

ОСНОВЫ CSS. Часть I



CSS

Елена Иванова
@liveldi90
Front-end developer
Artec3D

Содержание

- ➔ Способы подключения CSS к HTML.
- ➔ Селекторы и комбинаторы.
- ➔ Свойства и значения.
- ➔ Каскад.
- ➔ Типы верстки.
- ➔ Текст и шрифты.
- ➔ Кастомные шрифты и способы их подключения в CSS.

Способы подключения CSS к HTML

CSS

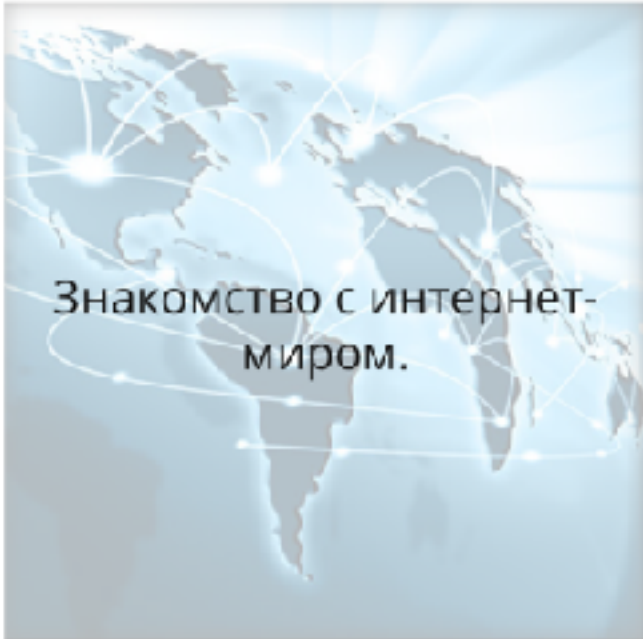
(Cascading Style Sheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа.

CSS используется для задания цветов, шрифтов, расположения отдельных блоков и других элементов представления внешнего вида html-страниц.



КОДИНГ ДЛЯ ДЕВУШЕК

(4 апреля 2017)



Знакомство с интернет-миром.



Основы HTML.



Основы CSS. Часть I.

Основы CSS. Часть II.

Продвинутый CSS.

Дополнительная
практика по CSS.



Кодинг для девушек

(4 апреля 2017)

- [Знакомство с интернет-миром.](#)
- [Основы HTML.](#)
- [Основы CSS. Часть I.](#)
- Основы CSS. Часть II.
- Продвинутый CSS.
- Дополнительная практика по CSS.
- Знакомство с программированием. JavaScript Начало.
- JavaScript: взаимодействие с DOM & Анимация.
- Готовые решения.
- Публикация и полный вперед!

Автор курса: [Иванова Елена](#) x [Moscow Coding School](#)

Основные понятия

Код CSS

Таблица стилей
(style sheet)

```
h1 {  
    color: #000000;  
}
```

Правило
(rule, rule set)

h1

Селектор (selector)

color

Свойство (property)

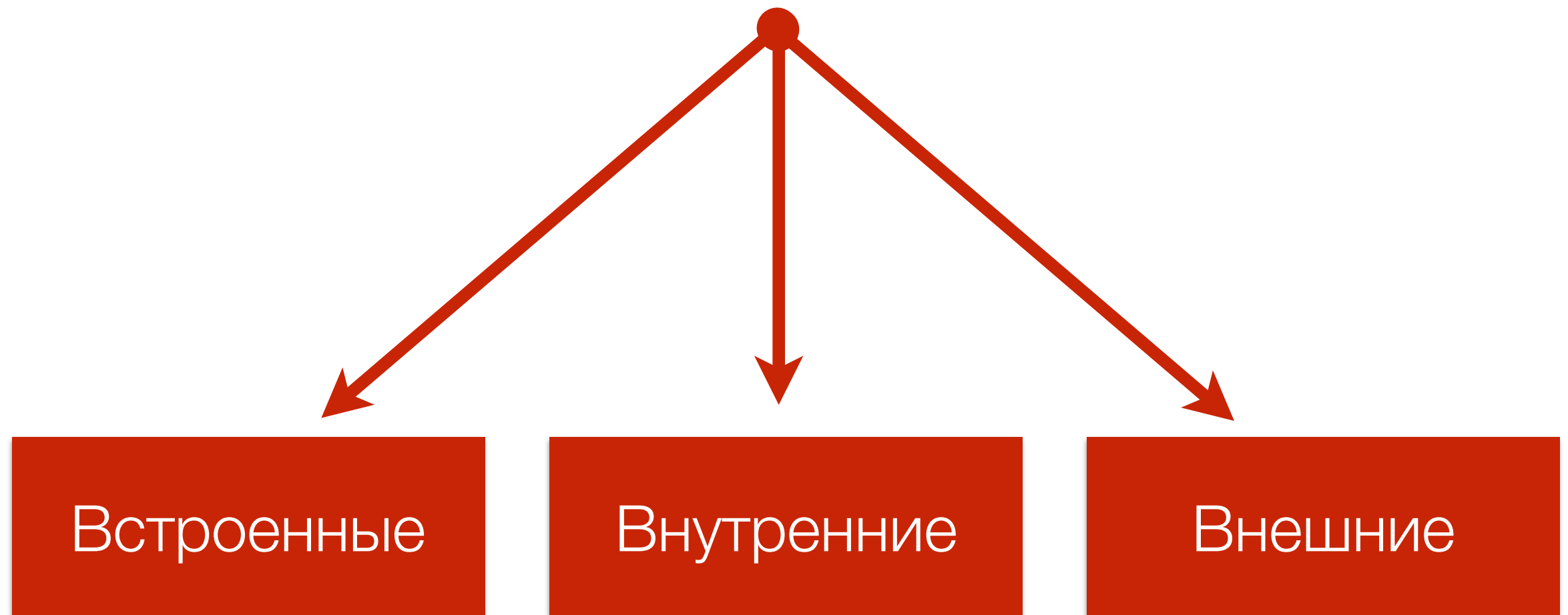
#000000

Значение (value)

Пример таблицы стилей

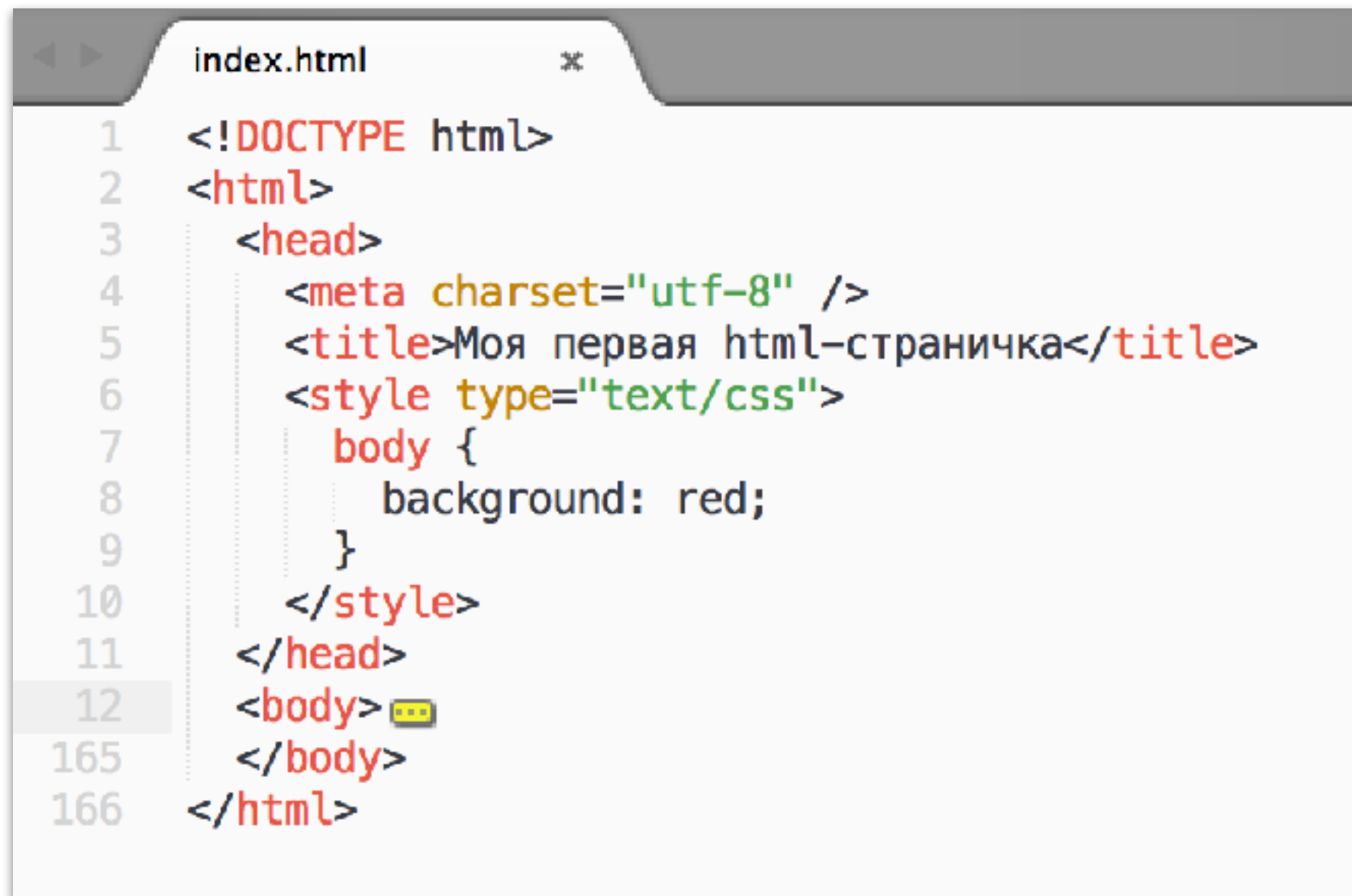
```
main.css
1  h1 {
2      font-size: 20px;
3      color: red;
4      margin-bottom: 10px;
5  }
6
7  p {
8      font-size: 14px;
9      color: gray;
10     margin-bottom: 5px;
11 }
12
13
```



Способы подключения CSS



Встроенные таблицы стилей

Таблица стилей встраивается в **head** html-страницы. Для этого в HTML существуют теги **<style></style>**, с параметром **type**, который указывает, что подключается именно таблица стилей CSS



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Моя первая html-страничка</title>
6     <style type="text/css">
7       body {
8         background: red;
9       }
10    </style>
11  </head>
12  <body> 
165 </body>
166 </html>
```

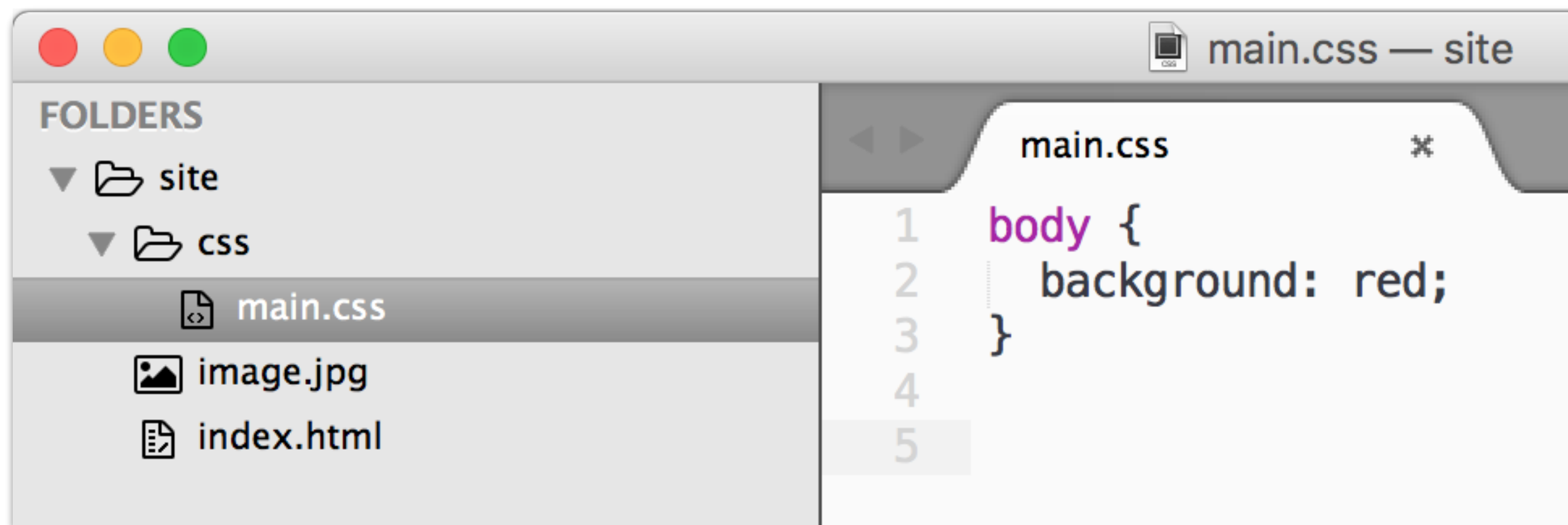
Внутренние таблицы стилей

— задается внутри элемента HTML, при помощи атрибута style:

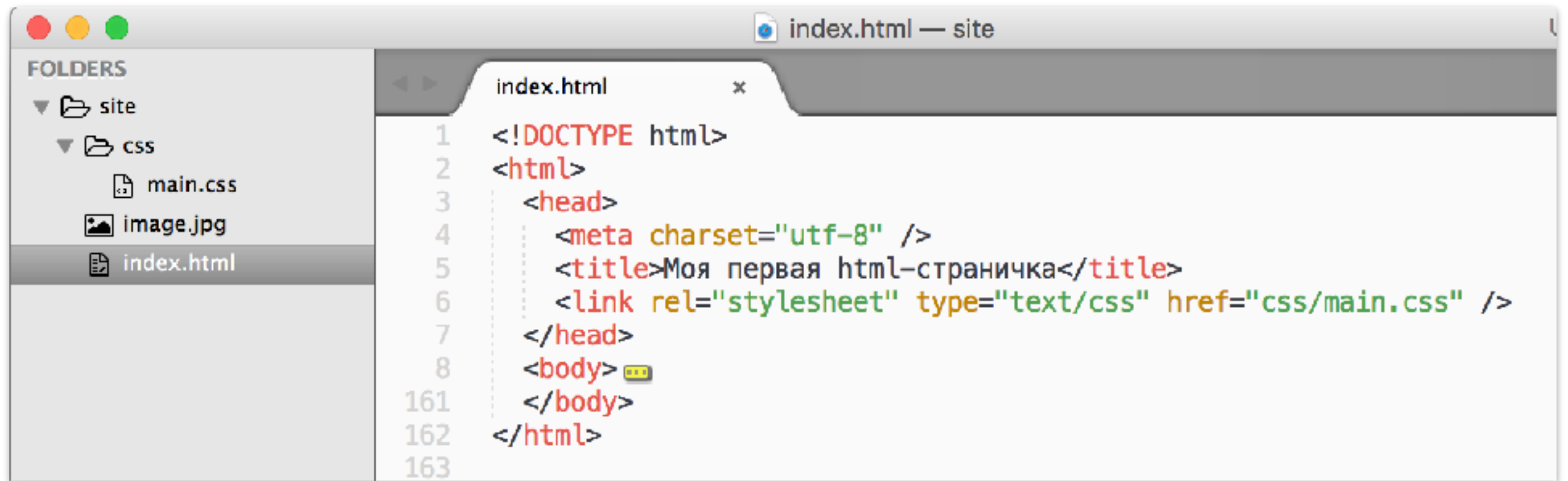
```
index.html *
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title>Моя первая html-страничка</title>
6   </head>
7   <body>
8     <hgroup style="background: red; color: green;">
9       <h2>Блочные элементы</h2>
10      <h3>Всегда начинаются с новой строки</h3>
11    </hgroup>
12
```

Создаем файл.css

Добавляем файл **main.css** в папку **css**



Подключаем main.css к index.html



Самый правильный способ подключения CSS к странице. HTML должен содержать только разметку, CSS — только стили. Следует придерживаться чистого метода разработки, чтобы проще было поддерживать код в будущем.

Тег link

— Самозакрывающийся (не имеет содержимого), но есть следующие атрибуты:

href - указывает URL-адрес подключаемого файла.

rel - указывает на тип отношения данного документа к внешнему (например: **rel="stylesheet"** указывает, что внешний файл определяет стиль текущего документа).

type - указывает тип и параметры присоединенной таблицы стилей (**type="text/css"**).

Сброс стилей reset.css

Каждый браузер устанавливает свои значения стилей по умолчанию для различных HTML-элементов. С помощью CSS Reset мы можем аннулировать эту разницу для обеспечения кроссбраузерности стилей.

Сброс стилей reset.css

Создаем файл reset.css и копируем код стилей со страницы: [reset.css](#).

Подключаем во всех файлах .html, перед всеми СТИЛЯМИ.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Мои истории</title>
    <link rel="stylesheet" type="text/css" href="css/reset.css">
  </head>
  <body> 
  </body>
</html>
```


Селекторы и комбинаторы

Селекторы

Селектор **определяет элемент или группу элементов** или другие компоненты веб-страницы, **к которым будут применены объявления в блоке.**

1. Селектор по типу элемента

```
1  h1 {  
2      color: red;  
3  }
```

Окрасит все заголовки 1-го уровня в красный цвет.

Селекторы

2. Универсальный селектор

```
1  * {  
2      margin: 0;  
3      padding: 0;  
4  
5      color: red;  
6  }  
7
```

Уберем все внешние и внутренние отступы у элементов.

Окрасит весь текст на странице в красный цвет.



Класс (class)

— позволяет **связать определенный тег со стилевым оформлением**. В значении допускается указывать сразу несколько классов, разделяя их между собой пробелом.

```
<h1 class="title">Заголовок</h1>  
<p class="text-red text-bold">  
    Некоторый параграф текста.  
</p>
```

Селекторы

3. Селектор по классу

```
1  .title {  
2      background: pink;  
3  }  
4  
5  .text-red {  
6      color: red;  
7  }  
8  
9  .text-bold {  
10     font-weight: bold;  
11 }
```

Позволяет применить одинаковый стиль к элементам различного типа. Для работы необходимо добавить к определенным тегам классы. Имя класса придумываем сами.

Идентификатор (id)

— **уникальное имя элемента**, которое **используется для изменения его стиля и обращения** к нему **через скрипты**.

Идентификатор в коде документа должен быть в единственном экземпляре.

```
<div id="page">  
  <header id="header"></header>  
  <main id="container"></main>  
  <footer id="footer"></footer>  
</div>
```

Селекторы

4. Селектор по идентификатору

```
1  #page {  
2      background: red;  
3  }
```

Аналогичен селектору класса. Вместо . используем #.

В одном HTML документе может быть только один id.

Селекторы

5. Селекторы атрибутов

```
1  [alt="Пушкин"] {  
2      border: 1px solid black;  
3  }  
4  
5  a[href^="http://"] {  
6      border: 1px solid black;  
7  }  
8  
9  a[href$=".pdf"] {  
10     border: 1px solid black;  
11 }
```

— Все элементы с атрибутом alt="Пушкин"

— Все ссылки атрибут href которых начинается на "http://"

— Все ссылки атрибут href которых заканчивается на ".pdf"

Селекторы

6. **Псевдоклассы** — это атрибуты, назначаемые строго к селекторам с намерением определить реакцию или состояние для данного селектора.

```
1  a {  
2      color: black;  
3  }  
4  
5  a:visited {  
6      color: blue;  
7  }  
8  
9  a:active {  
10     color: green;  
11 }  
12  
13 a:hover {  
14     color: pink;  
15 }
```

— Ссылка **по умолчанию** черного цвета

— Ссылка становится голубой, если **по ней переходили**. *

— **При клике**, ссылка окрашивается в зеленый цвет. *

— **При наведении**, ссылка приобретает розовый цвет.

* — псевдоклассы свойственный только для ссылок.

Селекторы

6. **Псевдоклассы** — это атрибуты, назначаемые строго к селекторам с намерением определить реакцию или состояние для данного селектора.

```
1  a {  
2      color: black;  
3  }  
4  
5  a:visited {  
6      color: blue;  
7  }  
8  
9  a:active {  
10     color: green;  
11 }  
12  
13 a:hover {  
14     color: pink;  
15 }
```

— Ссылка **по умолчанию** черного цвета

— Ссылка становится голубой, если **по ней переходили**. *

— **При клике**, ссылка окрашивается в зеленый цвет. *

— **При наведении**, ссылка приобретает розовый цвет.

* — псевдоклассы свойственны только для ссылок.

Селекторы

7. Псевдоэлементы — позволяют задать стиль элементов не определённых в HTML-страничке, а также генерировать содержимое, которого нет в исходном коде текста.

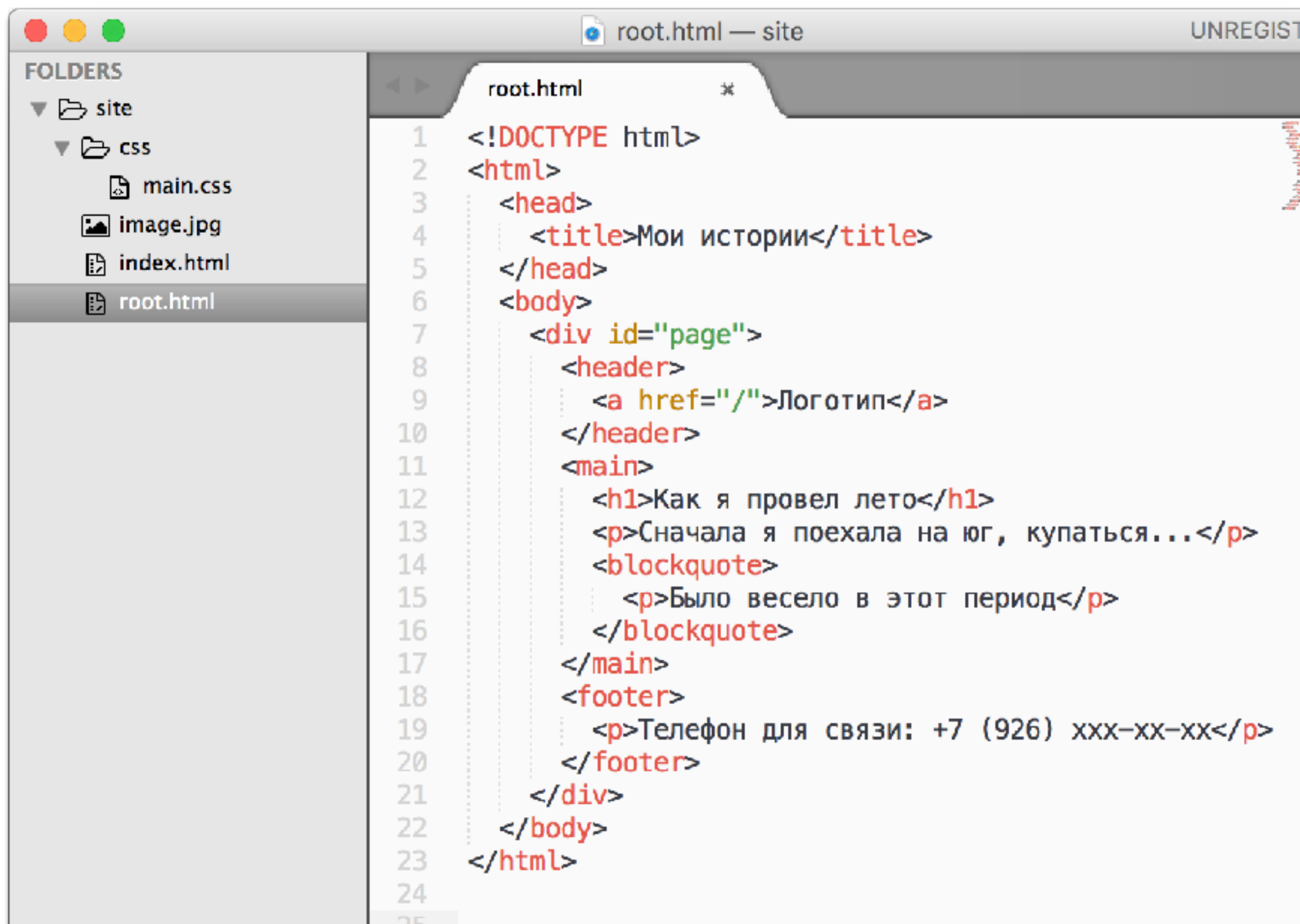
```
1  p::after {  
2      content: 'Любой текст/символ';  
3  }  
4  
5  p::before {  
6      content: 'Любой текст/символ';  
7  }  
8
```

— Добавляется после содержимого тега p

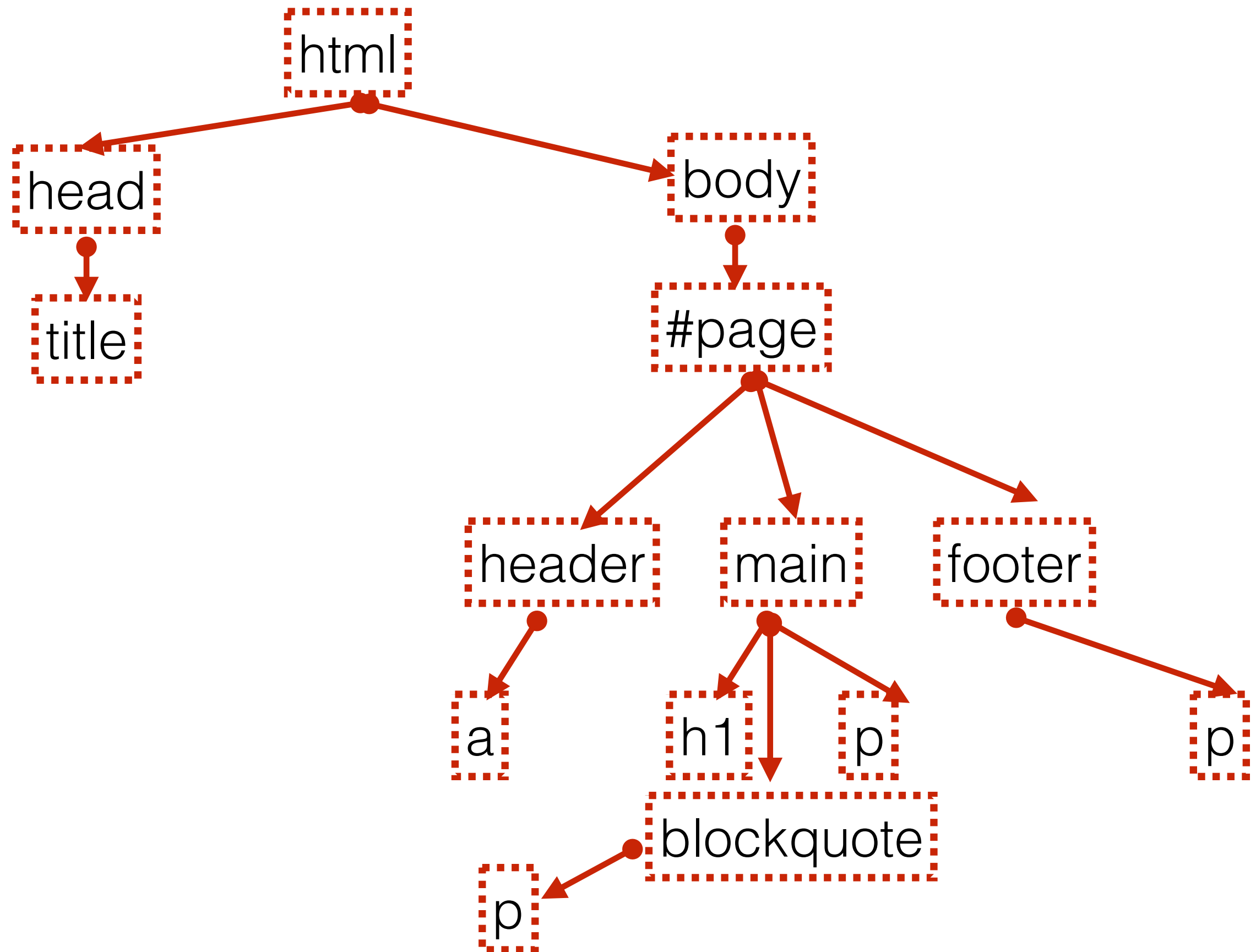
— Добавляется перед содержимым тега p

***content* — обязательный параметр для работы, он может быть пустым, но он нужен.**

HTML документ



Дерево HTML документа



Комбинаторы

1. Комбинатор элемента потока

```
1  main p {  
2    color: green;  
3  }  
4  
5  footer p {  
6    color: red;  
7  }
```

— все p в теге main
окрасить в зеленый цвет

— все p в теге footer
окрасить в красный
цвет

Разделение через пробел

Комбинаторы

2. Комбинатор дочернего элемента

т.е. комбинатор прямого потомка (дети элемента)

```
1  main > p {  
2      color: green;  
3  }
```

— в зеленый окрасятся,
только p первого уровня
в теге main.

Комбинаторы

3. Комбинатор ближайшего смежного элемента

```
1  h1 + p {  
2      color: red;  
3  }
```

— в красный окрасятся, только p которые идут сразу за h1 на одном уровне.

Группировка селекторов

Одни и те же свойства могут быть даны нескольким селекторам без необходимости повторять их.

```
1  main h1,  
2  footer p {  
3      color: red;  
4  }
```

— все h1 в теге main и все p в теге footer окрасятся в красный цвет.

Свойства и значения

Основные понятия

У каждого селектора есть свойства, которые находятся внутри фигурных скобок.

```
1  body {  
2      font-size: 14px;  
3      background: red;  
4  }
```

A diagram showing a snippet of CSS code. The code is: `1 body {`, `2 font-size: 14px;`, `3 background: red;`, `4 }`. Two orange arrows originate from the code. One arrow starts at the text `font-size` and points down and to the left towards the word 'Свойства'. The other arrow starts at the text `14px` and points down and to the right towards the word 'Значения'.

Свойства

Значения

За свойством следует знак двоеточия, а затем указывается значение этого свойства. В конце обязательно ставится ;

Длины и проценты

Значения, используемые в CSS, могут быть выражены в нескольких единицах измерений.

em

(font-size: 2em) — это единица измерения, которая определяет вычисляемый размер шрифта.

(2em указывает на то, что размер должен быть в два раза больше величины текущего шрифта.)

px

(font-size: 12px) — это единица измерения в пикселях.

rem

(font-size: 12pt) — типографский пункт (1/72 дюйма, приблизительно 0,35 мм).

%

(font-size: 80%) – единица измерения в процентах.

(200% указывает на то, что размер должен быть в два раза больше величины текущего шрифта. 200% = 2em)

Веб-страница – это динамичная, гибкая среда. По своему определению она изменчива.

Поэтому рекомендуется больше использовать em, rem и %.



Каскад

Cascading Style Sheets

Каскад — одновременное применение разных стилей к элементу, чтобы при подключении нескольких стилей браузер понимал, что использовать в итоге.

```
1  h1 {  
2      color: red;  
3  }  
4  
5  ...  
6  
7  .title {  
8      color: green;  
9  }
```

```
1  h1 {  
2      color: pink;  
3  }
```

Какой стиль применится?



Приоритеты

Руководство браузера при выборе стиля. Чем больше цифра, тем выше приоритет.

1

Стиль браузера

2

Стиль автора (в теге `<style></style>`, либо в файле `.css`)

3

Атрибута стиль (в атрибуте `style='...'`)

4

Стиль автора с добавлением `!important`
(`color: red !important;`)

5

Стиль атрибута с добавлением `!important`
(`style="color: red !important;"`)

!important

- **!important добавлен в автора стиль** — будет применяться стиль автора.
- **!important добавлен в атрибута стиль** — будет применяться стиль атрибута.
- **!important нет как в авторском стиле, так и стиле атрибута** — будет применяться стиль атрибута.
- **!important содержится в авторском стиле и стиле атрибута** — будет применяться стиль автора.

Свойство: значение !important

Типы верстки

Типы верстки

Фиксированная

Резиновая

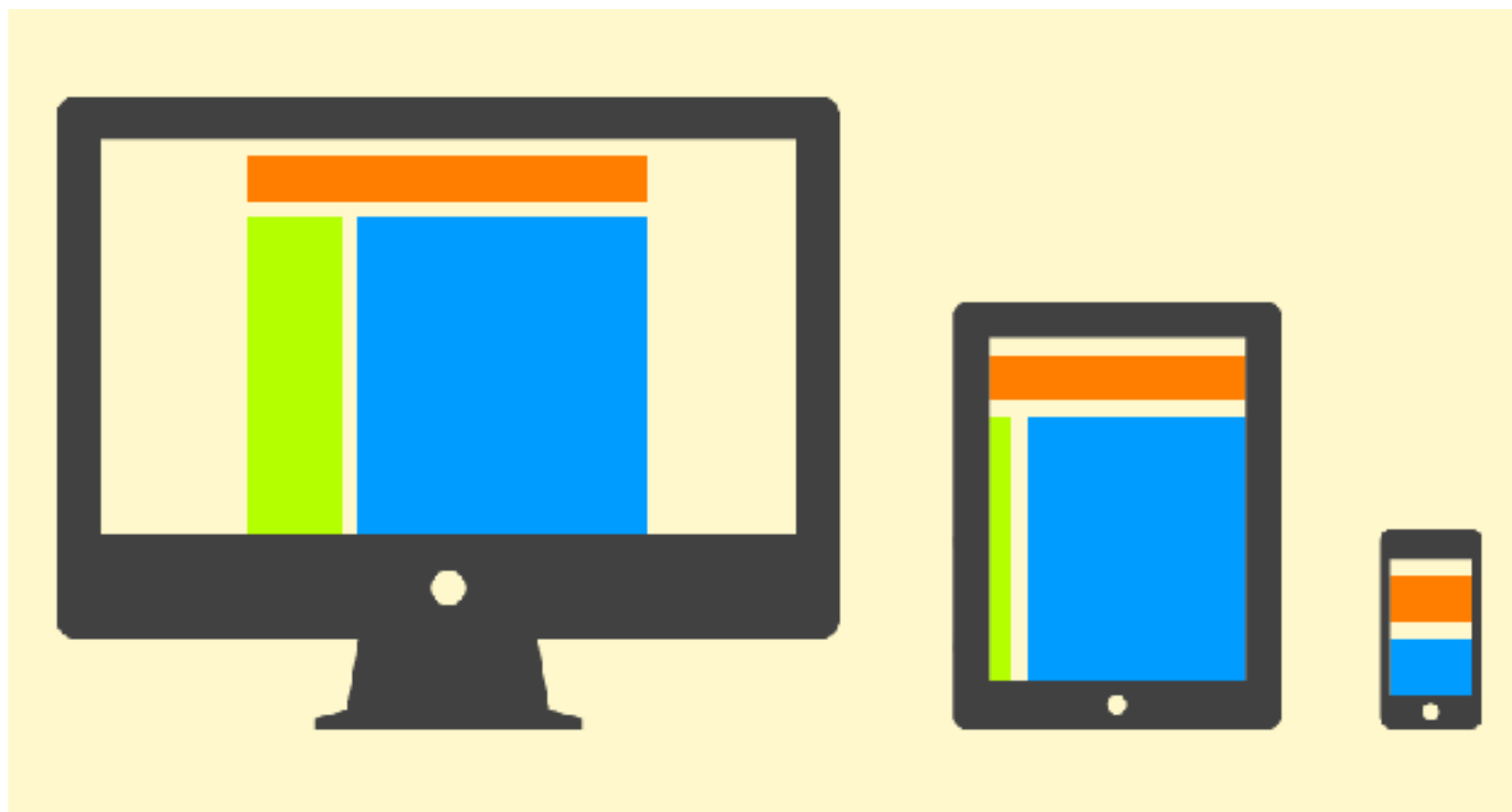
Адаптивная

Отзывчивая

Смешанная

Фиксированная верстка (в px)

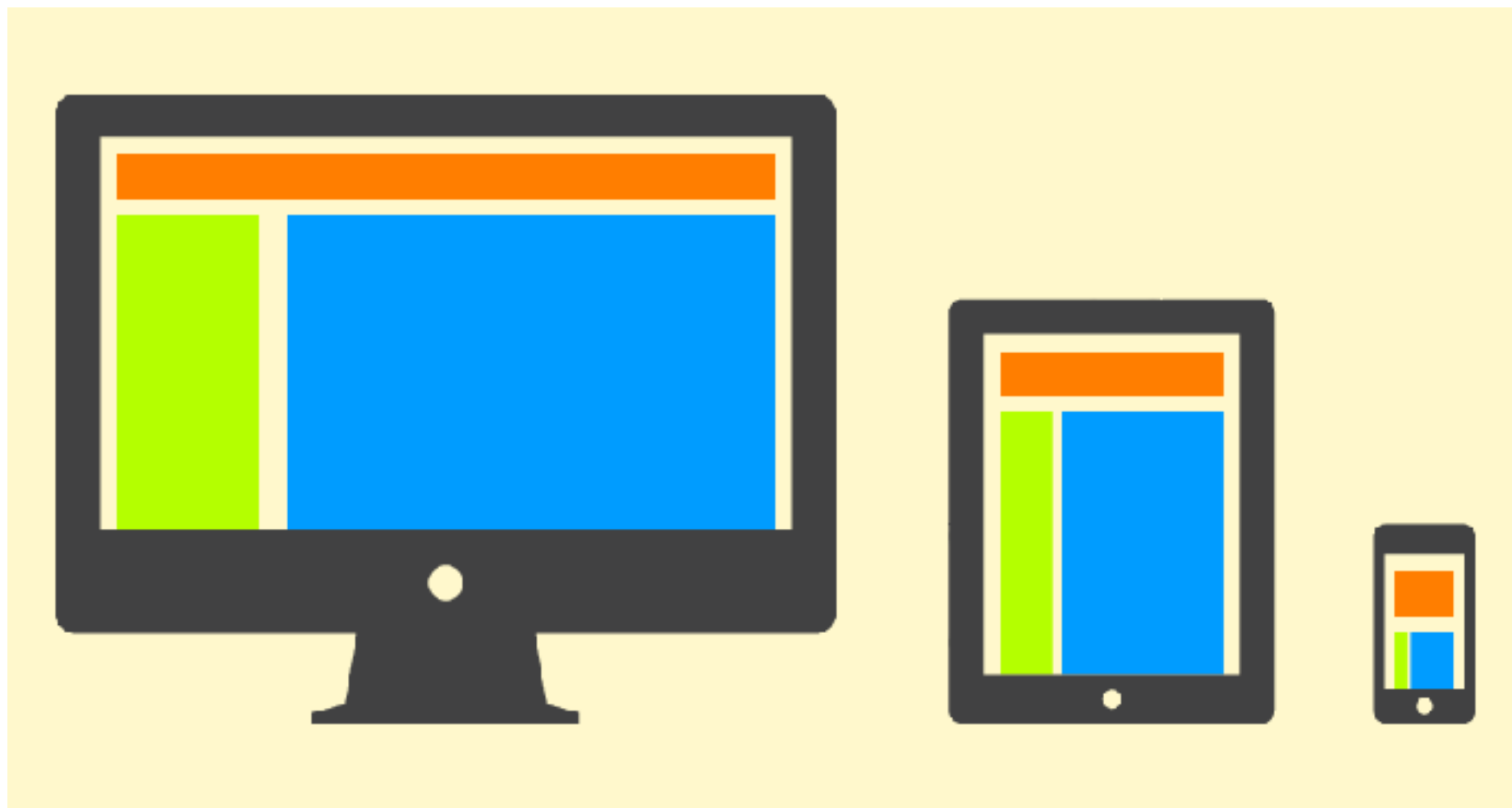
Ширина контента которой жестко задана в пикселях и не меняется в зависимости от размеров окна браузера.
(пример)



Устаревший

Резиновая верстка (в %)

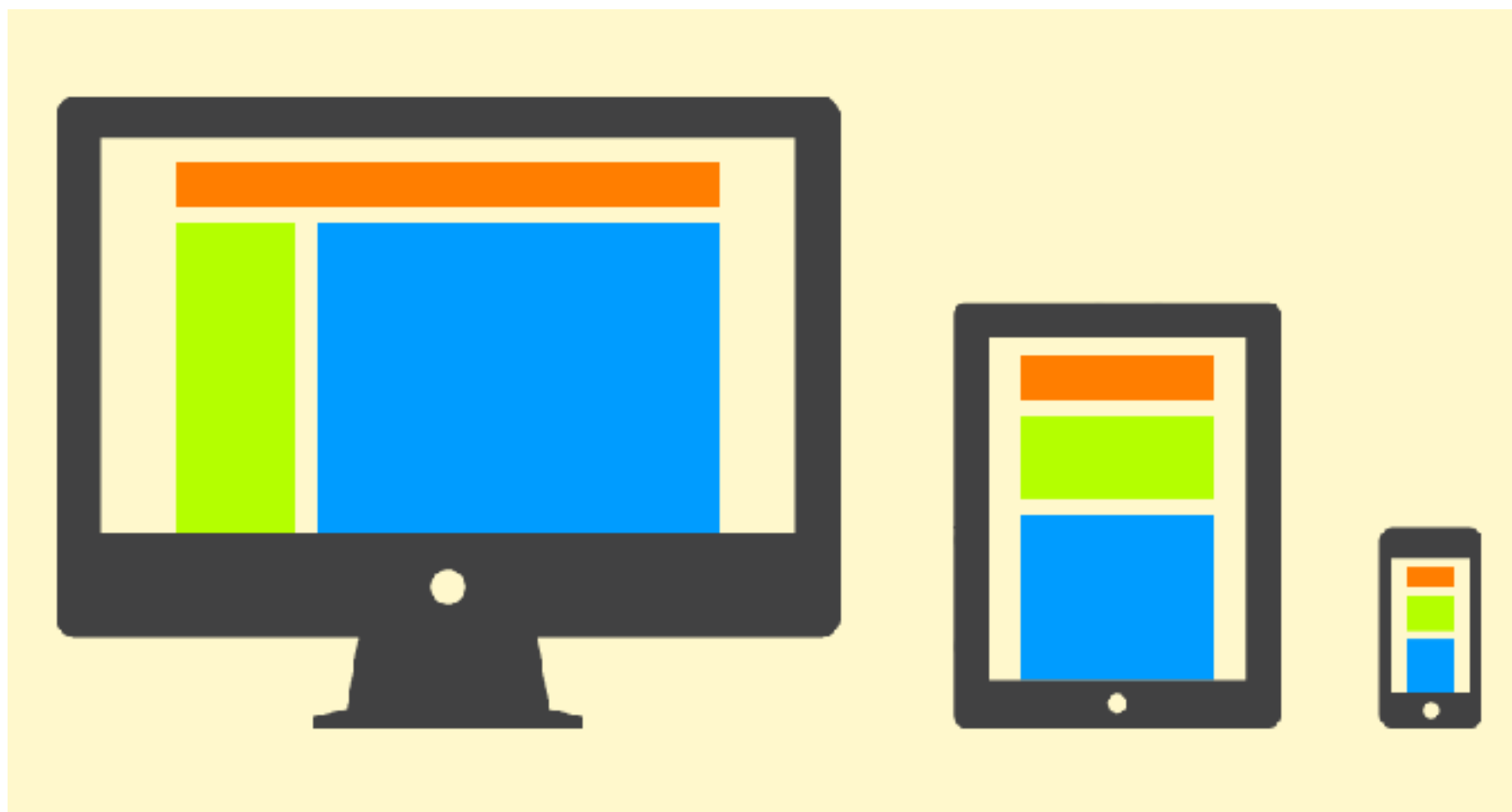
Контент принимает размер любого экрана за счет использования относительных (%) показателей для структурных элементов вместо px.



Устаревший: учитывает только один тип устройств.

Адаптивная верстка (в px, em + media-queries)

Базируется на использовании медиа запросов (англ. media queries) для адаптации контента под различные параметры экранов. Страница прыгает по контрольным точкам.



— не можем предугадать как контент будет выглядеть на всех устройствах

Media queries

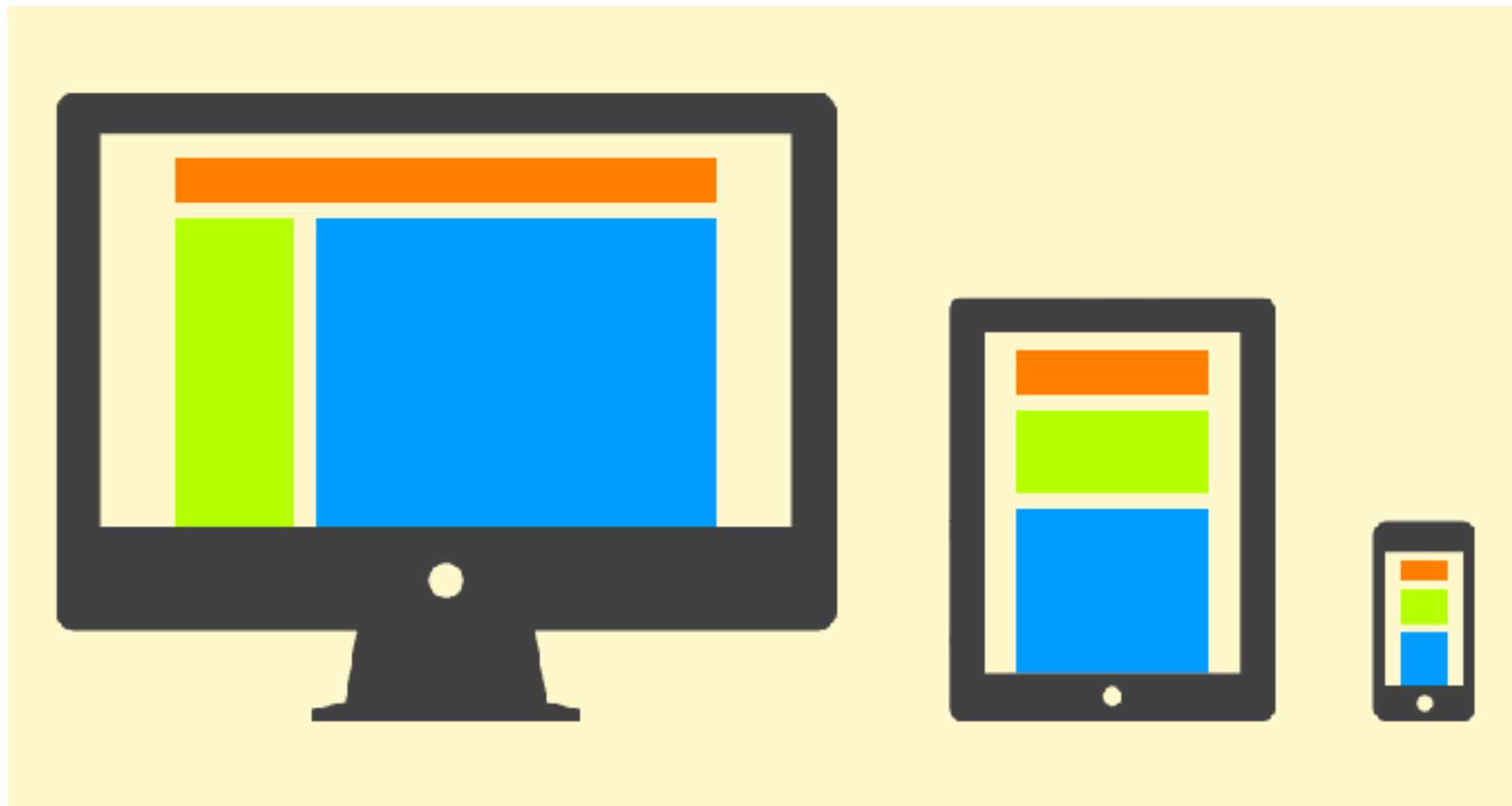
Позволяет в CSS описывать стили для разных типов устройств.

Если ширина окна браузера меньше 650 пикселей:

```
@media screen and (max-width: 650px) {}
```

Отзывчивая верстка (в % + media-queries)

В отличие от адаптивного основывается на принципе «резины», но так же как и он использует медиа запросы для приспособления контента под ширину устройства.



+ страница не «прыгает» по контрольным точкам, а плавно изменяется между ними.

Смешанная верстка (в px + % + media-queries)

Среднее между адаптивной и отзывчивой версткой страниц. Что-то плавно, что-то скачками уменьшается.



Текст и шрифты

Сброс стилей reset.css

Каждый браузер устанавливает свои значения стилей по умолчанию для различных HTML-элементов. С помощью CSS Reset мы можем аннулировать эту разницу для обеспечения кроссбраузерности стилей.

Сброс стилей reset.css

Создаем файл reset.css и копируем код стилей со страницы: [reset.css](#).

Подключаем во всех файлах .html, перед всеми СТИЛЯМИ.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Мои истории</title>
    <link rel="stylesheet" type="text/css" href="css/reset.css">
  </head>
  <body> 📺
  </body>
</html>
```

Свойства для шрифтов

- **font-family** — семейство шрифта.
- **font-style** — стиль шрифта.
- **font-weight** — вес шрифта.
- **font-size** — размер шрифта.
- **line-height** — межстрочный интервал.
- **font** — сокращенная форма записи.

font-family

В этом свойстве через запятую перечисляются шрифты в порядке приоритета.

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

Семейства шрифта

Существует 5 семейств шрифта:

serif

— шрифты с засечками

sans-serif

— шрифты без засечек

monospace

— моноширинные

cursive

— курсивные

fantasy

— декоративные

font-style

Значения:

1. `normal` — обычный (значение по умолчанию)
2. `italic` — курсив

```
p {  
  font-style: italic;  
}
```


font-weight

Это свойство устанавливает вес (насыщенность) шрифта.

1. 100
2. 200
3. 300
4. 400 или normal(по умолчанию)
5. 500
6. **600**
7. **700 или bold**
8. **800**
9. **900**

```
1  h1 {  
2      font-weight: bold;  
3  }  
4  
5  p {  
6      font-weight: 700;  
7  }
```

font-size

Размер шрифта. Основные единицы измерения: **px**, **em**, **rem**, **%**

px — жесткая единица измерения, по умолчанию ставят 14px, либо 16px (лучше избегать нечетных значений)

em — задается размер относительно родительского.

2em: размер шрифта в 2 раза больше родительского,

0,5em: размер шрифта в 2 раза меньше родительского

rem — задается размер относительно главного родителя.

% — задается размер относительно главного родителя

В %. 200% — в два раза больше родителя.

font-size

```
1  body {  
2      font-size: 16px;  
3  }  
4  
5  p {  
6      font-size: 1,5em;  
7      /*  
8          В полтора раза больше основного шрифта,  
9          т.е. 24px;  
10     */  
11 }
```

line-height

Межстрочный интервал, часто применяется вместе с `font-size`

1. **normal** — нормальное значение (по умолчанию).
2. **number** — число (≥ 0), на которое умножается текущий размер шрифта.
3. **length** — фиксированное значение в единицах измерения CSS (px, em, rem).
4. **%** — проценты от текущего размера шрифта.

line-height

```
1  body {  
2    font-size: 16px;  
3  }  
4  
5  p {  
6    font-size: 1.5em;  
7    line-height: 1.2; // 1.5em * 1.2 = 1.8em  
8  }
```

font

Это сокращённая форма записи свойств шрифта.



Свойства для текста

- **text-align** — выравнивание текста
- **letter-spacing/word-spacing**
— интервал между буквами.
- **text-decoration** — оформление текста.
- **text-transform** — трансформация текста.
- **text-shadow** — добавление тени к тексту (CSS3)

text-align

Выравнивание текста

1. `left` — по левому краю (значение по умолчанию)
2. `right` — по правому краю
3. `center` — по центру
4. `justify` — по ширине

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


letter-spacing/word-spacing

Интервал между буквами/словами

1. `normal` — нормальное значение (по умолчанию)
2. `length` — значение в единицах измерения CSS (допускаются отрицательные значения)

```
p {  
  letter-spacing: -1px;  
  word-spacing: 10px;  
}
```

text-decoration

Оформление текста

1. `underline` — подчёркнутый текст
2. `overline` — линия над текстом
3. `line-through` — перечёркнутый текст
4. `none` — отмена всех эффектов (значение по умолчанию)

```
a {  
  text-decoration: none;  
}
```

text-transform

Трансформация текста

1. `none` — нет трансформации (по умолчанию)
2. `uppercase` — преобразование всех строчных СИМВОЛОВ в заглавные
3. `lowercase` — преобразование всех заглавных СИМВОЛОВ в строчные
4. `capitalize` — преобразование первой буквы каждого слова в заглавную

```
h1 {  
  text-transform: uppercase;  
}
```

text-shadow

Добавление тени к тексту (CSS3)

1-3 обычно используются ед. измерения px

1. h-shadow — смещение тени по горизонтали
2. v-shadow — смещение тени по вертикали
3. blur — радиус размытия тени
4. color — цвет тени
5. none — нет тени (значение по умолчанию)

```
p {  
  text-shadow: 1px 1px 10px #000000;  
}
```

Таблица цветов

Кастомные шрифты и способы их подключения в CSS

Frontender Magazine

Адаптивно-отзывчивый: разбираемся в терминологии

Анонс [веб-сайта](#), позволяющего сравнить на примере простой страницы фиксированный, резиновый, адаптивный и отзывчивый типы макетов, вызвал большой резонанс в веб-сообществе. Несмотря на то, сколько всего уже было сказано об отзывчивом и адаптивном веб-дизайне, энтропия непонимания принципиальной разницы подходов в веб-разработке имеющих подобные названия только возрастает.

Я постараюсь разложить все по полочкам и объяснить принципиальную разницу между типами макетов и подходами к веб-разработке.

Типы HTML-макетов

Под макетом в данной статье подразумевается не изображение в формате PSD, а уже готовая разметка HTML-страницы.

Фиксированные макеты

Шрифт является неотъемлемой частью веб-дизайна, придаёт сайту выразительность и узнаваемость, выражает характерный стиль сайта!

Способы подключения

1. Скачиваем шрифт в формате .otf или .ttf. Для максимальной совместимости со всеми браузерами с помощью генератора шрифтов получаем woff, woff2, svg и **подключаем через CSS**.
2. Подключаем шрифт онлайн.

Подключение собственного шрифта

The screenshot shows the FontSquirrel website's 'Webfont Generator' tool. The browser address bar displays 'www.fontsquirrel.com/tools/webfont-generator'. The site has a purple header with the 'FONT SQUIRREL' logo and navigation links for 'Login', 'FAQ', and 'Contact'. A search bar is present with the text 'SEARCH FONTS & FOUNDRIES'. Below this is a navigation menu with links: 'Hot', 'Recent', 'Almost Free', 'Shop', 'Generator' (highlighted), 'Font Identifier', 'Font Talk', and 'Blog'.

Webfont Generator

Usage: Click the "Upload Fonts" button, check the agreement and download your fonts. If you need more fine-grain control, choose the **Expert** option.

UPLOAD FONTS ↑

HelveticaNeueCyr Roman	otf	232 glyphs	25 KB	✕
------------------------	-----	------------	-------	---

☐ **BASIC**
Straight conversion with minimal processing.

☐ **OPTIMAL**
Recommended settings for performance and speed.

☒ **EXPERT...**
You decide how best to optimize your fonts.

Font Formats:

☐ TrueType

☒ WOFF

☒ EOT Lite

☐ EOT Compressed

☒ SVG

NEW TO FONTS?

Download our free beginners guide to fonts.

FONT LISTS

Recently Added

Most Popular

Hot Today

Hot Web Fonts

Almost Free

Tags

Classifications

Languages

Foundries

FIND FONTS

Font Squirrel — настройки

Expert

Font Formats: WOFF, WOFF2, EOT Lite, SVG

X-height Matching: Arial

Subsetting: Custom Subsetting...

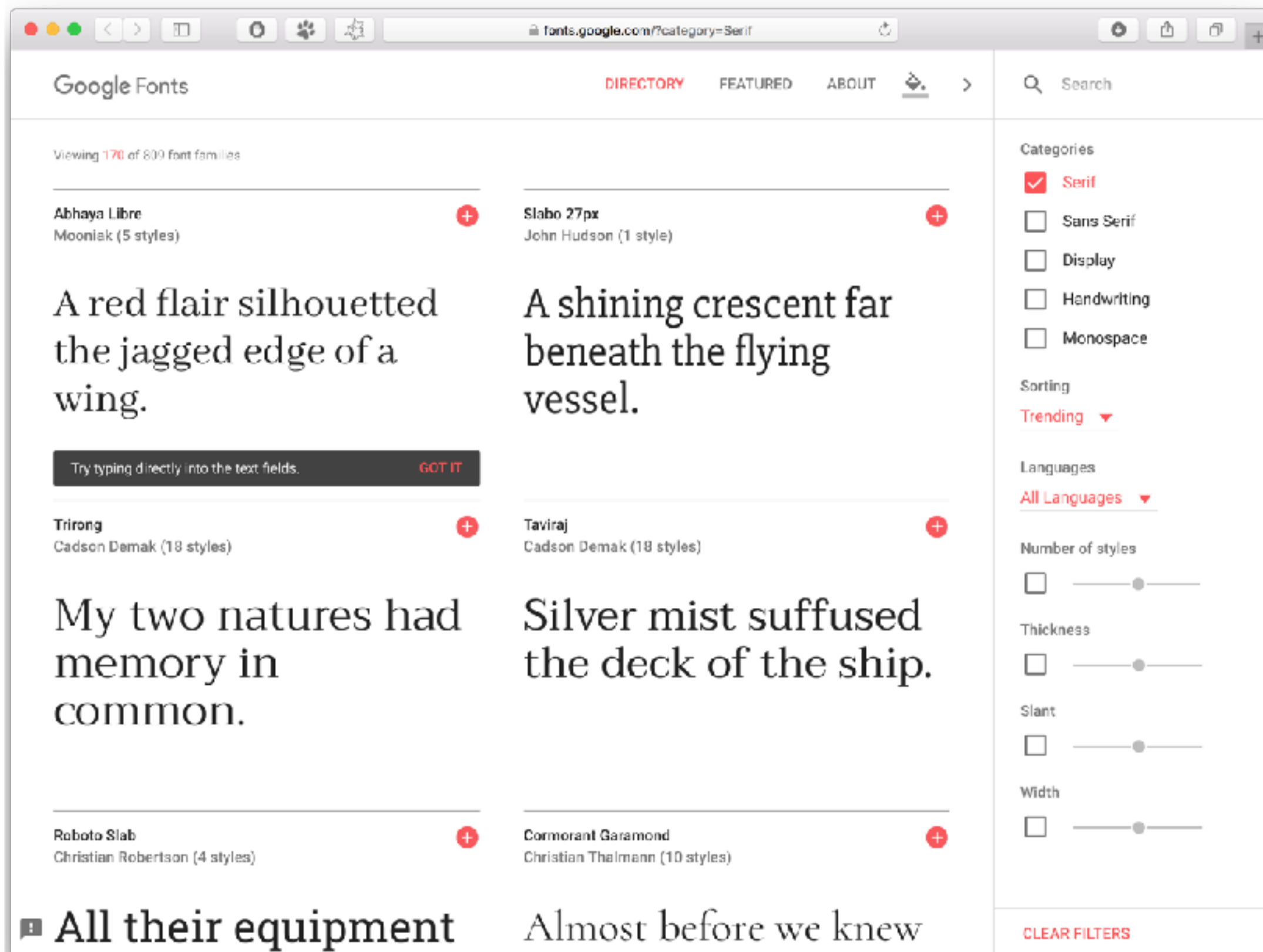
Language: Cyrillic, English

Shortcuts: Remember my settings

Agreement: Yes, the fonts I'm uploading are legally eligible for web embedding.

Скачиваем и подключаем к странице.

Подключение через google fonts



Google Fonts — настройки

Поиск

Languages: Cyrillic или Latin

Нажимаем на + и в embed можно посмотреть как его подключить. Перед этим можно его кастомизировать (customize), выбрать различные начертания шрифта.

Подключаем через link в html.

Варианты подключения

HTML:

```
main.css  root.html
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Мои истории</title>
6      <link href="https://fonts.googleapis.com/
      css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i&subset=cyrillic"
      rel="stylesheet">
7    </head>
8    <body>
24   </body>
25 </html>
26
```

CSS:

```
main.css  root.html
1  body {
2    font: normal 14px/1.2 'Open Sans', Arial;
3  }
4
```

Практика

Начинаем работу!

Street life



Домашнее задание

Домашнее задание

1. CSS курс на html academy (До урока свойства оформления текста включительно).

Вопросы?



Автор курса: Елена Иванова
lessons4.liveldi.ru