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.)

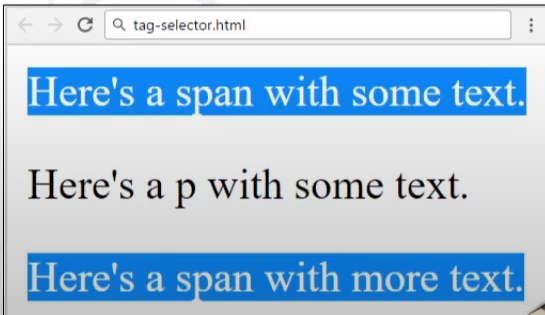
•••Select by Tag•••

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;

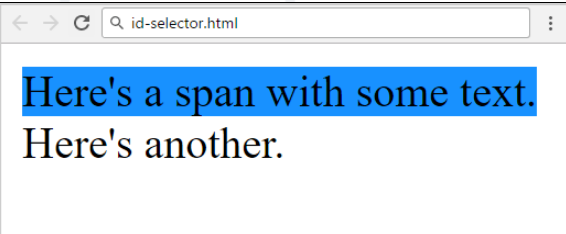
    pesho: red;

}

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;

}

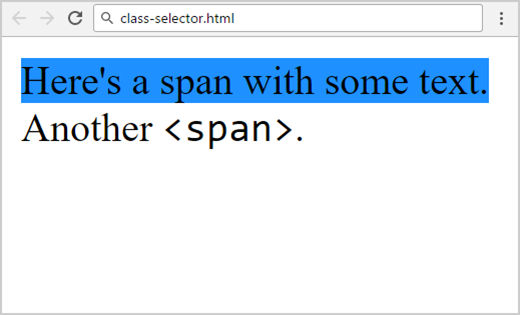
section#red-article {

    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 {

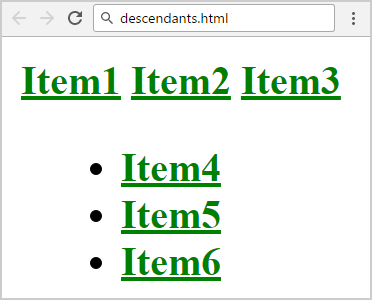
    text-align: center;

}

**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;

}

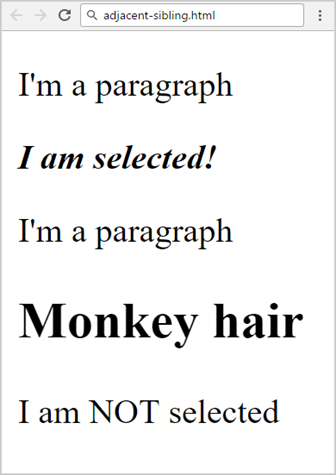
**Вземи ми div html тагът, който има class items.** **Като ни вземе този items** (class=”items”) в момента сме селектирали целияt контейнер <div> … </div>. Казваме вътре в него ни дай всички a тагове (div.items a {}) .

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:

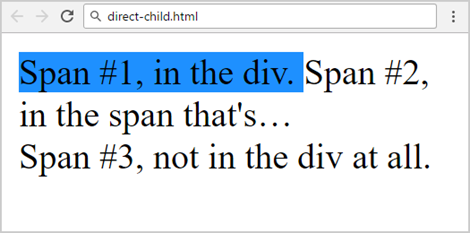
p+p означава – дай ми параграфа, който след него има параграф.

**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">

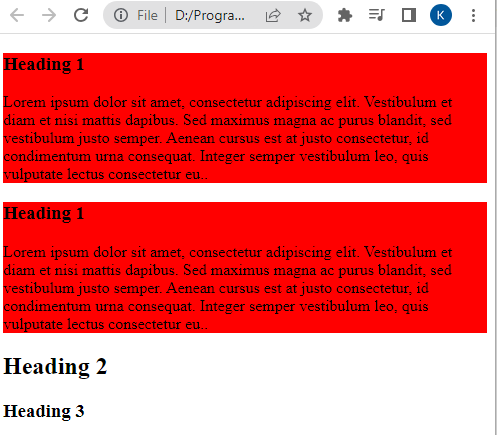
            <h1 class="text-center">Heading 1</h1>

            <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 {

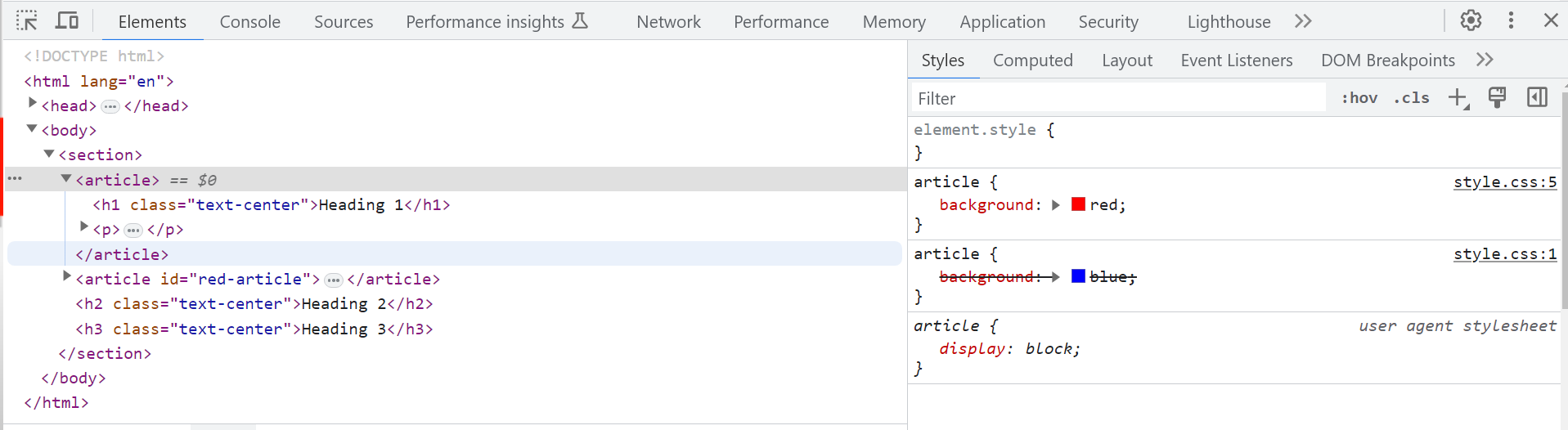
    background: blue;

}

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 към него има някакви допълнителни неща. Първото, което има е href, href – указваше пътят, от който да вземем съответният файл