport - width and height

Numeric values – unitless values

- margin, padding
- line-height

EM & REM

An explanation of EMS

Ems are so-called because they are thought to approximate the size of an uppercase letter M, although 1 em is actually significantly larger than this.

Bringhurst describes the em thus:

The **em** is a sliding measure. One em is a distance **equal to the type size**. In 6 point type, an em is 6 points; in 12 point type an em is 12 points and in 60 point type an em is 60 points. Thus a one em space is proportionately the same in any size.

em е релативна стойност към текущо избрания елемент font-а му.

Искаме глобално да работи проекта, с глобални настройки/дефиниции – от общото към частното. И си слагаме след това (**в css чете отгоре надолу: по-надолу по-специфични**) по-специфични css дефиниции за определени html елементи. И ако променим началната настройка за размер на шрифта от 12px на 16px, то всички елементи да се преизмерят на базата на **em**.

Настройки се сетват чрез **em** от текущия шрифт на дадения елемент, докато **rem** винаги взема от шрифта на **root елемента** (т.е. взема от html елемента шрифта – примера по-долу root фонта е 12px:)

Demo for em

typography.css

```
html { когато използваме тук body вместо html, то не на всички браузъри работи правилно.
  font: 12px/1.5 Georgia, Arial, Helvetica, sans-serif; font обединява в себе си долните три
  /* font-size: 28px;
  line-height: 1.5;
  font-family: Arial, Helvetica, sans-serif; */
}

/* сменяме scale всички шрифтове - спрямо големината на екрана */
@media (min-width: 800px) {
  html { font-size: 14px;}
}
@media (min-width: 1024px) {
  html { font-size: 16px;}
}
@media (min-width: 1280px) {
  html { font-size: 18px;}
}

/* Дефинираме шрифта глобално, и там където се налага сетваме на друг шрифт */
body, form, table, input, textarea, option, select {
  font: inherit; /*наследи го от родителя си/ от главния HTML*/ - иначе всеки отделен елемент
  може да има друг дефолтен шрифт
}

* {
  font: inherit; /*тук наследява всички възможни елементи от родителя си/ от главния HTML*/
}
```



```

p {
    margin: 0;
    padding: 0 0 1.5em 0;
}

h1, h2, h3, h4, h5, h6 {
    font-family: Georgia, 'Times New Roman', Times, serif;
    margin: 0;
    padding-bottom: 1em; /* за h1 ще е font-size-а на h1 (който е 2.8em);;; за h2 обаче ще е
font-size-а на h2 (който е 2.2em) */
    border: 1px solid #CCC;
}

h1 {font-size: 2.6em;}
h2 {font-size: 2.1em;}
h3 {font-size: 1.8em;}
h4 {font-size: 1.6em;}
h5 {font-size: 1.4em;}
h6 {font-size: 1.2em;}

p + h1,
p + h2,
p + h3,
p + h4,
p + h5,
p + h6 {
    padding-top: 0.5em;
}

body {
    padding: 50px 100px;
}

```

index.html

```

<html lang="en">
<head>
    <meta charset="UTF-8"><meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="typography.css">
    <title>CSS units</title>
</head>
<body>
    <h1>Main title brr</h1>

    <p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Nobis dolore facere eos quis ipsa
rerum quia amet illo, cupiditate animi, accusantium incidunt. Eum fugit optio id sed delectus
incidunt, culpa accusamus et quod</p>

    <h2>Secondary level title</h2>

```

```
<p>Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dolorum provident ab, reprehenderit culpa consequuntur molestiae quae repellendus eum qui possimus.</p>
```

```
<h3>Secondary level title</h3>
</body>
</html>
```

пиксели(px)/ проценти(%) / em/rem

Кога ползваме:

- пиксели(px) – почти никога, освен font на root-а 😊
- проценти(%) – много рядко, примерно 3 колони всяка по 33,33% от размера на подадения екран
- em/rem – почти винаги

buttons задачата

index.html

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

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- <script src="https://kit.fontawesome.com/aadeb913df.js"
crossorigin="anonymous"></script> -->
  <!-- <link rel="stylesheet" href="../fontawesome-free-6.2.1-web/css/all.css"> -->
  <link rel="stylesheet" href="buttons.css">
  <title>Buttons CSS</title>
</head>

<body>
  <h2>Normal Buttons</h2>
  <a class="button" href="#">Learn More</a>
  <button class="button">Learn More</button>
  <button class="button hover">Learn More</button>

  <h2>Filled Buttons</h2>
  <a class="button fill" href="#">Learn More</a>
  <button class="button fill">Learn More</button>
  <button class="button fill hover">Learn More</button>

  <h2>Icon buttons</h2>
  <a class="button fill icon" href="#"><i class="fa-solid fa-download"></i>Download</a>
  <button class="button fill icon"><i class="fa-solid fa-download"></i>Download</button>
  <button class="button fill icon hover"><i class="fa-solid fa-
download"></i>Download</button>

</body>
```

```
</html>
```

buttons.css

```
@import url("../fontawesome-free-6.2.1-web/css/all.css");

.button {
  /* display: inline-block;
  margin-right: 0.5em; */
  outline: none;
  /* border: 2px solid #000; */
  background: transparent;
  text-decoration: none; /* няма подчертавка на текста */
  color: inherit; /* цвета е дефалтно нормален на html root елемента */
  border-radius: 2em;
  padding: 1em 1.5em;
  font-weight: bold;
  /* text-transform: capitalize lowercase uppercase; */
  cursor: pointer; /* винаги има ръчичка курсора като посочим елементите с клас button*/
}

.hover {
  color: rgb(51, 51, 51);
}

.button.hover:hover {
  border-color: rgb(0, 102, 0);
  color: rgb(0, 102, 0);
}

.button.fill {
  /* ако използваме само background, то автоматично се пре-дефинират всички пропъртита, които съдържа background. Добра практика е да използваме конкретно само кое нещо сменяме от background - в случая променяме само background-color */
  background-color: rgb(51, 51, 51);
  color: white;
}

.button.fill.hover {
  color: rgb(251, 251, 251);
}

.button.fill.hover:hover {
  background-color: rgb(0, 102, 0);
  color: rgb(251, 251, 251);
}

.button.icon i {
  padding-right: 0.5em;
```

}