

HTML и CSS

Почему, зачем и что?

Козлов Антон



ТИНЬКОФФ



HTML



ТИНЬКОФФ

HTML



ТИНЬКОФФ

HTML. Что это?



HTML (от англ. HyperText Markup Language — «язык гипертекстовой разметки») — стандартизированный язык разметки документов во Всемирной паутине.

Структура HTML-документа

Любой HTML документ состоит из набора элементов, комментариев и спецсимволов.

Простейшая страница выглядит так:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Заголовок документа</title>
</head>
<body>
  <!-- Содержимое -->
</body>
</html>
```

Структура HTML-документа

Страничка посложнее:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>Заголовок документа</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Тинькофф</h1>
  <p>Он такой один!</p>
<script src="scripts.js"></script>
</body>
</html>
```

HTML-теги

Теги — строительные кирпичи HTML. Они определяют семантическое обозначение их содержимого.

`<имя_тега>` — тег

Примеры тегов:

`<html>` `<body>` `<section>` `<article>` `<div>`

`<h1>` `<nav>` `` `` `` `<p>` `<a>`

`` `` `<time>` `<mark>`

`` `<video>` `<audio>`

`<form>` `<input>` `<button>` `<label>`

HTML-теги

Состав тега



HTML-теги

Парные теги

`<h1>Тинькофф</h1>`

`<p>`

Он такой один!

`</p>`

Открывающий тег

Закрывающий тег

HTML-теги

Одиночные теги

`<hr>` `<!-- разделитель -->`

`
` `<!-- перенос строки -->`

`<input type="text">` `<!-- поле ввода -->`

`<link rel="stylesheet" href="style.css">` `<!-- ссылка на ресурсы -->`

`` `<!-- изображение -->`

HTML-теги

Теги содержания

<h1> <h2> <h3> <h4> <h5> <h6>

<p>

<div>

 <i> <cite> <ins>

<a>

 <video> <audio>

<form> <input> <textarea> <select>

<button>

<table> <tr> <td> ...

HTML-теги

Семантические теги

`<header>`
`<footer>`
`<nav>`
`<article>`
`<main>`
`<aside>`
`<section>`

HTML-теги

Какие теги самые используемые?

HTML-теги

`<div>`
``
`<p>`
`<h1>` ... `<h6>`

HTML: имеет множество тегов, чтобы верстать сайт

Все:



HTML-теги

Существует более сотни тегов, но обычно мы используем не более 10-20

Справочники:

webref.ru/html

htmlbook.ru/

w3schools.com/tags/

HTML-теги

Вложенные теги

Теги со всем их содержимым должны **полностью** вкладываться в другие теги, не оставляя "хвостов" снаружи.

Правильно

`<p>Какой-то текст с выделением</p>`

Неправильно

`<p>жирный</p> текст`

Не по правилам тега

```
<ol>
  <h2>название списка</h2>
  <li>элемент списка</li>
  <li>элемент списка</li>
</ol>
```


HTML-теги

Атрибуты тегов

Это дополнительные значения, которые настраивают элементы или регулируют их поведение различным способом, чтобы соответствовать критериям пользователей.

Атрибутов может быть несколько. Некоторые теги не могут быть без атрибутов.

```
  
<input type="text" name="email">  
<p class="important">...</p>  
<input type="submit" class="button">  
<div class="logo">...</div>  
<span class="date">...</span>
```

HTML-теги

Атрибуты тегов

Самый часто используемый атрибут – класс (class)

```
<p class="important">...</p>  
<div class="logo">...</div>
```

Классов может быть несколько. В таком случае они записываются **через пробел** и закрываются кавычками.

```
<span class="date birthday">...</span>
```

HTML-теги

Атрибутов много и постоянно вводятся новые

Справочники:

developer.mozilla.org/ru/docs/Web/HTML/Attributes

Структура HTML-документа

Комментарии

<!-- Это предложение будет игнорироваться браузером -->

<h1>Тинькофф</h1>

Структура HTML-документа

Комментарии

<!-- Это предложение будет игнорироваться браузером -->

<h1>Тинькофф</h1>

<!-- И это предложение
тоже будет игнорироваться браузером -->

<h1>Тинькофф</h1>

Структура HTML-документа

Спецсимволы

`>`; обозначает знак больше чем (>)

`<`; обозначает знак менее чем (<)

`&`; обозначает знак амперсанда (&)

` `; обозначает неразрывный пробел ()

Справочник

<http://html.spec.whatwg.org/multipage/named-characters.html#named-character-references>

Структура HTML-документа

Спецсимволы

> обозначает знак больше чем (>)

< обозначает знак менее чем (<)

& обозначает знак амперсанда (&)

** ** обозначает неразрывный пробел ()

5 000
руб.

Структура HTML-документа

Спецсимволы

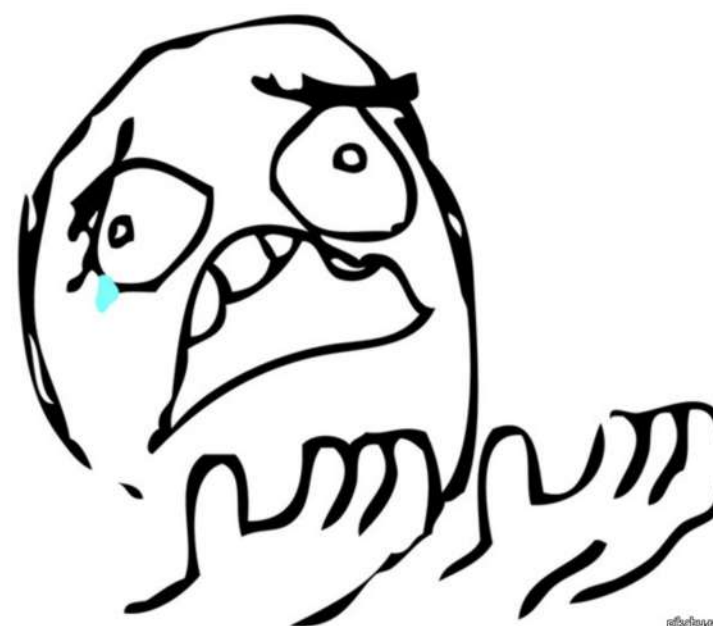
`>` обозначает знак больше чем (>)

`<` обозначает знак менее чем (<)

`&` обозначает знак амперсанда (&)

` ` обозначает неразрывный пробел ()

5 000
руб.



Структура HTML-документа

Спецсимволы

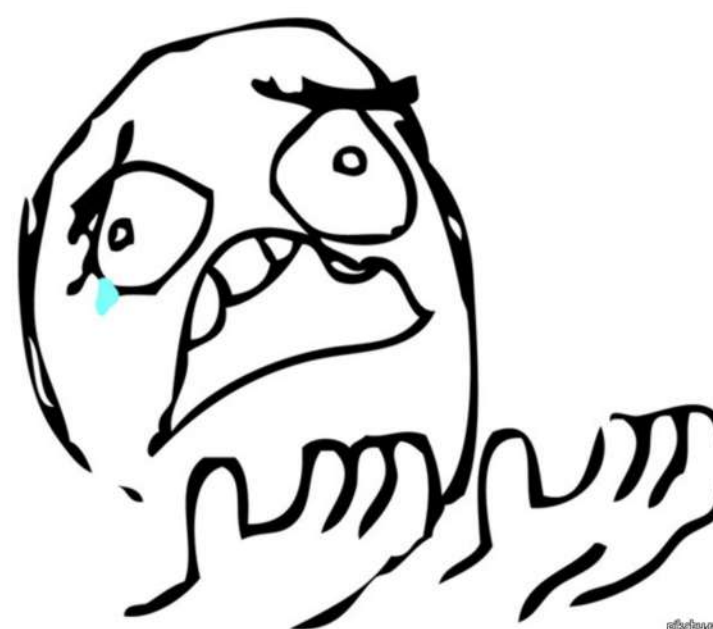
`>` обозначает знак больше чем (>)

`<` обозначает знак менее чем (<)

`&` обозначает знак амперсанда (&)

` ` обозначает неразрывный пробел ()

5 000
руб.



5 000 руб.

Элементы группировки

Элемент `<div>`

Служит для структуризации контента на веб-странице, заключения содержимого в отдельные блоки.

```
<body>
  <div>
    Блок 1
    <br>
    Содержимое блока 1
  </div>
  <div>
    Блок 2
    <br>
    Содержимое блока 2
  </div>
</body>
```

Блок 1
Содержимое блока 1

Блок 2
Содержимое блока 2



codepen.io/ngm/pen/MoLOvj

Элементы группировки

Параграфы `<p>`

Определяет текстовый абзац

```
<body>  
  <p>Параграф с текстом.</p>  
  <p>Ещё параграф с текстом.</p>  
</body>
```

Параграф с текстом.

Ещё параграф с текстом.



codepen.io/ngm/pen/MoLOvj

Заголовки

Элементы `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` и `<h6>` служат для создания заголовков различного уровня:

```
<body>
  <h1>Заголовок первого уровня</h1>
  <h2>Заголовок второго уровня</h2>
  <h3>Заголовок третьего уровня</h3>
  <h4>Заголовок четвертого уровня</h4>
  <h5>Заголовок пятого уровня</h5>
  <h6>Заголовок шестого уровня</h6>
</body>
```

Заголовок первого уровня

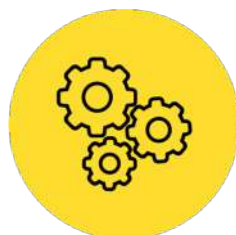
Заголовок второго уровня

Заголовок третьего уровня

Заголовок четвертого уровня

Заголовок пятого уровня

Заголовок шестого уровня



codepen.io/ngm/pen/xreOrg



ТИНЬКОФФ

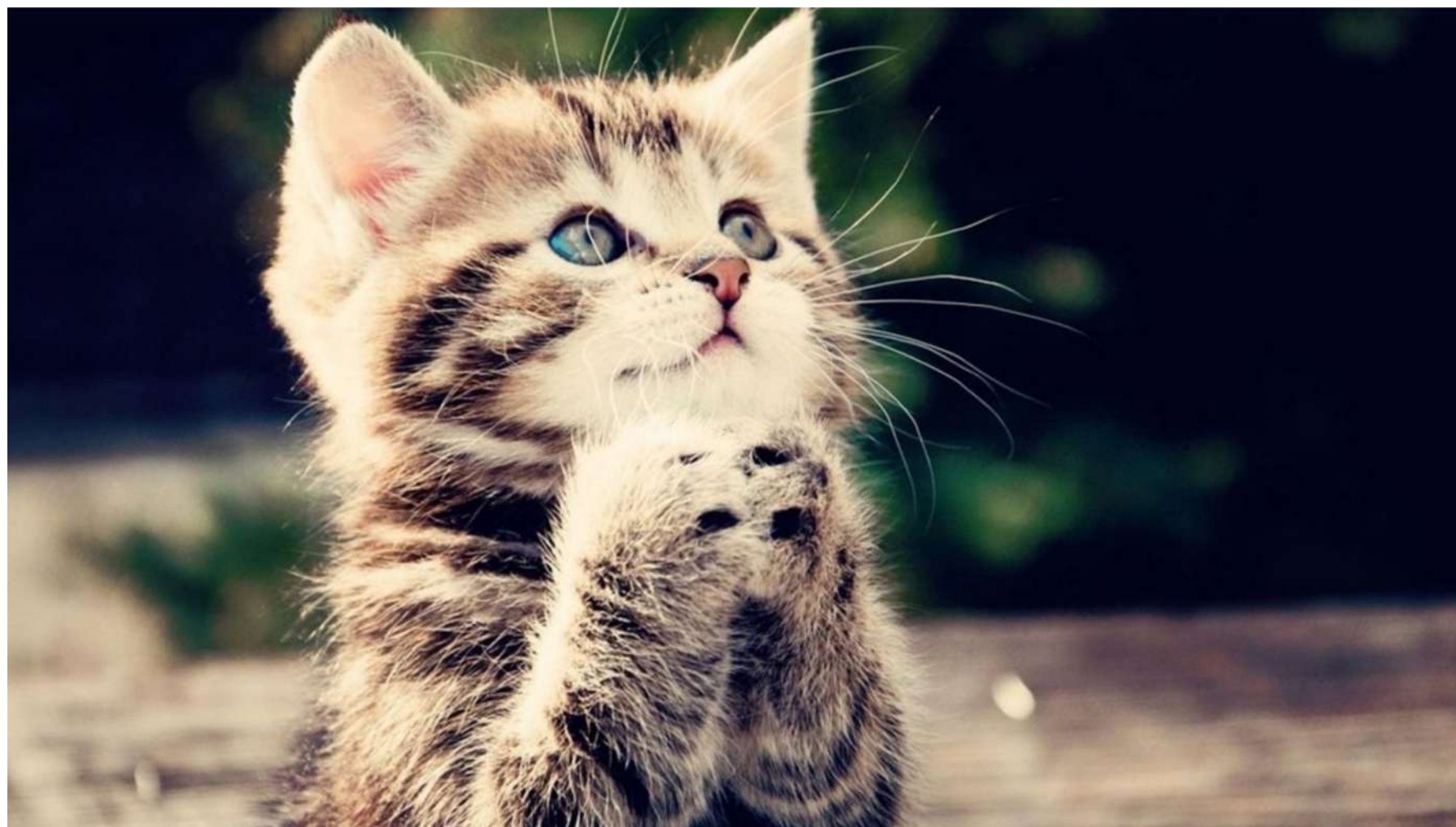
Форматирование текста

- **** : выделяет текст жирным
- **** : зачеркивает текст
- **<i>** : выделяет текст курсивом
- **** : выделяет текст курсивом, в отличие от тега **<i>** носит логическое значение, придает выделенному тексту оттенок важности
- **<s>** : зачеркивает текст
- **<small>** : делает текст чуть меньше размером, чем окружающий
- **** : выделяет текст жирным. В отличие от тега **** предназначен для логического выделения, чтобы показать важность текста.
- **<sub>** : помещает текст под строкой
- **<sup>** : помещает текст над строкой
- **<u>** : подчеркивает текст
- **<ins>** : определяет вставленный (или добавленный) текст
- **<mark>** : выделяет текст цветом, придавая ему оттенок важности

Работа с изображениями

Для работы с изображениями в HTML используется тег ``

```
<body>  
    
</body>
```



Таблицы

Для создания таблиц в html используется элемент `<table>`. Каждая таблица содержит строки, который представлены элементом `<tr>`. А каждая строка ячейки в виде элементов `<td>`.

```
<table>
  <tr>
    <td>Модель</td> <td>Компания</td> <td>Цена</td>
  </tr>
  <tr>
    <td>iPhone XS Max</td> <td>Apple</td>
    <td>127890</td>
  </tr>
  <tr>
    <td>Galaxy S9</td> <td>Samsung</td> <td>59990</td>
  </tr>
  <tr>
    <td>Huawei P20 Pro</td> <td>Huawei</td> <td>54990</td>
  </tr>
</table>
```

Модель	Компания	Цена
iPhone XS Max	Apple	127890
Galaxy S9	Samsung	59990
Huawei P20 Pro	Huawei	54990



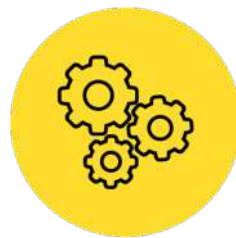
codepen.io/extremum/pen/MqdoEX

Ссылки

Для создания ссылок используется элемент `<a>`.

```
<a href="https://www.tinkoff.ru/" target="_blank">Решение вопросов</a>
```

Решение вопросов



codepen.io/ngm/pen/EXJgvE



ТИНЬКОФФ

Работа с формами

Все поля формы помещаются в парный тег `<form>`.

```
<form method="post" autocomplete="off" action="//mysite.com/login.php">  
  <input type="text" name="login">  
  <input type="password" name="password" autocomplete="on">  
  <input type="submit" value="Войти">  
input  
</form>
```



codepen.io/ngm/pen/xreEMQ



ТИНЬКОФФ

Элементы форм

Основной элемент форм - `input` и значение его атрибута `type` (`<input type="text">`)

- `text` : обычное текстовое поле
- `password` : тоже текстовое поле, только вместо вводимых символов отображаются звездочки
- `radio` : радиокнопка или переключатель.
- `checkbox` : элемент флажок, который может находиться в отмеченном или неотмеченном состоянии
- `submit` : кнопка отправки формы
- `color` : поле для ввода цвета
- `color` : поле для ввода даты
- `email` : поле для ввода адреса электронной почты
- `month` : поле для ввода года и месяца
- `number` : поле для ввода чисел
- `time` : поле для ввода времени
- `file` : поле для выбора отправляемого файла
- `tel` : поле для ввода телефона

Элементы форм

Кроме `input`:

- `button` : создает кнопку
- `Label` : выпадающий список
- `label` : создает метку, которая отображается рядом с полем ввода
- `textarea` : многострочное текстовое поле
- `fieldset` : для группировки элементов



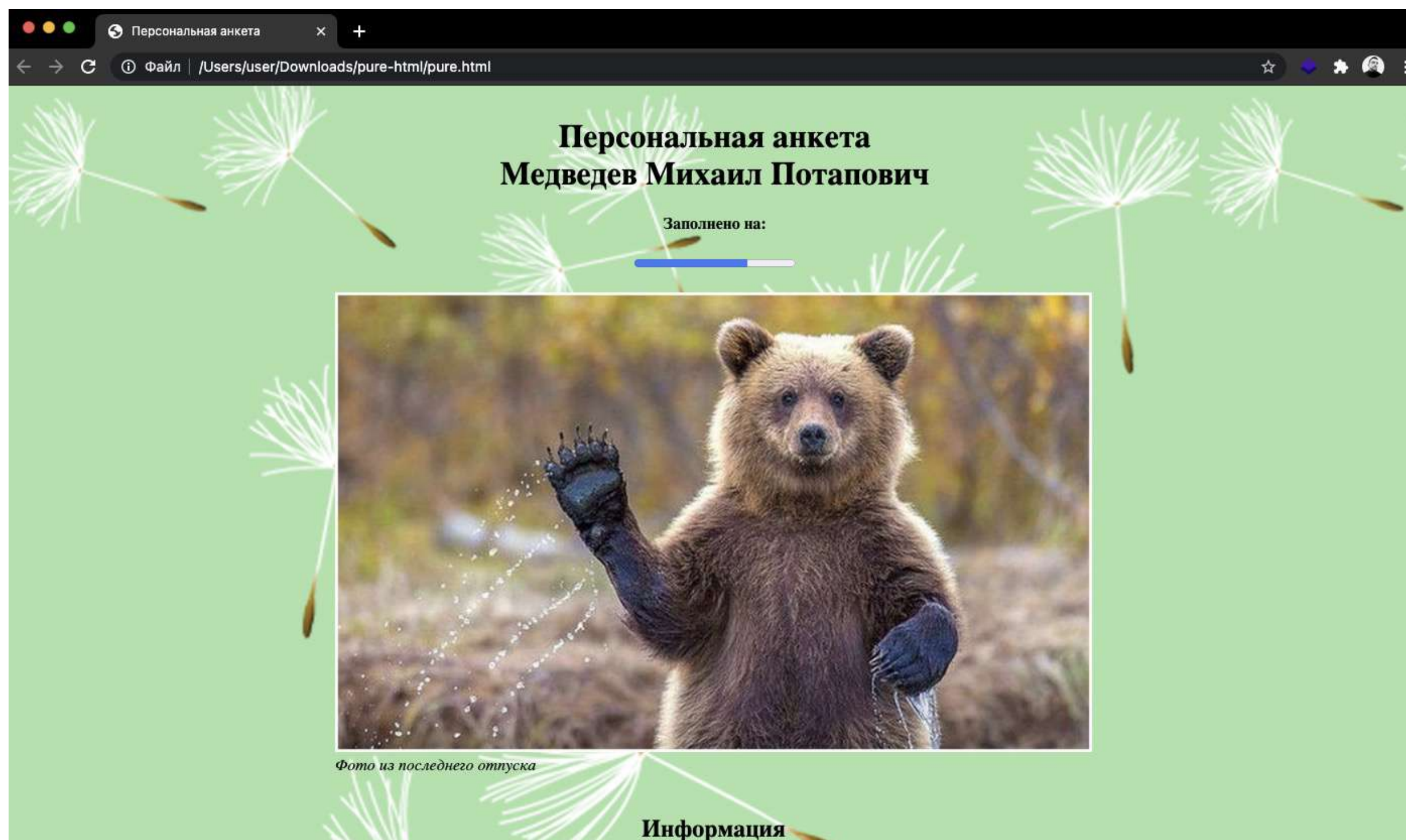
codepen.io/ngm/pen/xreEMQ

Атрибуты name и value

У всех элементов ввода можно установить атрибуты `name` и `value`.

По атрибуту `name` мы можем идентифицировать поле ввода, а атрибут `value` позволяет установить значение поля ввода.

Чистый HTML



[Скачать архив](#)



ТИНЬКОФФ

CSS



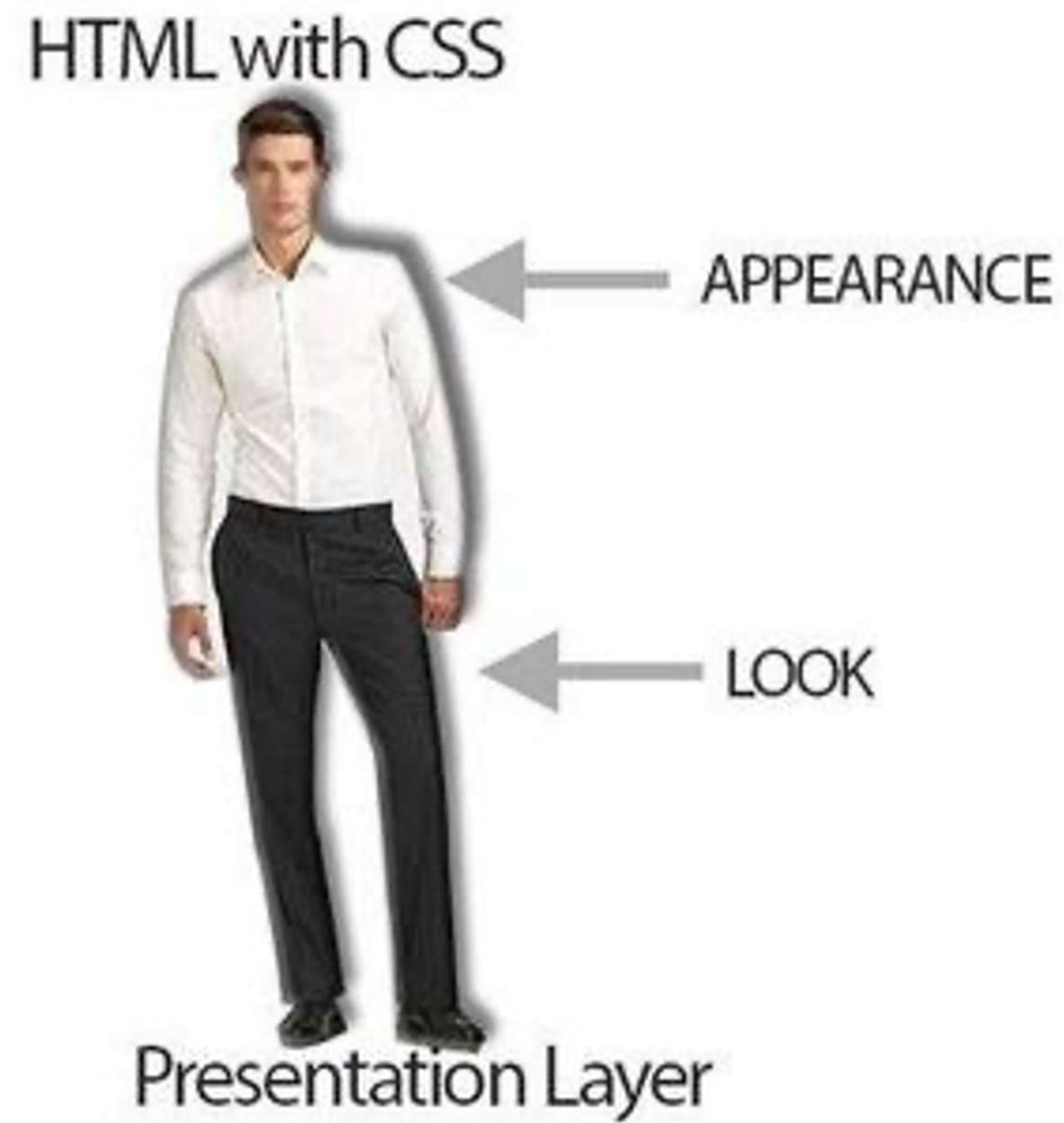
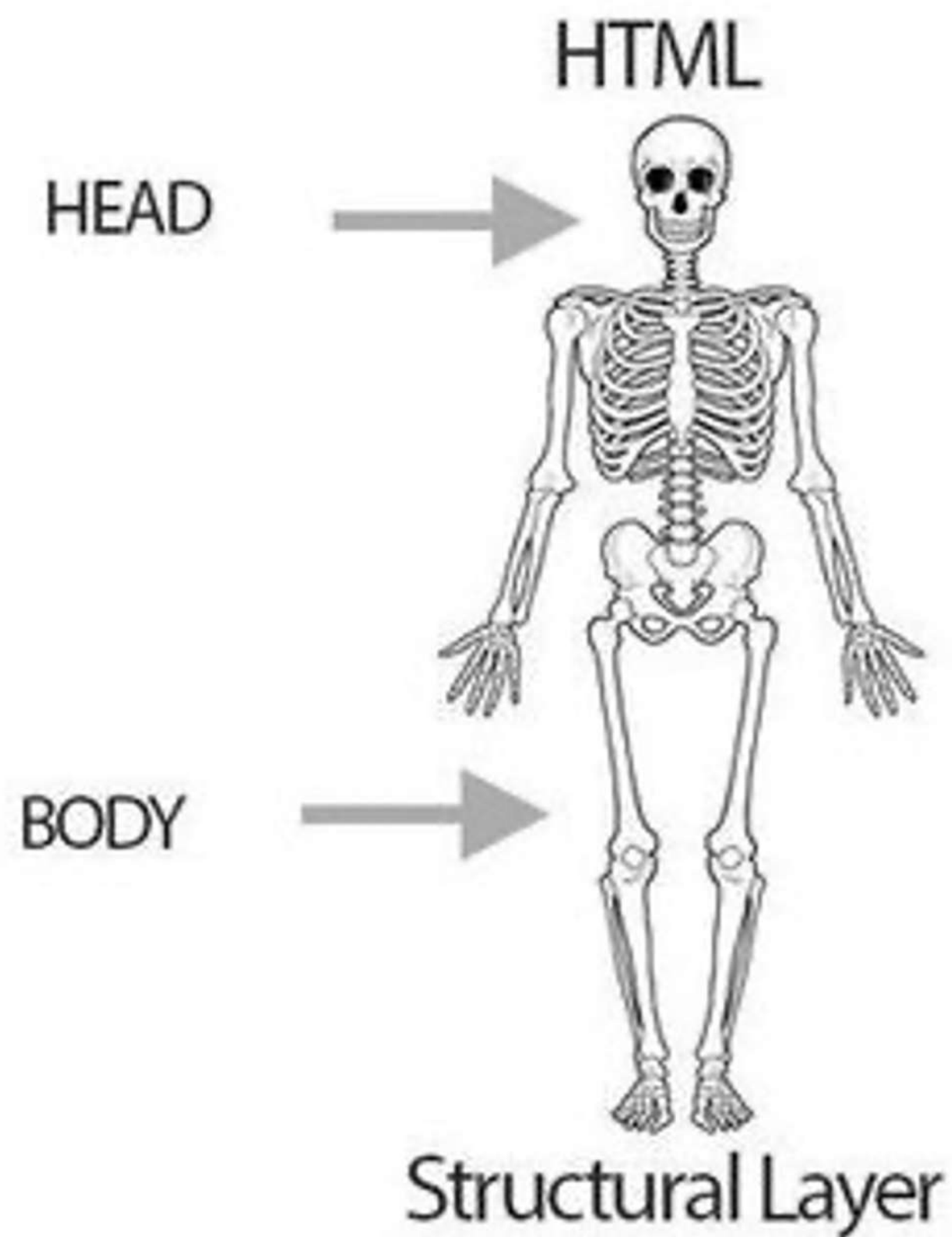
Каскадные таблицы стилей (Cascading Style Sheets = CSS) — это язык, который отвечает за визуальное представление документов пользователю

Для чего придумали CSS?

Основной целью разработки CSS являлось **разделение** описания логической **структуры** веб-страницы от описания **внешнего вида** этой веб-страницы.

Такое разделение может **увеличить доступность** документа, **предоставить большую гибкость** и **возможность управления** его представлением, а также **уменьшить сложность** и **повторяемость** в структурном содержимом.

Для чего придумали CSS?



Для чего придумали CSS?

Как выглядит HTML документ без CSS

Для чего придумали CSS?



Поиск

[Добавить ярлык](#)
[Ещё](#)

- [Blogger](#)
- [Документы](#)
- [Контакты](#)
- [Hangouts](#)
- [Google Keep](#)

Другие сервисы Google

[Изменить](#)
Евгений Мамаев
zhenya.mamaev@gmail.com
[Профиль Google](#)—[Конфиденциальность](#)
[Мой аккаунт](#)

[Евгений Мамаев](#)
[zhenya.mamaev@gmail.com](#) (по умолчанию)
[Все аккаунты брендов »](#)
[Добавить аккаунт](#)
[Выйти](#)

Распознавание команд не работает
[Перезапустить](#)[Справка](#)
Распознавание голосовых команд отключено
Распознавать "О'кей, Google"

Живой поиск Google недоступен. Для поиска нажмите "Ввод". [Подробнее...](#)

Живой поиск Google отключен из-за низкой скорости интернет-подключения. Чтобы выполнить поиск, нажмите "Ввод".
Для поиска нажмите "Ввод"

Поиск в Google

Мне повезёт!

[Пожаловаться на неприемлемые подсказки](#)

×

Россия
[Конфиденциальность](#) [Условия](#) [Настройки](#) [Настройки поиска](#) [Расширенный поиск](#) [История](#) [Справка по поиску](#) [Отправить отзыв](#) [Реклама](#) [Для бизнеса](#) [Все о Google](#)
Чтобы запустить голосовой поиск, скажите "О'кей, Google".

Голосовой поиск – это очень удобно! Чтобы найти нужную информацию, скажите "О'кей, Google" и произнесите запрос.

[Подробнее](#) [Нет](#) [Включить "О'кей, Google"](#)

[Карты](#)[Маркет](#)[Новости](#)[Переводчик](#)[Картинки](#)[Видео](#)[Музыка](#)
[ещё](#)
[Авиабилеты](#)
[Автобусы](#)
[Авто.ру](#)
[Афиша](#)
[Блоги](#)
[Деньги](#)
[Диск](#)
[Каталог](#)
[Музыка](#)
[Недвижимость](#)
[Погода](#)
[Поиск людей](#)
[Почта](#)
[Пробки](#)
[Путешествия](#)
[Работа](#)
[Радио](#)
[Расписания](#)
[Такси](#)
[Телепрограмма](#)
[Фотки](#)
[Для мобильного](#)
[Программы](#)
[Все сервисы](#)

Найдётся всё. Например, дата открытия америки

Найти

Погода+18 °C

- [Ночь +13, утро +14](#)

Пробки6 баллов

[Защита Покупателя](#)
[Помощь](#)

- [Гид покупателям](#)
- [Споры и жалобы](#)

Экономьте больше в приложении!
Доставка в / RUB
Выберите региональные настройки
Доставка в
Валюта

Сохранить

[Go to Global Site \(English\)](#)
CATEGORIES
[AliExpress](#)[Покупай умнее. живи веселее!](#)
[Корзина](#)0
[Мои желания](#)
[Войти](#)[Регистрация](#)
Мой AliExpress(0)
[Выйти](#)

Добро пожаловать на AliExpress.com

[Войти](#)

Войти через ...

Новый покупатель?
[Зарегистрироваться](#)

- [Мой AliExpress](#)
- [Мои заказы](#)
- [Центр сообщений](#)
- [Мои желания](#)
- [Любимые магазины](#)
- [Мои купоны](#)
- [Мой Alipay](#)

Все категории

Я ищу...

[Горячие товары](#)[Молт](#)[Молния](#)[Тао](#)[Бренд-фокус](#)[Hi-Tech](#)[Мода](#)[Бестселлеры](#)
Категории [Смотреть все >](#)

Первый раз на AliExpress?

Зарегистрируйся сегодня и получи купоны на сумму до \$100

купонами \$100
[Получить купон сейчас](#)

ТИНЬКОФФ

41

Немного истории



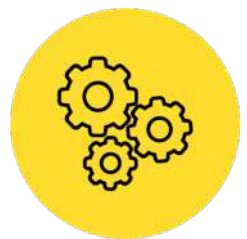
Термин «каскадные таблицы стилей» был предложен **Хокон Виум Ли** в 1994 году. Совместно с Бертом Босом он стал развивать CSS.

Версии:

- **Уровень 1 (CSS1) 1996 г**
Параметры шрифтов: размер, стиль. Цвета текста, фона, рамок и других элементов. Выравнивание для текста, изображений, таблиц и других элементов. Высота, ширина, внутренние (padding) и внешние (margin) отступы и рамки.
- **Уровень 2 (CSS2) 1998 г**
Блочная вёрстка. Относительное, абсолютное и фиксированное позиционирование. Расширенный механизм селекторов.
- **Уровень 2, ревизия 1 (CSS2.1) 2011 г**
- **Модули уровня 3 (CSS3) с 2011 г.** Возможность создавать анимированные элементы, поддержка линейных и радиальных градиентов, теней, сглаживания и многое другое.
- **Модули уровня 4 (CSS4) с 2011 г**

Как работает CSS?

1. Браузер парсит **HTML** и строит **DOM**
2. Браузер парсит **CSS** и строит **CSSOM**



<https://developers.google.com/web/fundamentals/performance/critical-rendering-path/constructing-the-object-model?hl=ru>

Как работает CSS?

HTML

```
<p>  
Используем  
<span>каскадные</span>  
<span>таблицы</span>  
<span>стилей</span>  
</p>
```

DOM

```
P  
├── "Используем"  
├── SPAN  
│   └── "каскадные"  
├── SPAN  
│   └── "таблицы"  
└── SPAN  
    └── "стилей"
```

CSS

```
span {  
  border: 1px solid black;  
  background-color: lime;  
}
```

Используем **каскадные** **таблицы** **стилей**

CSS

Что может CSS?

Что может CSS

Задавать внешний вид, размеры
и расположение элементов

Что может CSS

- Позиционировать элементы
 - Относительно самих себя
 - Относительно друг друга
 - Относительно окна браузера

Что может CSS

- Задавать размеры элементов (ширина, высота)
 - Минимально возможные
 - Максимально возможные
 - Менять алгоритм расчета высоты и ширины

Что может CSS

- Задавать цвет и фон для веб-страницы или отдельных элементов

Что может CSS

- Задавать отступы внутри или вокруг элементов

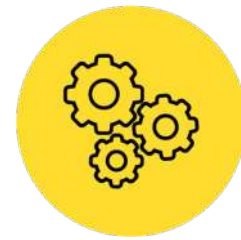
Что может CSS

Задавать отображение текста внутри страницы

- Размер
- Начертание
- Жирность
- Отступ между символами
- Выравнивание
- Обводка
- Тень
- и т. д.

Что может CSS

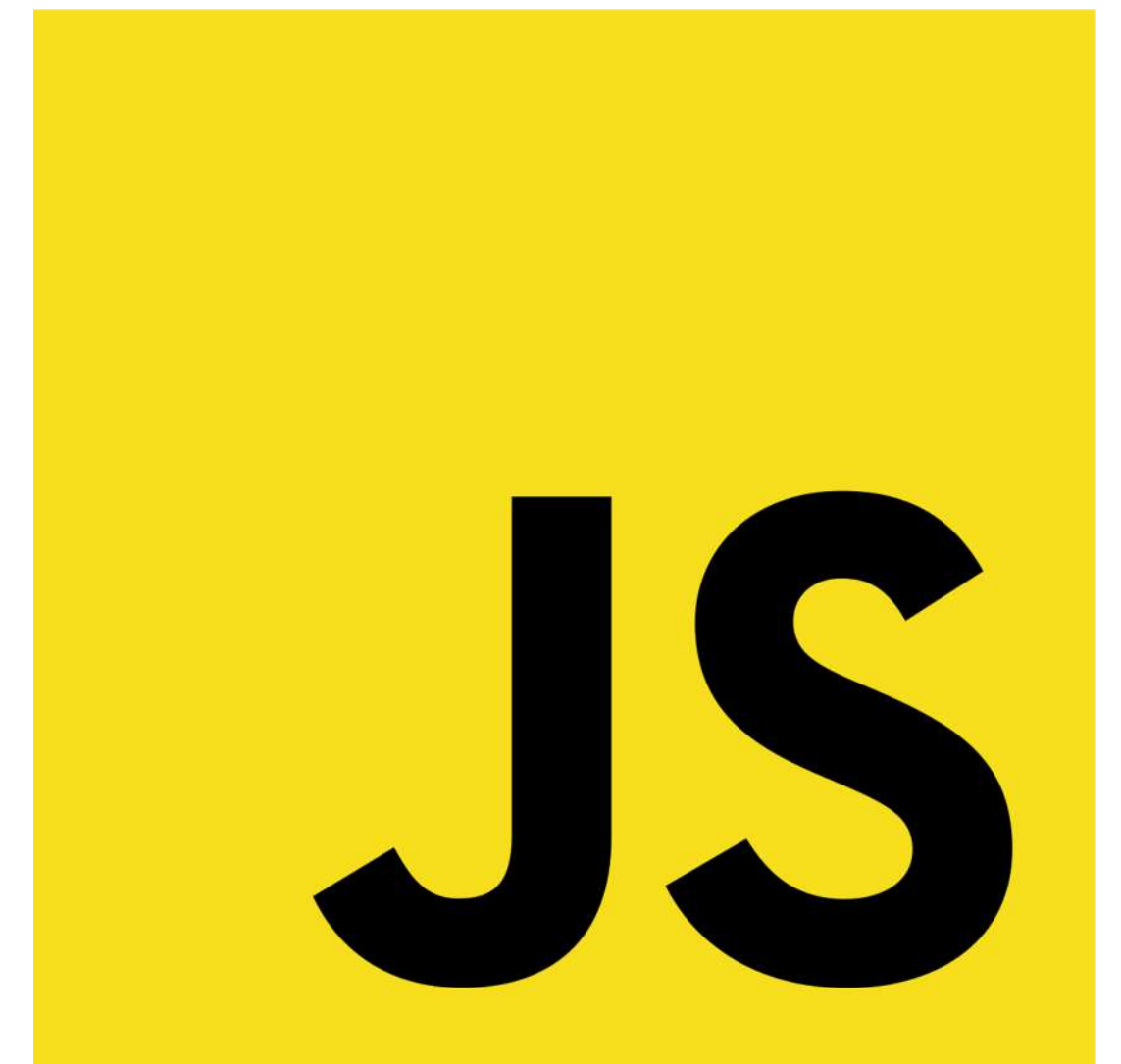
Посмотрим наглядно



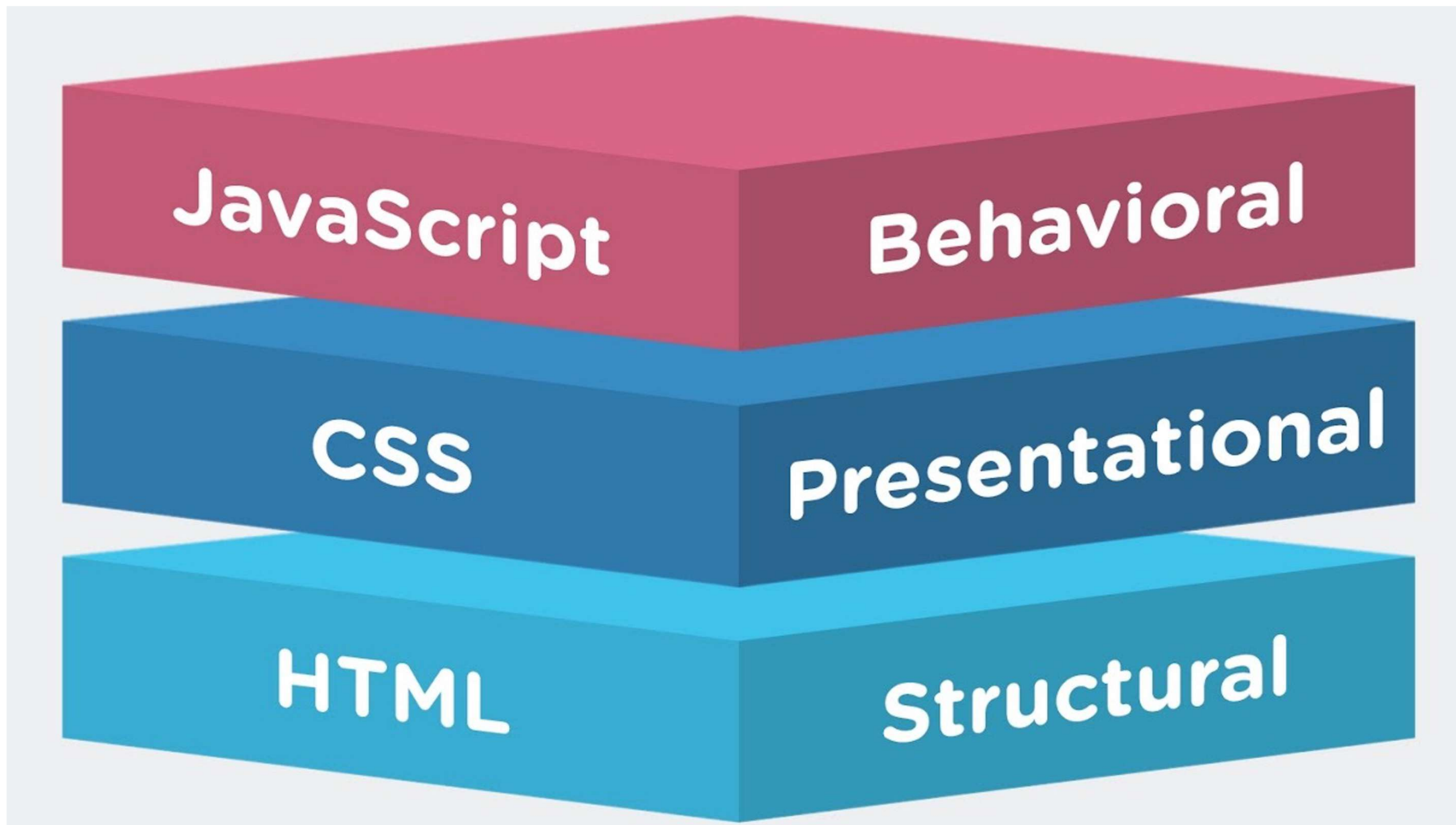
codepen.io/ngm/pen/bRQoOb

Что не может CSS

- **Создавать новые HTML-теги, удалять существующие, динамически менять стили элементов в зависимости от контекста**
- **Реагировать на действия посетителя, обрабатывать клики мыши, перемещения курсора, нажатия на клавиатуру и т.п.**



CSS



CSS

Как подключить стили к странице?

Есть 3 способа применить CSS
к HTML-документу

Как подключить стили к странице?

Внешняя таблица стилей

Это отдельный файл с расширением **.css**, в котором и хранится CSS. Вы ссылаетесь на него из HTML при помощи элемента **<link>**, который вы располагаете в секции **<head>**.

```
<html>
<head>
...
    <link rel="stylesheet" href="style.css">
</head>
...
</html>
```

Как подключить стили к странице?

Внутренняя таблица стилей

Содержится внутри элемента **<style>** и не требует подключения внешних файлов.

```
<html>
<head>
...
  <style>
    body {
      background-color: blue;
    }
  </style>
</head>
...
</html>
```

Как подключить стили к странице?

Внутренняя таблица стилей

Содержится внутри элемента **<style>** и не требует подключения внешних файлов.

```
<html>
<head>
...
  <style>
    body {
      background-color: blue;
    }
  </style>
</head>
...
</html>
```



Как подключить стили к странице?

Встроенные стили (inline css)

Это объявление CSS, которое применимо только к одному элементу и содержащееся в атрибуте **style**

```
<html>
...
p style="font-size: 20px; color: green; font-family: arial, helvetica, sans-serif">
...
</html>
```

Как подключить стили к странице?

Встроенные стили (inline css)

Это объявление CSS, которое применимо только к одному элементу и содержащееся в атрибуте **style**

```
<html>  
...  
p style="font-size: 20px; color: green; font-family: arial, helvetica, sans-serif">  
...  
</html>
```



CSS

Синтаксис CSS

Синтаксис CSS

Каждое правило CSS из таблицы стилей имеет две основные части — *селектор* и *блок объявлений*.

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

Синтаксис CSS

Комментарии в CSS

```
/* Это комментарий CSS */  
div {  
    background: lightgreen;  
    color: black;  
}  
/*  
    Комментарии предназначены только для чтения людьми и компьютер  
    их разбирать не будет  
*/
```


Синтаксис CSS

В CSS существует более **300** различных свойств и чуть ли не бесконечное количество значений.

Для каждого свойства определен некоторый список допустимых значений.

webref.ru/css

developer.mozilla.org/en-US/docs/Web/CSS/Reference

Синтаксис CSS

В CSS существует более **300** различных свойств и чуть ли не бесконечное количество значений.

Для каждого свойства определен некоторый список допустимых значений.

Обычно мы используем 50-100 свойств

webref.ru/css

developer.mozilla.org/en-US/docs/Web/CSS/Reference

Типы селекторов

Синтаксис CSS

Универсальный селектор

```
* {  
    margin: 0;  
    padding: 0;  
}
```

Синтаксис CSS

Универсальный селектор

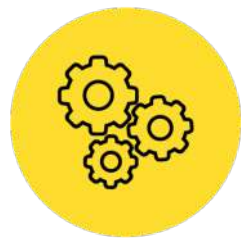
```
* {  
    margin: 0;  
    padding: 0;  
}
```

```
<div>  
    <span>  
        <a href="/"></a>  
    </span>  
    <h1></h1>  
    <ul>  
        <li></li>  
    </ul>  
</div>
```


Синтаксис CSS

Селектор тега

```
p {  
    font-family: arial, Helvetica, sans-serif;  
}
```



codepen.io/ngm/pen/qjGEeL



ТИНЬКОФФ

Синтаксис CSS

Селектор тега

```
p {  
    font-family: arial, Helvetica, sans-serif;  
}
```



codepen.io/ngm/pen/qjGEeL



ТИНЬКОФФ

```
<div>  
  <p>  
    <a href="/"></a>  
  <p></p>  
</p>  
<h1></h1>  
<ul>  
  <li>  
    <p></p>  
  </li>  
</ul>  
</div>
```

Синтаксис CSS

Селектор класса

```
.note {  
  font-weight: bold;  
  color: red;  
  background-color: yellow;  
}
```



codepen.io/ngm/pen/yXWNyP



ТИНЬКОФФ

```
<div>  
  <p class="note">  
    <a href="/"></a>  
    <p></p>  
  </p>  
  <h1></h1>  
  <ul>  
    <li>  
      <p></p>  
    </li>  
  </ul>  
</div>
```

Синтаксис CSS

Селектор по идентификатору (id)

```
#paragraph1 {  
    margin: 0;  
}
```



codepen.io/ngm/pen/mwYJVJ



ТИНЬКОФФ

Синтаксис CSS

Селектор по идентификатору (id)

```
#paragraph1 {  
    margin: 0;  
}
```

```
<div>  
  <p>  
    <a href="/"></a>  
  <p></p>  
</p>  
<h1 id="paragraph1"></h1>  
<ul>  
  <li>  
    <p></p>  
  </li>  
</ul>  
</div>
```

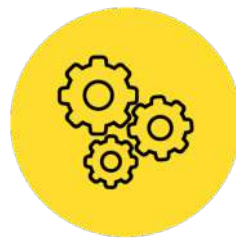


codepen.io/ngm/pen/mwYJVJ

Синтаксис CSS

Селектор атрибутов

```
a[href="https://www.somesite.com"] {  
    font-weight: bold;  
}
```



codepen.io/ngm/pen/RgmPoP



ТИНЬКОФФ

Синтаксис CSS

Селектор атрибутов

```
a[href="https://www.somesite.com"] {  
    font-weight: bold;  
}
```

```
<div>  
  <p>  
    <a href="http://www.somesite.com"></a>  
  <p></p>  
</p>  
<h1></h1>  
<ul>  
  <li>  
    <p></p>  
  </li>  
</ul>  
</div>
```



codepen.io/ngm/pen/RgmPoP



ТИНЬКОФФ

Синтаксис CSS

Контекстные селекторы (селекторы потомков)

```
div#paragraph1 p.note {  
    color: red;  
}
```



codepen.io/ngm/pen/pwmJrL

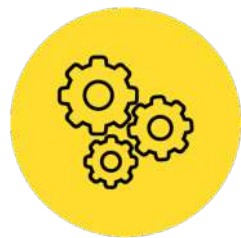


ТИНЬКОФФ

Синтаксис CSS

Контекстные селекторы (селекторы потомков)

```
div#paragraph1 p.note {  
    color: red;  
}
```



codepen.io/ngm/pen/pwmJrL



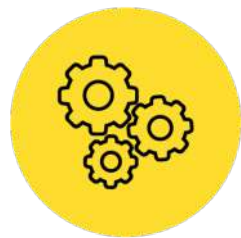
ТИНЬКОФФ

```
<div>  
  <p>  
    <a href="/"></a>  
  <p></p>  
</p>  
<h1 id="paragraph1"></h1>  
<ul>  
  <li>  
    <p></p>  
  </li>  
</ul>  
<div id="paragraph1">  
  <p class="note"></p>  
  <p></p>  
  <p class="note2"></p>  
</div>  
</div>
```

Синтаксис CSS

Селектор дочерних элементов

```
div.note > b {  
    color: red;  
}
```



codepen.io/ngm/pen/OgYVBg

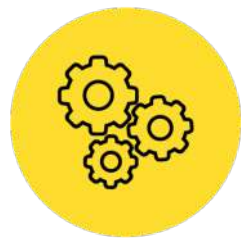


ТИНЬКОФФ

Синтаксис CSS

Селектор дочерних элементов

```
div.note > b {  
    color: red;  
}
```



codepen.io/ngm/pen/OgYVBg

```
<div>  
  <p>  
    <a href="/"></a>  
    <p></p>  
  </p>  
  <h1></h1>  
  <ul>  
    <li>  
      <p></p>  
    </li>  
  </ul>  
  <div class="note">  
    <p class="note">  
      <b></b>  
    </p>  
    <b></b>  
  </div>  
</div>
```

Синтаксис CSS

Селектор сестринских элементов

```
h1 + p {  
    margin-top: 20px;  
}
```



codepen.io/ngm/pen/qjGdeO



ТИНЬКОФФ

Синтаксис CSS

Селектор сестринских элементов

```
h1 + p {  
    margin-top: 20px;  
}
```



codepen.io/ngm/pen/qjGdeO

```
<div>  
  <p>  
    <a href="/"></a>  
    <p></p>  
  </p>  
  <h1></h1>  
  <p></p>  
  <p></p>  
  <ul>  
    <li>  
      <p></p>  
    </li>  
  </ul>  
  <p></p>  
</div>
```

Синтаксис CSS

Селектор псевдоклассов

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



codepen.io/ngm/pen/ModmKr

webref.ru/css#pseudo-class



ТИНЬКОФФ

Синтаксис CSS

Селектор псевдоклассов

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

```
<div>  
  <p>  
    <a href="/"></a>  
    <p></p>  
  </p>  
  <h1></h1>  
  <p></p>  
  <p></p>  
  <ul>  
    <li>  
      <p></p>  
    </li>  
  </ul>  
  <p></p>  
</div>
```



codepen.io/ngm/pen/ModmKr

webref.ru/css#pseudo-class

Синтаксис CSS

Селектор псевдоэлементов

```
p::first-letter {  
    font-size: 32px;  
}
```



codepen.io/ngm/pen/JJqNXw

webref.ru/css#pseudo-elements



ТИНЬКОФФ

Синтаксис CSS

Селектор псевдоэлементов

```
p::first-letter {  
    font-size: 32px;  
}
```



codepen.io/ngm/pen/JJqNXw

```
<div>  
  <p>  
    Текст  
  <p>Текст</p>  
</p>  
<h1></h1>  
<p></p>  
<p></p>  
<ul>  
  <li>  
    <p></p>  
  </li>  
</ul>  
<p></p>  
</div>
```

webref.ru/css#pseudo-elements

Каскадность, наследование, специфичность

Каскадность, наследование, специфичность

CSS - это сокращение от *Cascading Style Sheets* (Каскадные таблицы стилей)

Каскадность означает, что к одному и тому же элементу может применяться несколько CSS-правил.

Каскадность, наследование, специфичность

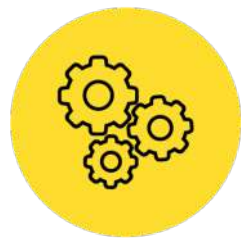
1. Важность
2. Специфичность
3. Исходный порядок

Каскадность, наследование, специфичность

Важность

Запись свойства с **!important** в конце всегда имеет приоритет перед остальными

```
.better {  
  background-color: gray;  
  border: none !important;  
}
```



codepen.io/ngm/pen/eRaWWK



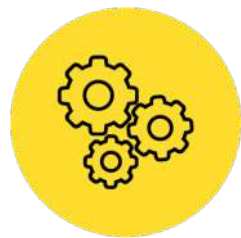
ТИНЬКОФФ

Каскадность, наследование, специфичность

Важность

Запись свойства с **!important** в конце всегда имеет приоритет перед остальными

```
.better {  
  background-color: gray;  
  border: none !important;  
}
```



codepen.io/ngm/pen/eRaWWK



ТИНЬКОФФ



Каскадность, наследование, специфичность

Специфичность

Побеждает тот селектор, который более специфичен.

4 составляющих специфичности:

1. Нахождение селектора внутри элемента **<style>** или внутри атрибута **style**
2. Наличие селектора ID
3. Наличие селектора класса, атрибута или псевдо-класса
4. Наличие селектора элемента или псевдо-элемента

Каскадность, наследование, специфичность

Селектор	style=""	#id	.class	<tag>	Специфичность
h1	0	0	0	1	0001
#id	0	1	0	0	0100
h1 + p::first-letter	0	0	0	3	0003
li > a[href="site"] > .inline-warning	0	0	2	2	0022
#id div > div > a:hover, внутри элемента <style>	1	1	1	3	1113



Каскадность, наследование, специфичность

Селектор	style=""	#id	.class	<tag>	Специфичность
span	0	0	0	0	0000

Какая специфичность у **span**?

Каскадность, наследование, специфичность

Селектор	style=""	#id	.class	<tag>	Специфичность
span	0	0	0	1	0001

Каскадность, наследование, специфичность

Селектор	style=""	#id	.class	<tag>	Специфичность
span	0	0	0	1	0001
div .class	0	0	0	0	0000

Каскадность, наследование, специфичность

Селектор	style=""	#id	.class	<tag>	Специфичность
span	0	0	0	1	0001
div .class	0	0	1	1	0011

Каскадность, наследование, специфичность

Селектор	style=""	#id	.class	<tag>	Специфичность
span	0	0	0	1	0001
div .class	0	0	1	1	0011
#id .class	0	0	0	0	0000

Каскадность, наследование, специфичность

Селектор	style=""	#id	.class	<tag>	Специфичность
span	0	0	0	1	0001
div .class	0	0	1	1	0011
#id .class	0	1	1	0	0110
div span	0	0	0	0	0000

Каскадность, наследование, специфичность

Селектор	style=""	#id	.class	<tag>	Специфичность
span	0	0	0	1	0001
div .class	0	0	1	1	0011
#id .class	0	1	1	0	0110
div span	0	0	0	2	0002

Каскадность, наследование, специфичность

Селектор	style=""	#id	.class	<tag>	Специфичность
span	0	0	0	1	0001
div .class	0	0	1	1	0011
#id .class	0	1	1	0	0110
div span	0	0	0	2	0002
#id span	0	0	0	0	0000

Каскадность, наследование, специфичность

Селектор	style=""	#id	.class	<tag>	Специфичность
span	0	0	0	1	0001
div .class	0	0	1	1	0011
#id .class	0	1	1	0	0110
div span	0	0	0	2	0002
#id span	0	1	0	1	0101

Каскадность, наследование, специфичность

Справочные материалы:

<http://css.yoksel.ru/specificity/>

<https://developer.mozilla.org/ru/docs/Web/CSS/Specificity>

Сброс CSS

На самом деле HTML без стилей не бывает.

Каждая веб-страница использует по крайней мере один CSS: **стиль клиентского приложения**.

Этот файл CSS включён в браузер и вызывается:

- **каждый раз**, когда веб-страница визуализируется;
- **до того**, как применяется любой из наших CSS.

Стили браузера по умолчанию могут **мешать** стилям, которые мы на самом деле хотим применить.

Сброс CSS

Это особенно полезно для нормализации размера текста и удаления всех отступов.

Вы можете найти (и скачать) HTML5 reset.css с сайта [Marksheet](#). Просто включите его в **head** перед вашей собственной таблицы стилей.

```
<head>
  <link rel="stylesheet" href="reset.css">
  <link rel="stylesheet" href="styles.css">
</head>
```

Блочная модель (Box model)

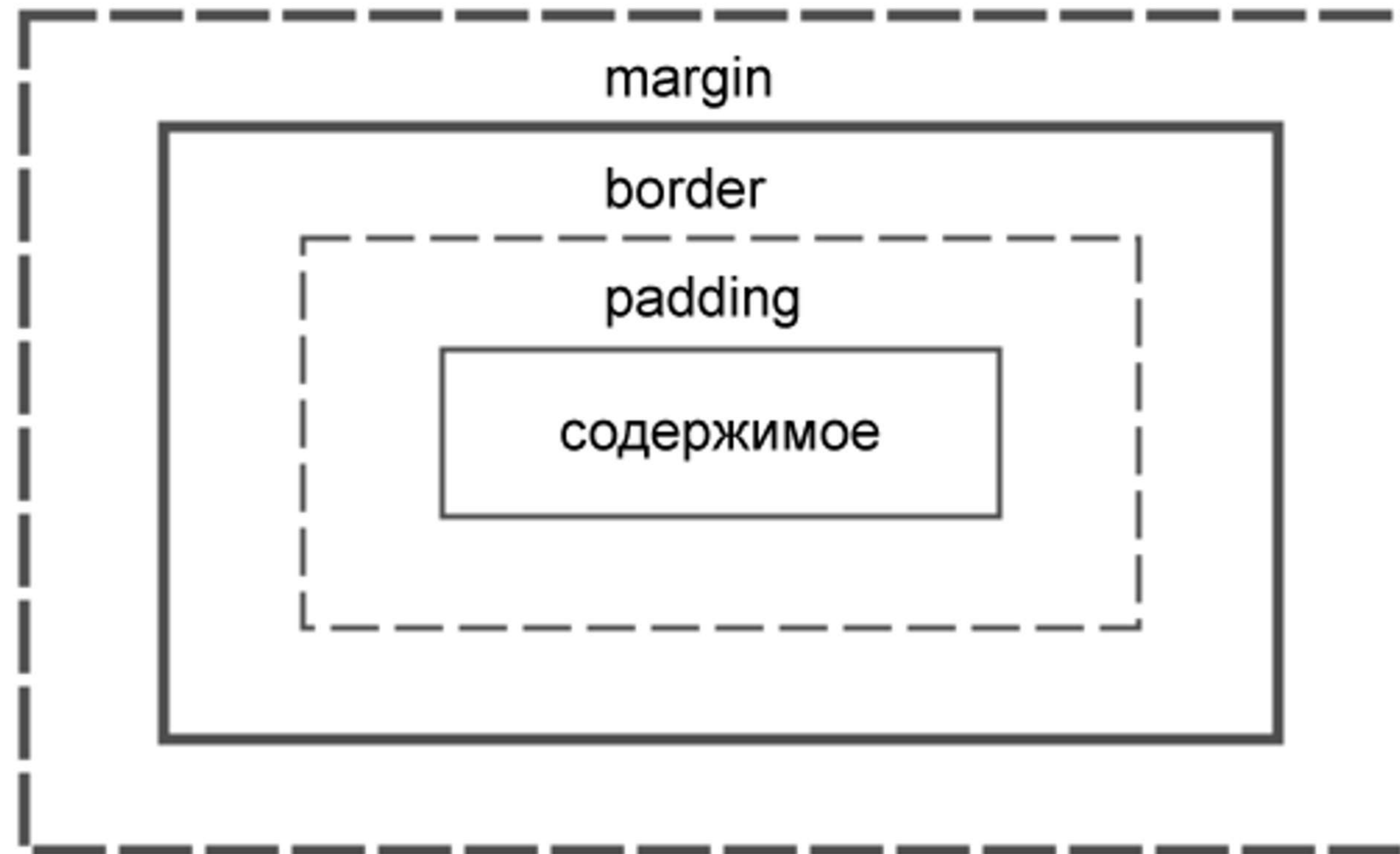
Блочная модель

Блочная модель (box model) — одно из фундаментальных понятий веб-верстки.

Элемент в HTML документе выводится внутри своего отдельного прямоугольного блока.

Блочная модель

Блок состоит из непосредственно контента (внутреннего содержимого), внутренних отступов (padding), границ (border) и, наконец, внешних отступов (margin).



Блочная модель

Размеры блока

Свойство **box-sizing** позволяет вам немного управлять моделью, применяемой для задания размеров блоков. Два возможных значения — это **content-box** и **border-box**.

content-box

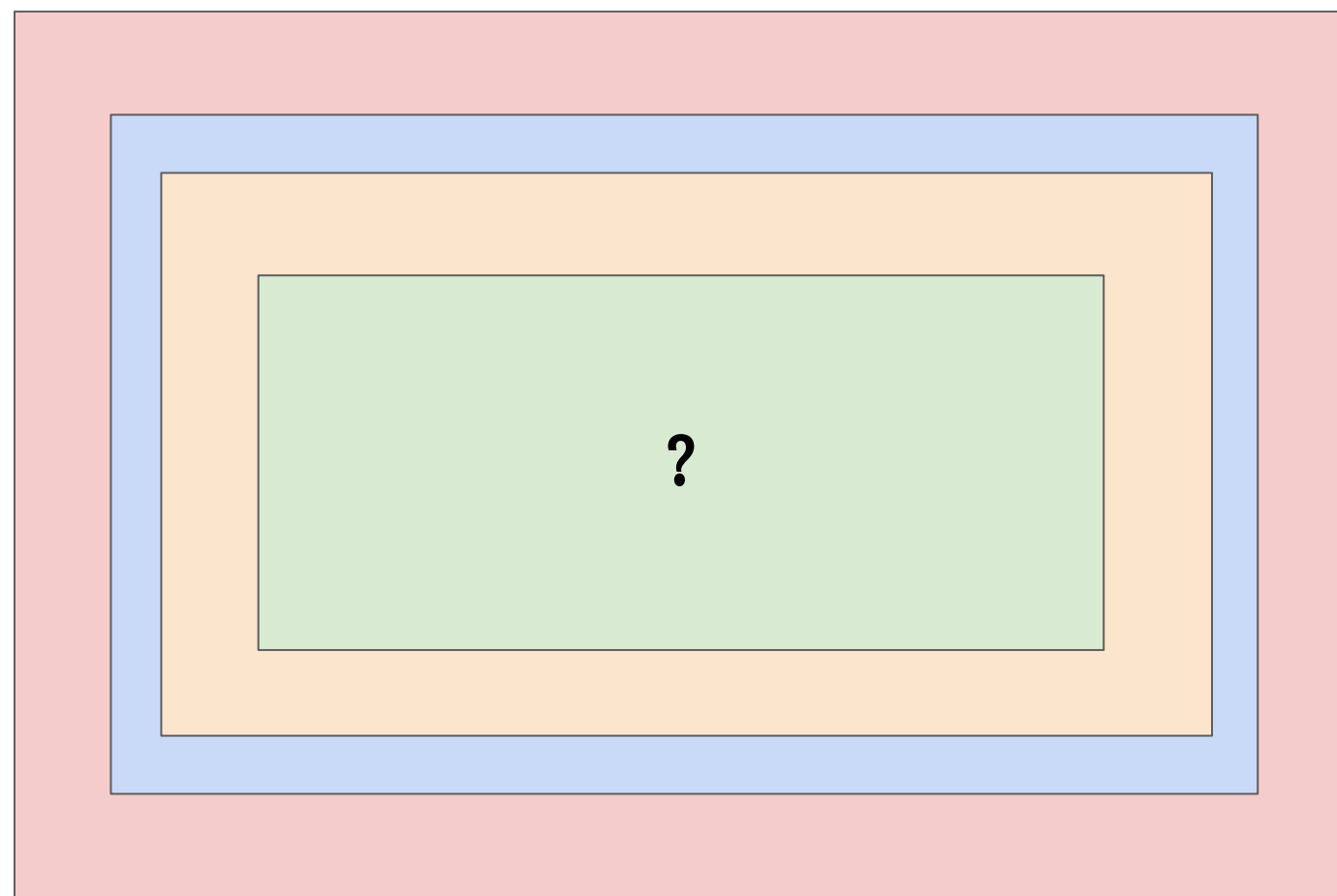
По умолчанию. При вычислении размера блока добавляются **padding** и **border**.

border-box

При вычислении размера блока **padding** и **border** не учитываются.

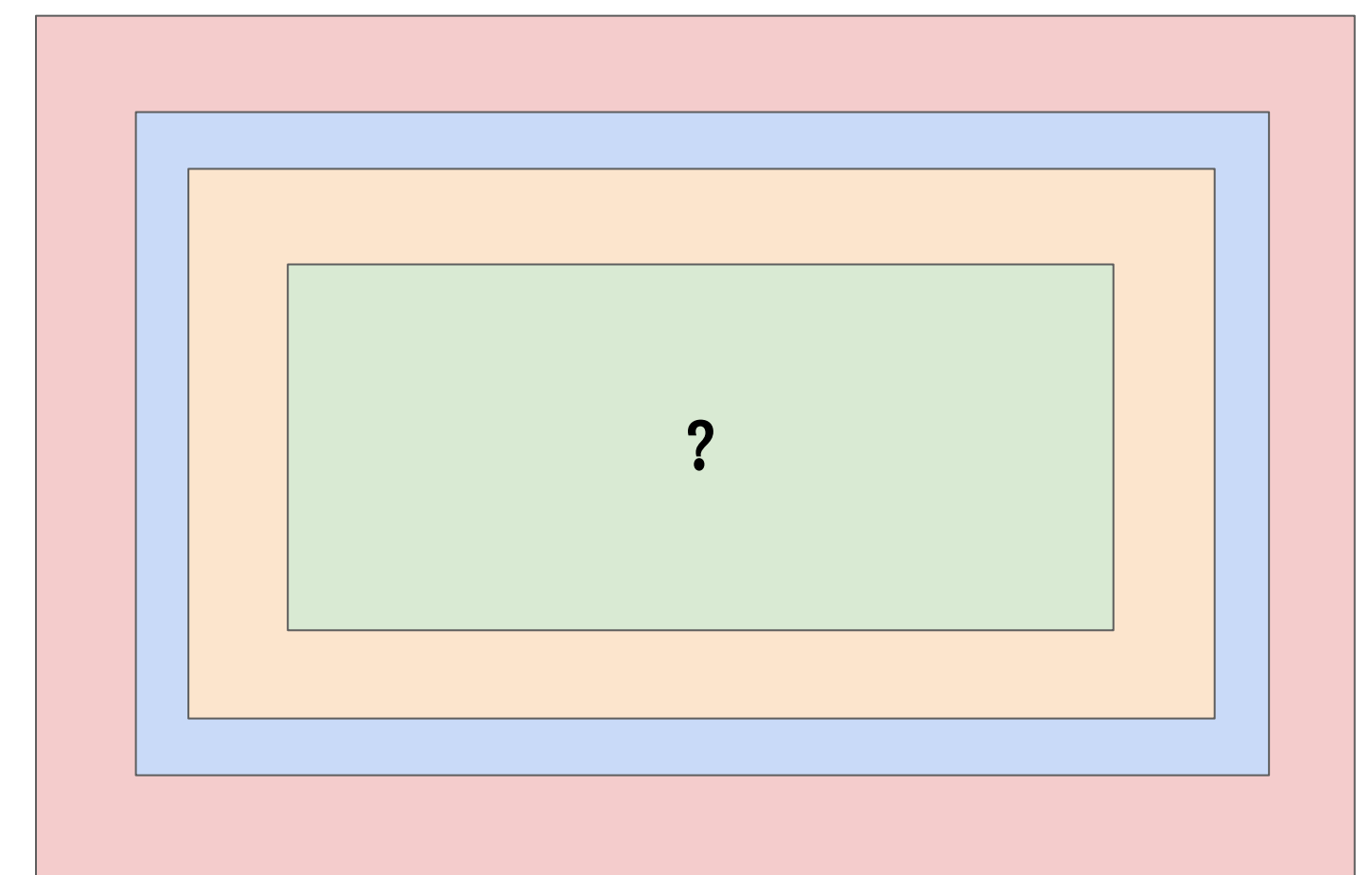
Блочная модель

content-box



```
{  
  width: 200px;  
  height: 100px;  
  padding: 20px;  
  border: 10px;  
  margin: 20px;  
}
```

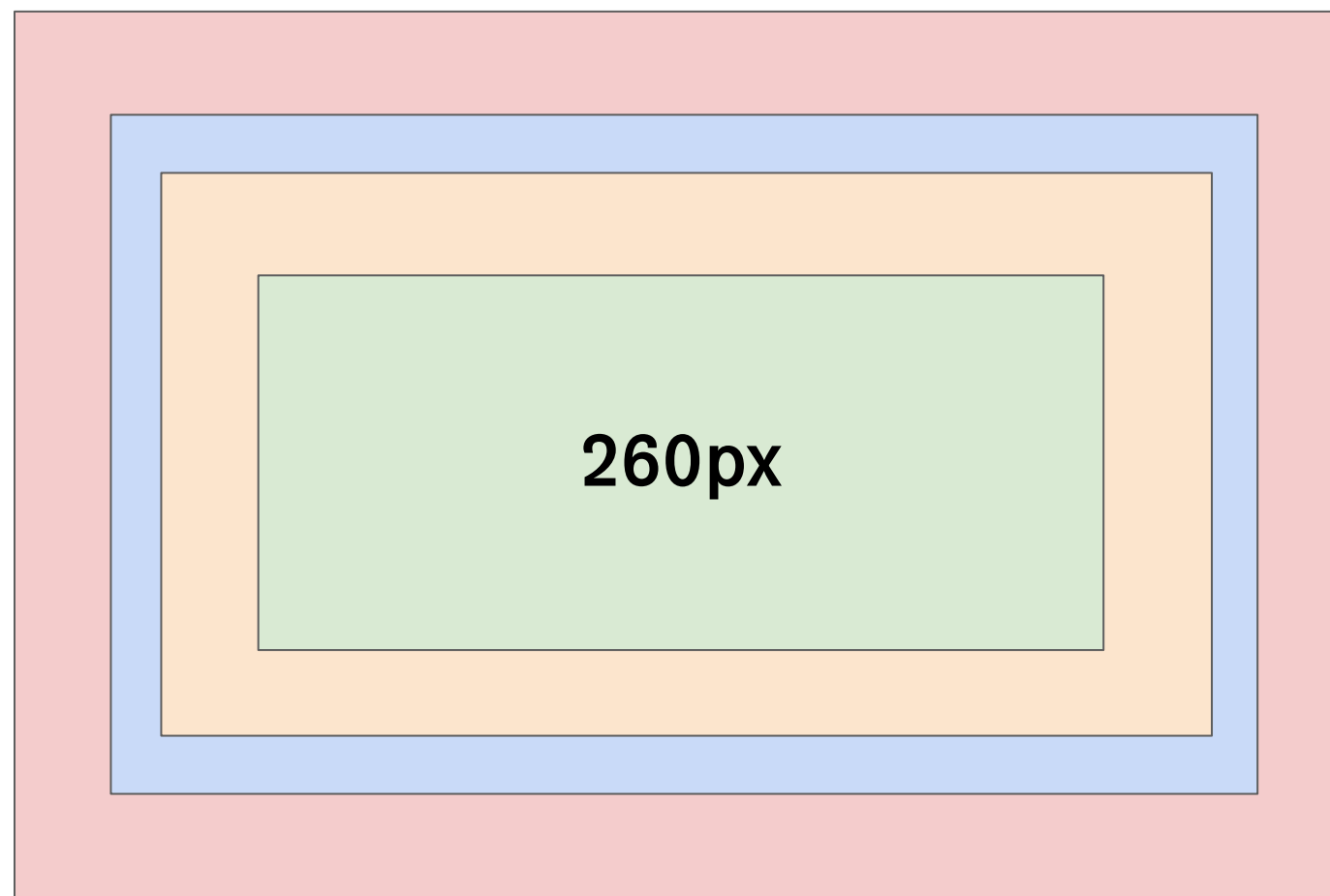
border-box



```
{  
  width: 200px;  
  height: 100px;  
  padding: 20px;  
  border: 10px;  
  margin: 20px;  
  box-sizing: border-box;  
}
```

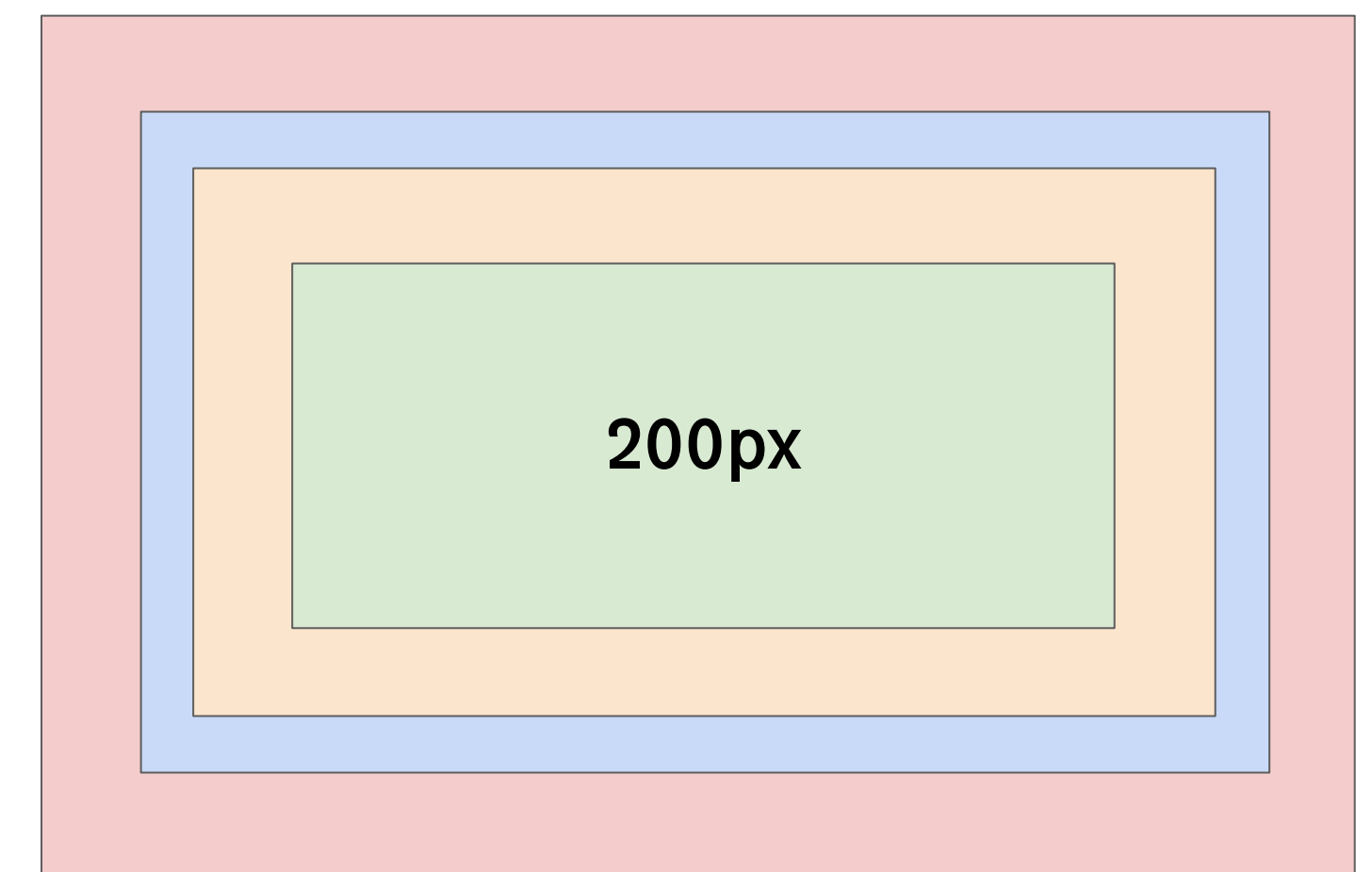
Блочная модель

content-box



```
{  
  width: 200px;  
  height: 100px;  
  padding: 20px;  
  border: 10px;  
  margin: 20px;  
}
```

border-box



```
{  
  width: 200px;  
  height: 100px;  
  padding: 20px;  
  border: 10px;  
  margin: 20px;  
  box-sizing: border-box;  
}
```

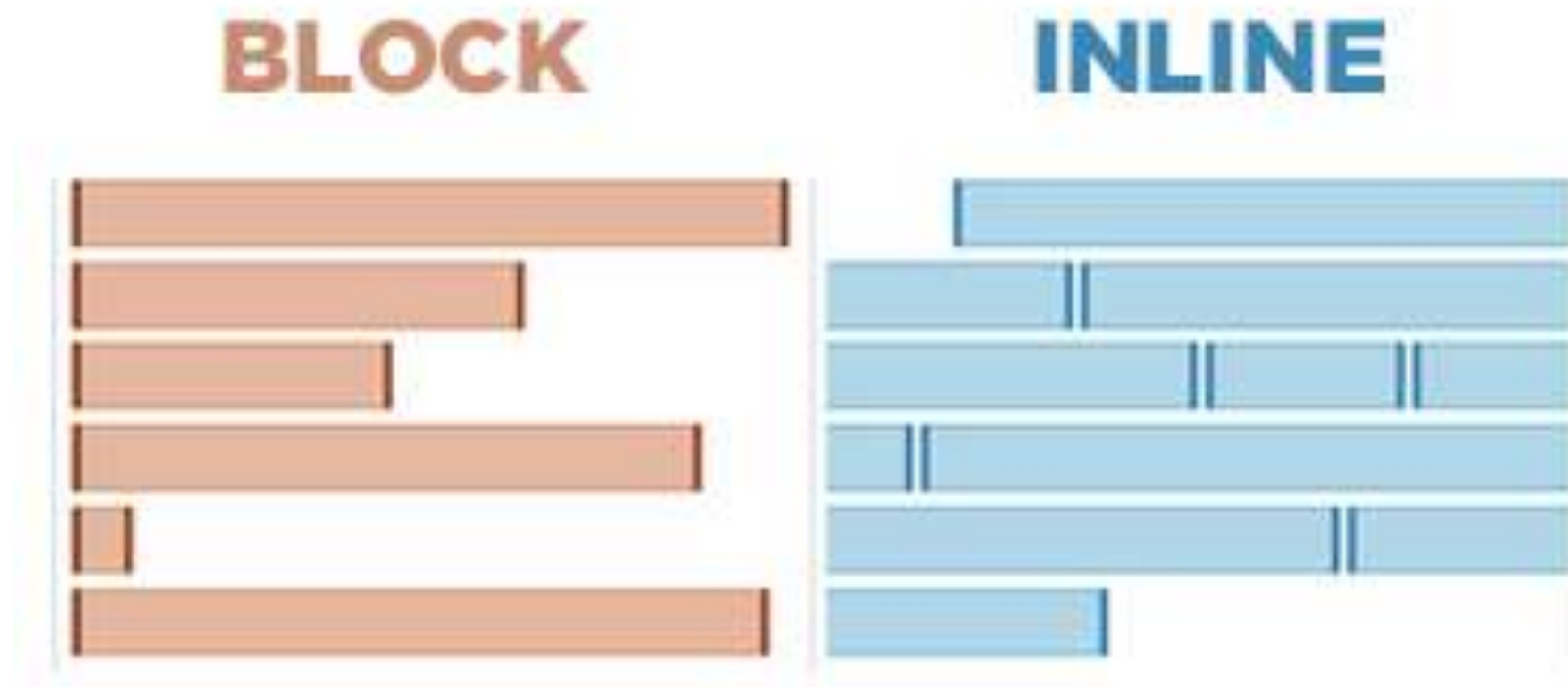
Блочная модель

Блочные и строчные элементы

Элементы в HTML делятся на блочные и строчные:

- **Блочные элементы (block)** занимают всю ширину своего родителя (контейнера), формально создавая «блок» (отсюда и название).
- **Строчные элементы (inline)** это те, которые занимают только то пространство, которое ограничено тегами, определяющими строчный элемент (необходимое для отображения их содержимого) и не нарушая потока содержимого (не требующее новой строки после каждого элемента).

Блочная модель



Блочная модель

Блочные элементы:

```
<h1> <h2> <h3> <h4> <h5> <h6>  
<p>  
<div>  
<ul> <ol> <li>  
<section>  
<article> ...
```

Строчные элементы:

```
<span>  
<strong> <em> <b> <i> <cite> <del>  
<a>  
<img>  
<input> ...
```

Блочная модель

При помощи CSS-свойства `display`, мы можем любой элемент сделать строчным или блочным. Поговорим о нем в следующей лекции.

Единицы размера и цвета в CSS

Единицы размера в CSS

Существует много свойств CSS, которые требуют размер в качестве единицы:

- **font-size** определяет размер текста
- **border-width** определяет толщину границ элементов
- **margin** определяет пространство между элементами
- **padding** определяет внутренние отступы элемента
- **top/left/right/bottom** позволяют позиционировать и перемещать элементы
- **width/height** определяет размеры элемента

Единицы размера в CSS

Наиболее часто используемые единицы:

- **px** для пикселей
- **%** для процентов
- **em/rem** для определения размера относительно родительского значения **font-size**
- **vw/vh** для определения размера относительно экрана

Единицы размера в CSS

Пиксели

Абсолютные единицы и не зависят от других наследуемых свойств. Самая распространенная единица размера в CSS.

```
.banner {  
  width: 300px;  
  padding: 20px;  
}
```

Единицы размера в CSS

Проценты

Относительные единицы: полагаются на родителя и/или предка элемента.

```
.container {  
  width: 80%;  
}  
  
.item {  
  margin-top: 10%;  
}
```

Единицы размера в CSS

em и rem

Относительные единицы: зависят от значения **font-size** родительского элемента (**em**) или корневого элемента (**rem**).

```
.container {  
    font-size: 20px;  
}  
  
h2 {  
    font-size: 2em; /* = 20px * 2 = 40px */  
}
```

```
html {  
    font-size: 20px;  
}  
  
h1 {  
    font-size: 3rem; /* = 20px * 3 = 60px */  
}  
  
p {  
    font-size: 0.8rem; /* = 20px * 0.8 = 16px */  
}
```


Единицы размера в CSS

vw и vh

Относительные единицы: зависят от размеров окна браузера (**viewport**).

- **1vw** – 1% ширины окна
- **1vh** – 1% высоты окна

```
h2 {  
    font-size: 3vw;  
}  
  
section {  
    height: 100vh;  
}
```

Единицы цвета в CSS

Цвета широко используются в **CSS** для изменения цвета текста, цвет фона, градиентов, теней, границ и др. Есть несколько способов задать цвета в **CSS**, каждый со своими плюсами и минусами.

Единицы цвета в CSS

Название цвета

На данный момент в CSS **147** названий цветов, от самых простых (black, white, orange, yellow, blue и др.) до более специфичных (lawngreen, orchid, crimson и др.).

```
body {  
    color: black;  
}  
  
a {  
    color: orange;  
}
```

<https://colorscheme.ru/html-colors.html>

Единицы цвета в CSS

rgb и rgba

Является результатом сочетания красного, зеленого и синего, у каждого из этих трёх цветов есть 256 возможных значений, существует $256 * 256 * 256 = 16777216$ возможных цветов.

```
body {  
    background-color: rgba(0, 0, 0, 0.8);  
}  
  
a {  
    color: rgb(219, 78, 68);  
}
```

Единицы цвета в CSS

Шестнадцатеричные значения (HEX)

Сочетание красного, зеленого и синего, причем каждый из них представляют в шестнадцатеричном значении, например DB для красного, 4E для зелёного и 44 для синего.

```
body {  
    background-color: #000000;  
}  
  
a {  
    color: #ADFF2F;  
}
```

Единицы размера и цвета в CSS

Справочные материалы:

<http://css.yoksel.ru/size-units/>

<https://learn.javascript.ru/css-units>

<https://habr.com/ru/company/ruvds/blog/505626/>

Единицы размера и цвета в CSS

Справочные материалы:

<http://css.yoksel.ru/size-units/>

<https://learn.javascript.ru/css-units>

<https://habr.com/ru/company/ruvds/blog/505626/>

Нужно всегда помнить

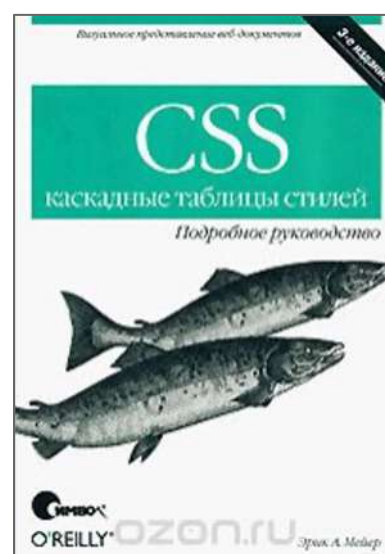
Браузерные войны

<https://habr.com/ru/company/ispsystem/blog/712370/>

Статьи на Medium: [Часть 1](#) и [Часть 2](#)



Что почитать?



CSS. Каскадные таблицы стилей. Подробное руководство
Эрик Мейер



Новая большая книга CSS
Дэвид Макфарланд



Секреты CSS.
Идеальные решения ежедневных задач
Леа Веру

Интернет-ресурсы

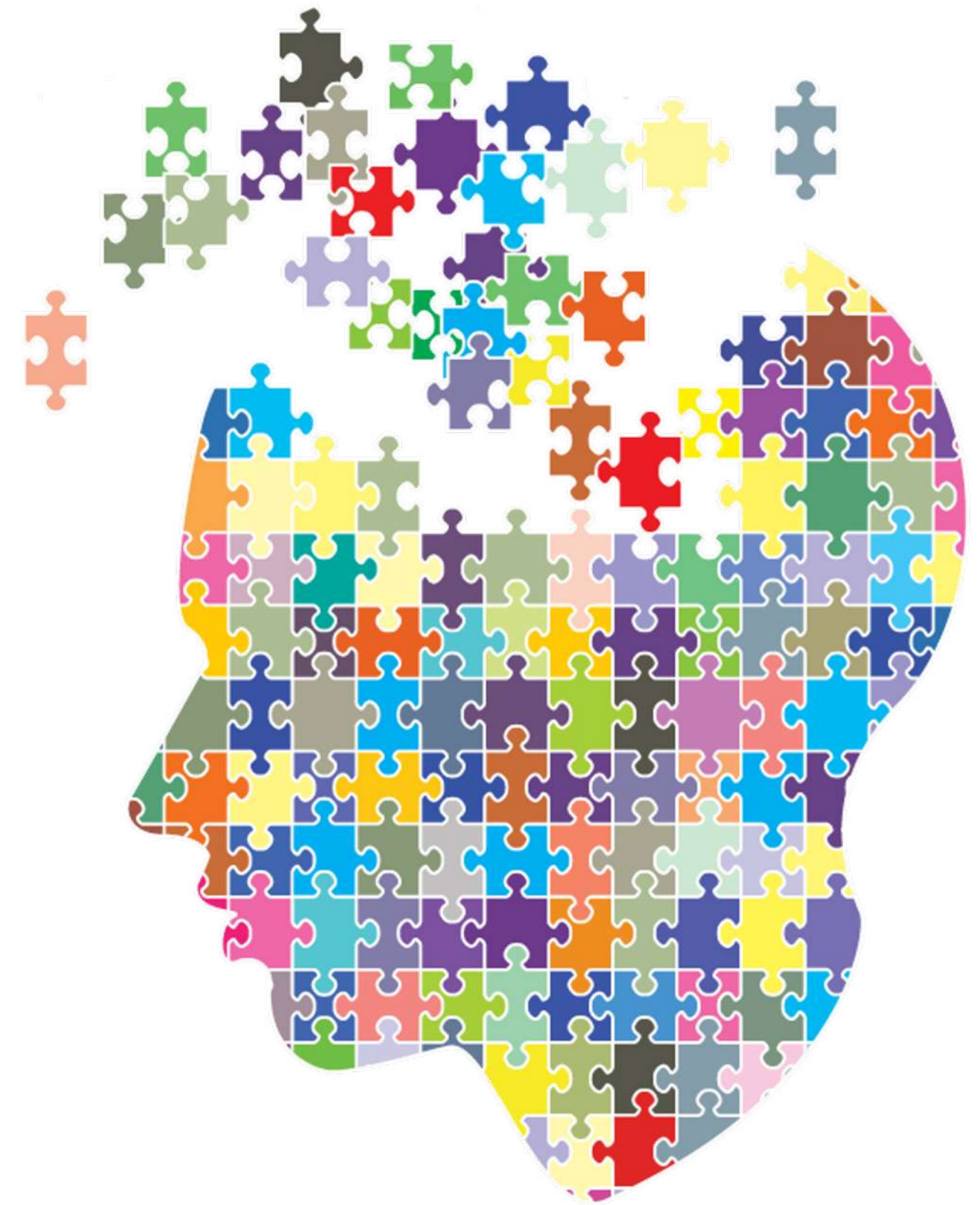
developer.mozilla.org/ru

webref.ru

htmlacademy.ru

html5book.ru

css-live.ru



Интернет-ресурсы

Какой самый главный интернет-ресурс?

Интернет-ресурсы

Какой самый главный интернет-ресурс?



или любой другой поисковик

Полезные ссылки

[Основы html](#)

[Фронтенд это не больно](#)

[Старый и новый CSS. История веб-дизайна](#)

[Как подключить кастомные шрифты](#)

[Параметры шрифта](#)

[Фоны](#)

[Скругления](#)

[Псевдоклассы и псевдоэлементы](#)



Спасибо!

