Доклад

на тема „CSS селектори“

**1.CSS (Cascading Style Sheets)** е език за описание на стилове (език за стилови файлове, style sheet language) – използва се основно за описание на онлайн представянето на уеббазиран документ, който написан на език за маркиране. Най-често се използва допълнително към чистия HTML, но се прилага и върху XML уебстраници и документи. Спецификацията на CSS официално се поддържа от W3C.

**2.Същност и характеристики на езика:**

CSS позволява да се определя как да изглеждат елементите на една HTML страница – шрифтове, размери, цветове, фонове, и др. CSS кодът се състои от последователност от стилови правила, всяко от които представлява селектор, последван от свойства и стойности. Например в следния CSS код:

p {font-size: 9pt;} - има едно правило. То се състои от селектора p и свойството font-size, на което е зададена стойност 9pt. Това правило ще направи размера на шрифта във всички параграфи 9 точки.

**3.Селектори:**

Селекторите (от английски: select, като при "select color") в CSS се използват като наименуват типа селекция и покажат съответно към кои елементи на HTML документа трябва да бъде прилаган съответният стил. Съществуват много видове селектори. Някои селектори позволяват постигане и на динамичност на страницата, макар и в определена степен. Например само с помощта на CSS могат да бъдат направени падащи (drop-down) или пък изскачащи менюта (за мобилни устройства), и особено за хипервръзки, при които може да се задава цвета на линка, и също при посочване на линка, той да променя цвета си и др.

* **Видове Селектори:**

В CSS има няколко вида селектори:

**I. Селектори за класове** – с помощта на CSS може да създавате собствени класове за всеки един селектор. След дефиниране на класа, то всички селектори, на които е присвоен този клас, ще показват един и същ ефект. Използва се следният синтаксис:

селектор.клас {атрибут: стойност}

Пример:

* Нека имаме клас заглавия h1 със син цвят и клас параграфи p, позиционирани вляво.
* Името на класа е произволно, но за удобство ще използваме blue. Следователно целият код ще изглежда така:

h1.blue {color: #0000ff}

p.left {text-align: left}

С това класовете са дефинирани и трябва да се поместят в секцията head на основната HTML страница, като се затворят между таговете <style> и </style>:

<style type="text/css">

h1.blue {color: #0000ff}

p.left {text-align: left}

</style>

* Вече дефинираните класове, могат да се използват и в тялото (body-то) на страницата (body) чрез следния код:

<h1 class="blue"> Синьо Заглавие </h1>

Пример:

<html>

<head>

<title>I. Класови селектори</title>

<meta http-equiv=Content-Type content="text/html; charset=windows-1251" />

<style type="text/css">

h1.blue {color: #FFFF00}

p.left {text-align: left}

</style>

</head>

<body>

<h1 class="blue"> Морето е синьо </h1>

<p class="left"> В Бургас лятото е топло и има много гларуси </p>

<h1> Статия 2 </h1>

<h1 class="blue"> ...и небето е синьо </h1>

</body>

</html>

* Класовете могат да бъдат декларирани и по друг начин – без обвързване с определен селектор:

.клас {атрибут: стойност}

* В този случай гореописаният код ще изглежда така:

.blue {color: #0000ff}

.left {text-align: left}

**II. ID селектори** – с помощта на този вид селектори могат да се декларират допълнителни параметри, които не са указани в декларираните класове. Имената отново са произволни. Използва се следният синтаксис:

#id {атрибут: стойност} /или също селектор#id {атрибут: стойност}/

След добавяне на id селектор в горния пример:

<head>

<title>css</title>

<meta http-equiv=Content-Type content="text/html; charset=windows-1251" />

<style type="text/css">

.left {text-align: left}

#rb {color:#ff0000; font-weight:bold}

</style>

</head>

* В тялото на страницата можем да въведем следния код:

<p class="left" id="rb"> Кръвта е тъмно червена и тече от ляво надясно </p>

* Може да се въведе и параграф без декларирания клас:

<p id="rb"> - Хубавото вино е червено, тъмно червено </p>

**III. Контекстуални селектори** – комбинация от няколко селектора, като зададения ефект се проявява в зависимост от подредбата им. Използва се следния синтаксис:

1-ви селектор 2-ри селектор... {атрибут: стойност}

Например в секцията head може да напишем кода:

<head>

<title>css</title>

<meta http-equiv=Content-Type content="text/html; charset=windows-1251"

/>

<style type="text/css">

p i b {color:#00ff00}

</style>

</head>

* Ако след това в тялото на страницата напишем:

<p><i><b> Жабите са зелени и наклонени </b></i> ..а рибите не са </p> то ефектът ще бъде удебелен и наклонен зелен текст, между и и стандартен текст за останалата част.

**4.Коментари в CSS** – в HTML се използва: <!--Коментари и обяснения-->

* В CSS коментарът трябва да е затворен между наклонена черта със „звезда“ и „звезда“ с наклонена черта: /\*Закоментиран текст\*/.

**5. Задаване на стилове на HTML документ:**

* **Вграден стил на елемента:**

Зададеният по този начин стил се нарича inline. Представлява записване на стилова информация в атрибута style на даден таг. Полученият стил е валиден директно само за елемента, на който е зададен (въпреки това негови поделементи могат да го наследят при определени условия). Този метод има редица ограничения, тъй като не позволява използването на CSS селектори.

<h2 style="font-size: 10pt; color: blue;">Hello</h2>

* **Блок със стилове в документа:**

Зададеният по този начин стил се нарича internal. Представлява блок със стилове, затворен в таг <style>. Този таг трябва да е поставен в заглавната част на HTML документа (тага <head>).

* **Файл със стилове:**

Зададеният по този начин стил се нарича external. Представлява самостоятелен файл, който се състои от стилове и към него е направено обръщение в HTML документа. Това е единственият способ, който отговаря на идеята за отделяне на съдържание от оформление.

Указването на такъв файл се случва между <head> таговете и има следният синтаксис:

<link rel="stylesheet" href="http://example.com/css/style.css" type="text/css" />

Могат да бъдат вкарани неограничен брой файла, но е препоръчително броят им да е максимално ограничен, заради бавното зареждане на уеб страницата.

**6.Позициониране:**

CSS 2.1 задава три възможни схеми за позициониране:

* **Нормален поток:**
* Генерираните от елементите кутии се позиционират в зависимост от контекста (блоков или поредов форматиращ контекст за кутия, но не и двата едновременно). Съществува възможност за относително позициониране на блокови и поредови кутии.
* **Плаваща схема:**
* Кутията първоначално се позиционира според нормалната потокова схема, а след това се отмества наляво/надясно в зависимост от указаната стойност.
* **Абсолютно позициониране:**
* Кутията изцяло се премахва от нормалния поток и се позиционира според зададените координати спрямо съдържащия я блок.

**Позиция: горе, долу, ляво и дясно**

Има четири възможности за свойството position. Ако даден елемент не е static, то следните свойства top, bottom, left, и right позволяват да му се зададат отстояния и позиция.

* **Статичен static:**
* Стойност по подразбиране, която поставя елемента в нормален поток.

#div-1 {

position:static;

}

* **Релативен relative:**
* Относителна позиция спрямо статичното местоположение на елемента.

#div-1 {

position:relative;

top:20px;

left:-40px;

}

**Абсолютен absolute:**

* Определя абсолютно позициониране. Елементът се позиционира по отношение на най-близкия си нестатичен предшественик.

#div-1a {

position:absolute;

top:0;

right:0;

width:200px;

}

**7.Плаващи и прозрачни елементи:**

Една от три различни стойности може да бъде присвоена на свойството float. Абсолютно- или фиксирано- позиционирани елементи не могат да бъдат плаващи. Други елементи обикновено се носят около плаващи такива освен ако не са възпрепятствани от свойството clear.

**Ляв left**

Елементът плава отляво на линията, на която би се появил, други елементи може да се застанат от дясната му страна

#div-1a {

float:left;

width:150px;

}

#div-1b {

float:left;

width:150px;

}

**Десен right**

* Eлементът плава отдясно на линията, на която би се появил, други елементи може да се застанат от лявата му страна

#div-1a {

float:right;

width:200px;

}

**Прозрачен clear**

* Принуждава елементът да се появи зад прозрачно ('clear') плаващи елементи отляво (clear:left), отдясно (clear:right) или и от двете страни (clear:both).

#div-1a {

float:left;

width:190px;

}

#div-1b {

float:right;

width:190px;

}

#div-1c {

clear:both;

}

Източници:

* <https://bg.wikipedia.org/wiki/CSS>

Изготвил: Валентин Кьосев 11,,а‘‘клас