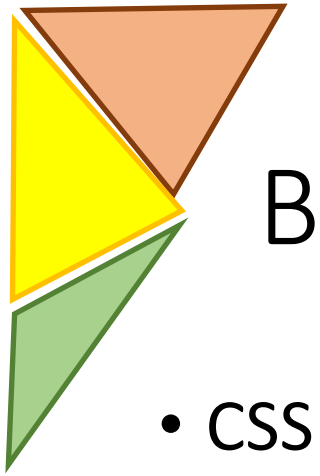
Основы CSS

Татаринова А.Г., каф. ПМИ

2025

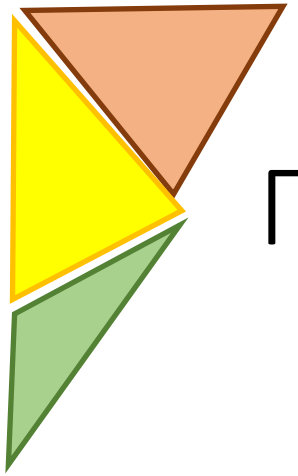


Введение в CSS

- CSS (Cascading Style Sheets) каскадные таблицы стилей
- формальный язык описания внешнего вида документа, написанного с использованием языка разметки
- язык стилей, определяющий отображение HTML-документов
- работает со шрифтами, цветом, полями, строками, высотой, шириной, фоновыми изображениями, позиционированием элементов и многими другим

История появления CSS

- начало 1990 - браузеры обладали собственными стилями для отображения HTML-страниц
- 1994 - Хокон Виум Ли предложил использовать для HTML-страниц концепцию «каскадных таблиц стилей»
- середина 1990-х - технологии CSS поддержана Консорциумом Всемирной Паутины



Преимущества

- расширенные способы оформления элементов
- единое стилевое оформление множества документов (разделение кода от оформления)
- разное оформление для разных устройств
- ускорение загрузки сайта
- централизованное хранение описание оформления элементов страниц

Синтаксис

- Селектор определяет элементы страницы, для которых задается стиль

```
селектор [, селекторы]
{
        свойство: значение
}
```

- Стиль описывается набором свойств и их значений
- Пример:

```
body {background: #ffc910}
```

- (!) CSS не чувствителен к регистру, переносу строк, пробелам и символам табуляции, поэтому форма записи зависит от желания разработчика

Добавление стилей

- Стили CSS описываются в самом документе
  - Описание стилей располагается в разделе **<head>** веб-страницы посредством тега **<style>**
  - Пример:

```
<html>
<head>
  <title>Style Test 2</title>
  <style>
       h1{
         font-size: 120%;
         color: green;
       }
  </style>
</head>
</html>
```

Правила применения стилей

- Допускается добавлять стилевое свойство и его значение по отдельности

  Пример:

```
td { background: olive;}
td {color: white;}
td {border: 1px solid black;}
```

- Удобнее группировать
  - Пример:

```
td {
      background: olive;
      color: white;
      border: 1px solid black;
}
```

Правила применения стилей

- Приоритет за тем, что ниже по коду
- (!) Повторные записи лучше избегать
- У каждого свойства могут быть только соответствующие его функции значения
- Комментарии:

```
/*
пример комментариев
они могут быть многостраничные
*/
```

Селекторы тегов

- В качестве селектора может выступать любой тег HTML, для которого определяются правила форматирования, такие как: цвет, фон, размер и т. д.
- Правила задаются в следующем виде:

```
Тег { свойство1: значение; свойство2: значение; ... }
```

- Пример:

```
p {
 text-align: justify; /* Выравнивание по ширине */
 color: green; /* Зеленый цвет текста */>
}
```

Классы

- Применяют, когда необходимо определить стиль для индивидуального элемента или задать разные стили для одного тега

  Синтаксис:

  ```
  Тег.Имя класса { свойство1: значение; свойство2: значение; ... }
  ```

  Пример 1.
- Использование русских букв в именах классов **недопустимо**
- Чтобы указать в коде HTML, что тег используется с определённым классом, к тегу добавляется атрибут `class="Имя класса"`
- Можно использовать классы без указания тега

  Синтаксис:

  ```
  .Имя класса { свойство1: значение; свойство2: значение; ... }
  ```

  Пример 2. и Пример 3.

Одновременное использование классов

- К любому тегу одновременно можно добавить несколько классов, перечисляя их в атрибуте `class` через пробел
- К элементу применяется стиль, описанный в правилах каждого класса
- Если при добавлении нескольких классов стили могут содержать одинаковые стилевые свойства, но с разными значениями, то берётся значение у класса, который описан в коде ниже

Пример 4.

Идентификаторы

- Идентификатор (называемый также «ID селектор») определяет уникальное имя элемента, которое используется для изменения его стиля и обращения к нему через скрипты

Синтаксис:

```
#Имя ид. { свойство1: значение; свойство2: значение; ... }
```

Обращение к идентификатору происходит аналогично классам, но в качестве ключевого слова у тега используется атрибут **`id`**, значением которого выступает имя идентификатора

(!) Символ решётки при этом не указывается

- Можно применять к конкретному тегу

Синтаксис:

```
Тег#Имя ид. { свойство1: значение; свойство2: значение; ... }
```

Пример 6.

Контекстные селекторы

- При создании веб-страницы часто приходится вкладывать одни теги внутрь другие
- Чтобы стили для вложенных тегов использовались корректно, необходимы селекторы, которые работают только в определённом контексте
- Например, задать стиль для тега `<b>` только когда он располагается внутри тега `<p>`
- Таким образом можно одновременно установить стиль для отдельного тега, а также для тега, который находится внутри другого

Контекстные селекторы

Синтаксис:

```
Тег1 Тег2 { ... }
```

будет применяться в случае

```
<Тег1> <Тег2> ... </Тег2> </Тег1>
```

Пример 7.

(!) Не обязательно, что контекстные селекторы содержат только один вложенный тег. В зависимости от ситуации допустимо применять два и более последовательно вложенных друг в друга тегов

Идентификаторы и классы можно использовать совместно

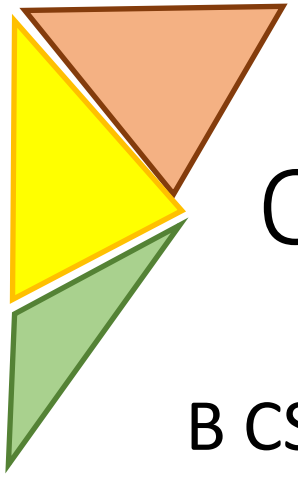
Пример 8.

Основные виды селекторов

- `*` - любые элементы
- `div` – элементы данного тега
- `#id` – элементы с данным id
- `.class` – элементы с данным классом
- `[name="value"]` – селекторы на атрибут
- `:visited` – “псевдоклассы”, остальные разные условия на элемент

Селекторы можно комбинировать, записывая последовательно, без пробелов:

- `.c1.c2` – элементы одновременно с двумя классами c1 и c2
- `a#id.c1.c2:visited` – элемент `a` с идентификатором `id` и классами `c1` и `c2`, а также псевдоклассом `visited`



Отношения

В CSS предусмотрено 4-е вида отношений между элементами:

- `div p` – *элементы* `p`, *являющиеся потомками* `div`
- `div > p` – *только непосредственные потомки*
- `div ~ p` – *правые соседи*: все `p` на том же уровне вложенности, которые идут после `div`
- `div + p` – *первый правый сосед*: `p` на том же уровне вложенности, который идёт сразу после `div` (если есть)

Пример

Балтославянские языки

1. Славянские языки
    1. Славянские микроязыки
    2. Праславянский язык
    3. *Восточнославянские языки* `#e-slavic`
    4. Западнославянские языки `#e-slavic ~ li`
    5. Южнославянские языки `#e-slavic ~ li`
    6. ... `#e-slavic ~ li`
2. Балтийские языки
    1. Литовский язык
    2. *Латышский язык* `#latvian`
        1. *Латгальский язык* `#latvian *`
    3. Прусский язык `#latvian + li`
    4. ... (следующий элемент уже не `#latvian + li`)

```
<style>
   #languages li {color: brown;}
   #languages > li {color: black;}
   #e-slavic {font-style: italic;}
   #e-slavic ~ li {color: red;}
   #latvian {font-style: italic;}
   #latvian * {font-weight: bold;}
   #latvian + li {color: green;}
   code {border: 1px solid black;}
 </style>
```

```
<h3>Балтославянские языки</h3>
  <ol id="languages">
    <li>Славянские языки
      <ol>
        <li>Славянские микроязыки</li>
        <li>Праславянский язык</li>
        <li id="e-slavic">Восточнославянские языки <code>#e-slavic</code></li>
        <li>Западнославянские языки <code>#e-slavic ~ li</code></li>
        <li>Южнославянские языки <code>#e-slavic ~ li</code></li>
        <li> ... <code>#e-slavic ~ li</code></li>
      </ol>
    </li>
    <li>Балтийские языки
      <ol>
        <li>Литовский язык</li>
        <li id="latvian">Латышский язык <code>#latvian</code>
          <ol>
            <li>Латгальский язык <code>#latvian *</code></li>
          </ol>
</li>
        <li>Прусский язык <code>#latvian + li </code></li>
        <li>... (следующий элемент уже не <code>#latvian + li</code>)</li>
      </ol></li></ol>
```

Фильтр по месту среди соседей

- При выборе элемента можно указать его место среди соседей
- Псевдоклассы:
  - `:first-child` – первый потомок своего родителя
  - `:last-child` – последний потомок своего родителя
  - `:only-child` – единственный потомок своего родителя, соседних элементов нет
  - `:nth-child(a)` – потомок номер *a* своего родителя
    Например `:nth-child(2)` – второй потомок
    (!) Нумерация начинается с 1
  - `:nth-child(an+b)` – указание номера потомка формулой, где `a`, `b` – константы, а под `n` подразумевается любое целое число
    Например `:nth-child(2n)` – потомки под чётными номерами
    `:nth-child(2n+1)` – потомки под нечётными номерами

Пример

- Древнерусский язык
- Древненовгородский диалект
- Западнорусский письменный язык
- Украинский язык
- Белорусский язык
- Другие языки

```
<!DOCTYPE HTML>
<html><head>
  <style>
    li:nth-child(2n) {background: #888;}
    li:nth-child(2n+1) {color: red;}
    code {border: 1px solid black;}
  </style>
</head><body>
  <ul>
    <li>Древнерусский язык</li>
    <li>Древненовгородский диалект</li>
    <li>Западнорусский письменный язык</li>
    <li>Украинский язык</code></li>
    <li>Белорусский язык</li>
    <li>Другие языки</li>
  </ul></body></html>
```

Фильтр по месту среди соседей с тем же тегом

Псевдоклассы, которые учитывают не всех соседей, а только с тем же тегом

- `:first-of-type`
- `:last_of_type`
- `:only-of-type`
- `:nth-of-type(odd | even | <число> | <выражение>)`
- `:nth-last-of-type(odd | even | <число> | <выражение>)`

Пример

Исторический турнир

```
<!DOCTYPE html>
<html> <head><title>nth-of-type</title>
  <style>
   img:nth-of-type(2n+1) {float: left;}
   img:nth-of-type(2n) {float: right;}
   div {width:600px;
        text-align: center;
        font-size: 24pt}
  </style>
 </head> <body>
  <div><img src="img2.png" width=100px>
   <img src="img1.png" width=100px>
   Исторический турнир
   </div> </body></html>
```

Селекторы атрибутов

- На атрибут целиком
  - `[attr]` – атрибут установлен
  - `[attr="val"]` – атрибут равен val
- На начало атрибута
  - `[attr^="val"]` – атрибут начинается с val, например «value»
  - `[attr|="val"]` – атрибут равен val или начинается с val
- На содержание
  - `[attr*="val"]` – атрибут содержит подстроку val, например, «myvalue»
  - `[attr~="val"]` – атрибут содержит val как одно из значений через пробел
- На конец атрибута
  - `[attr$="val"]` – атрибут заканчивается на val, например, «myval»

Пример

GOOGLE | Yandex

```
<!DOCTYPE HTML>
<html> <head>  <title>Селекторы атрибутов</title>
  <style>
   a[target="_blank"] {
    background: url(blank.png) 0 2px no-repeat
    padding-left: 25px;
   }
  </style>
  </head> <body>
   <p><a href="http://google.com">GOOGLE</a> |
   <a href="http://yandex.ru" target="_blank">Yandex</a></p>
 </body></html>
```

Другие псевдоклассы

- `:not(селектор)` – все кроме подходящих под селектор
- `:focus` – в фокусе
- `:hover` – под указателем мыши
- `:empty` – без потомков (даже без текстовых)
- `:checked`, `:disabled`, `:enabled` – состояния `input`
- `:target` – фильтр сработает для элемента `id`, которого совпадает с анкором `#...` текущего *url*
  Например, если на странице есть элемент с `id="intro"`,
  то правило `:target {color: red}` подсветит его в том случае, если текущий URL имеет вид http://...#intro

Псевдоэлементы ::before и ::after

- Псевдоэлементы – различные вспомогательные элементы, которые браузер может записать в докмуент
- При помощи ::before и ::after можно добавить содержимое в начало или конец элемента

```
<!DOCTYPE HTML>
<html><head>
<style>
        li::before{content: "**[["}
        li::after{content: "]]**"}
</style>
</head><body>
<ul>
        <li>First element</li>
        <li>Second element</li>
</ul></body></html>
```

- **[[First element]]**
- **[[Second element]]**

Значения свойств селекторов

- Строки
  - Любые строки необходимо брать в двойные или одинарные кавычки. Если внутри строки требуется оставить одну или несколько кавычек, то можно комбинировать типы кавычек или добавить перед кавычкой слэш
- Числа
  - Значением может выступать целое число, содержащее цифры от 0 до 9 и десятичная дробь, в которой целая и десятичная часть разделяются точкой
- Проценты
  - Процентная запись обычно применяется, когда надо изменить значение относительно родительского элемента или когда размеры зависят от внешних условий

Значения свойств селекторов

- Размеры
  - Для задания размеров различных элементов в CSS используются абсолютные и относительные единицы измерения

Относительные единицы

определяют размер элемента относительно значения другого размера

Единица	Описание
em	Размер шрифта текущего элемента
ex	Высота символа x
px	Пиксел
%	Процент

Абсолютные единицы

не зависят от устройства вывода

Единица	Описание
in	Дюйм (1 дюйм равен 2,54 см)
cm	Сантиметр
mm	Миллиметр
pt	Пункт (1 пункт равен 1/72 дюйма)
pc	Пика (1 пика равна 12 пунктам)

Значения свойств селекторов

- Цвет
  - задается:
    - по 16-ому значению,
    - по названию
    - в формате RGB

```
<!DOCTYPE HTML>
<html> <head>
  <style>
    body{background-color: #3366CC;}
      h1{background-color: rgb(249, 201, 16);}
      p{background-color: maroon;
        color: white;}
  </style>
</head>
<body>
<h1>Деконструктивизм</h1>
<p>...</p>
</body></html>
```

Значения свойств селекторов

- Адреса
  - URL, унифицированный идентификатор ресурсов
  - Применяются для указания пути к файлу
  - Используется ключевое слово url(), внутри скобок пишется адрес файла
- Ключевые слова
  - В качестве значений активно применяются ключевые слова, которые определяют желаемый результат действия стилевых свойств
  - Ключевые слова пишутся без кавычек

```
p {text-align: right;}
p {text-align: "right";}
```