

# HTML + CSS

Кузнецов Василий

# HTML

# Типичный html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>HTML</title>
  </head>
  <body>
    <p>
      <a href="https://www.w3.org/html/">HTML</a>
      – это стандартизированный язык
      разметки документов в Вебе.
    </p>
  </body>
</html>
```



# HTML-элементы (теги)

```
<a href="https://www.w3.org/html/">HTML</a>
```

# HTML-элементы (теги)

## Открывающий тег

```
<a href="https://www.w3.org/html/">HTML</a>
```

# HTML-элементы (теги)

## Закрывающий тег

```
<a href="https://www.w3.org/html/">HTML</a>
```

# HTML-элементы (теги)

## Атрибут

```
<a href="https://www.w3.org/html/">HTML</a>
```

# HTML-элементы (теги)

## Содержимое

```
<a href="https://www.w3.org/html/">HTML</a>
```



# HTML-элементы (теги)

```
<!-- Содержит другие теги -->
```

```
<p>
```

```
  <a href="https://www.w3.org/html/">HTML</a>
```

```
  - это стандартизированный язык разметки...
```

```
</p>
```

```
<!-- Только текст -->
```

```
<title>HTML</title>
```

```
<!-- Без содержимого -->
```

```

```

# Атрибуты

```
<!-- Их может быть несколько -->
```

```
<input type="button" value="Click me">
```

```
<!-- Кавычки необязательны... -->
```

```
<input type=button value=Click>
```

```
<!-- ... но для единообразия лучше ставить -->
```

```
<!-- Бывают булевы атрибуты без значений -->
```

```
<input disabled>
```

```
<!-- То же самое: -->
```

```
<input disabled="sometext">
```

# Универсальные атрибуты

# Универсальные атрибуты

id – задает **уникальный** идентификатор элемента

```
<input id="email-input" type="email">
```

```
/*В стилях*/  
#email-input { width: 100px; }
```

```
// Получаем доступ к элементу как к JS объекту по id  
var email = document.getElementById('email-input');
```

У элемента может быть только **один** id

# Универсальные атрибуты

class – задает **класс** элементов, позволяя группировать их

Список покупок:

```
<ul class="list">  
  <li class="food">Молоко</li>  
  <li class="food">Хлеб</li>  
  <li class="car">Audi TT</li>  
  <li class="car">Audi Q5</li>
```

```
/*В стилях*/  
.food { color: blue; }  
.car { color: red; }
```

# Универсальные атрибуты

У элемента может быть **несколько** классов, тогда они записываются через пробел

```
<a class="big red link">Большая красная ссылка</a>
```

# Универсальные атрибуты

Можно считать, что `id` задает класс из одного элемента, и для единообразия использовать `class`

```
<input id="email-input" type="email">
```

```
/*В стилях*/  
#email-input { width: 100px }
```

```
// Получаем доступ к элементу как к JS объекту по id  
var email = document.getElementById('email-input');
```

# Универсальные атрибуты

Можно считать, что id задает класс из одного элемента, и для единообразия использовать class

```
<input class="email-input" type="email">
```

```
/*В стилях*/  
.email-input { width: 100px }
```

```
// Получаем доступ к элементу как к JS объекту по классу  
var email = document.querySelector('.email-input');
```



# Универсальные атрибуты

title – добавляет пояснение к содержимому элемента, которое появляется при наведении

```
<abbr title="World Wide Web">www</abbr>
```



```
<p title="application programming interface">API</p>
```

# HTML Entity

<code>&lt;p&gt;&amp;copy;&lt;/p&gt;</code>	<code>&lt;!-- © --&gt;</code>
<code>&lt;p&gt;&amp;amp;&lt;/p&gt;</code>	<code>&lt;!-- &amp; --&gt;</code>
<code>&lt;p&gt;&amp;forall;&lt;/p&gt;</code>	<code>&lt;!-- ∀ --&gt;</code>
<code>&lt;p&gt;&amp;lt;&lt;/p&gt;</code>	<code>&lt;!-- &lt; --&gt;</code>
<code>&lt;p&gt;&amp;nbsp;&lt;/p&gt;</code>	<code>&lt;!-- Неразрывный пробел(no-break space) --&gt;</code>
<code>&lt;p&gt;&amp;#x0221E;&lt;/p&gt;</code>	<code>&lt;!-- ∞ --&gt;</code>

`<!-- &lt; используется для экранирования тегов: -->`  
`&lt;input type="button">`

Отображается как: `<input type="button">`

# HTML-элементы

# Структура html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8" />  
    <title>HTML</title>  
  </head>  
  <body>  
    <p>  
      <a href="https://www.w3.org/html/">HTML</a>  
      – это стандартизированный язык  
      разметки документов в Вебе.  
    </p>  
  </body>  
</html>
```

# Структура html

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8" />
```

```
    <title>HTML</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>
```

```
      <a href="https://www.w3.org/html/">HTML</a>
```

– это стандартизированный язык  
разметки документов в Вебе.

```
    </p>
```

```
  </body>
```

```
</html>
```

# !DOCTYPE

**!DOCTYPE** – директива, которая указывает на спецификацию, по которой был создан документ

Для стандарта HTML5 всегда используется:

```
<!DOCTYPE html>
```

Поэтому всегда используем его  
Обязательно надо указывать

# Структура html

```
<!DOCTYPE html>
<html>
  <head> <!-- содержит вспомогательную информацию -->
    <meta charset="utf-8" />
    <title>HTML</title>
  </head>
  <body>
    <p>
      <a href="https://www.w3.org/html/">HTML</a>
      – это стандартизированный язык
      разметки документов в Вебе.
    </p>
  </body>
</html>
```

# head

title – задает заголовок страницы, отображаемый во вкладке браузера

```
<head>  
  <!-- title этой презентации -->  
  <title>HTML + CSS<title>  
</head>
```



# head

link – описывает связь с внешним документом, например, файлом со стилями или шрифтами

```
<head>
  <link href="styles.css">

  <!-- media="all/screen/print/speech/braille" -->
  <link href="print-styles.css" media="print">

  <!-- rel - описывает тип связи -->
```

# head

script/style – описывают скрипты/стили

```
<head>  
  <script src="path/jquery-plugin.js"></script>  
  <style>  
    p {  
      color: red;  
    }  
  </style>  
</head>
```

# head

script/style могут находиться в body

```
<body>
  <script>
    console.log('Hello!');
  </script>
  <style>
    p {
      color: red;
    }
  </style>
</body>
```

# head

meta – метатеги, предназначенные для браузеров, поисковых систем, соцсетей

```
<head>
  <meta charset="utf-8">
  <!-- name="название" content="значение" -->
  <meta name="description" content="HTML lecture">
  <meta name="author" content="kvas">
</head>
```

Используется в [Open Graph](#)

# Структура html

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8" />  
    <title>HTML</title>  
  </head>  
  <body> <!-- содержимое страницы -->  
    <p>  
      <a href="https://www.w3.org/html/">HTML</a>  
      - это стандартизированный язык  
      разметки документов в Вебе.  
    </p>  
  </body>  
</html>
```

# Базовые теги

div – блочный элемент, span – строчный элемент

```
<div>
  <div><span title="HyperText Markup Language">HTML</span>
  - это ... </div>
  <div>Как правило, передаются ... </div>
  <div>Последнее изменение: 2001-04-23</div>
  <div>Автор: someone@example.com</div>
</div>
```

Несемантично – непонятно, какой элемент что значит  
Лучше использовать только для добавления стилей

# Семантическая верстка

```
<!-- Обозначаем статью -->
<article>
  <!-- Аббревиатура -->
  <p><abbr title="HyperText Markup Language">HTML</abbr>
  - это ... </p>
  <p>Как правило, передаются ... </p>
</article>
<!-- Информацию в подвале -->
<footer>
  <!-- Время -->
  <p>Последнее изменение: <time>2001-04-23</time></p>
  <!-- И контактную информацию -->
  <address>Автор: someone@example.com</address>
</footer>
```

# Семантические теги

- section/article
- aside
- header/main/footer
- figure и figcaption
- h1-h6
- nav
- ...



# Что будет?

<p>



,--. ,--. ,--. ,----. ,--.  
| \.' | / 0 \ | .--. " .-'  
| |'.' | | .-. | | '---'.' \-'  
| | | | | | | | | \ \ .-' |  
'--' '--' '--' '--' '--' '--' '--' '--' '--' '--'

</p>

Подряд идущие пробельные символы схлапываются в один пробел

```
pre(preformatted text)
```

```


```

```


```

# Списки

Теги ul, ol, li

```
<ol type="I">
  <li>Сельдь</li>
  <li>Хлеб</li>
  <li>Фрукты:
    <ul>
      <li>Бананы</li>
      <li>Яблоки</li>
    </ul>
  </li>
  <li>Молоко</li>
  <li>Audi TT</li>
</ol>
```

# Таблицы

Теги table, thead, tfoot, tbody, th, tr, td

```
<table border="1" cellspacing="0">  
  <tr>  
    <td>A 1.1</td>  
    <td>A 1.2</td>  
    <td>A 1.3</td>  
  </tr>  
  <tr>  
    <td>A 2.1</td>  
    <td>A 2.2</td>  
    <td>A 2.3</td>  
  </tr>  
</table>
```

# Таблицы

## Объединение ячеек

```
<table border="1" cellspacing="0">  
  <tr>  
    <!-- горизонтальные ячейки -->  
    <td colspan="2">A 1.1</td>  
    <!-- <td>A 1.2</td> -->  
    <!-- вертикальные ячейки -->  
    <td rowspan="2">A 1.3</td>  
  </tr>  
  <tr>  
    <td>A 2.1</td>  
    <td>A 2.2</td>  
    <!-- <td>A 2.3</td> -->  
  </tr>  
</table>
```



# Текстовые элементы

- Форматирование текста:
  - `em` - для акцентирования текста
  - `strong` – для обозначения важной информации
  - `mark` – помечает текст как выделенный
  - `small` – для различной уточняющей информации
  - `i` – для дополнительного выделения(экспрессивно-эмоциональным)
  - `b` – для выделения из окружающего его контекста(ключевые слова в выдержках)
  - `<sub>` `<sup>` `<ins>` `<del>`
- Для кода: `<code>` `<kbd>` `<samp>` `<var>` `<pre>`
- Перенос текста: `<p>` `<br>` `<hr>` `<wbr>`
- Цитаты/определения: `<abbr>` `<blockquote>` `<q>` `<dfn>` `<cite>` `<bdo>`
- Примеры

# ССЫЛКИ

```
<a href="https://yandex.ru">Yandex</a>
```

```
<!-- Открывается в новой вкладке/окне -->
```

```
<a href="https://yandex.ru" target="_blank">Yandex</a>
```

```
<!-- Относительно текущего пути -->
```

```
<a href="examples/title.html" >Yandex</a>
```

```
<!-- https://example.ru/01-html-css/examples/title.html -->
```

```
<!--Относительно текущего хоста-->
```

```
<a href="/examples/title.html" >Yandex</a>
```

```
<!-- https://example.ru/examples/title.html -->
```

# Ссылки

## Протоколы

```
<a href="http://images.yandex.ru">По http</a>
```

```
<a href="//images.yandex.ru">По тому же протоколу</a>
```

```
<a href="mailto:kvas@yandex-team.ru">Написать письмо</a>
```

```
<a href="skype:kvaser5">Позвонить по скайпу</a>
```

```
<a href="tel:+79000000000">Позвонить</a>
```



# Ссылки

## Якоря

```
<a href="#anchor">ссылка на якорь</a>
```

```
<article>Много текста...</article>
```

```
<!-- При нажатии на ссылку перейдем к этому элементу -->
```

```
<div id="anchor">якорь</div>
```

# Картинки

```

```

Картинка загрузилась:



Ошибка загрузки:

 image of a cute cat

# Картинки

## Атрибут alt

- Используется, если картинку не удалось загрузить
- Используется, если загрузка картинок отключена пользователем
- Используется [скрин-ридерами](#)

[Подробнее про alt](#)

# Размеры картинок

```

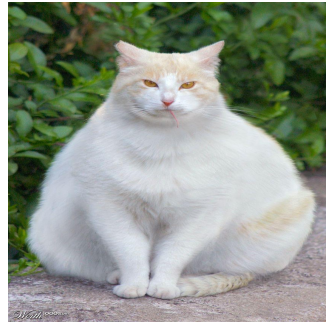
```

```

```

```

```



# Теги для вставки контента

- img
- iframe
- audio/video
- track

# Получить инвайт на курс

\* Имя

\* Фамилия

\* Login на GitHub

В том же регистре, как в GitHub'e

\* Login в Telegram

- \* ☐ Я даю свое согласие на передачу в ООО «ЯНДЕКС» анкеты, содержащей мои персональные данные, и согласен с тем, что они будут храниться в ООО «ЯНДЕКС» в течение 10 лет и будут использованы исключительно для целей приглашения меня к участию в мероприятиях группы компаний «ЯНДЕКС», в соответствии с Федеральным законом «О персональных данных».

Отправить

# Формы

```
<form action="/send-data">
  <input type="text" name="name">
  <input type="text" name="phone">
  <select name="option">
    <option selected value="a">A</option>
    <option value="b">B</option>
  </select>
  <textarea name="text" cols="30" rows="10"></textarea>
  <button>Submit</button>
</form>
```

# Формы

```
<form action="/send-data">  
  <input type="text" name="name">  
  <input type="text" name="phone">  
  <select name="option">  
    <option selected value="a">A</option>  
    <option value="b">B</option>  
  </select>  
  <textarea name="text" cols="30" rows="10"></textarea>  
  <button>Submit</button>  
</form>
```



# input

```
<input type="text">
```

```
<!-- Предустановленное значение "Текстовое поле" -->
```

```
<input type="text" value="Текстовое поле">
```

```
<!-- Отображается "Введите текст", если поле пустое -->
```

```
<input type="text" placeholder="Введите текст">
```

```
<!-- Все символы показываются звездочками -->
```

```
<input type="password">
```

```
<!-- Поле для указания файла -->
```

```
<input type="file">
```

# input

type="checkbox"/"radio"

```
<!-- Позволяют выбрать более одного  
варианта из предложенных -->  
<input type="checkbox" checked> Хлеб  
<input type="checkbox"> Молоко
```

```
<!-- Используются, когда следует выбрать  
один вариант из нескольких предложенных -->  
<input type="radio" name="drink"> Вода  
<input type="radio" name="drink" checked> Сок
```

input

ДЕМО: Атрибут type в HTML5

Подробнее про type

# select

```
<select name="hero">  
  <option selected>Выберите героя</option>  
  <option value="Чебурашка">Чебурашка</option>  
  <option value="Крокодил Гена">Крокодил Гена</option>  
  <option value="Шапокляк">Шапокляк</option>  
  <option value="Крыса Лариса">Крыса Лариса</option>  
</select>
```

# textarea

```
<textarea  
  rows="7"  
  cols="35"  
  name="text"  
  placeholder="Введите текст"  
>Текст</textarea>
```



# button

```
<button>Это кнопка</button>
```

```
<input type="button" value="Тоже кнопка">
```

```
<button disabled>Эта кнопка заблокирована</button>
```

```
<!-- type="button | reset | submit" -->
```

```
<button type="button">Это кнопка</button>
```



# label

```
<p>Выберите напиток</p>
```

```
<input type="checkbox" id="water-checkbox">
```

```
<!-- Устанавливает связь между определённой меткой и  
элементом формы -->
```

```
<label for="water-checkbox">Вода</label>
```

```
<label><input type="checkbox">Сок</label>
```



# Формы

```
<form action="/send-data">  
  <input type="text" name="name">  
  <input type="text" name="phone">  
  <select name="option">  
    <option selected value="a">A</option>  
    <option value="b">B</option>  
  </select>  
  <textarea name="text" cols="30" rows="10"></textarea>  
  <button>Submit</button>  
</form>
```



# form

Служит для передачи данных на сервер

```
<!-- method - указывает http-метод запроса -->
<!-- action - указывает обработчик, к которому обращаются
данные формы при их отправке на сервер -->
<form method="get" action="/handler">
    <input name="name" placeholder="Ваше ФИО">
    <input name="email" placeholder="Ваш email" type="email">
    <input name="secret" type="hidden" value="secretvalue">
    <button type="submit">Отправить</button>
    <button type="reset">Сбросить</button>
</form>
```

name=[значение1]&email=[значение2]&secret=[значение3]

Подробнее про формы

# Валидация

```
<form method="post">  
  <input type="text" required placeholder="Как вас зовут?">  
  
  <input type="email" placeholder="email" required>  
  
  <input type="url" placeholder="url" required>  
  
  <input type="tel" pattern="2-[0-9]{3}-[0-9]{3}"  
    placeholder="2-123-312" required>  
  
  <button>Отправить</button>  
  <button type="reset">Очистить</button>  
</form>
```

# Валидация

## Обязательное поле

```
<form method="post">  
  <input type="text" required placeholder="Как вас зовут?">  
  
  <input type="email" placeholder="email" required>  
  
  <input type="url" placeholder="url" required>  
  
  <input type="tel" pattern="2-[0-9]{3}-[0-9]{3}"  
    placeholder="2-123-312" required>  
  
  <button>Отправить</button>  
  <button type="reset">Очистить</button>  
</form>
```

# Валидация

## Проверка соответствия типу

```
<form method="post">  
  <input type="text" required placeholder="Как вас зовут?">  
  
  <input type="email" placeholder="email" required>  
  
  <input type="url" placeholder="url" required>  
  
  <input type="tel" pattern="2-[0-9]{3}-[0-9]{3}"  
    placeholder="2-123-312" required>  
  
  <button>Отправить</button>  
  <button type="reset">Очистить</button>  
</form>
```

# Валидация

## Проверка соответствия шаблону

```
<form method="post">  
  <input type="text" required placeholder="Как вас зовут?">  
  
  <input type="email" placeholder="email" required>  
  
  <input type="url" placeholder="url" required>  
  
  <input type="tel" pattern="2-[0-9]{3}-[0-9]{3}"  
    placeholder="2-123-312" required>  
  
  <button>Отправить</button>  
  <button type="reset">Очистить</button>  
</form>
```

# Полезные ссылки по HTML

- [HTML I](#)
- [HTML II](#)
- [div или section или article?](#)
- [W3C](#)
- [HTML5BOOK](#)
- [Челлендж на знание HTML-тегов](#)

Перерыв

CSS



**CSS** (каскадные таблицы стилей) - это язык,  
отвечающий за внешний вид  
HTML-документа

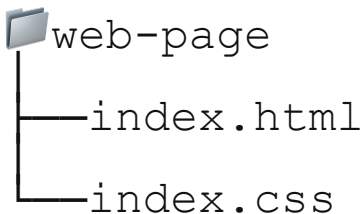
# Добавление стилей к HTML

## Определить стили в HTML

```
<style>
  .p {
    color: red;
  }
</style>
```

# Добавление стилей к HTML

## Отдельным файлом



```
<!-- index.html -->
<head>
  <link rel="stylesheet" href="./index.css">
</head>
```

# Типичный CSS

```
@import url('https://example.ru/font');
```

```
div {  
    color: white;  
}
```

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

```
#square {  
    width: 100px;  
    height: 100px;  
}
```

# Типичный CSS

```
@import url('https://example.ru/font');
```

```
div {  
    color: white;  
}
```

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

```
#square {  
    width: 100px;  
    height: 100px;  
}
```

# Правило

свойство: значение; /\* правило \*/

# Типичный CSS

```
@import url('https://example.ru/font');
```

```
div {
```

```
    color: white;
```

```
}
```

```
.red {
```

```
    background-color: red;
```

```
}
```

```
#square {
```

```
    width: 100px;
```

```
    height: 100px;
```

```
}
```

# Блок объявления стилей

```
селектор {  
    свойство: значение;  
}
```



# Блок объявления стилей

/\* можно указать набор правил \*/

селектор {

    свойство: значение;

    свойство: значение;

    свойство: значение;

}

# Блок объявления стилей

```
/* можно указать набор селекторов для набора правил */  
селектор,  
селектор,  
селектор {  
    свойство: значение;  
    свойство: значение;  
    свойство: значение;  
}
```

# Типичный CSS

```
@import url('https://example.ru/font');
```

```
div {  
    color: white;  
}
```

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

```
#square {  
    width: 100px;  
    height: 100px;  
}
```

# At-правила

- @charset – задает кодировку
- @import – подключает внешний css-файл
- @media
- @document
- @font-face
- @supports

# Селекторы

# Селектор тега

```
p {  
    color: red;  
}  
  
h1 {  
    font-size: 34px;  
}  
  
nav {  
    opacity: .5;  
}
```

```
<h1>Очень важный текст</h1>  
<nav>  
    <a href="/">Главная</a>  
    <a href="/profile">Профиль</a>  
</nav>  
  
<p>Вот вам яркий пример  
современных тенденций – реализация  
намеченных плановых заданий является  
качественно новой ступенью  
модели развития!</p>
```

# Селектор тега

```
p {  
  color: red;  
}
```

```
h1 {  
  font-size: 34px;  
}
```

```
nav {  
  opacity: .5;  
}
```

```
<h1>Очень важный текст</h1>
```

```
<nav>
```

```
  <a href="/">Главная</a>
```

```
  <a href="/profile">Профиль</a>
```

```
</nav>
```

```

```

```
<p>Вот вам яркий пример
```

```
современных тенденций – реализация
```

```
намеченных плановых заданий является
```

```
качественно новой ступенью
```

```
модели развития!</p>
```

# Селектор класса

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

```
.title {  
  font-size: 34px;  
}
```

```
.info {  
  opacity: .5;  
}
```

```
<h1 class="title">Очень важный текст</h1>  
<nav class="nav">  
  <a href="/" class="red">Главная</a>  
  <a href="/profile" class="blue">Профиль</a>  
</nav>  
  
<p class="info red">Вот вам яркий пример  
современных тенденций – реализация  
намеченных плановых заданий является  
качественно новой ступенью  
модели развития!</p>
```



# Селектор класса

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

```
.title {  
  font-size: 34px;  
}
```

```
.info {  
  opacity: .5;  
}
```

```
<h1 class="title">Очень важный текст</h1>
```

```
<nav class="nav">
```

```
  <a href="/" class="red">Главная</a>
```

```
  <a href="/profile" class="blue">Профиль
```

```
</nav>
```

```

```

```
<p class="info red">Вот вам яркий пример
```

```
современных тенденций – реализация
```

```
намеченных плановых заданий является
```

```
качественно новой ступенью
```

```
модели развития!</p>
```

# Селектор идентификатора

```
#red {  
  color: red;  
}
```

```
#title {  
  font-size: 34px;  
}
```

```
#info {  
  opacity: .5;  
}
```

```
<h1 id="title">Очень важный текст</h1>  
<nav id="nav">  
  <a href="/" id="red">Главная</a>  
  <a href="/profile" id="blue">Профиль</a>  
</nav>  
  
<p id="info">Вот вам яркий пример  
современных тенденций – реализация  
намеченных плановых заданий является  
качественно новой ступенью  
модели развития!</p>
```

# Селектор идентификатора

```
#red {  
  color: red;  
}
```

```
#title {  
  font-size: 34px;  
}
```

```
#info {  
  opacity: .5;  
}
```

```
<h1 id="title">Очень важный текст</h1>
```

```
<nav id="nav">
```

```
  <a href="/" id="red">Главная</a>
```

```
  <a href="/profile" id="blue">Профиль</a>
```

```
</nav>
```

```

```

```
<p id="info">Вот вам яркий пример  
современных тенденций – реализация  
намеченных плановых заданий является  
качественно новой ступенью  
модели развития!</p>
```

# Универсальный селектор

```
* {  
    font-size: 24px;  
}
```

```
<h1>Очень важный текст</h1>  
<nav>  
    <a href="/">Главная</a>  
    <a href="/profile">Профиль</a>  
</nav>  
  
<p>Вот вам яркий пример  
современных тенденций – реализация  
намеченных плановых заданий является  
качественно новой ступенью  
модели развития!</p>
```

# Универсальный селектор

```
* {  
  font-size: 24px;  
}
```

`<h1>`Очень важный текст`</h1>`  
`<nav>`  
    `<a href="/">`Главная`</a>`  
    `<a href="/profile">`Профиль`</a>`  
`</nav>`  
``  
`<p>`Вот вам яркий пример  
современных тенденций – реализация  
намеченных плановых заданий является  
качественно новой ступенью  
модели развития!`</p>`

Стоит использовать с осторожностью

# Селекторы атрибутов

```
/* применится, если атрибут есть */
```

```
[href] {
```

```
    color: red;
```

```
}
```

```
<a href="https://yandex.ru">Яндекс</a>
```

```
<a>Пустая ссылка</a>
```

Яндекс

Пустая ссылка

# Селекторы атрибутов

```
/* применится, если равен "https://yandex.ru" */  
[href="https://yandex.ru"] {
```

```
    color: red;  
}
```

```
<a href="https://yandex.ru">Яндекс</a>  
<a href="https://google.com">Google</a>
```

Яндекс  
Google

# Селекторы атрибутов

```
/* применится, если начинается с "https://" */  
[href^="https://"] {  
  
    color: red;  
}
```

```
<a href="https://yandex.ru">Яндекс</a>  
<a href="https://google.com">Google</a>
```

Яндекс  
Google



# Селекторы атрибутов

```
/* применится, если кончается на "yandex.ru" */  
[href$="yandex.ru"] {  
  
    color: red;  
  
}
```

```
<a href="https://yandex.ru">Яндекс</a>  
<a href="https://yandex.ru/maps">Яндекс.Карты</a>
```

Яндекс  
Яндекс.Карты

# Селекторы атрибутов

```
/* применится, если содержит на "yandex.ru" */  
[href*="yandex.ru"] {  
  
    color: red;  
}
```

```
<a href="https://yandex.ru">Яндекс</a>  
<a href="https://yandex.ru/maps">Яндекс.Карты</a>
```

Яндекс  
Яндекс.Карты

# Псевдоклассы

- :active
- :any-link
- :checked
- :default
- :dir()
- :disabled
- :empty
- :enabled
- :first
- :first-child
- :first-of-type
- :fullscreen
- :focus
- :hover
- :indeterminate
- :in-range
- :invalid
- :lang()
- :last-child
- :last-of-type
- :left
- :link
- :not()
- :nth-child()
- :nth-last-child()
- :nth-last-of-type()
- :nth-of-type()
- :only-child
- :only-of-type
- :optional
- :out-of-range
- :read-only
- :read-write
- :required
- :right
- :root
- :scope
- :target
- :valid
- :visited

# Динамические псевдоклассы

```
a:visited { color: green; }
```

```
a:focus { color: blue; }
```

```
a:hover { color: red; }
```

```
<a>Ссылка</a>
```

Ссылка

# Динамические псевдоклассы

```
input:disabled {  
    background: red;  
}
```

```
input:enabled {  
    background: green;  
}
```

```
<input type="text" disabled placeholder="Неактивный">  
<input type="text" placeholder="Активный">
```

Неактивный

Активный

# Динамические псевдоклассы

```
input:valid {  
    background: green;  
}
```

```
input:invalid {  
    background: red;  
}
```

```
<input type="text" pattern="^[a-zA-Z]*$">
```

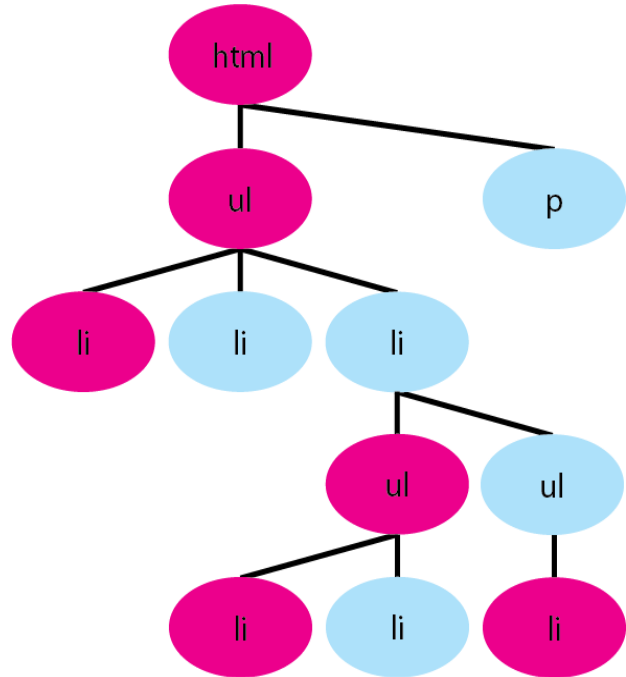
Поле с валидацией

# Структурные псевдоклассы

- :root
- :first-child/last-child
- :nth-child/nth-last-child( $2n+1$ )
- :nth-of-type/nth-last-of-type( $-n+4$ )
- :only-child/only-of-type
- :empty

# :first-child

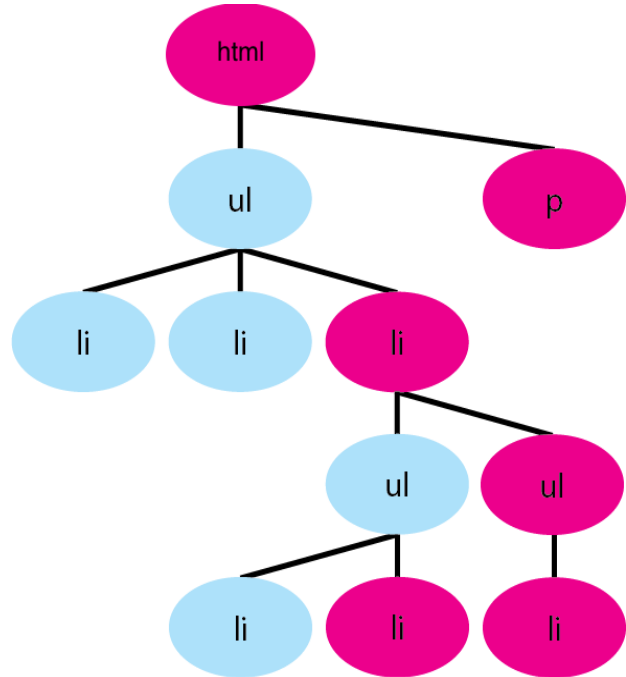
```
<html>  
  <ul>  
    <li></li>  
    <li></li>  
    <li>  
      <ul>  
        <li></li>  
        <li></li>  
      </ul>  
      <ul>  
        <li></li>  
      </ul>  
    </li>  
  </ul>  
  <p></p>  
</html>
```





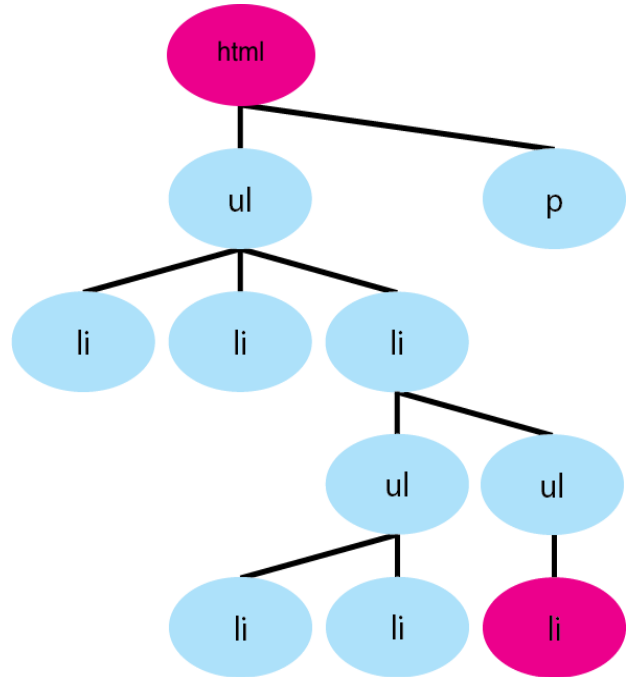
# :last-child

```
<html>
  <ul>
    <li></li>
    <li></li>
    <li>
      <ul>
        <li></li>
        <li></li>
      </ul>
      <ul>
        <li></li>
      </ul>
    </li>
  </ul>
  <p></p>
</html>
```



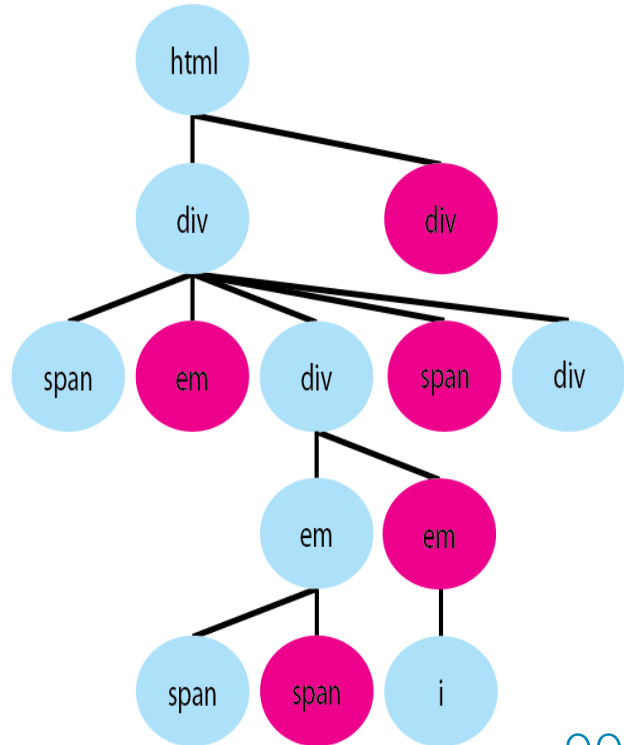
# :only-child

```
<html>
  <ul>
    <li></li>
    <li></li>
    <li>
      <ul>
        <li></li>
        <li></li>
      </ul>
      <ul>
        <li></li>
      </ul>
    </li>
  </ul>
  <p></p>
</html>
```



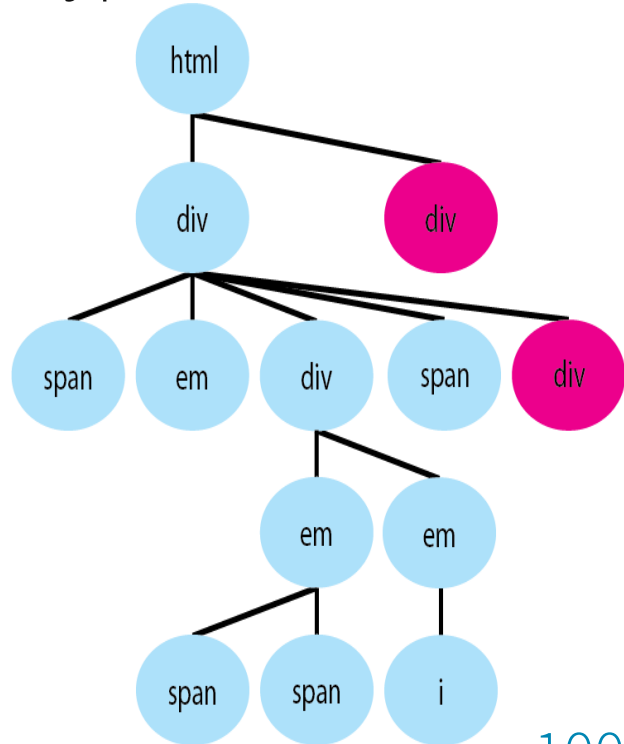
# :nth-child(2n)

```
<html>
  <div>
    <span></span>
    <em></em>
    <div>
      <em>
        <span></span>
        <span></span>
      </em>
      <em>
        <i></i>
      </em>
    </div>
    <span></span>
    <div></div>
  </div>
  <div></div>
</html>
```



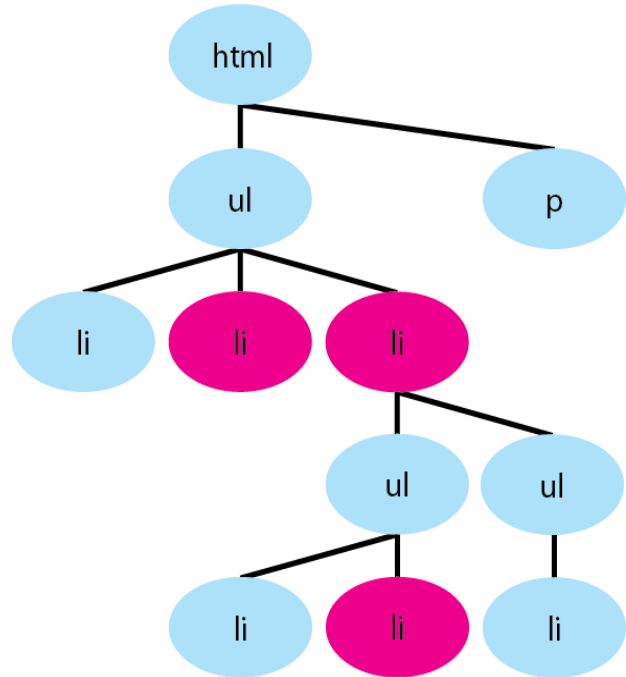
# div:nth-of-type(2n)

```
<html>  
  <div>  
    <span></span>  
    <em></em>  
    <div>  
      <em>  
        <span></span>  
        <span></span>  
      </em>  
      <em>  
        <i></i>  
      </em>  
    </div>  
    <span></span>  
    <div></div>  
  </div>  
  <div></div>  
</html>
```



# li:not(:first-child)

```
<html>
  <ul>
    <li></li>
    <li></li>
    <li>
      <ul>
        <li></li>
        <li></li>
      </ul>
      <ul>
        <li></li>
      </ul>
    </li>
  </ul>
  <p></p>
</html>
```



# Псевдоэлементы

- ::before
- ::after
- ::first-letter
- ::first-line
- ::selection
- ::backdrop

# Псевдоэлементы

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

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

Первая строка

Вторая строка

Первая строка

Вторая строка

# Псевдоэлементы

```
<div>::beforeCSS is awesome::after</div>
```



# Псевдоэлементы

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

CSS is awesome

«CSS is awesome»

# Комбинации

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

```
.info.link { color: green; }
```

```
a[href].link { opacity: .9; }
```

```
a.link:hover { color: blue; }
```

```
#link#big { opacity: .9 }
```

```
<a href="/" class="link">На главную</a>
```

```
<div class="info">
```

Вот вам яркий пример

современных тенденций –  
реализация намеченных  
плановых заданий является  
качественно новой ступенью  
модели развития!

```
</div>
```

```
<div></div>
```

# Комбинации

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

```
.info.link { color: green; }
```

```
a[href].link { opacity: .9; }
```

```
a.link:hover { color: blue; }
```

```
#link#big { opacity: .9 }
```

```
<a href="/" class="link">На главную</a>
```

```
<div class="info">
```

Вот вам яркий пример

современных тенденций –  
реализация намеченных  
плановых заданий является  
качественно новой ступенью  
модели развития!

```
</div>
```

```
<div></div>
```

# Комбинации

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

```
.info.link { color: green; }
```

```
a[href].link { opacity: .9; }
```

```
a.link:hover { color: blue; }
```

```
#link#big { opacity: .9 }
```

```
<a href="/" class="link">На главную</a>  
<div class="info">
```

Вот вам яркий пример

современных тенденций –  
реализация намеченных  
плановых заданий является  
качественно новой ступенью  
модели развития!

```
</div>
```

```
<div></div>
```

# Комбинации

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

```
.info.link { color: green; }
```

```
a[href].link { opacity: .9; }
```

```
a.link:hover { color: blue; }
```

```
#link#big { opacity: .9 }
```

```
<a href="/" class="link">На главную</a>
```

```
<div class="info">
```

Вот вам яркий пример

современных тенденций –  
реализация намеченных  
плановых заданий является  
качественно новой ступенью  
модели развития!

```
</div>
```

```
<div></div>
```

# Комбинации

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

```
.info.link { color: green; }
```

```
a[href].link { opacity: .9; }
```

```
a.link:hover { color: blue; }
```

```
#link#big { opacity: .9 }
```

```
<a href="/" class="link">На главную</a>
```

```
<div class="info">
```

Вот вам яркий пример

современных тенденций –  
реализация намеченных  
плановых заданий является  
качественно новой ступенью  
модели развития!

```
</div>
```

```
<div></div>
```

\*Применится при наведении

# Комбинации

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

```
.info.link { color: green; }
```

```
a[href].link { opacity: .9; }
```

```
a.link:hover { color: blue; }
```

```
#link#big { opacity: .9; }
```

```
<a href="/" class="link">На главную</a>
```

```
<div class="info">
```

Вот вам яркий пример

современных тенденций –  
реализация намеченных  
плановых заданий является  
качественно новой ступенью  
модели развития!

```
</div>
```

```
<div></div>
```

\*У одного элемента не может быть двух id

# Комбинаторы

- $\text{div div}$  – потомок
- $\text{div} > \text{div}$  – сын
- $\text{div} + \text{div}$  – младший брат
- $\text{div} \sim \text{div}$  – младшие братья

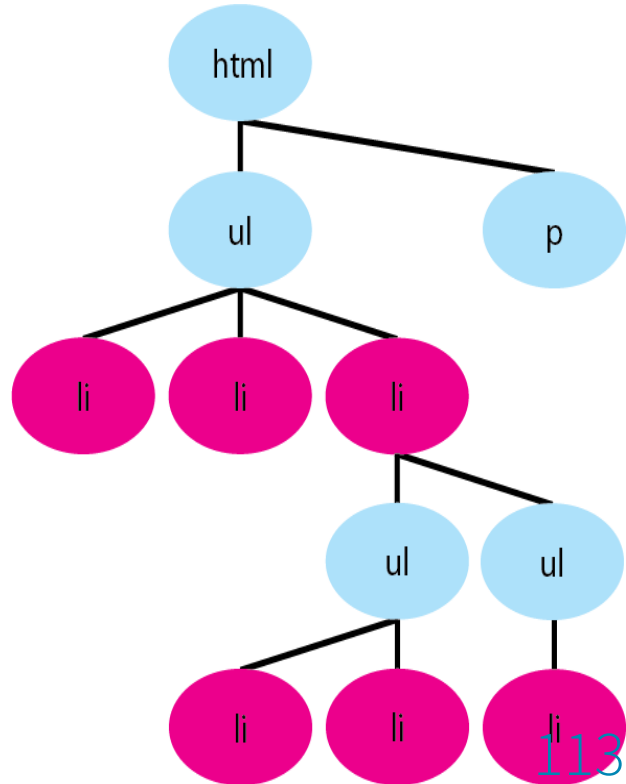


# ul li

```
<html>
  <ul>
    <li></li>

    <li></li>
    <li>
      <ul>
        <li></li>
        <li></li>
      </ul>
    <ul>
      <li></li>
    </ul>
  </li>
</ul>

  <p></p>
</html>
```

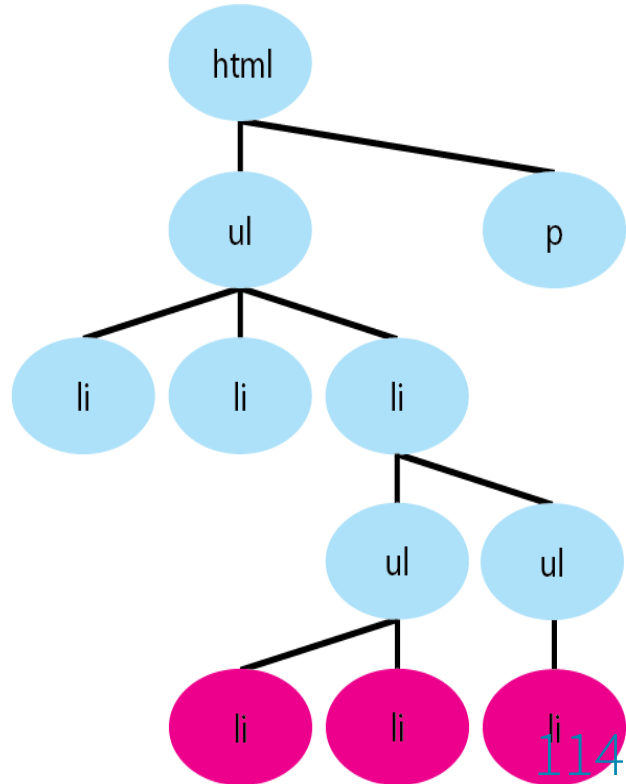


li li

```
<html>
  <ul>
    <li></li>

    <li></li>
    <li>
      <ul>
        <li></li>
        <li></li>
      </ul>
      <ul>
        <li></li>
      </ul>
    </li>
  </ul>

  <p></p>
</html>
```

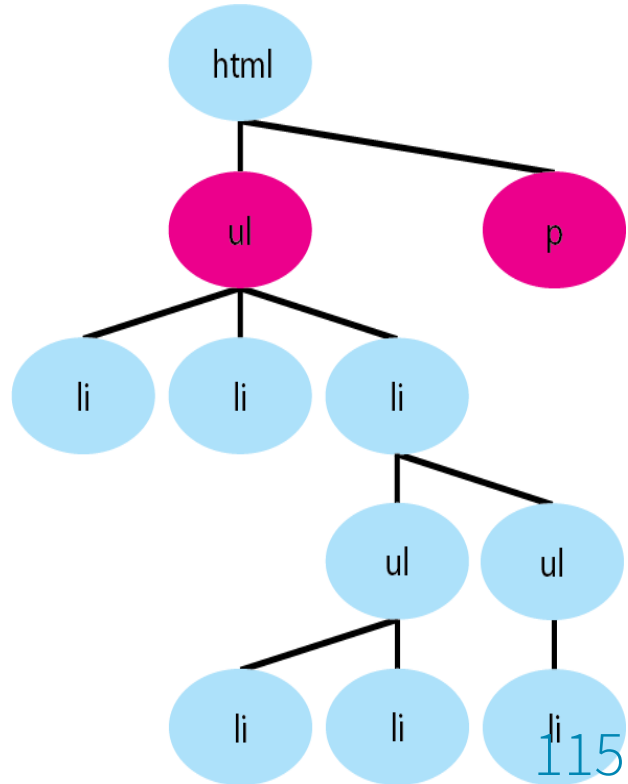


html > \*

```
<html>
  <ul>
    <li></li>

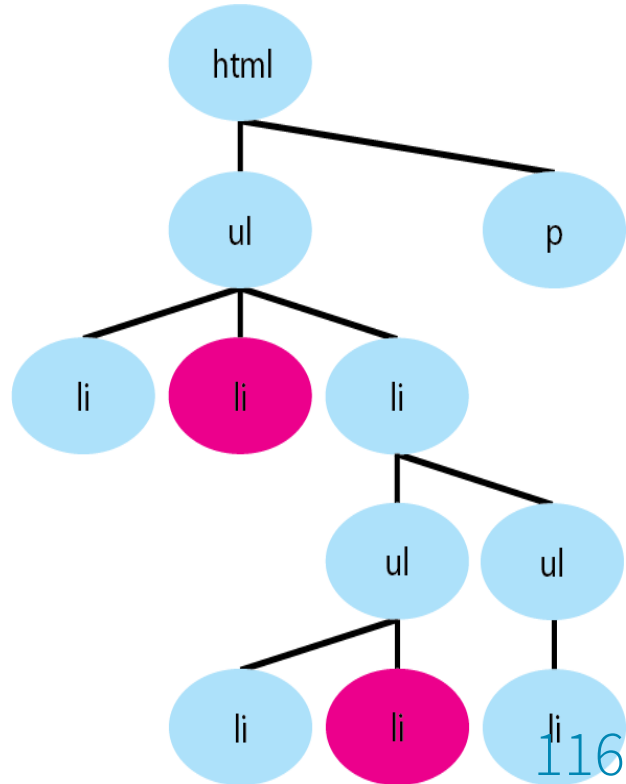
    <li></li>
    <li>
      <ul>
        <li></li>
        <li></li>
      </ul>
      <ul>
        <li></li>
      </ul>
    </li>
  </ul>

  <p></p>
</html>
```



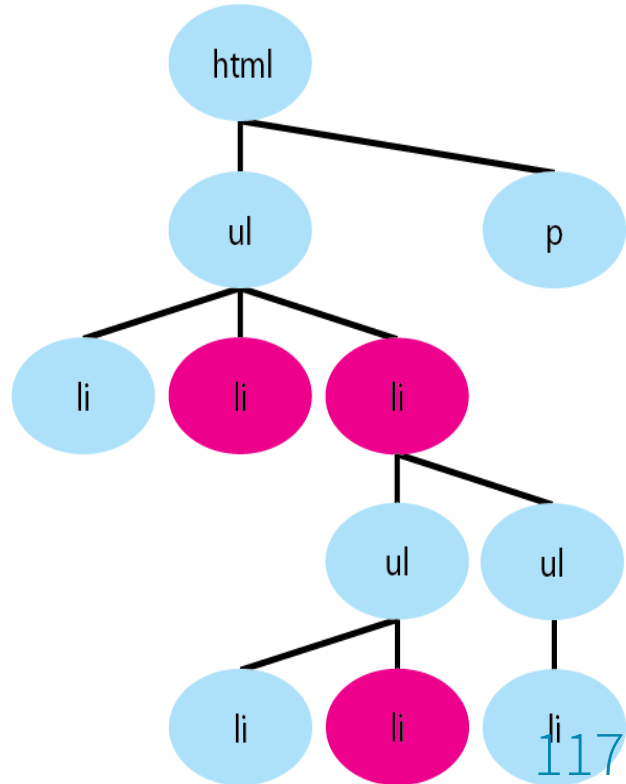
# li:first-child + li

```
<html>
  <ul>
    <li></li>
    <li></li>
    <li>
      <ul>
        <li></li>
        <li></li>
      </ul>
      <ul>
        <li></li>
      </ul>
    </li>
  </ul>
  <p></p>
</html>
```



# li:first-child ~ li

```
<html>
  <ul>
    <li></li>
    <li></li>
    <li>
      <ul>
        <li></li>
        <li></li>
      </ul>
      <ul>
        <li></li>
      </ul>
    </li>
  </ul>
  <p></p>
</html>
```



# Пример

```
<p class="red blue">  
    Красный или синий?  
</p>
```

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

Красный или синий?

# Пример

```
<p class="blue red">  
    Красный или синий?  
</p>
```

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

Красный или синий?

# Пример

```
<p class="red blue">  
    Красный или синий?  
</p>
```

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

Красный или синий?



# Пример

```
<p class="red blue">  
    Красный или синий?  
</p>
```

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

Красный или синий?

# Пример

```
<p id="red" class="blue">  
    Красный или синий?  
</p>
```

```
html .blue {  
    color: blue;  
}  
  
#red {  
    color: red;  
}
```

Красный или синий?

Специфичность

идентификаторы

классы  
псевдоклассы  
атрибуты

элементы (теги)  
псевдоэлементы

универсальный селектор и комбинаторы не  
влияют на специфичность

# Применяем высчитанные значения:

1. Каждому HTML-элементу находим соответствующие ему селекторы
2. Сортируем селекторы по специфичности
  - От менее приоритетных к более приоритетным
  - Если специфичность совпадает, то раньше тот, который встретился раньше
3. Применяем правила последовательно

# Специфичность

```
div { /* 0 0 1 */  
  color: blue;  
  font-weight: bold;  
  font-size: 12px;  
}
```

```
div#id { /* 1 0 1 */  
  font-size: 15px;  
}
```

```
html div { /* 0 0 2 */  
  color: red;  
}
```

# Специфичность

```
div { /* 0 0 1 */  
    color: blue;  
    font-weight: bold;  
    font-size: 12px;  
}
```

```
html div { /* 0 0 2 */  
    color: red;  
}
```

```
div#id { /* 1 0 1 */  
    font-size: 15px;  
}
```

# Специфичность

```
div {  
  color: blue; /* 0 0 1 */  
  font-weight: bold; /* 0 0 1 */  
  font-size: 12px; /* 0 0 1 */  
}
```

```
html div {  
  color: red; /* 0 0 2 */  
}
```

```
div#id {  
  font-size: 15px; /* 1 0 1 */  
}
```



# Специфичность

```
color: blue; /* 0 0 1 */  
font-weight: bold; /* 0 0 1 */  
font-size: 12px; /* 0 0 1 */  
color: red; /* 0 0 2 */  
font-size: 15px; /* 1 0 1 */
```

# Специфичность

```
color: blue; /* 0 0 1 */
font: 13px Arial; /* 0 0 1 */
font-size: 12px; /* 0 0 1 */
font-weight: bold; /* 0 0 1 */
color: red; /* 0 0 2 */
font-size: 15px; /* 1 0 1 */
```

# Атрибут style и !important

```
<div style="color: blue;"></div>
```

```
div {  
  color: green !important;  
}
```

```
<div style="color: white !important;"></div>
```

1. Атрибут style по умолчанию приоритетнее стилей в CSS
2. Стили в CSS с !important приоритетнее атрибута style
3. Атрибут style с !important приоритетнее всего (на самом деле нет)

# Атрибут style и !important

```
color: blue; /* 0 0 1 */
font-weight: bold; /* 0 0 1 */
font-size: 12px; /* 0 0 1 */
font-size: 15px; /* 0 0 1 */
color: red; /* 0 0 2 */
<div style="color: blue;"></div>
color: green !important;
<div style="color: white !important"></div>
```

# Атрибут style и !important

- Атрибут style стоит использовать, только если стили высчитываются динамически
- Вместо !important чаще всего можно указать бóльшую специфичность

# Тренировка

```
p.class /* 0 1 1 */
```

```
.class-1.class-2:first-child /* 0 3 0 */
```

```
#id#id /* 2 0 0 */
```

```
p.class-1.class-2:nth-child(2n)[href]::after /* 0 4 2 */
```

# Наследование

- Некоторые стили применяются не только к целевому элементу, но и к его потомкам
- Унаследованные стили не имеют специфичности, то есть их всегда перебивает любой селектор

# Наследование

```
<div>  
  Привет,  
  <abbr title="Уральский федеральный университет">  
    УрФУ  
  </abbr>  
</div>
```

```
div {  
  color: red; /* 0 0 1 */  
  /* abbr { color : red } - нет специфичности */  
}
```

Привет, УрФУ



# Наследование

```
<div>
  Привет,
  <abbr title="Уральский федеральный университет">
    УрФУ
  </abbr>
</div>
```

```
* {
  color: blue; /* 0 0 0 */
}
div {
  color: red; /* 0 0 1 */
  /* abbr { color : red } - нет специфичности */
}
```

Привет, УрФУ

# Наследование

Стили, которые наследуются ([Подробнее](#))

color	direction
cursor	empty-cells
font-family	list-style
font-size	line-height
font-weight	quotes
font-style	text-align
font-variant	text-indent
font	text-transform
letter-spacing	visibility
list-style-type	white-space
list-style-position	word-spacing
list-style-image	

# Порядок применения стилей

- Стили браузера
- Стили пользователя и/или плагинов браузера
- Стили страницы
- Стили пользователя и/или плагинов браузера с !important

Значения свойств

# Числовые значения

- Целые числа (1, 2, 3...)
- Дробные числа (.5, 1.5)
- Процентные значения (50%)

# Абсолютные единицы измерения

- Миллиметры (mm)
- Сантиметры (cm)
- Дюймы (in)
- Пункты – используется в типографиях, 1 дюйм = 72 пункта (pt)
- Пики – 12 пунктов (pc)

Абсолютные единицы удобны при определении CSS для печати. В вебе же практически не используются

# Относительные единицы измерения

- пиксели (px)
- em – зависит от размера шрифта (em)
- rem – зависит от размера шрифта корневого элемента (rem)
- ex – зависит от высоты символа x в данном шрифте
- ch – зависит от ширины символа 0 в данном шрифте
- vh/vw – 1/100 высоты и ширины viewport'a соответственно
- vmax – 1/100 от максимума между высотой и шириной viewport'a
- vmin – 1/100 от минимума между высотой и шириной viewport'a

# Способы задания цвета

- Именованные цвета – red
- Функциональный – rgb(255, 0, 0) или rgba(255, 0, 0, 1)
- Шестнадцатиричный – #ff0000 или #f00
- Шестнадцатиричный с прозрачностью - #ff000000



# Ещё типы значений

- Именованные слова: `bold`, `underline`
- Комбинации: `1px solid red`
- Функциональные: `url()`, `attr()`, `rgb()`, `calc()`

# Полезные ссылки по CSS

- [MDN](#)
- [HTML5BOOK](#)
- [Игра про селекторы](#)
- [Специфичность через Звездные войны](#)

Вопросы?

# Все ссылки

- [HTML Entity](#)
- [!DOCTYPE](#)
- [Списки](#)
- [Таблицы](#)
- [Текстовые элементы](#)
- [Подробнее про URL](#)
- [Скрин-ридеры](#)
- [Подробнее про alt](#)
- [iframe](#)
- [Open Graph](#)
- [Семантические теги](#)
- [audio/video](#)
- [track](#)
- [Подробнее про атрибут type](#)
- [Подробнее про формы](#)
- [HTML I](#)
- [HTML II](#)
- [div или section или article?](#)

# Все ссылки

- [W3C](#)
- [HTML5BOOK](#)
- [Псевдоклассы](#)
- [Наследуемые стили](#)
- [MDN](#)
- [Челлендж на знание HTML-тегов](#)
- [At-правила](#)
- [HTML5BOOK](#)
- [Игра про селекторы](#)
- [Специфичность через Звездные войны](#)

Спасибо!