

ОСНОВЫ CSS

CSS (Cascading Style Sheets)

HTML

```
<body bgcolor="#f1f1f1">
  <h1 align="center"><font size="16px" color="red"
  <face="Tahoma">Основной заголовок</font></h1>
  <p><font size="8px" color="gray" face="Arial">Текст параграфа</font> </p>
  <h2><font size="10px" face="Tahoma">Подзаголовок 1</font></h2>
  <p><font size="8px" color="gray" face="Arial">Текст параграфа</font></p>
  <h2><font size="10px" face="Tahoma">Подзаголовок 2</font></h2>
  <p><font size="8px" color="gray" face="Arial">Текст параграфа </font></p>
</body>
```

HTML +

```
<body>
<h1>Основной заголовок</h1>
<p>Текст параграфа 1</p>
<h2>Подзаголовок 1</h2>
<p>Текст параграфа 2</p>
<h2>Подзаголовок 2</h2>
<p>Текст параграфа 3</p>
</body>
```

Преимущества CSS

Разграничение кода и оформления

При подобном разделении работа над дизайном и версткой сайта может вестись параллельно.

Разное оформление для разных устройств

С помощью стилей можно определить вид веб-страницы для разных устройств вывода: монитора, принтера, смартфона, планшета и др.

Расширенные по сравнению с HTML способы оформления элементов

В отличие от HTML стили имеют гораздо больше возможностей по оформлению элементов веб-страниц.

Ускорение загрузки сайта

При хранении стилей в отдельном файле, он кэшируется и при повторном обращении к нему извлекается из кэша браузера. За счёт кэширования и того, что стили хранятся в отдельном файле, уменьшается код веб-страниц и снижается время загрузки документов.

Единое стилевое оформление множества документов

Сайт это не просто набор связанных между собой документов, но и одинаковое расположение основных блоков, и их вид. Разработчикам же использование стилей существенно упрощает проектирование дизайна.

Централизованное хранение

Стили, как правило, хранятся в одном или нескольких специальных файлах, ссылка на которые указывается во всех документах сайта.

Применение CSS к HTML-документу

1 способ:

Инлайн/In-line (атрибут style)

```
<html>
<head> <title>Example</title>
</head>
<body style="background-color: #FF0000;"> <p>This is a red page</p>
</body>
</html>
```

Применение CSS к HTML-документу

2 способ: Внутренний (тэг style)

```
<html>
  <head>
    <title>Example</title>
    <style type="text/css">
      body {background-color: #FF0000;}
    </style>
  </head>
  <body>
    <p>This is a red page</p>
  </body>
</html>
```

Применение CSS к HTML-документу

3 способ:

Внешний (ссылка на таблицу стилей)

```
<html>  
  <head>  
    <title>My document</title>  
    <link rel="stylesheet" type="text/css" href="style/style.css" />  
  </head>  
<body> ...
```

Применение CSS к HTML-документу

3 способ:

Внешний (ссылка на таблицу стилей)



Синтаксис CSS

Одиночный селектор

```
селектор {  
    правило  
}
```

Группа селекторов

```
селектор,  
селектор,  
селектор { /* блок объявления стилей  
*/  
    СВОЙСТВО: значение;  
    СВОЙСТВО: значение;  
    СВОЙСТВО: значение;  
    СВОЙСТВО: значение;  
    СВОЙСТВО: значение;  
}
```

Пример:

```
H1 {color: blue; font-size: 20px;}
```

```
H1 {  
    font-family: Helvetica;  
    font-style: italic;  
    font-size: 30pt;  
    color: black;}
```

Пример:

```
H1, H2, H3, H4 {  
    font-family: Helvetica;  
    color: red;}
```


Селекторы CSS

Селектор тегов

```
P {  
font-family: arial, verdana, sans-serif;  
font-size: 12px;  
}
```

```
H1{  
font-family: Helvetica;  
font-style: italic;  
font-Size: 30pt;  
color: black;  
}
```

Селекторы CSS

Селектор класса

Пример:

HTML

```
<div class="red">Красный заголовок</div>
```

CSS

```
.red {  
  color: red;  
}
```

Пример:

HTML

```
<div class="heading red">Красный заголовок</div>
```

```
<div class="red">Красный заголовок</div>
```

CSS

```
div.heading {  
  font-size: 50px;  
}
```

```
div.red {  
  color: red;  
}
```

Селекторы CSS

Селектор идентификатора

```
<div id="red">Красный заголовок</div>
```

```
div#heading {  
    font-size: 50px;  
}  
div#red {  
    color: red;  
}
```

Селектор id может быть только один!!!

Пример, который работать не будет

```
<div id="heading-red">Красный заголовок</div>
```

```
div#heading {  
    font-size: 50px;  
}  
  
div#red {  
    color: red;  
}
```

Селекторы CSS

Селекторы Атрибутов

/ Элементы содержащие атрибут */*

```
[href] { color: red; }
```

/ Значение атрибута в точности равно заданному */*

```
[href="http://ya.ru"] { color: red; }
```

/ Значение атрибута содержит заданное значение */*

```
[href*="http://ya.ru"] { color: red; }
```

/ Значение атрибута начинается с заданного значения */*

```
[href^="https://"] { color: red; }
```

/ Значение атрибута заканчивается заданным значением */*

```
[href$="ya.ru"] { color: red; }
```

Псевдоклассы

Псевдоклассы — такие селекторы, которые позволяют выбрать элементы в зависимости от их состояния.

```
a:link { color: blue }  
a:active { color: red }  
a:hover { color: green }  
a:visited { color: purple }  
a:focus { color: yellow }
```

- «:enabled» и «:disabled» позволяют выбирать элементы в зависимости от их доступности для взаимодействия с пользователем.
- «:checked» позволяет выбрать все элементы, в которых есть атрибут checked.
- «:indeterminate» соответствует неопределенному состоянию checkbox'ов.

Его нет в HTML, но оно доступно из javascript. Чаще всего оно употребляется во вложенных списках с чекбоксами, в которых одним кликом по флажку можно выбрать всю категорию. Такой флажок принимает неопределенное состояние, если в соответствующей категории есть выбранные и не выбранные элементы.

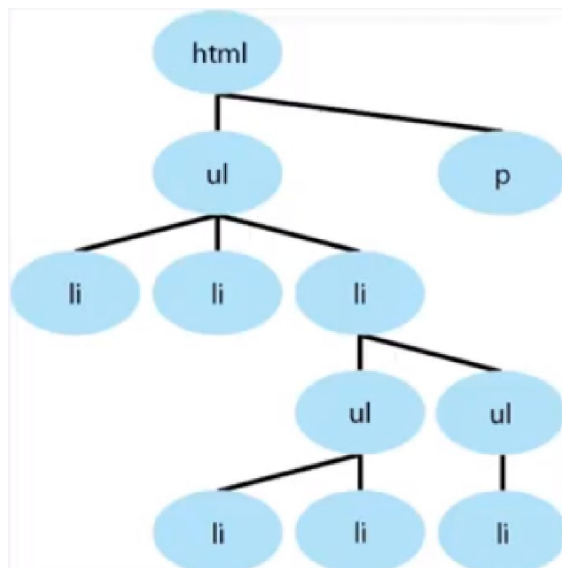
- :read-only позволяет выбрать элементы с атрибутом "только для чтения".
- :valid позволяет выбрать валидные элементы форм.

Селекторы CSS

Структурные псевдоклассы

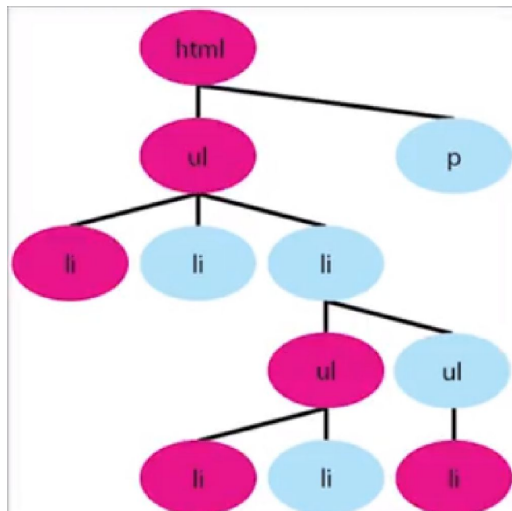
- `:root` выбирает корневой элемент HTML-документа
- `:first-child`/`:last-child` выбирают первый или последний дочерний элемент.
- `:nth-child`/`:nth-last-child`($2n+1$) позволяют выбрать n -ый элемент или n -ый элемент с конца. Здесь дополнительно указано, как строится номер элемента. В выражении $2n + 1$, n принимает значение от нуля до бесконечности. Таким образом, выбираются все нечетные элементы.
- `:nth-of-type`/`:nth-last-of-type`($-n+4$)
- `:only-child`/`:only-of-type` выбирает элемент, который является единственным дочерним для своего родительского элемента, или единственным данного типа.
- `:empty` выбирает пустые элементы.

Структурные псевдоклассы



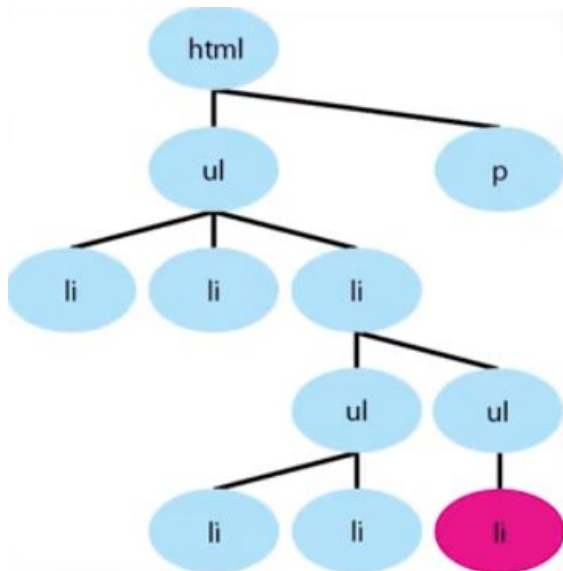
Структурные псевдоклассы

:first-child



Структурные псевдоклассы

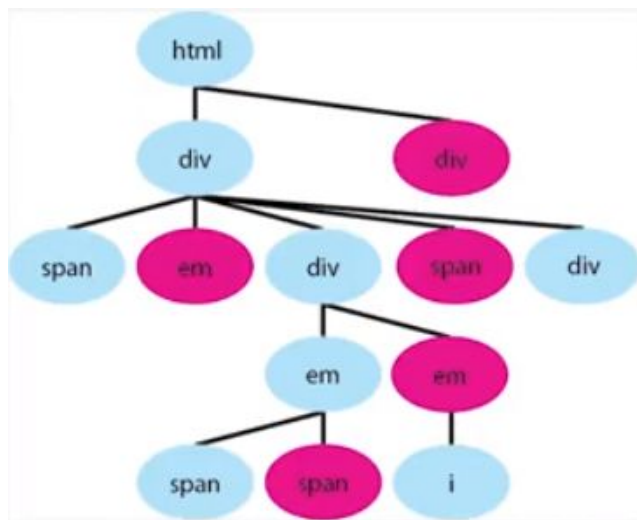
:only-child



Селекторы CSS

Структурные псевдоклассы

:nth-child(2n)



Селекторы CSS

Псевдоэлементы

- `::first-letter` который выделяет первую букву текста:

```
::first-letter { color: red }
```

- `::first-line` который выделяет первую строку текста:

```
::first-line { color: blue }
```

- `::before`, `::after` выделяют создаваемые в структуре документа элементы сразу после открывающего тега и сразу до закрывающего соответственно.

- `::before`,
- `::after` {
- `color: red;`
- `font-weight: bold;`
- `content: '»';`
- }
- `::before { content: '«'; }`

Селекторы CSS

Отношения элементов в CSS

- Потомок. Например, следующий селектор выделяет все div которые являются потомками (другого) div.

div div

div >> div

- Непосредственный потомок.

div > div

- Элемент, следующий за данным элементом (брат):

div + div

- Элементы, следующие за данным элементом (братья):

div ~ div