Introduction to Css

**1. CSS Selectors • Apply CSS Rules**

Името CSS е съкращение от Cascading Style Sheets. Идеята е, че CSS пряко работи с HTML и по някакъв начин украсява това, което имаме.

Последнарта версия CSS4 е една доста напреднала версия.

Как е базирана работата на CSS с HTML?

- Това е на базата на тези CSS селектори.

CSS Selector означава ние да вземем един HTML елемент и да му придадем някакви стилизации. Как го взимаме - като го селектираме.

Селектирането с най-най основните селектори се случва по няколко начина.

**2. Primary Selectors** **• Define the Elements to Which a Set of CSS Rules Apply**

Primary Selector-ите са най-обикновените и прости селектори, които общо взето могат да са например по таг.

В предната лекция, когато свързахме html-а и css-a като файлове, първото нещо което направихме за да проверим дали са свързани е, че написахме:

body {

background: #red;

}

Какво означава това?

- Казахме на CSS дай ми body-то като html таг, таг- цялата същинска част на html-a, т.е. всичко което е видимо за потребителя.

Казахме дай ми цялото боди и му сложи бекграунд в червен цвят. Идеята е, че това беше за селектиране на таг (селектирахме body тага и го променихме със css.)

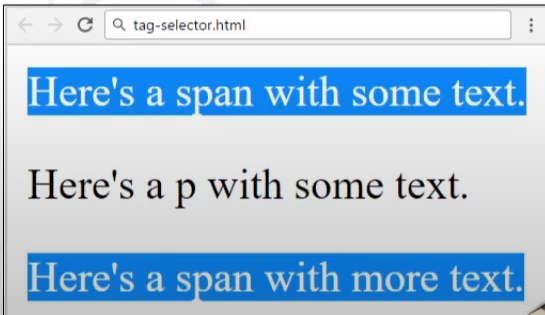
•••Select by Tag•••

Пример 2: със <span> тага

<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>



span {

background: DodgerBlue;

color: #ffffff; /\* цвят на буквите \*/

}

- В повечето случаи използваме <span> тага, когато не знаем кой друг таг да използваме.

- <span> тага е inline елемент.

    <link rel="stylesheet" href="style.css">    /\*  \*/

Връзваме html и css

link + tab – autofill in vscode

href="|" ctrl+enter ни отваря подсказка с файловетв от текущата директория във vscode.

* Остана да направим същото и с Html-a

Пр. Да си представим, че имаме една секция и в тази секция имаме <article>:

* Имаме 2 <article> например, третият искаме да бъде <div> , четвъртият в случая ще бъде footer.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="style.css">    /\*  \*/

    <title>Fitness-Site</title>

</head>

<body>

    <section>

        <article>

            <h1>Heading 1</h1>

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

        </article>

        <article>

            <h1>Heading 1</h1>

            <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi mattis dapibus. Sed maximus magna ac purus blandit, sed vestibulum justo semper. Aenean cursus est at justo consectetur, id condimentum urna consequat. Integer semper vestibulum leo, quis vulputate lectus consectetur eu..</p>

        </article>

        <div>

            <h1>Heading 1</h1>

            <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi mattis dapibus. Sed maximus magna ac purus blandit, sed vestibulum justo semper. Aenean cursus est at justo consectetur, id condimentum urna consequat. Integer semper vestibulum leo, quis vulputate lectus consectetur eu..</p>

        </div>

        <footer>

            <h1>Heading 1</h1>

            <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi mattis dapibus. Sed maximus magna ac purus blandit, sed vestibulum justo semper. Aenean cursus est at justo consectetur, id condimentum urna consequat. Integer semper vestibulum leo, quis vulputate lectus consectetur eu..</p>

        </footer>

    </section>

</body>

</html>

Имаме артикъл с h1 и p, пак артикъл с h1 и p, div контейнер и footer с h1 и p. Обаче да кажем, че искаме да селектираме само параграфите, които са в article. Може ли да го направим?

Отиваме в CSS:

p { /\* оцветява всички параграфи \*/

    background: yellow;

}

/\* Това вече е нестнато селектиране \*/

article p { /\* оцветява само article параграфите \*/

    background: yellow;

}

* Това, което CSS прави в този случаш е, минава, събира всички посочени параграви и им прилага някакво CSS Property, което е с ключ и стойност. Ключът не е нещо, което си измисляме, това са предефинирани CSS стойности, но можем да зададем каквато си стойност пожелаем. CSS не е като програмните езици – не се компилира, не се изпълнява и няма как да ни гръмне с някаква грешка, т.е. ако кажем:

article p {

    background: yellow;

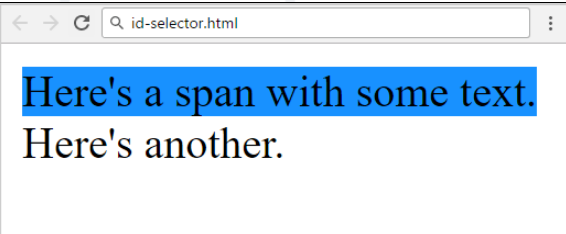
    pesho: red;

}

е ок и не ни се кара, просто няма да се изпълни самото пропърти (pesho: red) тъй като то не съществува.

Най-най-простата селекция е селектирането по таг, казваме вземи ми html таговете и приложи следните неща върху тях, дали ще бъдат span, p, article няма никакво значение.

Следващото нещо което можем да имаме е селектиране по ID. ID-то е нещо уникално, защото: Когато имаме тагове, те могат да са повече от един. Може да имамe 2ва span тага, 2 параграфа, 2 секции, даже в повечето случаи ще имаме повече от един, много рядко се случва да имаме уникален html таг в документа. С ID-то не е така. Когато искаме например много параграфи като <p> елементи в нашият html документ, обаче искаме само един от тях да бъде уникален, например само на един от тях да сложим някакъв бекграунд, а на другите не искаме да слагаме, тогава е доста уместно да използваме ID-то. То се използва, когато искаме да „хванем“ нещо единствено и не върху други елементи, т.е. трябва ни уникалния елемент, който сме искали да си „хванем“.

Select by ID

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

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

span#top {

background: DodgerBlue;

}

**Как се задава ID?**

-В примера горе имаме 2 span тага. Първият span има **id=”top”**, ето така се задава ID като името на ID-то, което стои в кавички няма никакво значение какво ще бъде.

-Селектирането може да бъде само по ID или по html таг и ID. В горният пример е показано по html таг и ID. Този пример би могъл да се изпише и по друг начин, тъй като няма да имаме и друго нещо с **id=”top”** и не е нужно да казваме, че ще бъде span в повечето случаи.

Id + таб подсказка от vscode

<body>

    <section>

        <article>

            <h1>Heading 1</h1>

            <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi mattis dapibus. Sed maximus magna ac purus blandit, sed vestibulum justo semper. ..</p>

        </article>

        <article id="red-article">

            <h1>Heading 1</h1>

            <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi mattis dapibus. Sed maximus magna ac purus blandit, sed vestibulum justo semper. .. <p>

        </article>

    </section>

</body>

Отиваме в style.css и как може да достъпим нещо по id – c #.

**Пример:** **#red-article**

**style.css**

#red-article {

    background: red;

}

article#red-article {

    background: red;

}

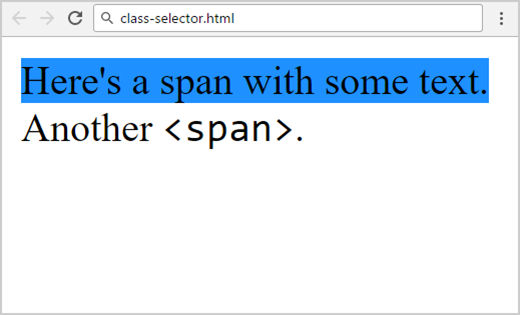
section#red-article {

    background: red;

}

Последното не работи, защото в този случай css първо търси тагът section и после търси секция с това id. Използвайт е ги когато имате нужда само веднъж от тях.

**Select by 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;

}

Нещо което е доста по-използвано от id-тата са класовете, те са нещо, което повтаряме много много пъти са css класовете. Те имат различна цел от id-тата. Класът има за цел, когато искаме да повторим някакво css property да му зададем клас и да го зададем на повече от един елементи, т.е. ако искаме да приложим някакув css само на един уникален елемент използваме id, ако искаме да приложим някакъв css на повече от един елемент използваме класове.

Как се задават класовете?

-По същият начин като id. Просто вземаме елемента, на който искаме да зададем съответният клас и казваме class = “име на самоят клас”;

В случая работят по същият начин като id-тата просто начинът им на достъпване е различен, тъй като той е с . , а не с #.

**Пример**: по-горе

Обяснение:

В css сме казали: Дай ми всеки един span елемент, който да има класът sky и му сложи за background син цвят. После ми дай всичко,

ID използваме върху един HTML таг, но класовете можем да ги използваме върху различни html тагове, т.е. ако имаме клас red, този клас red може да го сложим и върху заглавие и върху параграф и върху article, няма никакво значение. Идеята е, че може на повече от един елементи.

**Съвет:**

Изписвайте и името на html тага отстрани, за да сте сигурни на кое точно прилагате css-а.

span.sky {} Дай ми всеки един span таг, който да има класът sky и приложи кода м/у {}

.code {} Дай ми абсолютно всичко, без значение какъв таг е, но има клас code и изпълни {…}

Пр: да предположим, че навсякъде където имаме заглавие, искаме то да бъде центрирано. Как може да стане това? -По много начини.

1)Може да си изброим всички елементи, които искаме да са центрирани, т.е. може да селектираме по таг, но може и да изброяваме тагове и ако искаме да хванем абсолютно всички заглавия, какво може да кажем:

h1, h2, h3, h4, h5, h6 {

    text-align: center;

}

**Еквивалент:**

h1 {

    text-align: center;

}

h2 {

    text-align: center;

}

h3 {

    text-align: center;

}

…

2) Друг начин по който може да го направим е със css класове. Може да минем в html-a и да сложим някакъв клас на всяко едно заглавие:

<body>

    <section>

        <article>

            <h1 class="text-center">Heading 1</h1>

        <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi...</p>

        </article>

        <article id="red-article">

            <h1 class="text-center">Heading 1</h1>

        <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi...</p>

        </article>

        <h2 class="text-center">Heading 2</h2>

        <h3 class="text-center">Heading 3</h3>

        <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi...</p>

        <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi...</p>

        <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi...</p>

        <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi...</p>

    </section>

</body>

**style.css**

.text-center {

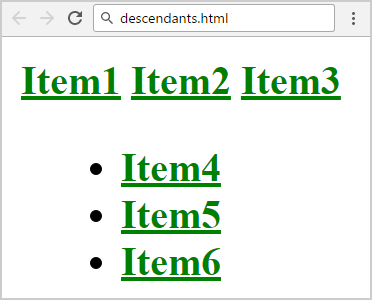
    text-align: center;

}

**3. Nested Selectors** • More than one simple selector

Нестнати селектори – преди малко показахме с article, но как може да селектираме повече от един simple selector?

Ако имаме 100 параграфа, но някои от тези 100 параграфи са в article и ние искаме да вземем само тези параграфи, които са в article, тогава вече ще трябва да селектираме нестнато, т.е. ще трябва да кажем „Вземи ми всички параграфи, които са в някакъв article“ и css да ни ги върне и да ги стилизираме.



**Descendant**

<div class="items">

<a href="#">Item1</a> /\* Те са first

<a href="#">Item2</a> child –

<a href="#">Item3</a> директни наследници \*/

<ul>

<li><a href="#">Item4</a></li>

<li><a href="#">Item5</a></li>

<li><a href="#">Item6</a></li>

</ul>

</div>

div.items a { /\* Вземи всички а елементи и направи буквите зелени и болд \*/

color: green;

font-weight: bold;

}

До тук видяхме какво са класове и какво са id-та. Да разгледаме примера по-горе.

<div> e html таг, който дефинира празен контейнер. Той е като article, section, като всички тези тагове, които сами по себе си не носят значение, но ни дефинират празен контейнер.

В примера този div има някакъв клас, който се казва items и вътре в него имаме 3 хипервръзки , които са Item1, Item2, Item3. След това имаме един unordered list <ul> и в него имаме list item-и с хипервръзки. В тези хипервръзки имаме още Item4, Item5, Item6 (общо 6 тага <a>). Eдните от тях са директно в items, а другите от тях са в ul, после в li.

Идеята е – Как и кога мога да ги селектирам разделно и как и кога мога да ги селектирам всички?

-В примера имаме

div.items a {

color: green;

font-weight: bold;

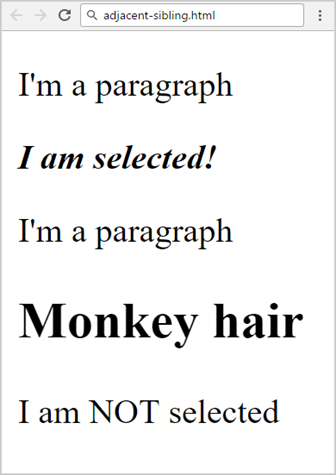
}

**Вземи ми div html тагът, който има class items.** **Като ни вземе този items** (class=”items”) в момента сме селектирали целияt контейнер <div> … </div>. Казваме вътре в него ни дай всички a тагове (div.items a {}) .

Какво правим в момента, ами селектираме си всички елементи, но по никакъв начин не сме посочили дали те да бъдат директен child или да бъдат нестнати надолу по дървото.

- Вземи всички а елементи и направи буквите зелени и болд.

Когато имаме space и a всъщност ние казваме дай ми навътре, без значение дали има и други елементи над тях, дали са нестнати, дали не са нестнати. Идеята е, че когато селектираме по този начин **space a,** ние просто казваме: ела и ми намери този контейнер div class=”items” и вътре в него колкото и надолу по дървото да има някакви елементи ти ми ги дай. При **space a** няма конкретика и така търсим абсолютно всички наследници.



**Adjacent Sibling – не се използва често**

<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>

p + p {

font-style: italic;

font-weight: bold;

}

В css-a:

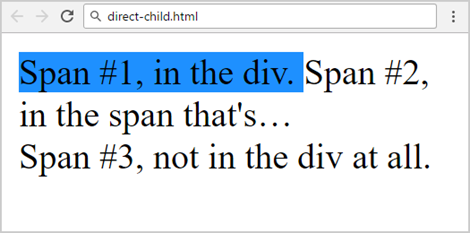
p+p означава – дай ми параграфа, който след него има параграф.

**Direct Child**

<div>

<span>Span #1, in the div. /\* директен чайлд на div\*/

<span>Span #2, in the span that's…</span> /\* директен child на спана над него \*/

 </span>

</div>

<span>Span #3, not in the div at all.</span>

div > span {

background: DodgerBlue;

}

span { background: #fff; }

Директен чайлд - когато имаме някакъв div например и в него имаме spam таг, след него също има спам таг.

Кой е директен наследник на div елемента? – този, който не е нестнат по дървото надолу, а да кажем, че тага започва директно след него.

Как се хваща директен наследник в css-а -> ами хваща се с по-голям.

Какво означава по-голямото? – Дай ми всеки един директен child на елемента. Директен child имаме когато са на един таг разстояние, нямаме нестване и така нататъка.

В css какво имаме: **div > span**

Тук сме казали дай ми div контешнера и span тага, но тук нямаме само space между тях, както беше преди малко (**div span**).

Ако имахме **div > span** щеше да ни оцвети Span #1…Span#2…. , но ние сме казали

**div > span** дай ми само директния child на този div и то ни връща само първият span, тъй като вторият е нестнат надолу.

**Multiple Classes**

<h2 class="apple orange small">Apple + Orange</h2>

<h2 class="apple">Apple</h2>

<h2 class="orange">Orange</h2>

.apple {

color: red;

}

.orange {

color: orange;

}

.small {

font-size: 16px;

}

.apple.orange { /\* избира елемент, който има и 2та класа \*/

font-style: italic;

}

Multiple classes означава да имаме повече от един класа. ID по принцип не е препоръчително да имаме повече от едно id-та. Ако имаме повече, значи че не знаем да пишем css и не е добра практика, но класовете могат да имат изброяване.

Ако сте виждали bootstrap, за което има мини кур също. Там имаме изброяване на много класове, които правят различни неща с нашите html елементи. Тези bootstrap класове всъщност прилагат различни пропъртита върху един елемент, това е и идеята на css класовете. Не е добре да имаме по 100 класа например, защото става доста притрупано и не се чете и ориентира. Идеята е, че всеки един добър layout работи на базата на класове. Ако искаме всички наши заглавия да са центрирани, да са болднати, да са с някакъв определен шрифт, например Consolas и т.н., искаме да кажем 4-5 неща за всички заглавия е по-удачно да си направим един class=”heading” и навсякъде, където имаме заглавие да викаме него и той всеки път ще идва и ще ни прилага css пропъртитата на съответно текущият елемент. В практиката реално така се прави и затова се използват css класовете.

В примера горе имаме:

Мултипъл класове на няколко елемента.

Първият ред имаме:

**<h2 class="apple orange small">Apple + Orange</h2>**

**class="apple orange small"** тук изброяваме няколко css класа.

Какво имаме в css-а:

**.apple, .orange, .small, .apple.orange**

**.apple.orange** това прилича на последното, което показвахме за имам таг и име на клас.

Например: Дай ми всички **p.red** (p тагове, които имат клас red). Така взимаме тагът p.

Тук (**.apple.orange**) сме казали: дай ми всички apple класове, които да имат и клас orange, т.е. ние просто търсим елемент, без значение от тага, който да има и двата класа едновременно и го правим italic.

<h2 class="apple orange small">Apple + Orange</h2>

Има клас apple orange и small. Тук какво ще се приложи:

първо) цветът ще стане червен

после) класът ще стане оранжев

Какъв ще е цветът в крашна сметка: В css има презаписване и последното записано нещо печели.

Пример:

Index.html

<body>

    <section>

        <article>

            <h1 class="text-center">Heading 1</h1>

            <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi mattis dapibus. Sed maximus magna ac purus blandit, sed vestibulum justo...</p>

        </article>

        <article id="red-article">

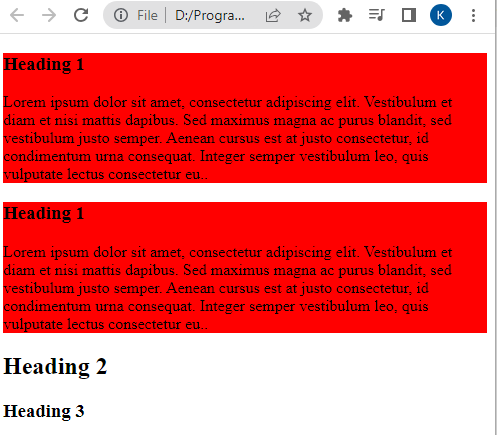
            <h1 class="text-center">Heading 1</h1>

            <p> Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum et diam et nisi mattis dapibus. Sed maximus magna ac purus blandit, sed vestibulum justo...</p>

        </article>

        <h2 class="text-center">Heading 2</h2>

        <h3 class="text-center">Heading 3</h3>

    </section>

</body>

style.css

article {

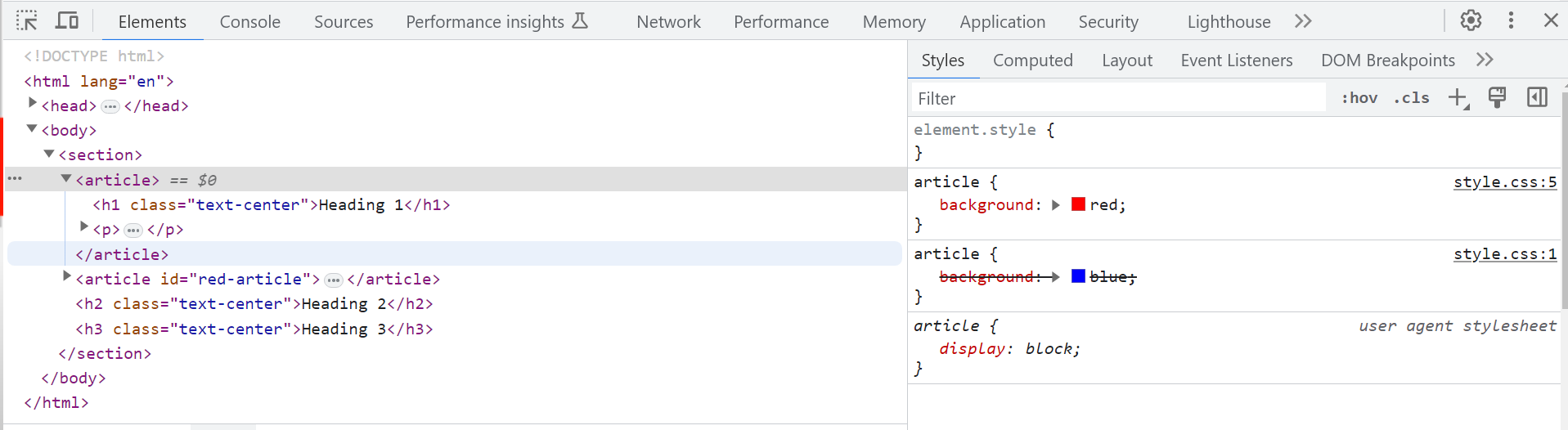
    background: blue;

}

article {

    background: red;

}

Ако дадем F12 на браузъра (което се препоръчва да се прави от всеки) и цъкнем на Elements може да разгледаме какво се случва

Презаписването се случва и с класовете.

**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"] {

text-decoration: none;

color: #962103;

font-size: 22px;

}

More about CSS Selectors: <https://css-tricks.com/pseudo-class-selectors/>

Елементи също може да селектираме по атрибути.

HTML елементите могат да имат някави атрибути. Това го видяхме при мултимедия таговете, които бяха за видео и за аудио и там казахме, че може някои от тях да имат опции за аутолууп, за спиране на аудиото и т.н. Идеята е, че това нещо в HTML го наричаме атрибути.

В примера горе имаме един ul списък и вътре в него имаме list item-и. Последният таг, който имаме нестнат е <a> тагът, който е хипервръзка и той има някакви допълнителни неща.

Първото нещо, което има тагът <a> е href, href – указва пътят, по който да вземем съответния файл.

Второто нещо, което има е title, този title сме му дали в случая името „Menu“.

В **style.css**

Как го селектираме този таг?

-Казваме вземи ми всички <a> тагове, които имат title “Menu”.

a[**title="menu"**] {

text-decoration: none; /\* Зададени 3 пропъртита \*/

color: #962103;

font-size: 22px;

}

Как по друг начим можем да селектираме по атрибути?

-В първата лекция се пропусна разглеждането на формите. Следват няколко примера за input полета.

Input полета:

<body>

    <!--пример за статична форма, която не работи-->

    <form>

        <input type="text"> <!-- дава ни просто едно празно поле -->

        <input type="submit"> <!-- изпраща нашата форма -->

<button>Submit</button> <!-- създава бутон -->

    </form>

</body>

Как може да селектираме тази форма по атрибут?

* <input type="text"> типът му е атрибут на даденият html елемент, иначе имаме просто форма с 2 input тага. <input> тагът е самозатварящ се таг.

Как може да селектираме втория input (бутона) и да го стилизираме със css?

Казваме му „дай ми инпут полето“

input {

    background: red; /\* Така оцветява и двете инпут полета. \*/

}

На нас обаче ни трябва само второто поле и трябва да го селектираме по атрибут.

Първо [], тъй като така казваме да ни даде атрибут на конкретният html елемент.

input[type="submit"] { /\* в [] не задаваме **title="…." Както е в примера, a**  type="submit" защото ни трябва този атрибут \*/

    background: red; /\* Така оцветява и двете инпут полета. \*/

}

input[type="submit"] {

    background: red;

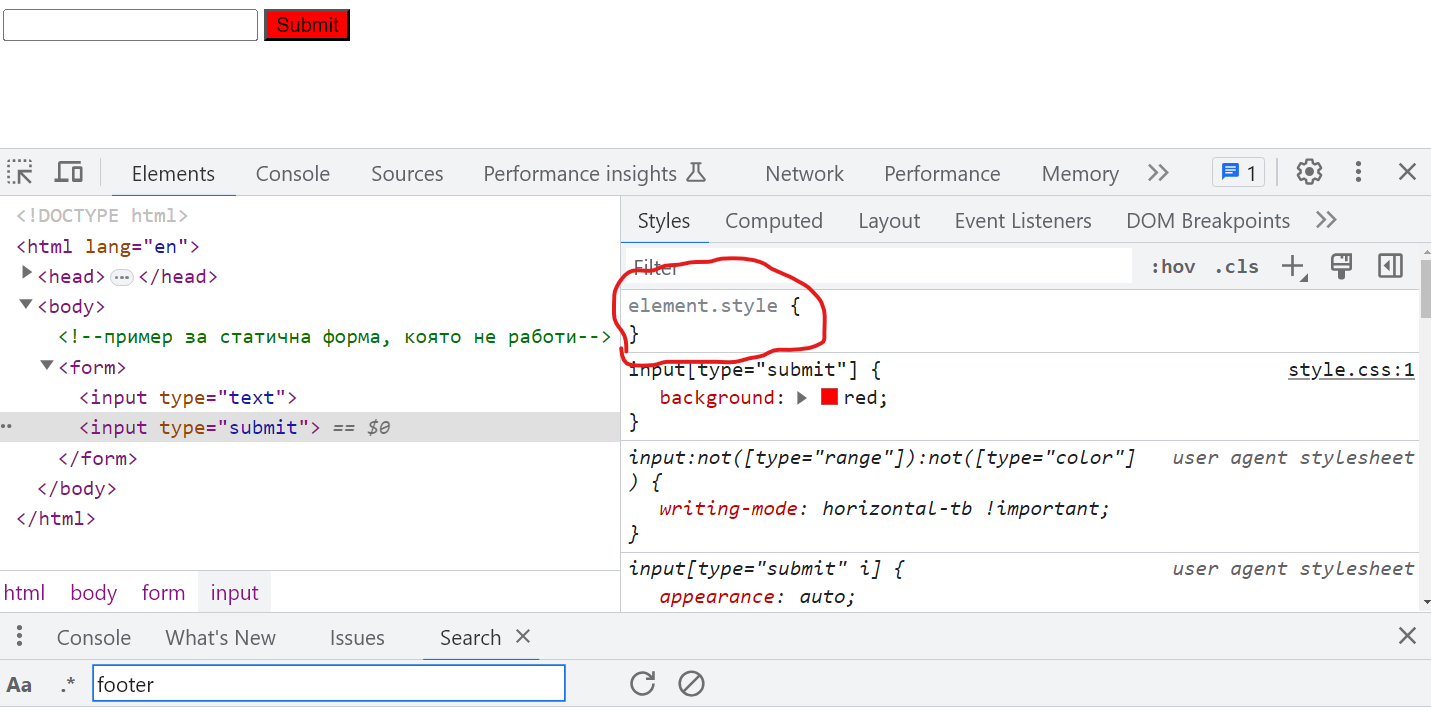
}

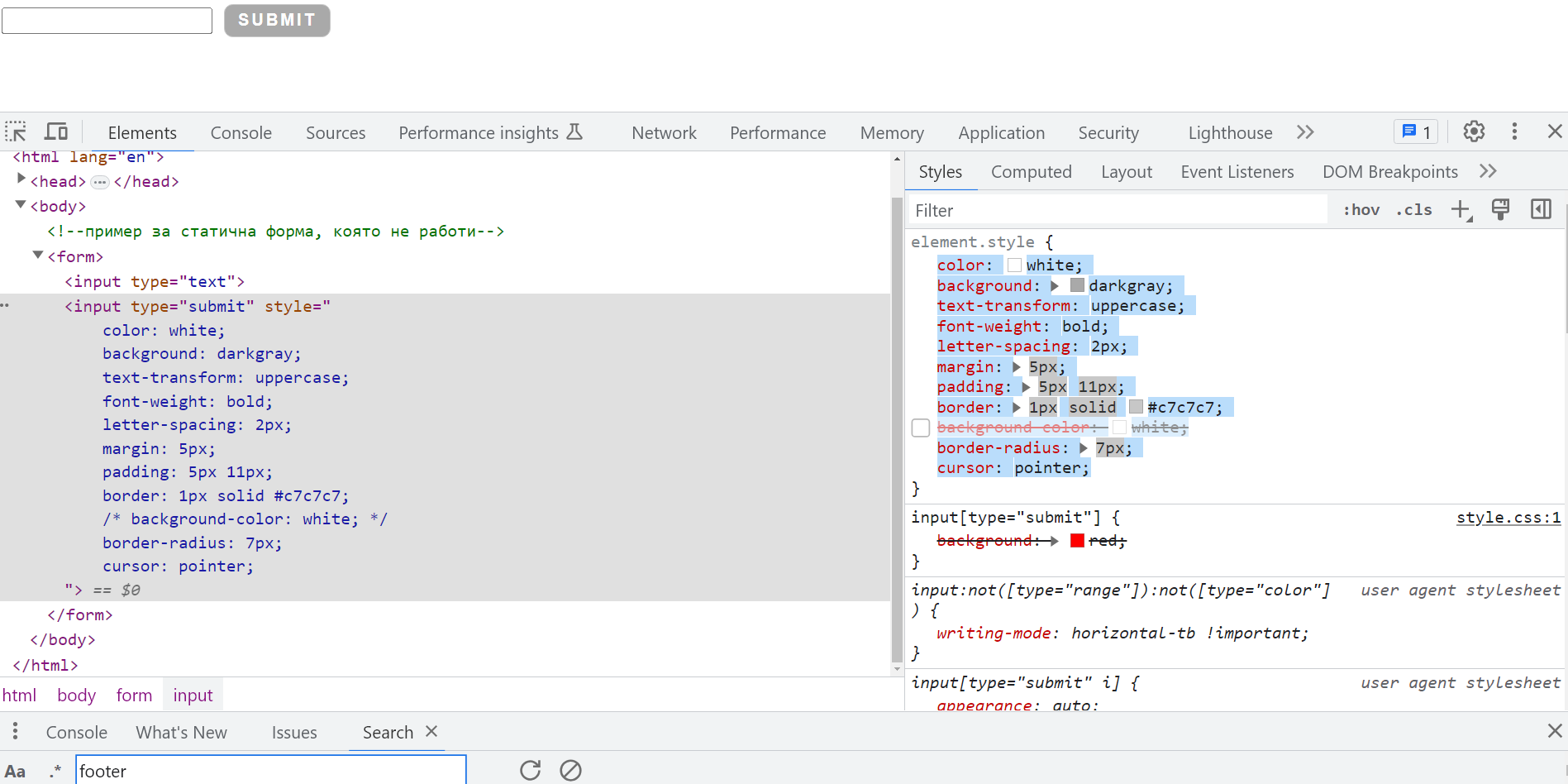
Остана само да направим бутона да изглежда добре

Как лесно да стилизираме?

-Даваме десен бутон на това което искаме да стилизираме в браузъра (в този случай бутона **Submit**) => inspect.

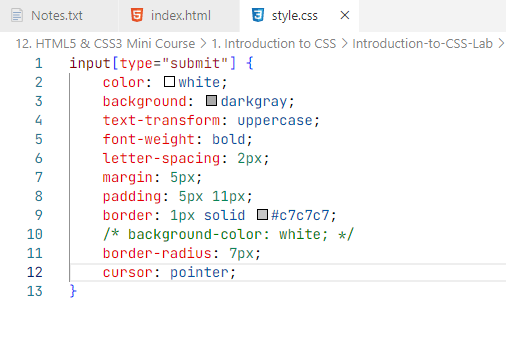
Отиваме на element.style {} и почваме да стилизираме конкретно този html елемент.





Какво правим след това?

Копираме си го това цялото и го пренасяме в нашия файл style.css. В повечето случаи работим така. Правим го в браузъра и след това пренасяме кода във файла и съхраняваме файла. Много по лесно е да стилизираме в браузъра защото виждаме промените на момента.



**Повече за CSS Селекторите на:**

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



**Multiple Selectors**

<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;

}

Вече споменахме, че при multiple селекторите просто имаме изброяване на html елементи, които съответно прилага css пропъртита върху множество елементи.

В примера горе имаме h1, h2, p p-тата са две на брой, но ние селектираме по таг.

Казваме h1, h2, p { background: yellow; } което ще приложи върху всички **h1, h2** и **p** тагове бекграунд yellow.

**Combining Multiple Selectors**

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



h1#header.intro {

text-decoration: underline;

color: #C00;

}

Как се комбинират мултипъл селектори?

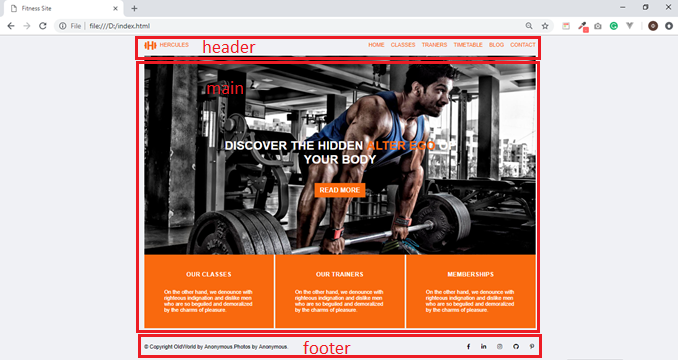
* Може да имаме h1, който да има id=”header”, id-то е уникално и него го имаме само веднъж. Обаче може да имаме и class=”intro”, т.е. върху един html елемент можем да имаме и id и class.
* Как можем да го селектираме по най-подробния начин ( h1#header.intro {…} ): Можем да кажем „Дай ми h1 тагът, който да има id=”header“ и class=”intro”, т.е. ние в момента селектираме и по трите неща и ако едно от тях не е спазено ние няма да си приложим съответно тези css пропъртита върху елемента.

**Problem: Fitness Site**

* Create an "index.html" and a "style.css" files
* Divide your content into <header>, <main> and <footer> tags

Hints:

Трябва да разделим съдържанието си на header, на main и footer тагове.



Ctrl+shift+v отваря .md форматът, в който са дадени условията.

Трябва текста HERCULES и навигацията да са на една линия:

1. Единият начин е да кажем на параграфа <p>HERCULES</p> да има float: left;

Тъй като параграфът е блоков елемент по дефолт, той става inline блоков елемент и просто пуска елементите, които са след него да се качът от дясната му страна, тъй като ние му казваме да се float-не отляво.

element.style { /\* пишем в браузъра и после пействаме в css \*/

float: left;

}

И сега, това което ни трябва е целият навигейшън да има text-align: right;

nav {

    text-align: right;

}

Кода до тук:

**html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

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

    <title>Fitness Site</title>

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

</head>

<body>

    <header>

        <p>HERCULES</p>

        <nav>

            <ul>

                <li><a href=#">HOME</a></li>

                <li><a href="#">CLASSES</a></li>

                <li><a href="#">TRANERS</a></li>

                <li><a href="#">TIMETABLE</a></li>

                <li><a href="#">BLOG</a></li>

                <li><a href="#">CONTACT</a></li>

            </ul>

        </nav>

    </header>

    <main></main>

    <footer></footer>

</body>

</html>

/\* Fitness-Site \*/

body {

    background: #F0F1F5;

    margin: 0;

    padding: 2rem 12rem;  /\* за да не е залепено за екрана \*/

    font-family: "Helvetica", sans-serif;

}

header > p {    /\* искаме фърст чайлда p \*/

    float: left;

    color: orange;

}

nav {

    text-align: right;

}

nav ul li {

    display: inline-block;

}

nav ul li a {   /\* Тук работим с <a> тагът и на него променяме цвета. \*/

    color: orange;

    text-decoration: none;  /\* За да премахнем подчертаното и декорацията от хиперлинковете. \*/

}

Казахме, че падинга работи отгоре, отдолу, отляво и отдясно.

**padding: 2rem 12rem; 2rem – отгоре и отдолу 12rem – oтляво и отдясно**

****

Защо не са подравнени? Какво можем да направим, за да ги подравним?

-Първо можем да кажем на ul-а vertical-align: middle; (не сработи) и затова даваме:

nav ul {

    float: right;

};

До тук css-а

/\* Fitness-Site \*/

body {

    background: #F0F1F5;

    margin: 0;

    padding: 2rem 12rem;  /\* за да не е залепено  за екрана \*/

    font-family: "Helvetica", sans-serif;

}

header > p {    /\* искаме фърст чайлда p \*/

    float: left;

    color: orange;

}

nav {

    text-align: right;

}

nav ul {

    float: right;

}

nav ul li {

    display: inline-block;

    padding: 0 8px; /\* разстояние между отделните li => отгоре и отдолу е 0, а отляво и отдясно е 8пь \*/

}

nav ul li a {   /\* Тук работим с <a> тагът и на него променяме цвета. \*/

    color: orange;

    text-decoration: none;  /\* За да премахнем подчертаното и декорацията от хиперлинковете. \*/

}

Следващото нещо, което ни трябва е main часта:

-Ще направим 2 секции. 1вата секция ще държи снимката, текста и т.н., а втората секция ще държи 3те артикъла

<main>

        <section>

            <h1>DISCOVER THE HIDDEN <span>ALTER EGO</span> OF YOUR BODY</h1>

            <button>READ MORE</button>

        </section>

        <section>

            <article></article>

            <article></article>

            <article></article>

        </section>

    </main>

Отиваме в style.css и сега на нас вече ни трябва main часта, в тази main част обаче ни трябва секцията, която да е първа и тук идват на помощ псевдо селекторите, с които можем да селектираме първи от типа, н-чаилд, фърст чаилд и т.н. В случая ще използваме :nth-child(1)

main section:nth-child(1) {

}

Как задаваме за бекграунд снимка, тъй като тази снимка ние трябва да я зададем зад елементите. Ние вече сме хванали секцията като елемент и там ще си добавим снимката.

main section:nth-child(1) {

    background-image: url("../Requirements-and-Resources/Images/picture.jpg");

}

Снимката не е на правилното място, защото когато зададохме float на горните елементи (в header) те също станаха inline елементи и снимката просто е застанала върху тях.

* Това което можем да направим в случая е да зададем няккакво разстояние отгоре, което да ни избута снимката (margin-top: 3rem;)

/\* main \*/

main section:nth-child(1) {

    margin-top: 3rem;

    background-image: url("../Requirements-and-Resources/Images/picture.jpg");

}

* Другото нещо, което ни трябва е да fit-нем снимката, така че тя да се вижда. Това ще постигнем с background-size: cover;
* Следващото нещо, което искаме да му дадем е     background-position: center;

До тук за мейн имаме:

main section:nth-child(1) {

    margin-top: 3rem;

    background-image: url("../Requirements-and-Resources/Images/picture.jpg");

    background-size: cover;

    background-position: center;

}

Но контейнера не е достатъчно широк, т.е. това което ще ни трябва от тук нататък е някакъв падинг (padding: 11rem;)

main section:nth-child(1) {

    margin-top: 3rem;

    background-image: url("../Requirements-and-Resources/Images/picture.jpg");

    background-size: cover;

    background-position: center;

    padding: 11rem;

}

* Това, което ни трябва сега е този h1 да стане в бели букви а span тагът в оранжеви

Ще ни трябва малко повече padding отляво и отдясно, за да слезе на два реда текст. Сменяме падинга на цялата секция. padding: 11rem 15rem;

* Следва да стилизирамe READ MORE бутона.

main section:nth-child(1) button {

    background: orange;

    border: none;

    color: white;

    font-size: 22px;

    padding: 12px;

    font-weight: bold;

}

* Следва на направим 3те артикъла в следващата секция.

<section>

            <article>

                <h3>

                    OUR CLASSES

                </h3>

                <p>

                    On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure.

                </p>

            </article>

            <article>

                <h3>

                    OUR TRAINERS

                </h3>

                <p>

                    On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure.

                </p>

            </article>

            <article>

                <h3>

                    MEMBERSHIPS

                </h3>

                <p>

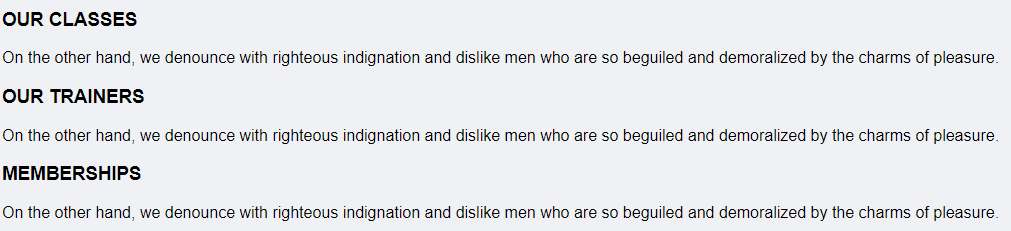
                    On the other hand, we denounce with righteous indignation and dislike men who are so beguiled and demoralized by the charms of pleasure.

                </p>

            </article>

        </section>

Следва да ги направим да изглеждат като на снимката.

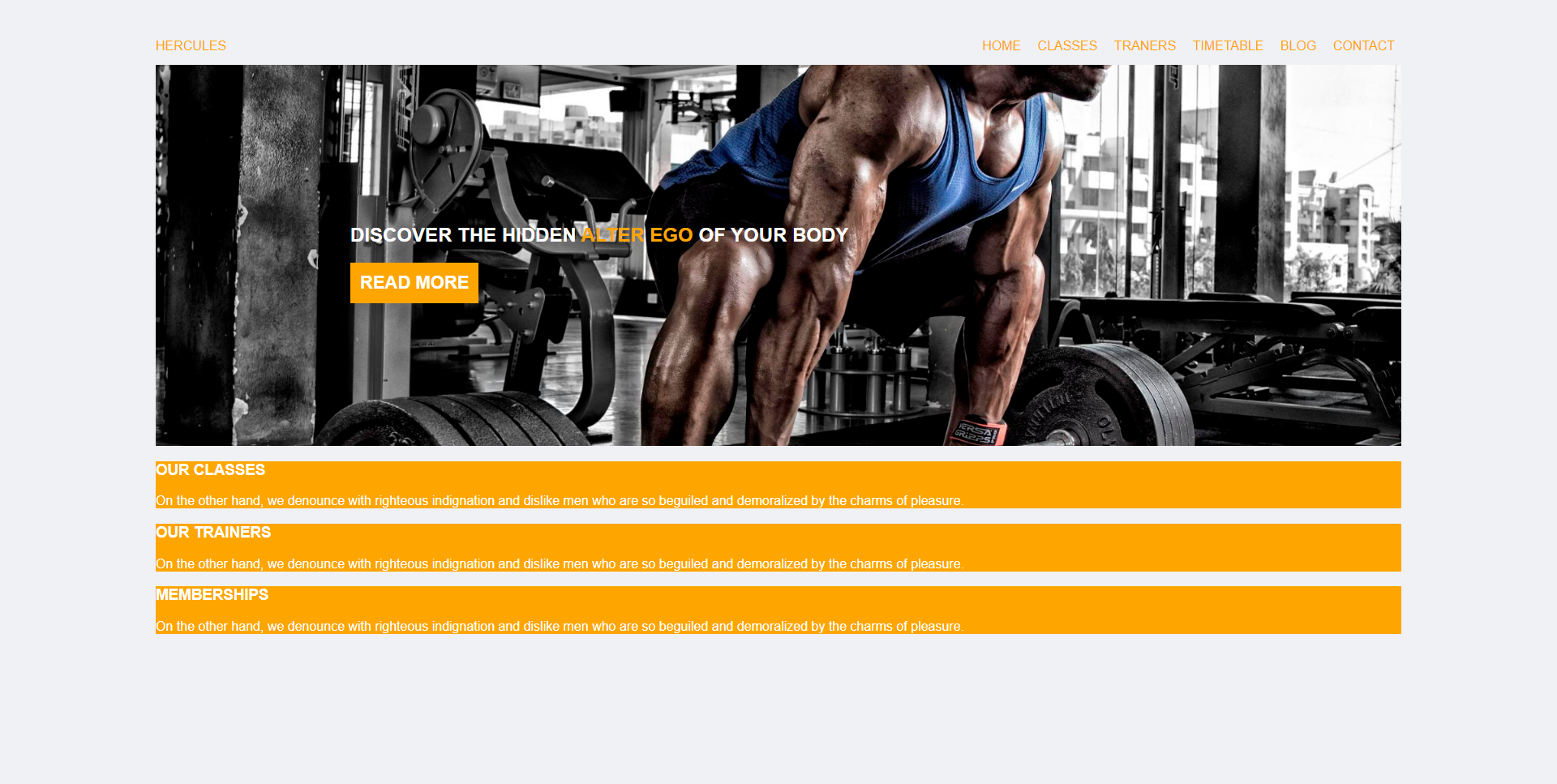


main section:nth-child(2) article {

    background: orange;

    color: white;

}



* Сега ни трябва тези артикъли да имат някаква дължина, защото те в момента заемат full вид, а на нас ни тряба да имат ограничена дължина, която може да я зададем с пропъртито width и да зададем в проценти.
* Може да задаваме проценти стойности в уида и ги подреждаме едно до друго ().
* Ще добавим и падинг примерно 20px. В момента чисто визуално просто наместваме нещата така че те да изглеждат сходно със снимката.
* Трябва да центрираме заглавията на артикълите.

main section:nth-child(2) article h3 {

    text-align: center;

}

Остана ни фуутъра:



1ви артикъл 2ри артикъл

Във втория артикъл има иконки, които може да вземем от един шрифт, който се казва **Font Awesome**.

Font Awesome е една библиотека, която си я импортваме първо, казваме коя иконка искаме и то ни дава някакъв код за нея, т.е. дава ни html таг буквално. Добавяме го и иконката се рисува.

Включваме <link rel="stylesheet" href="<script src="https://kit.fontawesome.com/da42e13f9f.js" crossorigin="anonymous"></script>">

в хед частта

<footer>

        <article>

            &copy; Copyright OldWorld by Anonymous.Photos by Anonymous.

        </article>

        <article>

            <i class="fa-brands fa-facebook-f"></i>

        </article>

    </footer>

За да се визиолизират един до друг.

footer article {

    width: 46%;

    display: inline-block;

}

    <footer>

        <article>

            &copy; Copyright OldWorld by Anonymous.Photos by Anonymous.

        </article>

        <article>

            <i class="fa-brands fa-facebook-f"></i>

            <i class="fa-brands fa-linkedin-in"></i>

            <i class="fa-brands fa-instagram"></i>

            <i class="fa-brands fa-github"></i>

            <i class="fa-brands fa-pinterest"></i>

        </article>

    </footer>

* Следва да добавим някакво разстояние между самите иконки, които са просто тагове <i>.

/\* footer \*/

footer article {

    width: 46%;

    display: inline-block;

}

footer article:nth-child(2) {

    text-align: right;

}

footer i {

    padding: 16px;

}

Остана да се добави логото в хедъра

===============================================================================================

**Solution (HTML)**

<header>

<section>

<p><i class="fas fa-dumbbell"></i> HERCULES</p>

<nav>

<ul>

<li><a href="#">HOME</a></li>

<!-- TODO: put the next list items here -->

</ul>

</nav>

</section>

<section>

<p>DISCOVER THE HIDDEN <span>ALTER EGO</span> OF YOUR BODY</p>

<button>READ MORE</button>

</section>

</header>

**Solution (HTML)**

<main>

<article>

<h3>OUR CLASSES</h3>

<p>On the other hand ...</p>

</article>

<!-- TODO: put the next articels here -->

</main>

<footer>

<p>© Copyright OldWorld ...</p>

<ul>

<li><a href="#"><i class="fab fa-facebook-f"></i></a></li>

<!-- TODO: put the next list items here -->

</ul>

</footer>

**Solution (CSS)**

body {

font-family: "Helvetica",

sans-serif;

width: 1180px;

margin: 0 auto;

padding: 0;

font-size: 16px;

background: #F0F1F5;

}

header section:nth-child(1) {

color: #f9690e;

}

header section:nth-child(1) i {

font-size: 30px;

padding-right: 5px;

vertical-align: middle;

}

nav {

display: inline-block;

width: 675px;

}

nav ul {

text-align: right;

margin: 0;

padding: 0;

}

ul li {

display: inline-block;

}

ul li:not(:last-child) {

padding-right: 20px;

}

**Solution (CSS)**

nav ul li a {

display: block;

text-align: center;

text-decoration: none;

color: #f9690e;

}

header section:nth-child(1) p {

width: 500px;

display: inline-block;

}

header section:nth-child(2) {

background: url("img/picture.jpg")

no-repeat;

background-size: cover;

...

...

height: 350px;

text-align: center;

padding-top: 250px;

}

header section:nth-child(2) p{

width: 65%;

margin: 0 auto;

font-size: 36px;

font-weight: bold;

color: white;

}

header section:nth-child(2) span {

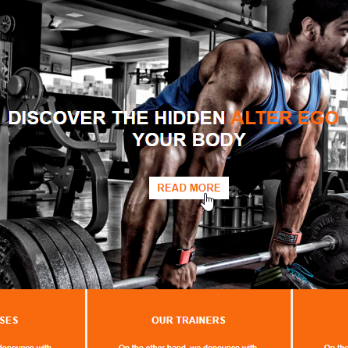
color: #f9690e;

}

**Solution (CSS)**

header section:nth-child(2) button {

margin-top: 50px;

 padding: 10px 15px;

font-size: 20px;

font-weight: bold;

color: white;

background: #f9690e;

border: none;

}

header section:nth-child(2) button:hover {

background: white;

color: #f9690e;

cursor: pointer;

}

**Solution (CSS)**

main article {

width: 270px;

display: inline-block;

padding: 30px 60px;

background: #f9690e;

color: white;

}

main article h3 {

text-align: center;

color: white;

font-weight: bold;

padding-bottom: 15px;

}

footer {

margin: 0;

padding: 30px 0 10px 0;

}

footer p {

width:730px;

display: inline-block;

}

footer ul {

width: 400px;

display: inline-block;

text-align: right;

}

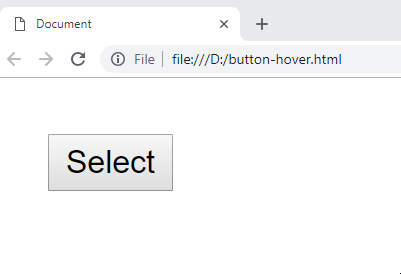
footer a i {

color: black;

}

**Pseudo Selectors • Relative to Element Content or State**

Псевдо селекторите ни дават някакво състояние на html елемент. Това е css, който прилагаме само при някакво състояние на елемента.

Например **:hover**, който имаме в този пример какво ни казва:

:hover

button:hover {

background: blue;

color: white;

}

-Имаме бутон, който е Select. Казали сме „Нека при hover“, което означава нека докато мишката/курсора ми е в/у бутона, той да има бекграунд син и цвят бял.

\*\*\* Псевдо селекторите се достъпват с **:**

input[type="submit"]:hover {

}

**-**Имайте в предвит, че когато селектирате това състояние (hover), css-ът който вече имате на даденият елемент не се изтрива/променя. Ние просто добавяме някакъв css или overwrite-ваме старият. В случая с overwrite-а в примера:

input[type="submit"] {

    color: white;

    background: darkgray;

    text-transform: uppercase;

    font-weight: bold;

    letter-spacing: 2px;

    margin: 5px;

    padding: 5px 11px;

    border: 1px solid #c7c7c7;

    /\* background-color: white; \*/

    border-radius: 7px;

    cursor: pointer;

}

input[type="submit"]:hover {

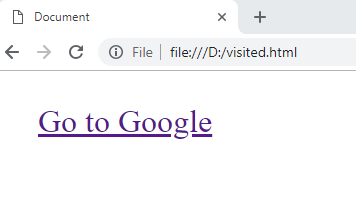
    color: darkgray;

    background: white;

    border: 1px solid darkgray;

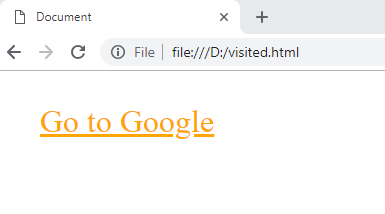
}

Тук сме задали на нашият бутон да има background да има darkgray цвят бяло, но искаме когато курсорът е на бутона бекграунда да бъде бял и цветът да е тъмносив.

**:visited**

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

Друг псевдо селектор в css е **:visited**. **:visited** работи при хиперлинкове като там можем да стилизираме нещата, които вече са били кликнати/посетени.

a:visited {

color: orange;

}

<body>

HTML

<p>

        <a href="https://open.spotify.com/">Spotify</a>

    </p>

    <p>

        <a href="https://music.youtube.com/">Youtube</a>

    </p>

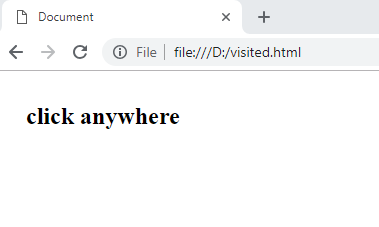
</body>

CSS

a:visited {

    color: red;

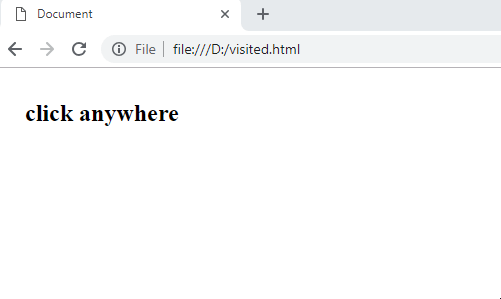
}



**:active**

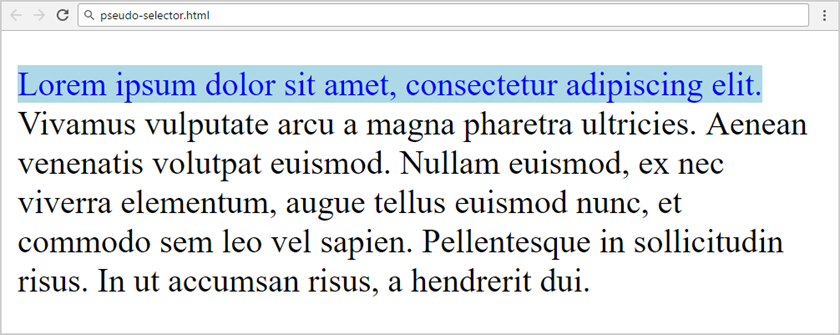
<h2>click anywhere</h2>

html:active {

 background: grey;

}

:active работи само когато курсора е върху него. Става active когато кликба върху бекграунда и задържа, като пуснем бутона пак става бяло.

**:first-line**

<p>Lorem ipsum dolor

sit amet, consectetur

adipiscing elit.</p>

p:first-line {

color:blue;

background-color: lightblue;

}

:fitst-line работи при всякакъв вид текст и винаги остава първи ред, т.е. когато преоразмеряваме екран и независимо от какво устройство отваряме даденото нещо, винаги първият ред ще остане селектиран, т.е. този псевдоселектор е достатъчно умен, че да знае до къде се простира първият ни ред.

**:before and :after**

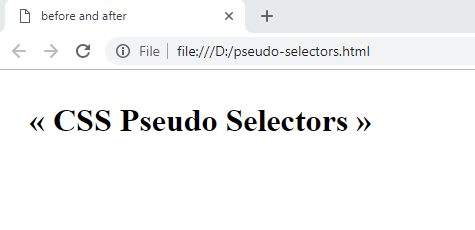
<h1>CSS Pseudo Selectors</h1>

h1:before { /\* "« " \*/

content: "\00AB\0020"

}

h1:after { /\* " »" \*/

 content: "\0020\00BB"

}

:before и :after са псевдо селектори, които ни добавят някакъв контент. Този контент в случая е дадено като някакво html entity, което просто ще ни даде някакъв символ.

h1:before { /\* "« " \*/

content: "\00AB\0020"

}

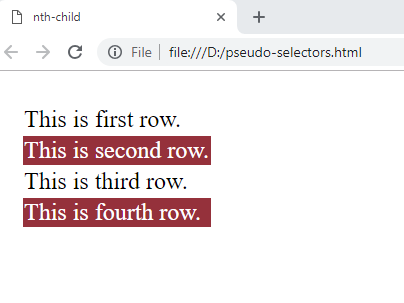
/\* Нека преди h1 да имаме един content, който да ни бъде следниято код "\00AB\0020"

После имаме h1 after => нека след h1 да имаме един content, който да ни бъде следниято код "\00AB\0020"

Резултата е << и >> между h1. T

\*/

Това е единственият начин, по който може да добавим някакви елементи в html-а чрез css, но може и на място в html да си ги добавим.

: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) { /\* За четните редове \*/

background: #95313b;

color: #fff;

}

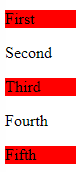
\*\*\*Тук просто изброяваме елементите, които имаме като последователност. Започваме да броим от 1 нагоре редовете.

**tr:nth-child(2n) {…}** => Дай ми всички четни редове.

Пример2:

<body>

    <p>First</p>

    <p>Second</p>

    <p>Third</p>

    <p>Fourth</p>

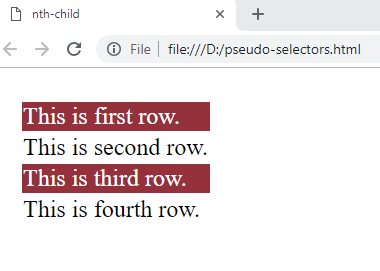
    <p>Fifth</p>

</body>

p:nth-child(2n-1) {

    background: red;

}

:nth-child(n+1) за нечетните редове

<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>

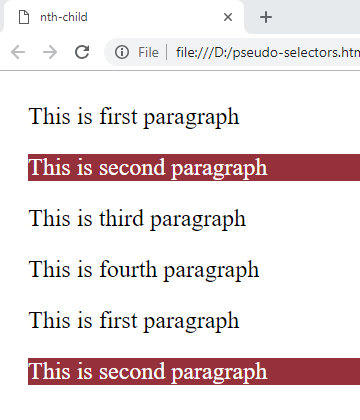
</table>

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

background: #95313b;

color: #fff;

}

:nth-of-type(n)

<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;

}

\*\*\*nth-of-type(2) => “Дай ми вторият от такъв тип”, т.е. „вземи всички параграфи, които са от тип параграфи и на 2рият му сложи бекграунд“. Тук работим само с direct child, за нестнатото започваме да броим отначало.

И тук:

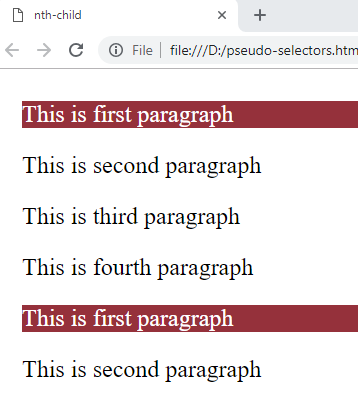
<div>

<p>This is first paragraph</p>

<p>This is second paragraph</p>

</div>

Почва да брои от начало и оцветява пак вторият.

:first-of-type

<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;

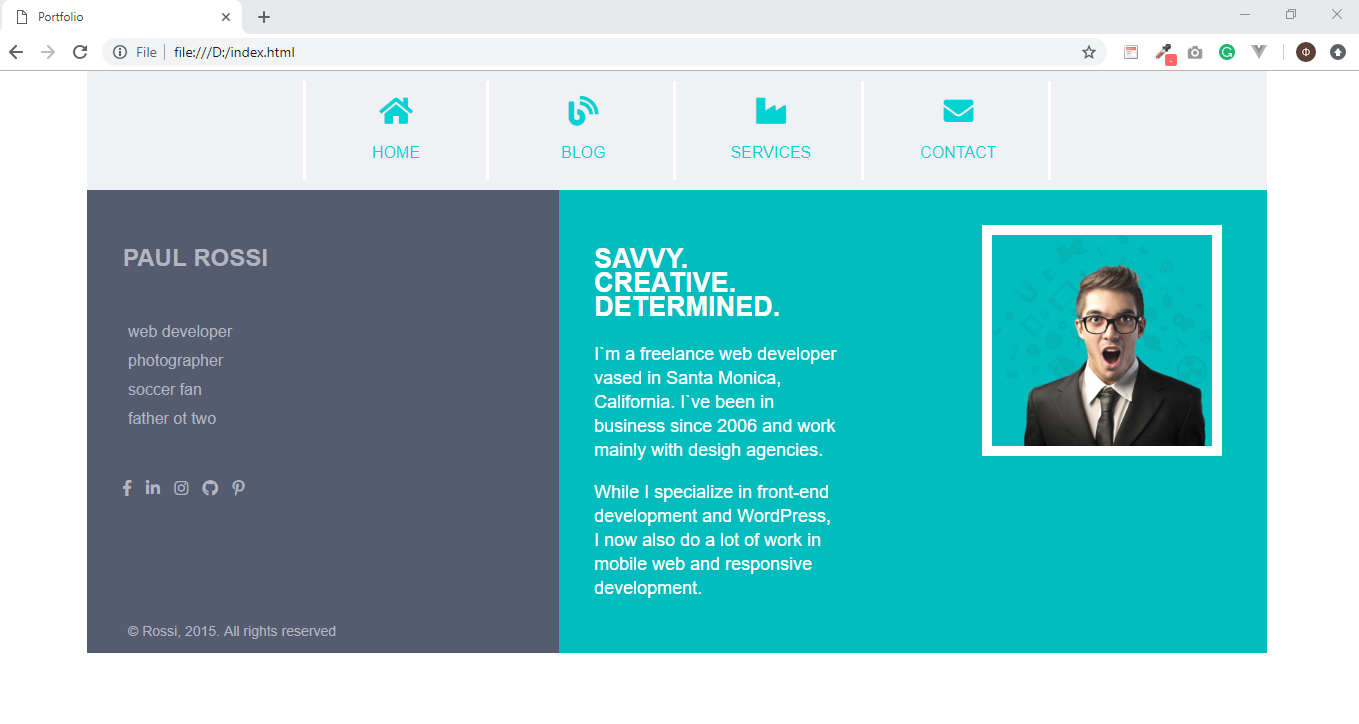
color: #fff;

}

\*\*\* :first-of-type е доста сходно с nth-of-type() просто винаги взима първото, а не второто. При nth-of-type() може да вземеме някое подред, а тук взимаме винаги първото.

Problem: Portfolio

* Create an "**index.html**" and a "**style.css**" files
* Divide your content into **<header>**and **<main>** tags



Solution (HTML)

<header>

<nav>

<ul>

<li><i class="fas fa-home"></i>

<a href="#">HOME</a>

</li>

<li><i class="fas fa-blog"></i>

<a href="#">BLOG</a>

</li>

<li><i class="fas fa-industry"></i>

<a href="#">SERVICES</a>

</li>

<li><i class="fas fa-envelope"></i>

<a href="#">CONTACT</a>

</li>

</ul>

</nav>

</header>

Solution (HTML)

<main>

<section>

<h2>PAUL ROSSI</h2>

<p>web developer</p>

<!-- TODO: put the next p tags here -->

<ul>

<li><a href="#"><i class="fab fa-facebook-f"></i></a></li>

<!-- TODO: put the next list items here -->

</ul>

<p>&copy Rossi, 2015. All rights reserved</p>

</section>

<section>

<article>

<h2>SAVVY. CREATIVE. DETERMINED.</h2>

<p>I`m a freelance ...</p>

<p>While I specialize ...</p>

</article>

<article>

<img src="img/..." alt="picture">

</article>

</section>

</main>

Solution (CSS)

body {

font-family: "Helvetica",

sans-serif;

font-size: 16px;

width: 1180px;

line-height: 24px;

margin: 0 auto;

padding: 0;

}

nav {

background: #f0f1f5;

}

nav ul {

text-align: center;

margin: 0;

padding: 0;

}

nav ul li {

display: inline-block;

width: 180px;

margin: 10px 0;

padding: 15px 0;

border-right: 3px solid white;

}

nav ul li:first-child {

border-left: 3px solid white;

}

nav ul li a {

display: block;

text-align: center;

text-decoration: none;

color: #00d3d4;

}

Solution (CSS)

nav ul li i {

padding-bottom: 15px;

font-size: 30px;

color: #00d3d4;

}

main section:nth-child(1) {

width: 400px;

display: inline-block;

background: #555c6f;

color: #b2b6c0;

padding: 36px;

padding-bottom: 5px;

}

main section:nth-child(1) h2 {

padding-bottom: 30px;

}

main section:nth-child(1) p {

margin: 5px;

}

main section:nth-child(1) ul {

padding: 30px 0 102px 0;

}

main section:nth-child(1) ul li {

display: inline-block;

padding-right: 10px;

}

main section:nth-child(1) ul li a i {

color: #b2b6c0;

}

Solution (CSS)

main section:nth-child(1)

> p:nth-child(7) {

font-size: 14px;

}

main section:nth-child(2) {

width: 638px;

display: inline-block;

float: right;

background: #01bdbe;

color: white;

padding: 35px;

}

main section:nth-child(2)

article:nth-child(1) {

width: 243px;

display: inline-block;

padding-right: 20px;

}

main section:nth-child(2)

article:nth-child(1) {

font-size: 18px;

}

main section:nth-child(2)

article:nth-child(2) {

width: 250px;

display: inline-block;

float: right;

}

img {

width: 220px;

display: inline-block;

border: 10px solid white;

}

Summary

**Primary Selectors:**

* select by **tag**, **id**, **class**

**Nested Selectors:**

* descendant
* direct child
* attribute and multiple

**Pseudo Selectors:**

* hover
* visited
* active

* nth-child
* after
* before

Questions?

<https://softuni.bg/trainings/2286/html-css-mini-course-january-2019>

1:13:24