

Организационные моменты



Отключите микрофоны



Не забывайте включать микрофон, когда задаете вопрос



Не стесняйтесь задавать вопросы



По возможности включайте камеры



- Роман Дмитерко
- Фронтенд-разработчик в hh.ru
- Занимаюсь web-разработкой с 2012 года



План

1. Лекция
 - HTML
 - CSS
 - CSS-свойства
 - БЭМ
 - Адаптивная верстка
 - CSS-препроцессоры
2. Практика – часть 1
3. Практика – часть 2
4. Домашнее задание

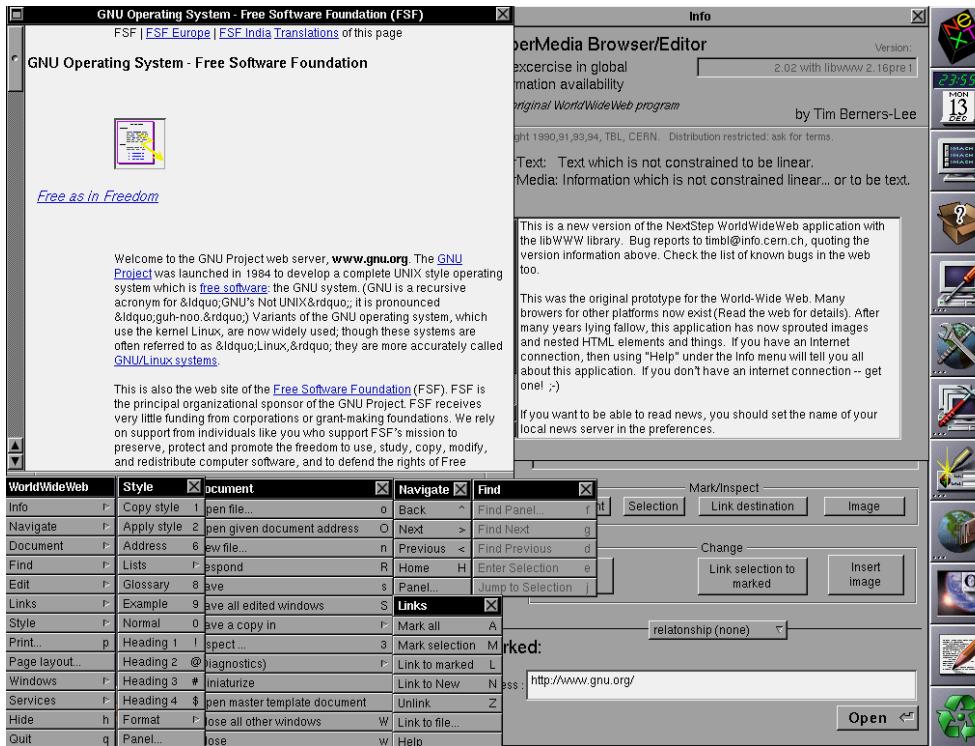
HTML

Язык разметки документов, который определяет, как и какие элементы должны располагаться на веб-странице.

Был разработан Тимом Бернерсом-Ли приблизительно в 1986–1991 годах на основе SGML, а [первая в истории страница](#) появилась в сети CERN

- Простой в работе для тех, кто не ничего не понимает в верстке
- Однаковое отображение на любом устройстве

World Wide Web



HTML дескриптор (тег)

Anatomy of an HTML element



[Список HTML-тегов](#)

Атрибуты

- позволяют изменять свойства и поведение элемента, для которого они заданы
- Обязательные
- Необязательные
- Пользовательские

С чего начинается HTML-документ?

DOCTYPE

Указывает версию спецификации HTML, которой соответствует документ.

Сегодня он нужен только для того, чтобы браузер не переключался в режим совместимости.

Единственный актуальный doctype - `<!DOCTYPE html>`

С чего начинается HTML-документ?

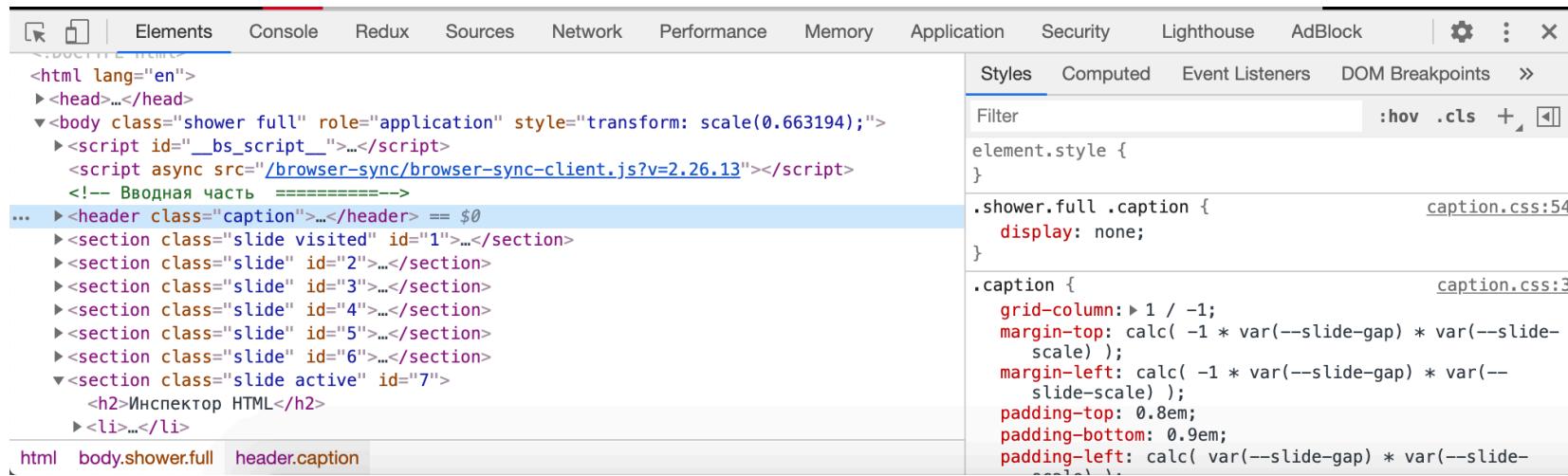
<html></html> - контейнер для всего документа. Имеет атрибут lang, который позволит браузеру понять как отображать контент

<head></head> - метаданные страницы: заголовок, иконка, стили, скрипты, информация для шаринга

<body></body> - содержит весь контент, который появится на странице: текст, изображения, теги.

Инспектор HTML

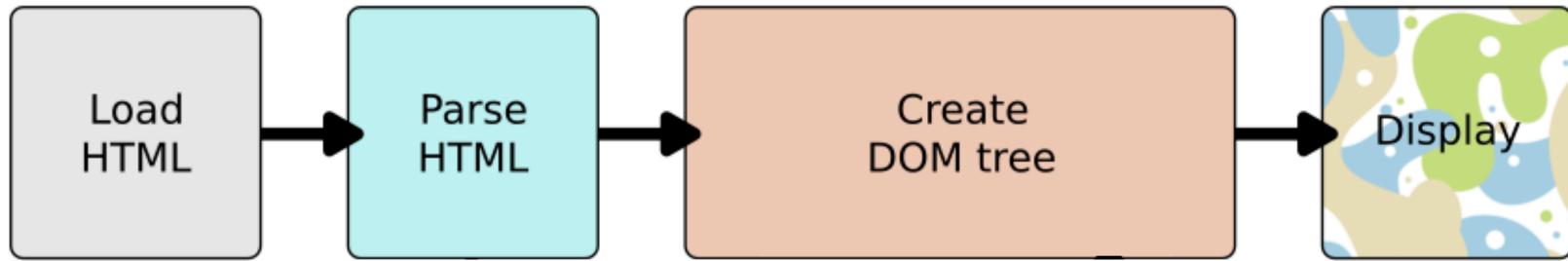
- Правой кнопкой → Inspect Element / Просмотреть код
- Mac → Cmd + Alt + I или F12
- Windows, Ubuntu → Ctrl + Shift + I или F12



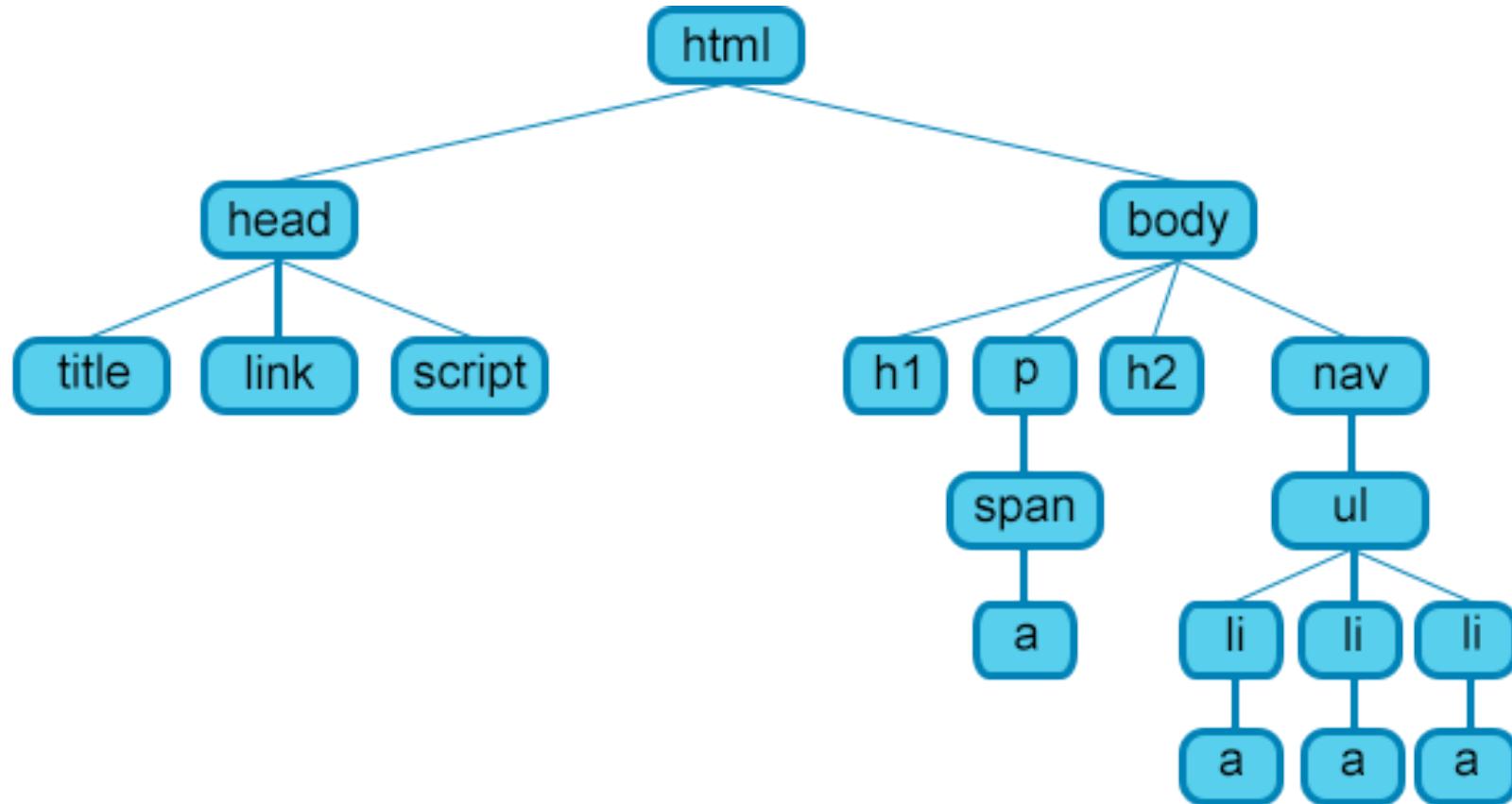
Спецификация HTML

- Спецификация HTML – html.spec.whatwg.org
- Список тегов – webref.ru
- Моя любимая документация на русском языке – doka.guide

Схема работы браузера



DOM (Document Object Model)



Типы HTML-тегов

- Строчные - располагаются на одной строке
- Блочные - располагаются с новой строки
- Строчно-блочные - располагаются на одной строке, но имеют поведение блочных

Нельзя вкладывать блочные теги в строчные.

В строчно-блочные - можно

Валидная верстка

- Документ имеет doctype
- Содержимое документа внутри тега html, контент внутри body
- Все закрывающиеся теги закрыты
- Теги имеют только допустимые атрибуты

Как проверить?

- [Онлайн-валидатор](#)
- Плагин для редактора

Семантическая верстка - что это?

В общем случае - использование тегов по назначению:

для кнопки - `button`, для таблиц - `table`, заголовков - `h1-h6`

Зачем?

- Доступность для скринридеров
- Отображение при отключенных стилях и скриптах
- Поисковая оптимизация
- Удобство поддержки

Рекомендации по написанию HTML

- Страйте использовать теги по назначению
- Учитывайте доступность интерфейсов
- Проверяйте html валидаторами

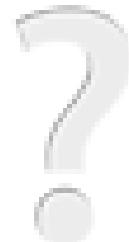


Что почитать



- Про теги – [webref.ru](#) или [htmlbook.ru](#)
- Про HTML – [doka.guide](#)
- Про accessibility [Статья на MDN](#)

Вопросы



CSS

CSS

– появившийся в 1996 году, формат описания визуальных стилей, который позволяет применять визуальный стиль к элементам структурированных документов (например, документам HTML и XML)

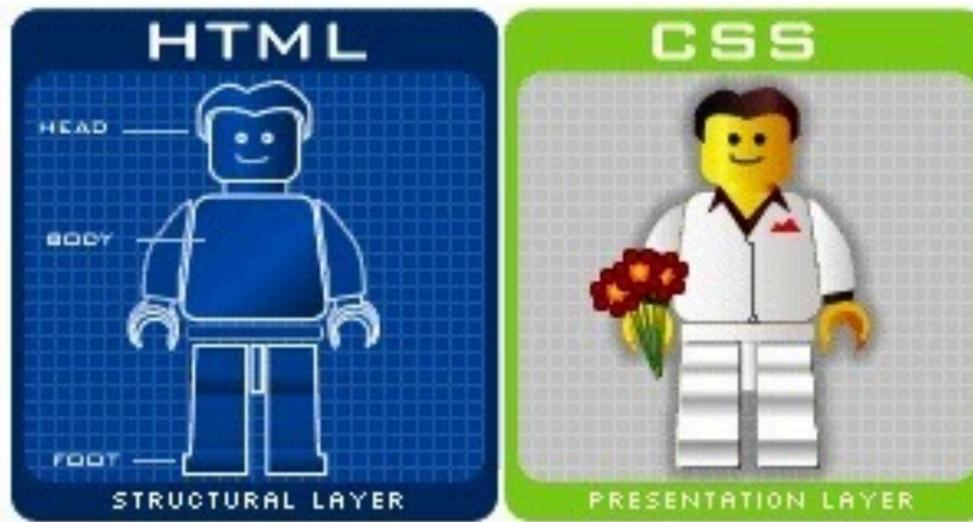
Почему "cascading"?

Браузер загружает все стили, описанные на странице и применяет их "сверху-вниз", добавляя или заменяя свойства элемента

Для чего?

- Идея была в том, что пользователь и разработчик могут управлять отображением
- Понятный алгоритм определения стилей элементов - каскадность
- Возможности стилизации, которых нет в HTML
- Различная стилизация одного HTML-документа

Разделение структуры и внешнего вида



Как это выглядит

```
.custom-slide-code h2 {  
    margin: 10px;  
}
```

```
.custom-subtitle {  
    font-size: 50%;  
    color: #999;  
}
```

```
.custom-subtitle_white {  
    color: #fff;  
}
```

Синтаксис CSS-правила

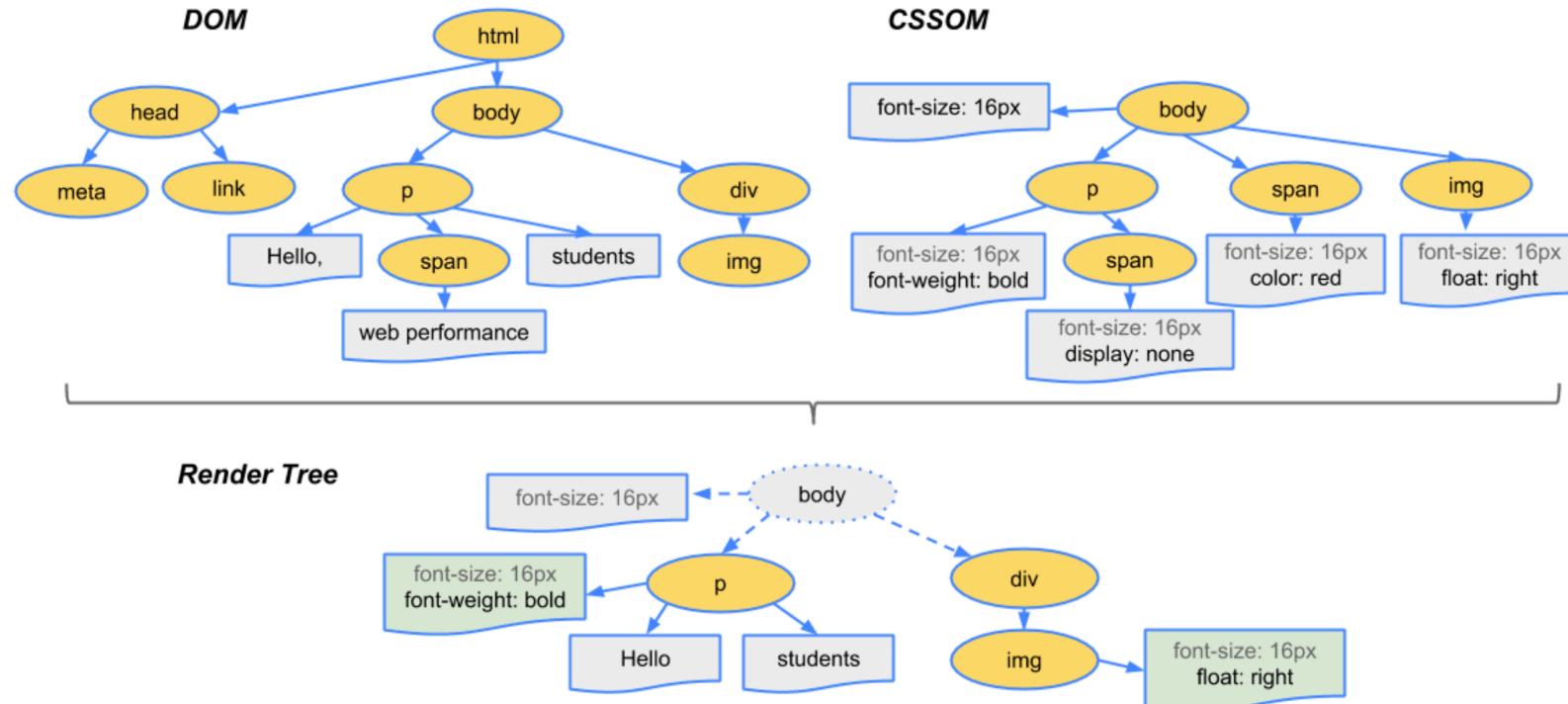


Как применить?

- Подключить внутри HTML `<link rel="stylesheet" href="styles/main.css">`
- Описание внутри тега `<style> .class {...} </style>`
- Применение инлайн-стилей

```
<div style="width: 100px;" height= "200px">
```


Схема работы браузера



Типы селекторов

- Базовые селекторы
- Псевдоклассы
- Псевдоэлементы
- Комбинаторы

Наследуемые стили:

color, font, letter-spacing, line-height, list-style,
text-align, text-indent, text-transform, visibility,
white-space и word-spacing

Ненаследуемые стили:

background, border, display, float и clear, height и width, margin, min-max-height и -width, outline, overflow, padding, position, text-decoration, vertical-align и z-index

Как применяются CSS-свойства

1. Могут быть указаны в CSS для самого элемента
2. Наследуются от родительского элемента
3. Браузер устанавливает некоторые свойства по-умолчанию
4. К одному элементу могут применяться стили из разных мест и/или от разных селекторов
 - Учитывается специфичность селекторов
 - Стили, расположенные в коде ниже, переопределяют стили, расположенные выше

Расчет специфичности селекторов

Селектор	Вес
тег, псевдо-элемент(span, ::after)	0 0 0 1
класс, псевдокласс, атрибут (.header, :hover)	0 0 1 0
идентификатор (#footer)	0 1 0 0
инлайн-стили (style="color: red;")	1 0 0 0

[Гайд по специфичности селекторов](#)

[Расчет специфичности](#)

!important

```
.text #name div.red.small { ... }
```

А что плохого?

- сложно читать
- сложно сравнивать специфичность
- сложно увеличивать
и уменьшать специфичность
- усложняет поиск
селекторов браузерами



Что же делать?

Стараться использовать только селекторы классов

```
.text #name div.red.small { ... } ->  
.specific-element { ... }
```

Проблема: сложно придумывать уникальные названия CSS-классов

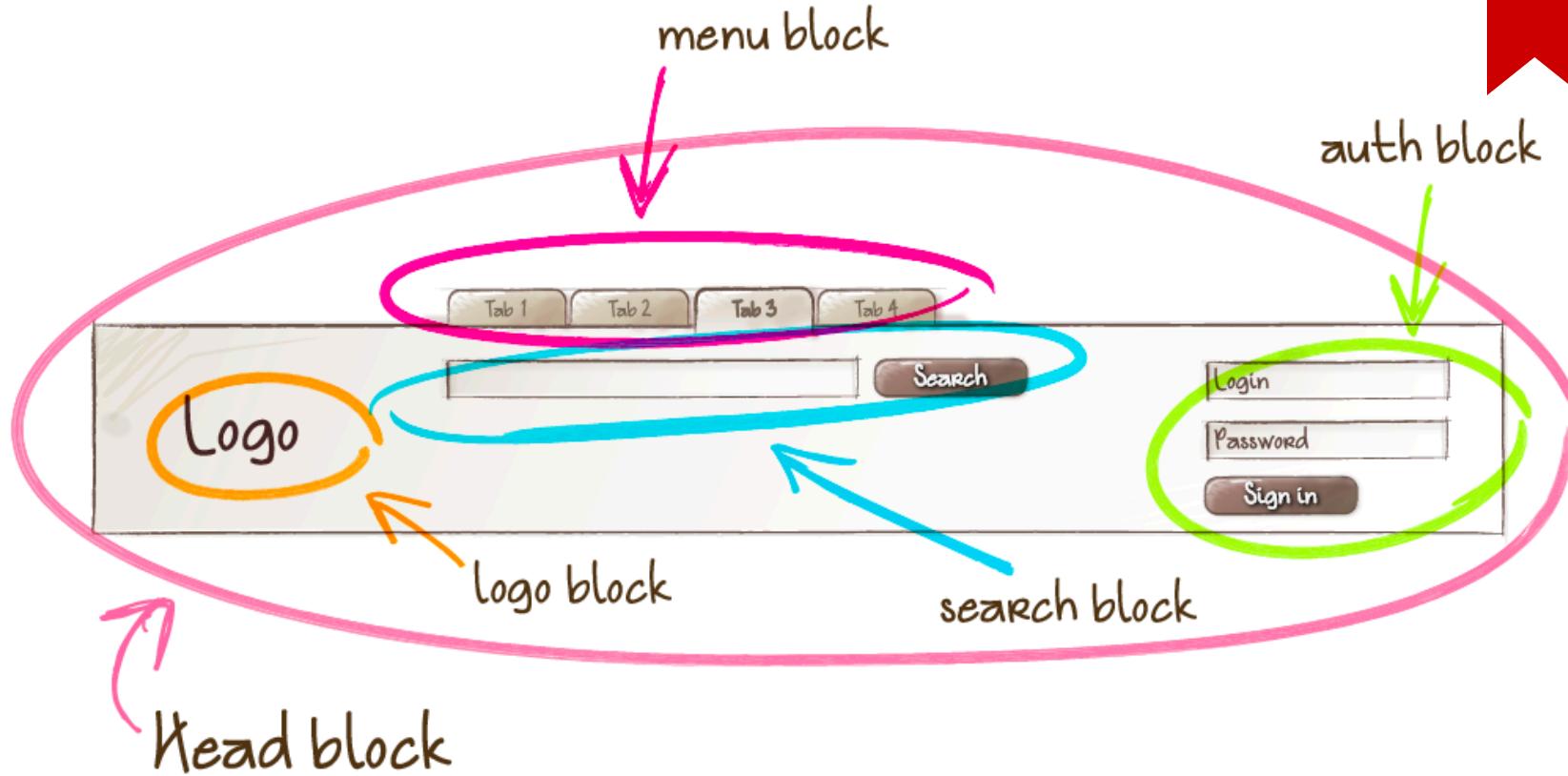
БЭМ (Блок, Элемент, Модификатор)

Компонентный подход к веб-разработке. В его основе лежит принцип разделения интерфейса на независимые блоки

ru.bem.info

[Основные понятия](#)

[Соглашения по именованию](#)



БЛОК

- Функционально независимый компонент страницы
- Название – английское слово или словосочетание

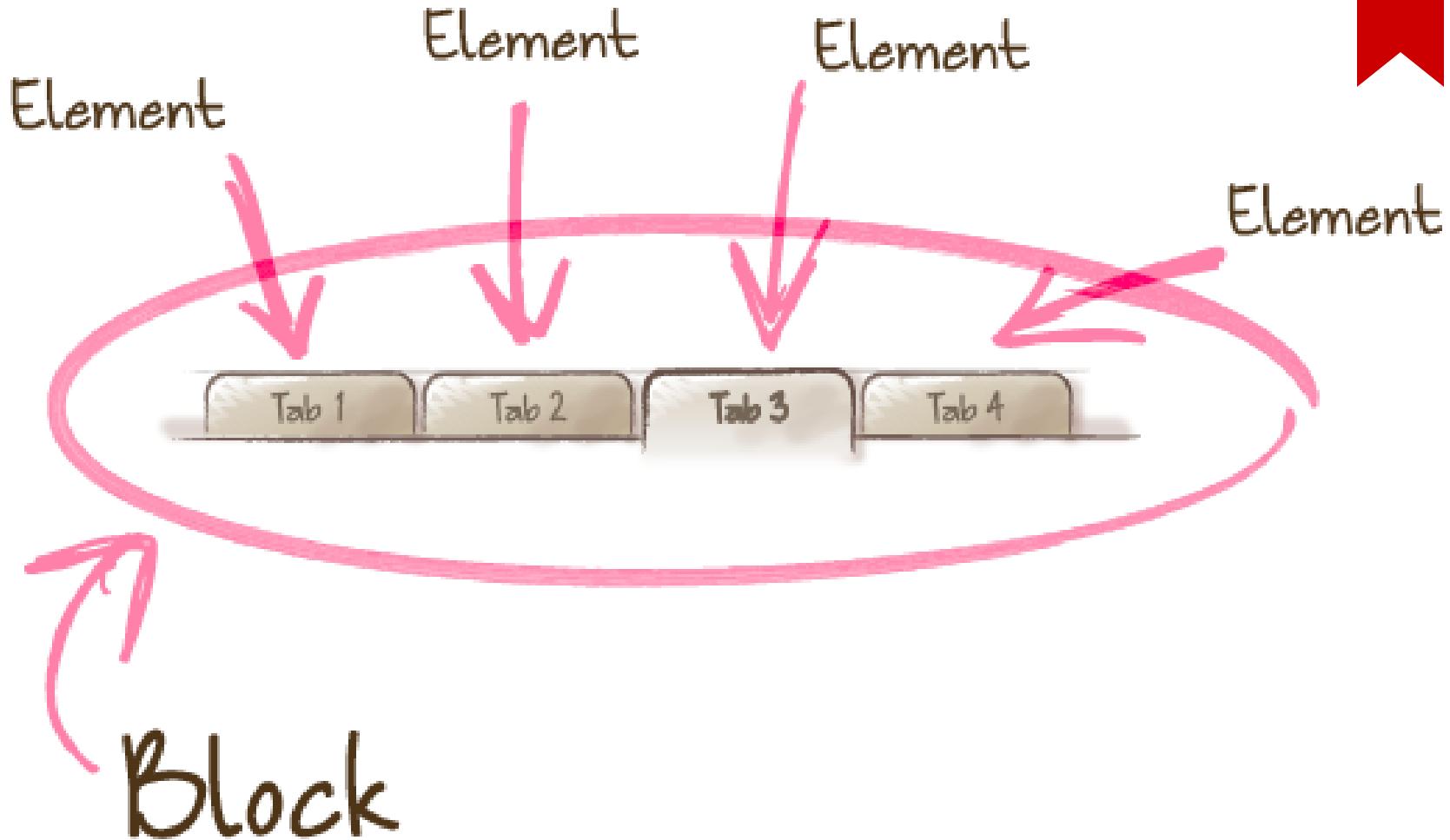
.header, .logo, .main-menu

- Название характеризует смысл, а не состояние

.red-text

.error

- Можно вкладывать друг в друга



Элемент

- Составная часть блока, которая не может использоваться без него
- Название соответствует схеме

`.имя-блока__имя-элемента`, например `.main-menu__item`

- Имя элемента характеризует смысл, а не состояние
- Элементы можно вкладывать друг в друга.
- Элемент всегда часть блока, а не другого элемента

`.block__elem1__elem2`

`.block__elem1` и `.block__elem2`



Page Title



Модификатор

- Определяет внешний вид, состояние или поведение блока или элемента
- Название соответствует схеме
 - .имя-блока_имя-модификатора или
 - .имя-блока__имя-элемента_имя-модификатора
 - .main-menu__item_active
- Имя модификатора характеризует внешний вид
- Нельзя использовать самостоятельно

Правила именования CSS-классов

- Блок

```
.block-name
```

- Элемент

```
.block-name__element-name
```

- Модификатор

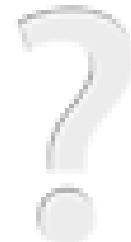
```
.block-name_mod-name
```

```
.block-name__element-name_mod-name
```


Преимущества

- Не повышается специфичность селекторов
- Блоки легко переиспользовать
- Легко масштабировать верстку

Вопросы

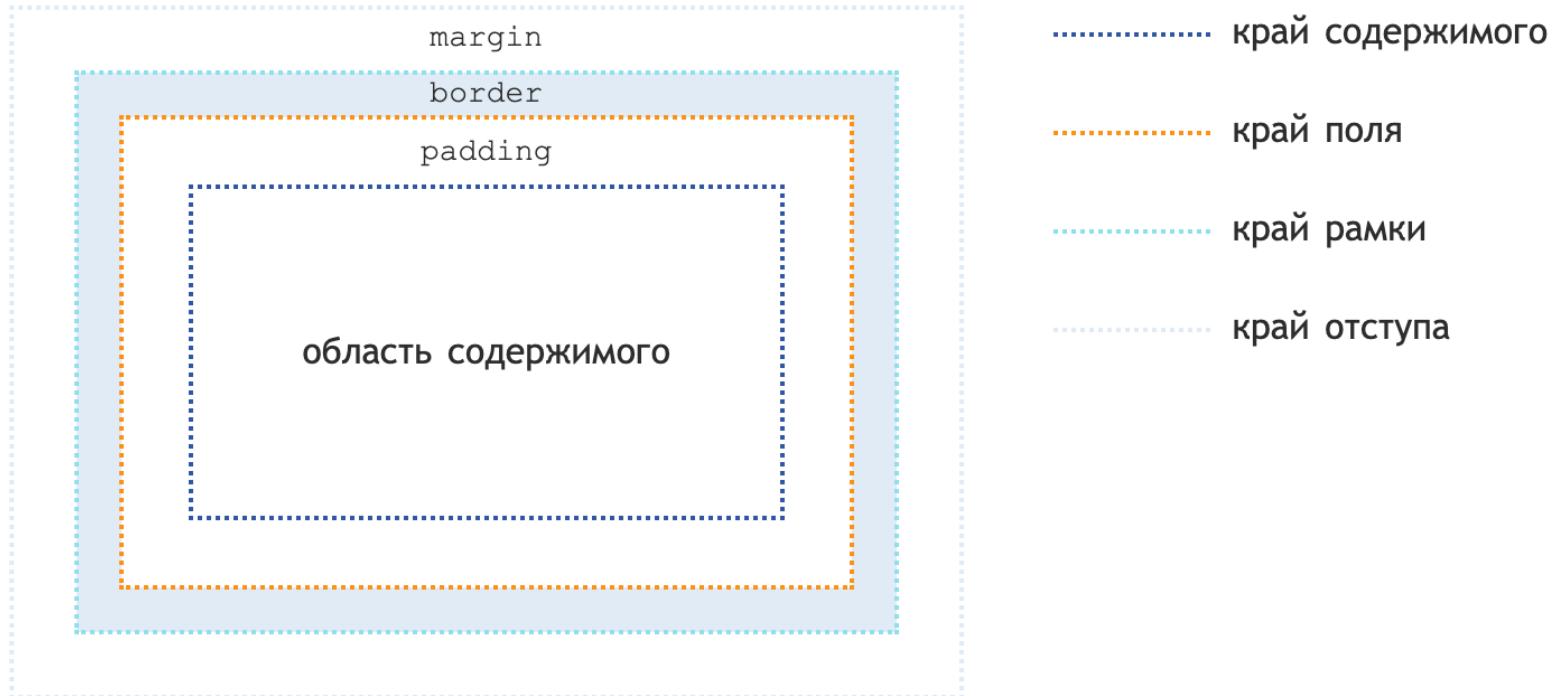


Перерыв



CSS свойства

Блочная модель



Свойства

box-sizing: <'border-box' | 'context-box'>

border: <ширина в px> <solid | dotted | dashed> <цвет>

margin -top, -right, -bottom, -left: <px | em | rem>

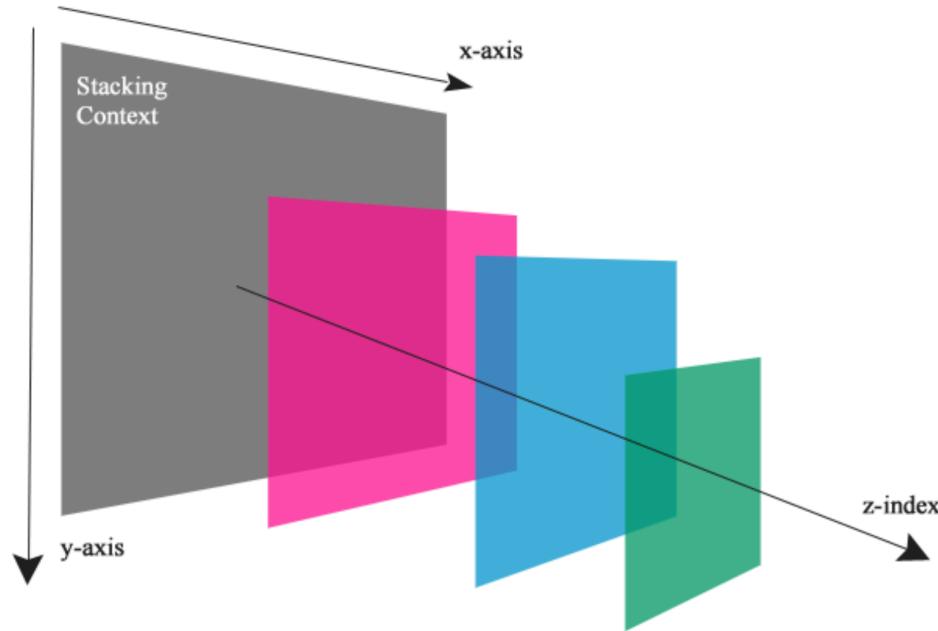
padding -top, -right, -bottom, -left: <px | em | rem>

Поток документа

Position

- **static** (*по умолчанию*)
- **relative** - отступ от того места, где элемент должен быть отрисован
- **absolute** - позиция внутри ближайшего родителя с relative или absolute
- **fixed** - позиция внутри документа
- **sticky** - позиция, меняющаяся при скролле родителя

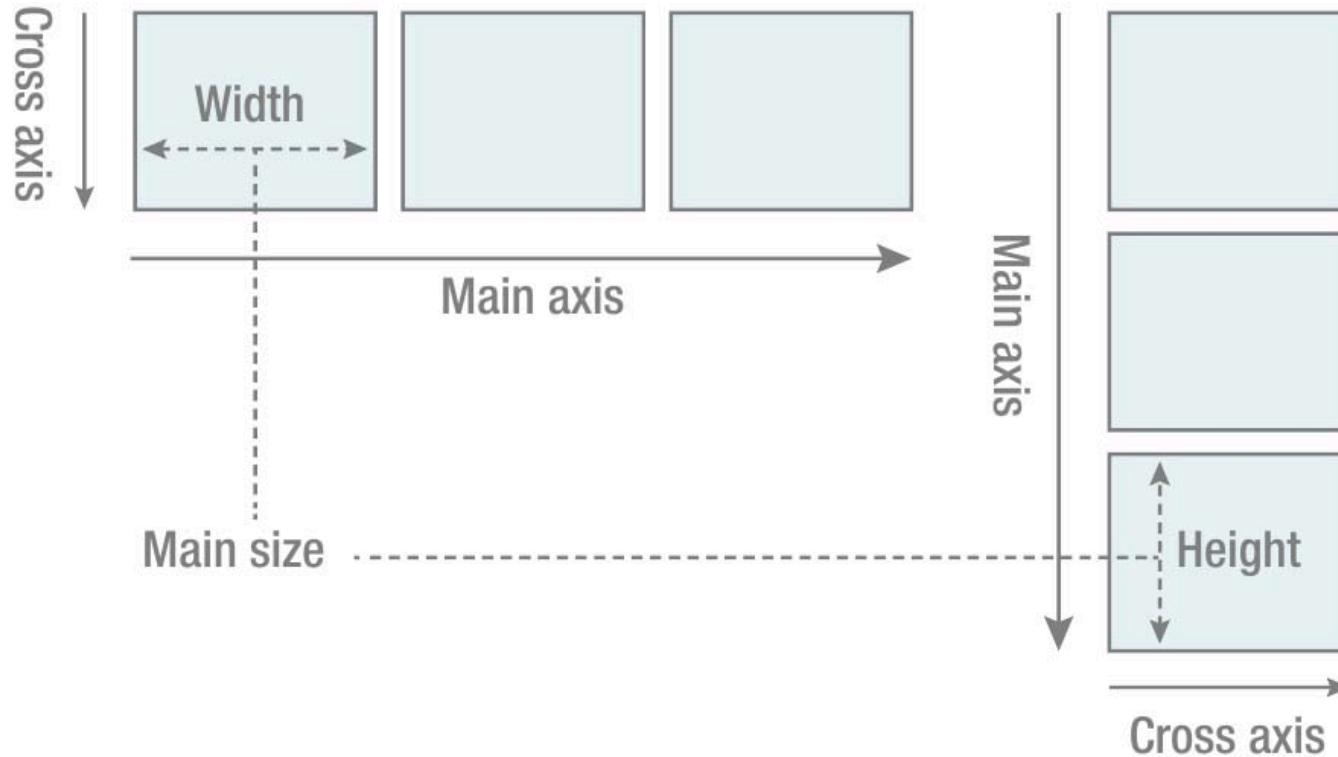
Z-index



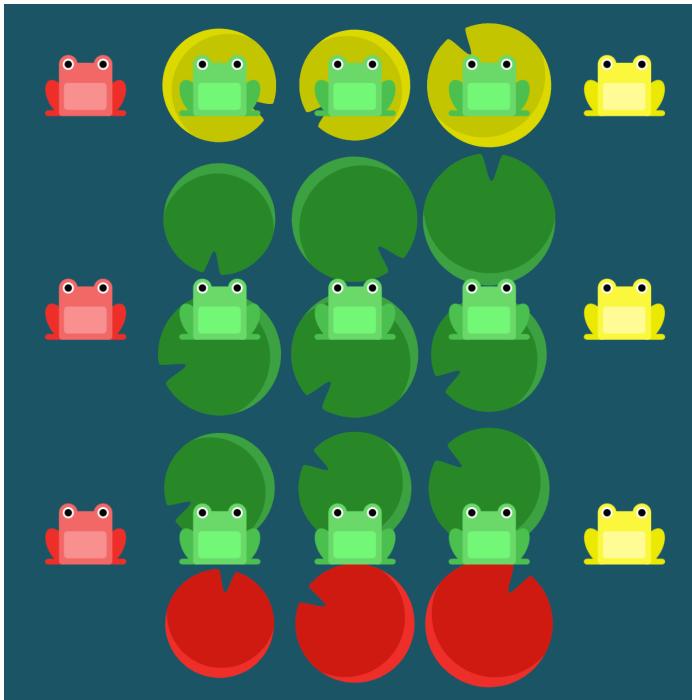
Отображение элемента: display

- `block` - блочный элемент
- `inline` - строчный(инлайн) элемент
- `inline-block` - блочно-строчный элемент
- `flex` - ?
- `grid` - ??

Flexbox



flexbox



[A Complete Guide to Flexbox](#)

[flexboxfroggy.com](#)

[Все про flexbox на русском языке](#)

Grid

Похожи на flexbox, но работают с двумя осями сразу и позволяют описать сетку

Можно сравнить с таблицами

[A Complete Guide to Grid](#)

[Все про грид на русском языке](#)

Часто используемые CSS-свойства

- font-size: <num px/em>
- width, height: <num px/ % /em>
- color: <hex | rgb | rgba>
- top, right, left, bottom: <num px>
- background: <url | hex| rgba>

Единицы измерения

- px - самый простая и понятная величина
- % - от размеров родительского элемента
- vh,vw - процент высоты или ширины вьюпорта (экрана)
- em - вычисляется относительно размера шрифта родительского элемента
- rem - вычисляется относительно размера шрифта корневого элемента
(html)

[О других единицах измерения](#)

Анимации

Простые анимации

transition - плавное изменение свойства

Гибкие анимации

- Описывают зависимость CSS-свойств от времени
- Требуют больше памяти при рендеринге и могут быть посчитаны на GPU



```
@keyframes <название анимации> {  
    from {  
        /* css-свойства в первом кадре */  
        width: 50px;  
    }  
    to {  
        /* css-свойства в последнем кадре */  
        width: 200px;  
    }  
}
```

```
.circle {  
    animation-name: circle-to-square; // название анимации  
    animation-duration: 5s; // продолжительность  
    animation-iteration-count: infinite; // кол-во повторений  
    animation-timing-function: linear // функция скорости анимации  
}
```

Еще про свойства анимаций

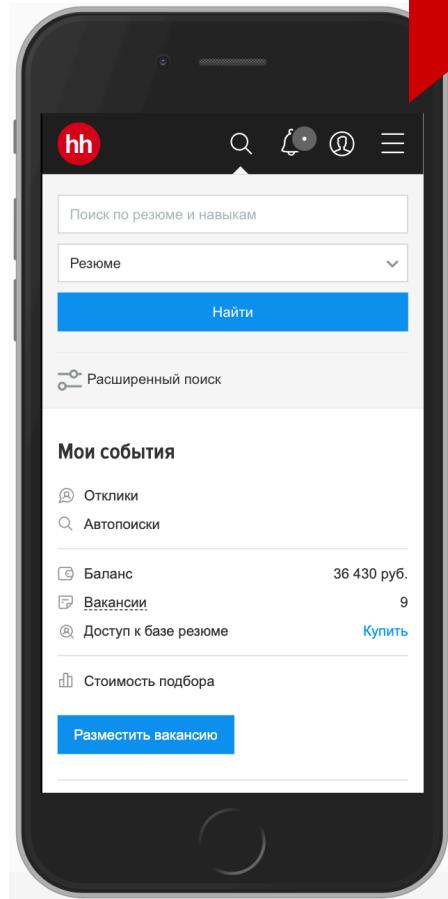
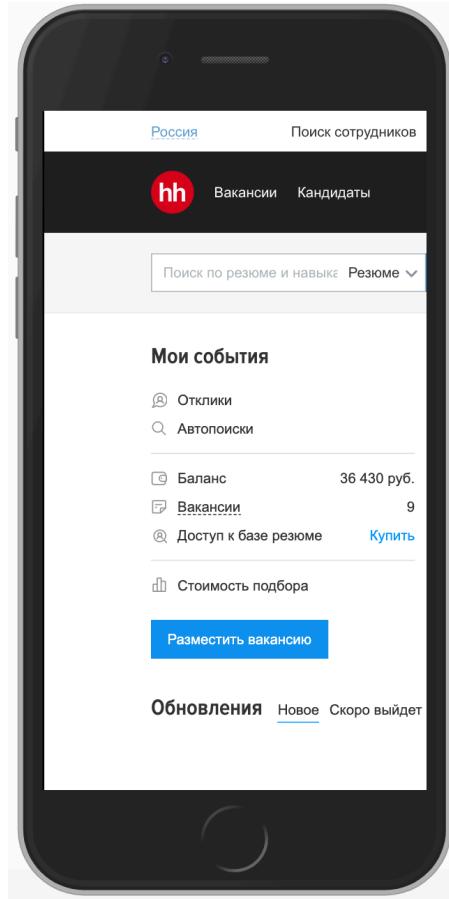
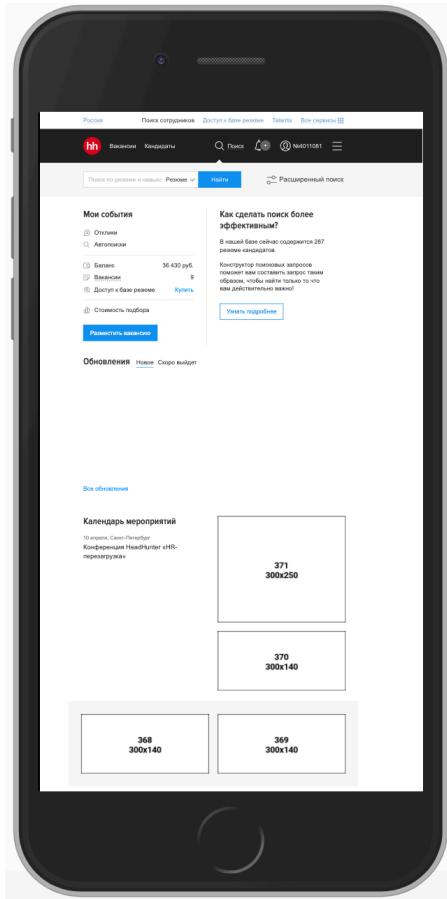
Визуализация скорости анимации с помощью кривых Безье

Отличная статья о выделении слоя анимации и рендеринге на GPU

Переменные

Адаптивная

верстка





Media queries (Медиа запросы)

CSS правила, которые позволяют применять разные CSS-стили, для разных устройств по типу отображения (например: для принтера или монитора), или для конкретных характеристик устройства (например: ширины окна просмотра, ориентация устройства)

```
@media screen and (max-width: 480px) {  
    .block {  
        background-color: #2196f3;  
        color: #fff;  
    }  
    ...  
}
```

[Про разные виды медиа запросов](#)

Примеры адаптивной верстки

[hh.ru](#)

[dzen.ru](#)

[чемпионат.ком](#)

[DNS](#)

[Хабр](#)

[The Times](#)

90

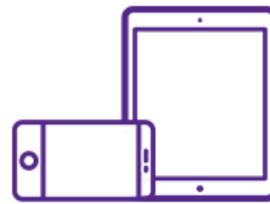


Breakpoints



0-480

Smaller
smartphones



481-768

Tablets & larger
smartphones



769-1279

Laptops, larger tablets
in landscape, and small
desktops



1280+

Larger desktops
and monitors

CSS пиксель ≠ физический пиксель

name	phys. width	phys. height	CSS width	CSS height
Apple iPhone 12 Pro Max	1284	2778	428	926
Apple iPhone 12, 12 Pro	1170	2532	390	844
Apple iPhone 11 Pro Max, XS Max	1242	2688	414	896
Apple iPhone 11, XR	828	1792	414	896
Apple iPhone X, XS, 11 Pro	1125	2436	375	812
Apple iPhone 7, iPhone 8	750	1334	375	667
Apple iPhone 6+, 6S+, 7+, 8+	1080	1920	414	736
Apple iPhone 6, 6S, SE2	750	1334	375	667
Apple iPhone 5, SE	640	1136	320	568
Apple iPhone 4	640	960	320	480

Реализация адаптивной верстки



```
.header {
```

```
    ...
```

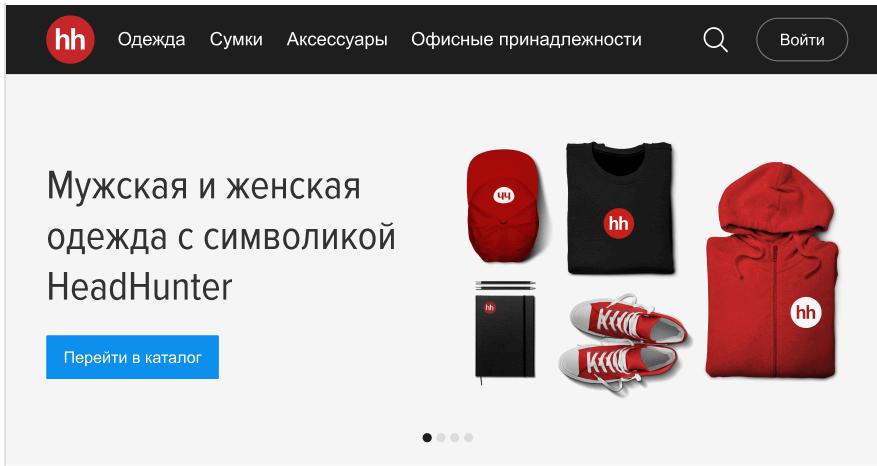
```
}
```

```
.promo-carousel {
```

```
    ...
```

```
}
```

Реализация адаптивной верстки



```
@media (min-width: 480px){
```

```
.header {
```

```
...
```

```
}
```

```
.promo-carousel {
```

```
...
```

```
}
```

```
}
```

Реализация адаптивной верстки

The screenshot displays a desktop view of the HeadHunter website. At the top, there is a dark navigation bar with the HeadHunter logo (a red circle containing 'hh') on the left, followed by menu items: Одежда (Clothing), Сумки (Bags), Аксессуары (Accessories), and Офисные принадлежности (Office supplies). To the right of the menu are a search icon, a search input field, and a 'Войти' (Log in) button. Below the navigation bar, there is a promotional section featuring several items of clothing and accessories, all featuring the 'hh' logo. The items include a red baseball cap, a black t-shirt, a red hoodie, a black notebook, and a pair of red sneakers. To the left of these items, there is俄文 text: 'Мужская и женская одежда с символикой HeadHunter'. Below this text is a blue button labeled 'Перейти в каталог' (Go to the catalog). At the bottom of the promotional section, there are three small dots indicating more content.

```
@media (min-width: 1000px) { ... }
```

Препроцессоры

Проблемы CSS

- Нет функций
- На CSS нельзя программировать. Например, писать циклы
- Использование переменных ограничено свойствами

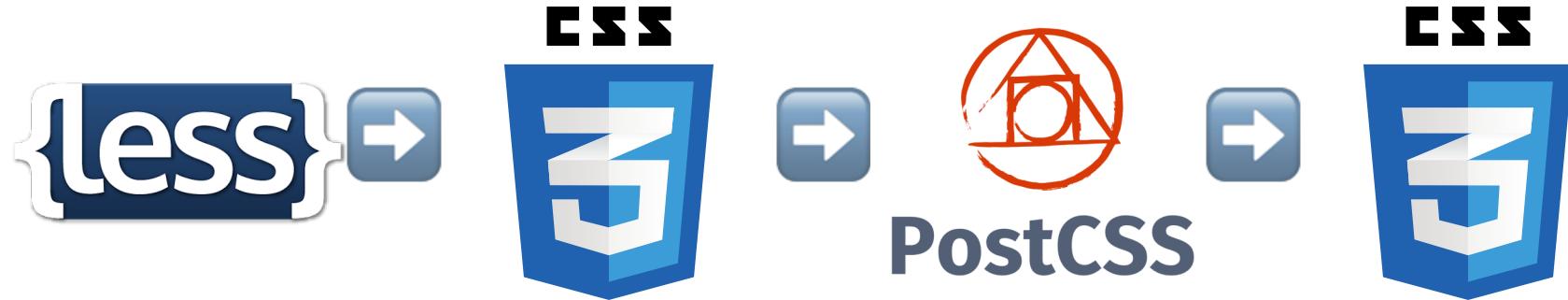
CSS-препроцессор

– это надстройка над CSS, которая добавляет новые возможности в синтаксис CSS и на выходе генерирует CSS код.

CSS-постпроцессор

– функция, которая позволяет изменять CSS с помощью JavaScript на основе его синтаксического дерева и добавлять для этого пользовательские плагины

Как это работает



CSS-препроцессоры

- [Less](#)
- [Sass/SCSS](#)
- [Stylus](#)

CSS-постпроцессор

- [PostCSS](#)

Использование less

1.

```
<link rel="stylesheet/less" type="text/css"
      href="styles.less" />

<script src="//cdn.jsdelivr.net/npm/less" ></script>
```
2.

```
npm install -g less

lessc styles.less styles.css
```
3. настроить компиляцию less с помощью сборщиков: webpack, gulp, parcel или др.

Вложенность селекторов

Переменные

Миксины

Другие возможности

Другие возможности

- Условия
- Циклы
- Функции

lesscss.org

Рекомендации по CSS

- Используйте селекторы по классу
- Применяйте методологию БЭМ
- Именуйте классы по смыслу, а не по значению
- Не повышайте специфичность селекторов без надобности
- Страйтесь не использовать непонятные сокращения
- Учитывайте переполнение или отсутствие контента
- Не перегружайте стили лишними свойствами и анимациями

Что почитать



- [doka.guide](#)
- [БЭМ](#)
- [Less](#)
- [PostCSS](#)

Вопросы

