HTML + CSS

Методичка будущего супергероя

полная версия

Впиши сюда свое имя, чтобы не потерять

Шпаргалки, практические советы, подсказки
стандарты разработки и любопытные схемы.

Версия 2.7 от 24 сентября 2014

Road Map

Занятие 1 Основы HTML, структура, теги, и семантика

Занятие 2 Основы CSS, сокращения emmet, цвета, фоны

Занятие 3 Формы, загрузка на сервер

Занятие 4 Блочная модель, верстка сеток, выравнивания

Занятие 5 Нарезка макетов, спрайты

Занятие 6 HELPDAY

Занятие 7 Псевдоэлементы, позиционирование

Занятие 8 Анимации и трансформации CSS

Занятие 9 Кроссбраузерность

Занятие 10 Адаптивность, Bootstrap, jQuery

Дополнение Типичные ошибки верстальщика

Дополнение Набор юного верстальщика

{EPIC SKILLS}

Основные теги **HTML**

▶ Одиночный тег

▷ Тег-контейнер

Оформление

Индекс	▷	`<sup><sub>`
Важное	▷	`<strong>`
Новое	▷	`<ins>`
Акцентрировать	▷	`<em>`
~~Неактуально~~	▷	`<del>`
Перенос строки	▶	` `

Таблицы

Таблица	▷	`<table>`
Строка	▷	`<tr>`
Столбец	▷	`<td>`
Заголовок	▷	`<th>`
Подпись	▷	`<caption>`

Форматирование

Ссылка	▷	`<a href="#">`
Параграф	▷	`<p>`
Картинка	▶	`<img src="a.png" />`
Заголовок	▷	`<h1>...<h6>`
Список	▷	`<ul><li>`
Цитата	▷	`<blockquote>`
Определения	▷	`<dl><dt><dd>`

Контейнеры

Контейнер	▷	`<div>`
Статья	▷	`<article>`
Секция	▷	`<section>`
Сайдбар	▷	`<aside>`
Шапка	▷	`<header>`
Футер	▷	`<footer>`
Меню	▷	`<nav>`

Семантическая разметка

<nav> - навигационный блок по сайту или странице

<header> - задает «шапку» сайта или раздела

<footer> - задаёт «подвал» сайта или раздела, в нём может располагается имя автора, контактная и правовая информация

<section> - раздел страницы, в котором элементы объеденены общим смысом, должен содержать заголовок

<aside> - сайдбар, раздел комментариев, реклама, примечания

<figure> - один или несколько рисунков, графиков, примеров кода

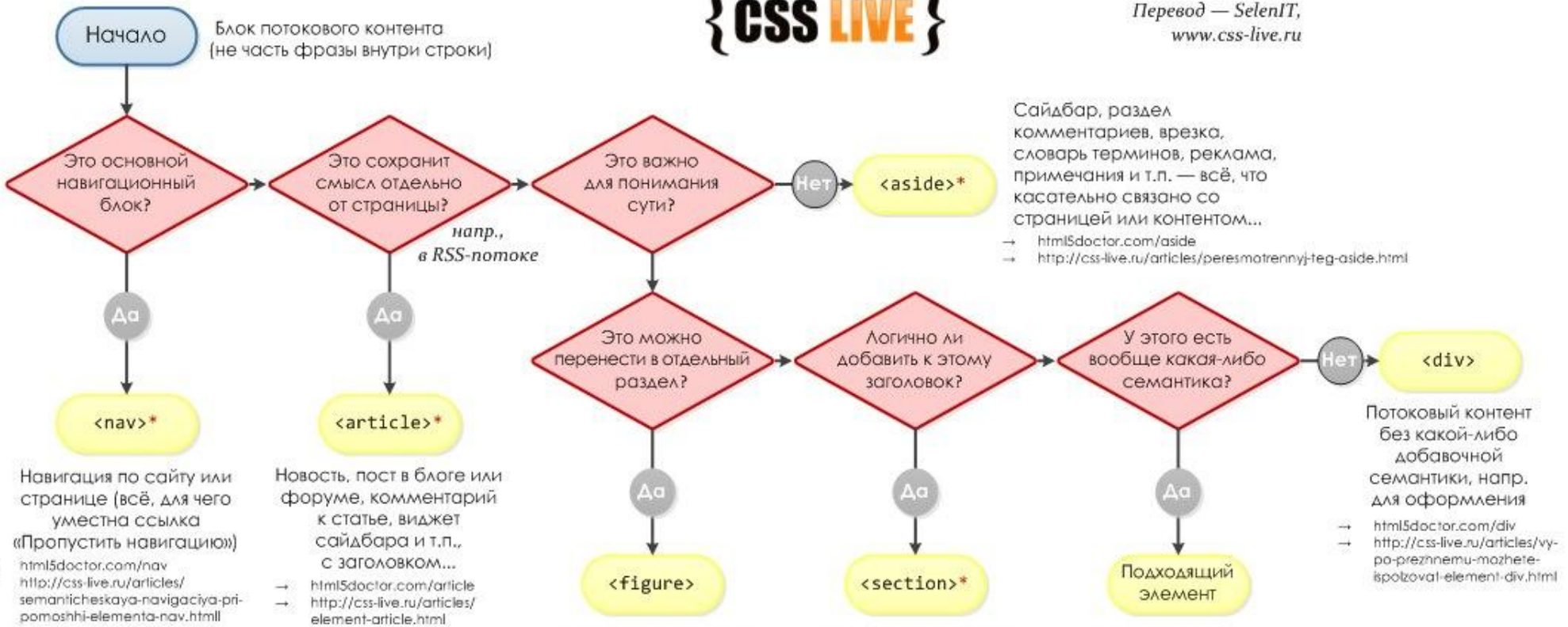
<article> - независимая часть сайта, задает содержание сайта вроде новости, статьи, записи блога, форума или др.

Выбор элемента HTML5

Секционные элементы и им подобные

Авторы — @riddle и boblet,
www.html5doctor.com

Перевод — SelenIT,
www.css-live.ru



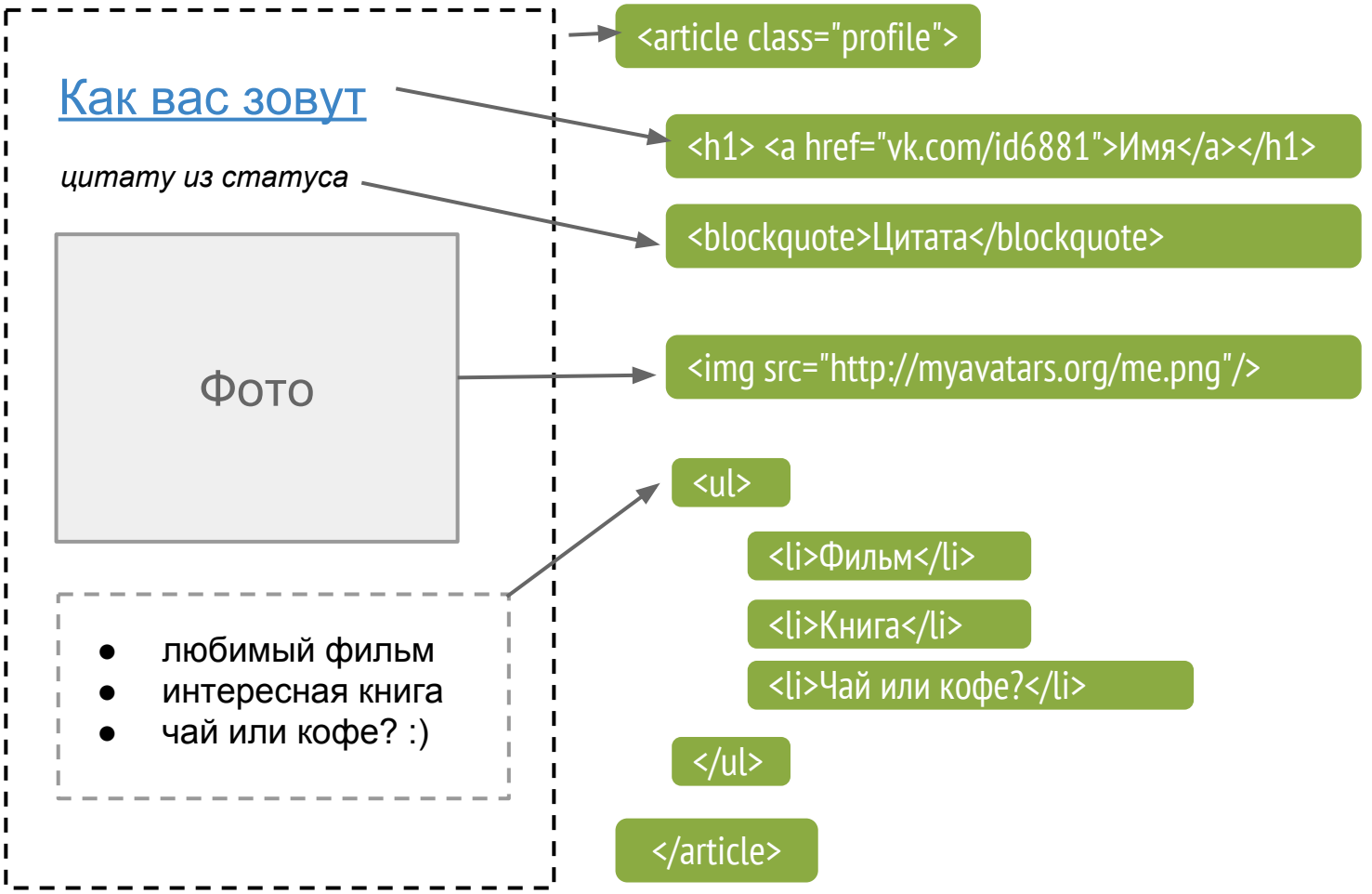
*Элементы «секционного контента»

Эти четыре элемента (и их заголовки) используются алгоритмом построения схем (планов) документа HTML5

→ html5doctor.com/outline
→ http://css-live.ru/articles/plany-dokumentov.html

Анкетка

Откройте cssdeck.com, нажмите New и вперед!



Копируем структуру сайта

Страница сайта

Структура сайта

Кое-что о львах

Привычки

Известные львы

Как превратиться в льва

Картинки



Исторический ареал льва был значительно шире современного: ещё в раннем средневековье лев встречался на всей территории Африки, кроме пустынь и тропических лесов

Львы для ягнят без регистрации и смс

```
<header>
    <h1>Кое-что о львах</h1>
    <a href="#">Привычки</a>
    <a href="#">Известные львы</a>
    <a href="#">Как превратиться в льва</a>
</header>
<section>
    <h2>Картинки</h2>
    <img src="mylion.png" height="200" />
    <img src="mylion.png" />
    <img src="mylion.png" />
    <p>
        Исторический ареал льва
        был значительно шире современного:
    </p>
</section>
<footer>
    <p>
        <a href="#">Львы для ягнят</a>без
        регистрации и смс
    </p>
</footer>
```

Что такое **EMMET** ?

Emmet - это инструмент для ускорения работы с HTML и CSS. В основе проекта лежит механизм динамических аббревиатур, которые разбираются «на лету» и из которых генерируется готовый фрагмент кода.

Выражение раскрывается по нажатию **TAB**

div →

div {А вот и я} → А вот и я

div.first →

div>lorem → Lorem ipsum dolor sit, consectetur…..

div.column$*3 →

ul>li>a[href=”#”] →

А теперь расшифруйте выражения

ul.first>li*4 →

div.first+div.second →

table>tr*2>td*2>p →

Резюме первого занятия

HTML - это язык разметки, отвечающий за структуру и содержание страницы

Основные теги оформления- это теги, меняющие внешний вид текста, позволяя не прописывать некоторые CSS свойства (например)

Одиночные теги не нуждаются в закрывании, (например <hr>), **парные теги** (например) нуждаются в этом

Ссылки в HTML имеют вид текст ссылки (парный тег)

Изображения в HTML создаются тегом <img src="" alt=""> (одиночный тег)

Контейнеры (DIV) это единицы структуры HTML, в которые можно укладывать наши теги, как вещи в чемодан

Семантические теги нужны, чтобы лучше воспринимать код и взаимодействовать с ним.

Emmet - это супер-фича, которая сэкономит вам кучу времени за счет сокращений вместо написания полного кода

Правильное оформление кода заключается в предусмотрении вложенности, табуляции и общего понимания и красоты кода.

Комментарии нужны, чтобы у верстальщика была возможность пояснить какой-либо кусок кода своими словами, и выглядят так <!--привет-->

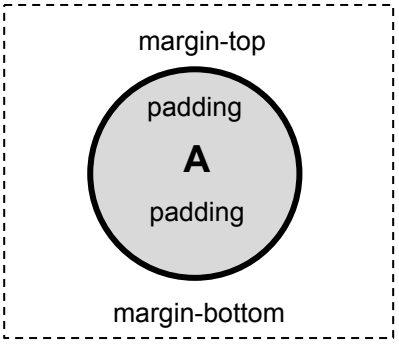
Семантическая верстка - верстка, в которой используются семантические теги. Желательно, чтобы верстка была именно такой

Домашнее задание

Время выполнения: **3 часа**

- ❏ Выполнить практическое задание "Тест по HTML"
- ❏ Выполнить практическое задание "Тест по Emmet"
- ❏ Расчертить макеты и подписать теги

Задания находятся в рабочей тетради и за каждый выполненный тест получаете поинты :)



CSS

Оформление текста

ff font-family: **Georgia;**
выбирает начертание (семейство шрифтов)

fz font-size: **16px;**
задает размер шрифта

fw font-weight: **bold;**
жирное начертание

fs font-style: **italic;**
устанавливает курсив

td text-decoration: **underline;**
подчеркивает текст

tt text-transform: **uppercase;**
прописной текст

Отступы и границы

m20 margin: **20px;**
задает внешние отступы

p40 padding: **40px;**
задает внутренние отступы

bd+ border: **1px solid #000;**
задает рамку

Цвет и фон

c color: **#fff;**
задает цвет текста

bgc background-color: **#fff;**
задает цвет фона

bgi background-image: **url();**
задает фон картинкой

bgp background-position: **0 0;**
задает позиционирование картинки

bgs background-size: **100%;**
задает размер картинки

Стилизация

bdrs border-radius: **6px;**
радиус закругленных уголков

bxs box-shadow: **0px 5px 5px #ccc;**
тень от элемента

ts text-shadow: **1px 2px 5px #000;**
тень от текста

-webkit-transform: **rotate(30deg);**
поворот

Расположение

ta text-align: **center;**
горизонтальное выравнивание внутри блока
left\right\center

m-a margin: **0px auto;**
горизонтальное выравнивание блока

va vertical-align: **top;**
вертикальное выравнивание
элементов top/middle/bottom

Отображение блока

h40 height: **40px;**
задает высоту блока

w20 width: **20px;**
задает ширину блока

lh40px line-height: **40px;**
задает высоту строки

oh overflow: **hidden;**
скрывает содержимое, выпавшее из контейнера

vh visibility: **hidden;**
делает видимым все содержимое

db display: **block;**
блочный элемент

di display: **inline;**
строчный элемент

dib display: **inline-block;**
строчно-блочный элемент

CSS Basics

HTML		CSS		Браузер
Вот дом	+	p { color:#f00; } c#f00	=	Вот дом
Который	+	p { font-style: italic; } fsi	=	*Который*
Построил джек	+	p { font-size: 32px; } fz32	=	Построил джек

А это пшеница	+	p {
		background-color:#ccc;
		}
		bgc#ccc
			=	А это пшеница

Которая	+	p {
		border-radius: 20px;
		border: 2px solid #444;
		}
		bdrs20+bd2-s-#444
			=	Которая

В темном чулане	+	p {
		background-color: #000;
		color: white;
		}
		bgc#000+c#fff
			=	В темном чулане

Хранится	+	p { border: 2px dashed cyan; } bd2-dashed-cyan	=	Хранится
в доме	+	p { border - top: 2px solid #444; } bdt2-s-#444	=	В доме
который	+	p { font-family: Georgia; } ff	=	Который
построил Джек	+	p { padding: 15px } p15	=	Построил Джек

Грамматика CSS

Простое описание

```
<дверь>
```

Нора начиналась идеально круглой, как иллюминатор, дверью, выкрашенной зеленой краской...

```
Дверь {
    форма : круглая ;
    цвет : зеленый ;
}
```

Классы и идентификаторы

```
<хоббит>
```

Бороды у хоббитов нет. Волшебного в них тоже, в общем-то, ничего нет, если не считать волшебным умение быстро и бесшумно исчезать

```
хоббит {
    борода : отсутствует ;
    волшебность : отсутствует ;
}
```

```
<хоббит class="Бэггинс">
```

Наш хоббит был весьма состоятельным хоббитом по фамилии Бэггинс. Бэггинсы проживали в окрестностях Холма с незапамятных времен и считались очень почтенным семейством не только потому, что были богаты, но и потому, что с ними никогда и ничего не приключалось и они не позволяли себе ничего неожиданного

```
.Бэггинс {
    почтенность : очень ;
    предсказуемость : высокая ;
    ноги : аккуратно причесаны ;
}
```

```
<хоббит class="Бэггинс" id="Бильбо">
```

Бильбо Бэггинс успел стать взрослым хоббитом, лет этак около пятидесяти; Бильбо Бэггинс стоял после завтрака в дверях и курил свою деревянную трубку, такую длинную, что она почти касалась его мохнатых ног (кстати, аккуратно причесанных щеткой)

```
#Бильбо {
    возраст : около 50 ;
}
```

```
<хоббит class="Норохолм Бэггинс" id="Фродо">
```

```
.Норохолм {
    одежда : капюшон ;
    ноги : растрепанные ;
}
```

Как называть классы, если вы не знаете, как их назвать

Задавать названия классов можно исходя из назначения, типа содержимого или внешнего вида. **Правило только одно: по селектору должно быть легко догадаться, к какому объекту он относится.**

Узкий выделенный элемент списка может получить классы **featured-item** и **narrow-item**

А маленькая кнопка удаления получит классы **micro-button** и **button-remove**

Типы элементов

switcher,
progress,
box, area

Содержание

message, article, entry, content, post, page, result

Контейнеры

container, column, canvas, wrapper, promo

Таблицы

table, row, cell,
caption, heading

Роли

main, secondary, featured, new, extra, branding, recent, navigation, popular, old, other, highlighted

Списки и элементы

list, item, unit

Внешний вид

transparent, shadowed, magnified

Состояния

success, alert, danger, active, removed, transparent, colored
inactive, current, default, toggle, animate

Ориентации и формы

landscape, portrait, square, rectangle, rounded, circle

Размеры

micro, small, narrow, medium, large, full-width, super

Цвета в CSS

RGB color

color: **rgb(255,0,0);**

классическая модель
цвет от 0 до 255

RGBA color

color: r**gba(255,255,255,0.4);**

последний параметр
задает прозрачность

классическая модель
цвет от 0 до 255

hex color

color:**#ffcc33;**

можно сократить до #fc3

классическая модель
цвет от 0 до ff

color name

color: **red;**

ограниченное
количество имен

rgb(153,80,160) =

#9944a0 =

DarkOrchid =

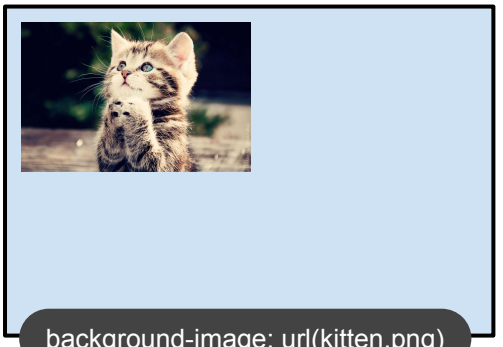
rgba(153,80,160,**0.8**)

rgba(153,80,160,**0.5**)

rgba(153,80,160,**0.2**)

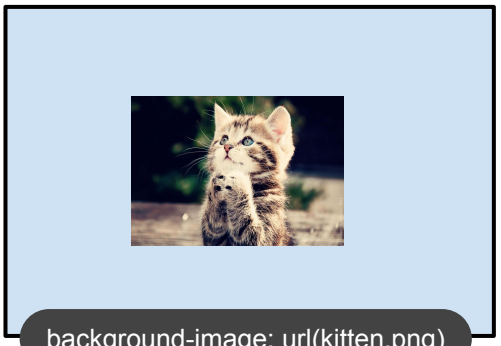
CSS Backgrounds

background-image: url(kitten.png)



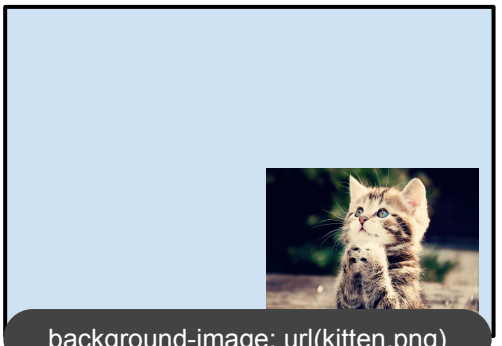
background-image: url(kitten.png)

background-repeat: no-repeat;



background-image: url(kitten.png)

background-position: 50% 50%;

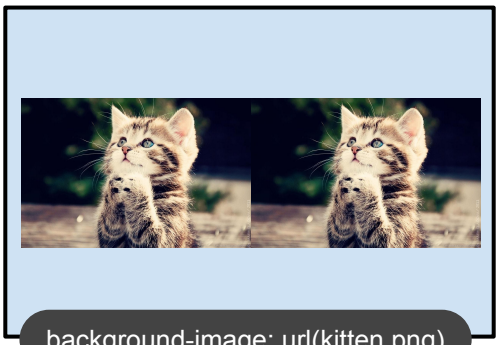


background-image: url(kitten.png)

background-position: bottom right;

background-image: url(kitten.png)

background-size: 100% 50%;



background-image: url(kitten.png)

background-repeat: repeat-x;

background-position: 50%;

Multiple backgrounds

```
background-image: url(left.png), url(right.png);
background-repeat: repeat-y, repeat-y;
background-position: left, right;
```

Тени с помощью CSS

1 Тень для текста

text-shadow: **1px 1px 5px gray**; → Я текст :)

- **1px** — Двигаем тень вправо\влево
- **1px** — Двигаем тень вверх\вниз
- **5px** — Размываем тень
- **gray** — Цвет

text-shadow: **10px 10px 3px gray;** → Я текст :)

2 Тень для блока

Синтаксис такой же как и у text-shadow

Стандартная тень в 5px

box-shadow: 1px 1px 5px gray;

Размываем тень на 50px

box-shadow: 1px 1px 50px gray;

Делаем внутреннюю тень с помощью inset

box-shadow: 0 0 15px gray **inset**;

Добавляем 2 типа тени: внешнюю и внутреннюю

box-shadow: 5px 5px 5px gray **inset,** 5px 5px 15px gray;

Градиенты с помощью CSS

Важно: вендорные префиксы для отображения разными браузерами! *(-webkit-, -o-, -moz-, -ms-)*

Пишем для Safari и Chrome:

Результат:

background-image: **-webkit-linear-gradient**(**bottom**, **red 1%**, **white 51%**, **black 76%**);

Тип градиента:
linear - линейный,
radial - радиальный

Направление градиента:
bottom - снизу
top - сверху
bottom right - правый нижний угол и.т.д.

Цвет и точка начала
% - с какой позиции начнется цвет градиента?
(Можно в %, px)

Пишем для Firefox

background-image: **-moz-radial-gradient(center, white 20%, gray 51%, white 70%);**

Пишем для Opera

background-image: **-o-linear-gradient(right top, white 10%, gray 30%, white 50%, gray 70%, white 90%);**

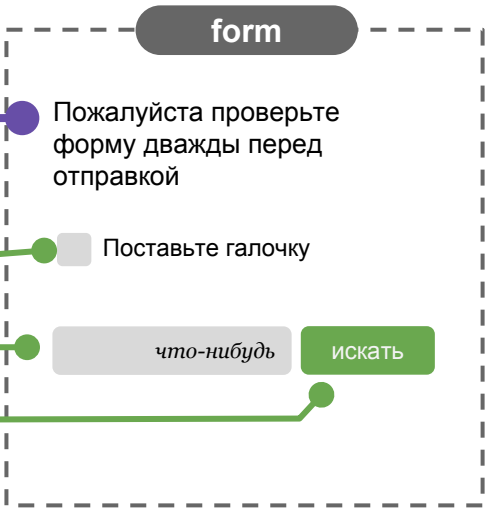
Генератор градиентов на CSS3 http://gradients.glrzad.com/

Верстка форм

1 Создайте форму, укажите action

Аттрибут action отвечает за адрес, куда будет отправлена форма

```
<form action="http://kushedow.ru/go">
    <input type="checkbox" name="string">
    <input type="text" placeholder="что-нибудь">
    <input type="submit" value="Искать">
</form>
```



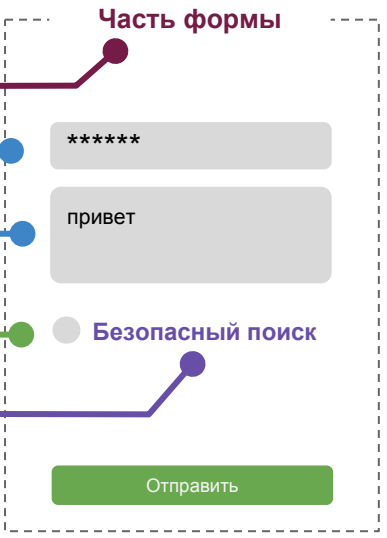
2 Добавьте еще элементов для формы

Большая их часть - разные виды input

```
<fieldset>
    <legend>Часть формы</legend>
    <input type="password" name="pass">
    <textarea name="message">привет</textarea>
    <input type="radio" name="string" id="radio1">
    <label for="radio1">Безопасный поиск</label>
    <input type="submit" value="Отправить">
</fieldset>
```

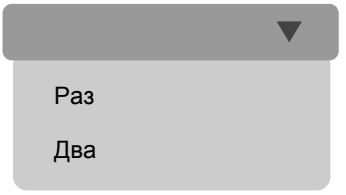
Объединяем элементы в группы

заголовок для fieldset



3 Используем выпадающие списки

```
<select name="pass">
    <option value="1"> Раз </option>
    <option value="2"> Два </option>
</select>
```



Файловая система

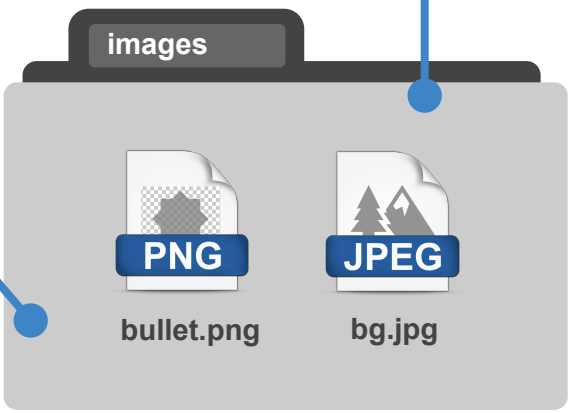
Чтобы опубликовать сайт на сервере нужно правильно расположить файлы по папочкам. HTML-файлы хранятся в .html, а css-файлы в css.

index.html

1 В корне сайта размещается файл index.html и другие html-страницы. Все ресурсы прячутся по папочкам


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

<img src="images/bullet.png"/>
```

```
a {
    background-image: url(../images/bg.jpg);
}
```



2 В именах файлов настоятельно рекомендуется использовать только строчные латинские символы, тире, подчеркивание и цифры. Пробелы, знаки препинания и спецсимволы исключены. Понмните, что .jpeg и jpg - разные файлы

3 Нормативный способ подключения CSS и JS-файлов - располагать их в секции <head>. Скрипты и стили в теле документа сурово осуждаются

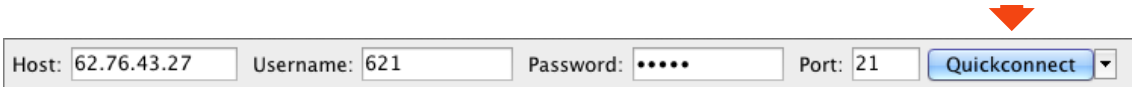
Загрузка файлов по FTP

Чтобы сайт был доступен в сети все его страницы нужно загрузить на сервер хостера. Это делается с помощью протокола передачи файлов (ftp) и специально программы (ftp-клиента)

Установить клиент можно по ссылке:

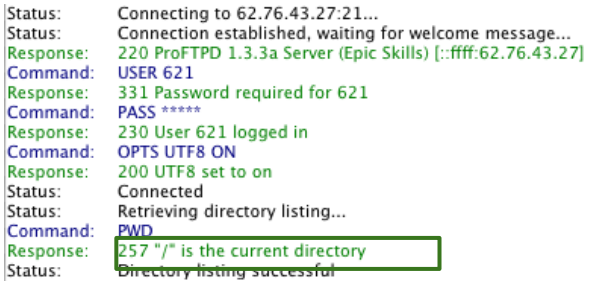
https://filezilla-project.org/

1



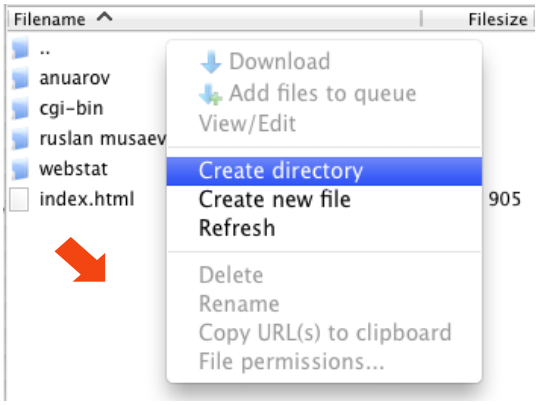
Вам понадобятся адрес хоста, логин и пароль. Укажите также порт на всякий случай.

2



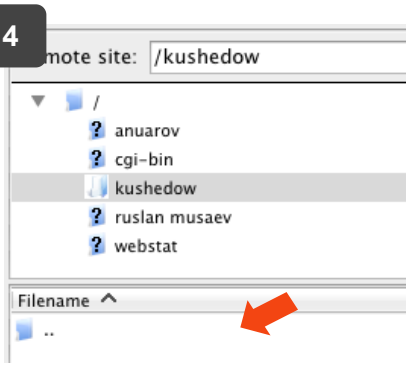
Увидев быстро появляющиеся надписи дождитесь появления в правом окне каталогов. Если появляющиеся надписи красного цвета - проверьте логин и пароль.

3



Создайте новую папку и перейдите в нее

4



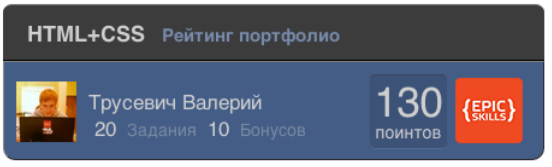
Перетащите файлы в папку. Дождитесь завершения загрузки

5

Если ваш файл лежит здесь:
/kushedow/portfolio/index.html

Открыть его можно тут:

http://621.epixx.ru/kushedowportfolio/index.html



! Важно: используйте только строчные латинские буквы без пробелов и знаков препинания. Цифры, тире, подчеркивания и точки разрешены.

Создание сетки

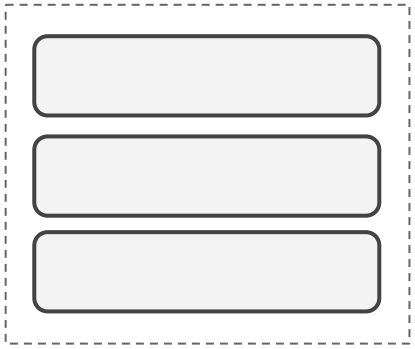
Пустые блоки

```
.box{
  width: 30px;
  height: 30px
  background-color: #444;
  border-radius: 6px;
}
```

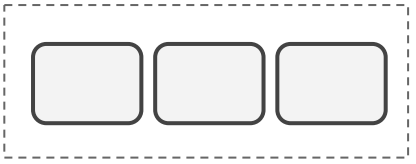
Выравнивание по горизонтали

```
h1{text-align:center;}
div {text-align:center;}
div{text-align:right;}
```

Колонки

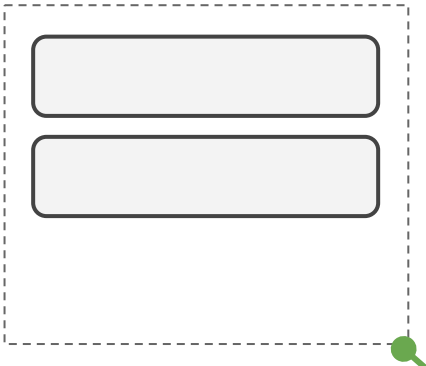


```
.box{
  width: 29%;
  display: inline-block;
  vertical-align: top;
  margin: 1%;
}
```

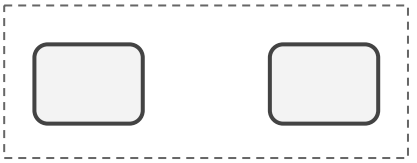


Меняя поведение в потоке и определяя ширину, мы убеждаем блоки расположиться горизонтально

Расположение



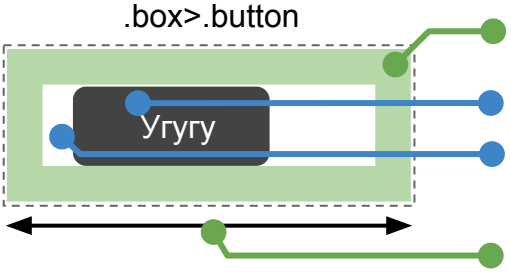
```
.box1{
  width: 29%;
  float:left;
  }

.box2{
  width: 29%;
  float: right;
  }
```



Задавая обтекание элементам мы можем расположить их справа или слева.

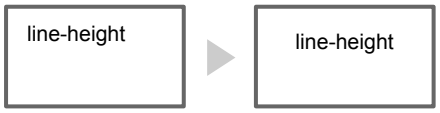
Если вы заметите. что родительский контейнер подозрительно изменил высоту, используйте для него **overflow:hidden**

Отступы



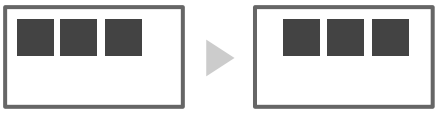
Горизонтальное выравнивание

Контент внутри блока



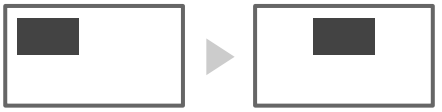

```
p {
    text-align:center;
    height: 40px;
    line-height: 40px;
}
```

Один/несколько инлайн-блоков



```
.parent {
    text-align:center;
}
```

Один блочный элемент



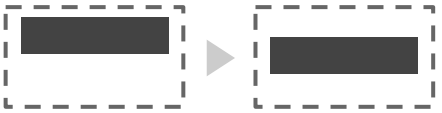
```
.child{
    width: 30%;
    margin: 0px auto;
}
```

Вертикальное выравнивание

Одна строка, высота известна


```
p {
    height: 40px;
    line-height: 40px;
}
```

Блок внутри блока



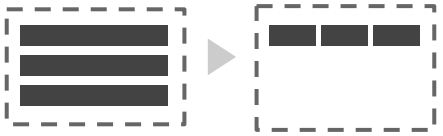
```
.parent {  display: table; height: 400px}
.child{
    display: table-cell;
    vertical-align: middle;
}
```

Блок (или инлайн- блок) по центру

```
.parent {  position: relative;}
.child{
    width: 400px; height: 200px;
    position: absolute;
    top:0; bottom:0; margin: auto;
}
```

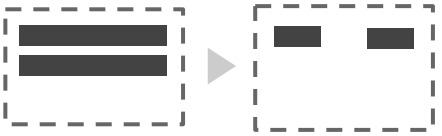
Сетка

Создание колонок



```
.parent {  font-size: 0px;  }
.column{
    vertical-align:top;
    display: inline-block;
    width: 33.3%;
    font-size: 14px;
}
```

Приклеивание налево и направо



```
.parent {  overflow: hidden;  }
.left{
    width: 200px;   float:left;
}
.right{
    width: 200px;   float:right;
}
```

Правильная блочная модель

```
.parent {  overflow: hidden;  }
.column{
    box-sizing: border-box;
    padding: 20px;
    margin:  1%;
    width: 31.3%
    float: left;
}
```

Создаем четыре колонки (инлайн-блоки)

Полезная шпаргалка

Пляжный отдых

Походы

Вкусная еда

Вечеринки

Шаг 1 - создаем html-разметку

.row>(.column>img+h2)*4

```
<div class="row">
  <div class="column">
    <img src="" alt=""><h2>Пляжный отдых</h2>
  </div>
  <div class="column">
    <img src="" alt=""><h2>Походы</h2>
  </div>
  <div class="column">
    <img src="" alt=""><h2>Вкусная еда</h2>
  </div>
  <div class="column">
    <img src="" alt=""><h2>Вечеринки</h2>
  </div>
</div>
```

Шаг 2 - Превращаем список в колонки

.row{fz0}

.column{dib+w23%+tac+fz10+vat+m1%}

```
.row{font-size: 0;}
.column{
  display: inline-block;
  width: 23%;
  text-align: center;
  font-size: 10px;
  vertical-align: top;
  margin: 1%
}
```

Оставим по 1% на отступы. В сумме всегда должно получаться 100%

Между инлайн блоками остается расстояние в один пробел. Установим размер шрифта 0, пробел исчезнет, а мы востановим размер шрифта до 10

Шаг 3 - Раскрашиваем рамочки

.column{bd1-s-#dfdacf+bc#f4f3ee+bxbb}

```
.column{
  border: 1px solid  #dfdacf;
  background-color: #f4f3ee;
  box-sixing: border-box;
}
```

Чтобы границы не увеличивали ширину элемента, мы меняем алгоритм блочной модели

Шаг 4 - Фиксируем размер картинок

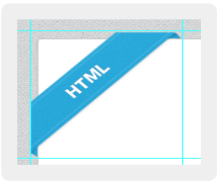
.column {pt20}

.column img{maxw100%}

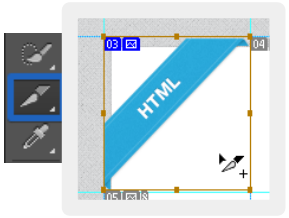
```
.column {
  padding-top: 20px;
}

.column img{
  max-width: 100%;
}
```

Нарезка макета в Photoshop

1. Расставляем направляющие вокруг нужного фрагмента

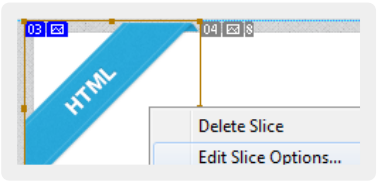


2. По направляющим выделяем фрагмент с помощью инструмента **slice tool**



3. Даем имя фрагменту через **edit slice options**

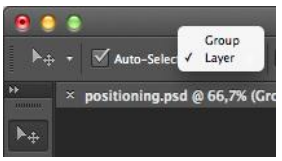
Имя фрагмента обязательно латинскими буквами



4. Сохраняем фрагмент **file - save for web**

Если нужна прозрачность, не забываем скрыть фоновое изображение фрагмента

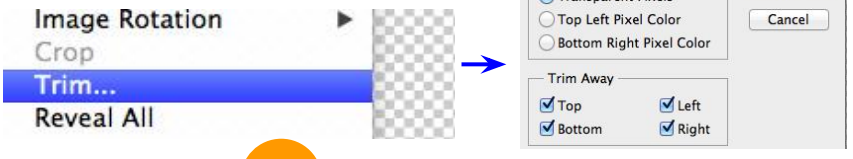
1. Для поиска объекта настраиваем и используем инструмент **select tool**



2. Выбираем кликом нужный элемент, программа выделяет нужный слой

3. Правой кнопкой мыши щелкаем на иконку с глазом слева от миниатюры слоя и выбираем **show/hide all other layers**

4. В пункте меню **Image** выбираем **Trim, Transparent pixels** и жмем ок



5

Выбор формата изображения

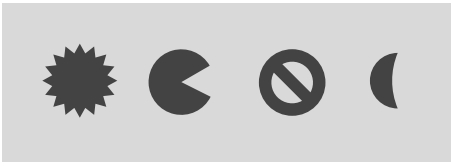
JPG Сохраняет яркость и оттенки цветов неизменными. **Использовать:** отлично подходит для фотографий

PNG-8 Поддерживает прозрачность, но использует 8-битную палитру (256 цветов). **Использовать:** для иконок, текста, логотипов

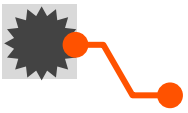
PNG-12 Использует 24-битную палитру (16,7 млн цветов) и поддерживает полупрозрачность. Позволяет получить изображение с наилучшим качеством, но, часто это сказывается на объеме конечного файла. **Использовать:** изображения с прозрачными и полупрозрачными элементами, рисунки с большим количеством цветов, фотографии.

Что такое спрайты

Спрайт - это картинка, содержащая в себе несколько иконок, для экономии места и более удобного редактирования



sprite.png



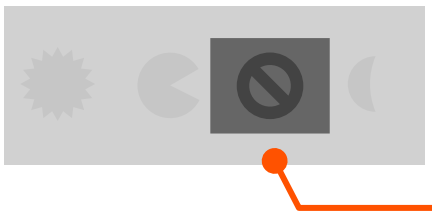
```
.icon{
    width: 20px;
    height: 20px;
    background-image: url(sprite.png);
    background-position:  0px 0px;
}
```

Чтобы использовать спрайты создайте небольшой контейнер и задайте ему ширину высоту, затем используйте спрайт в качестве фонового изображения

-25px

```
.icon.pacman{
    background-position: -25px 0px;
}
```

-45px



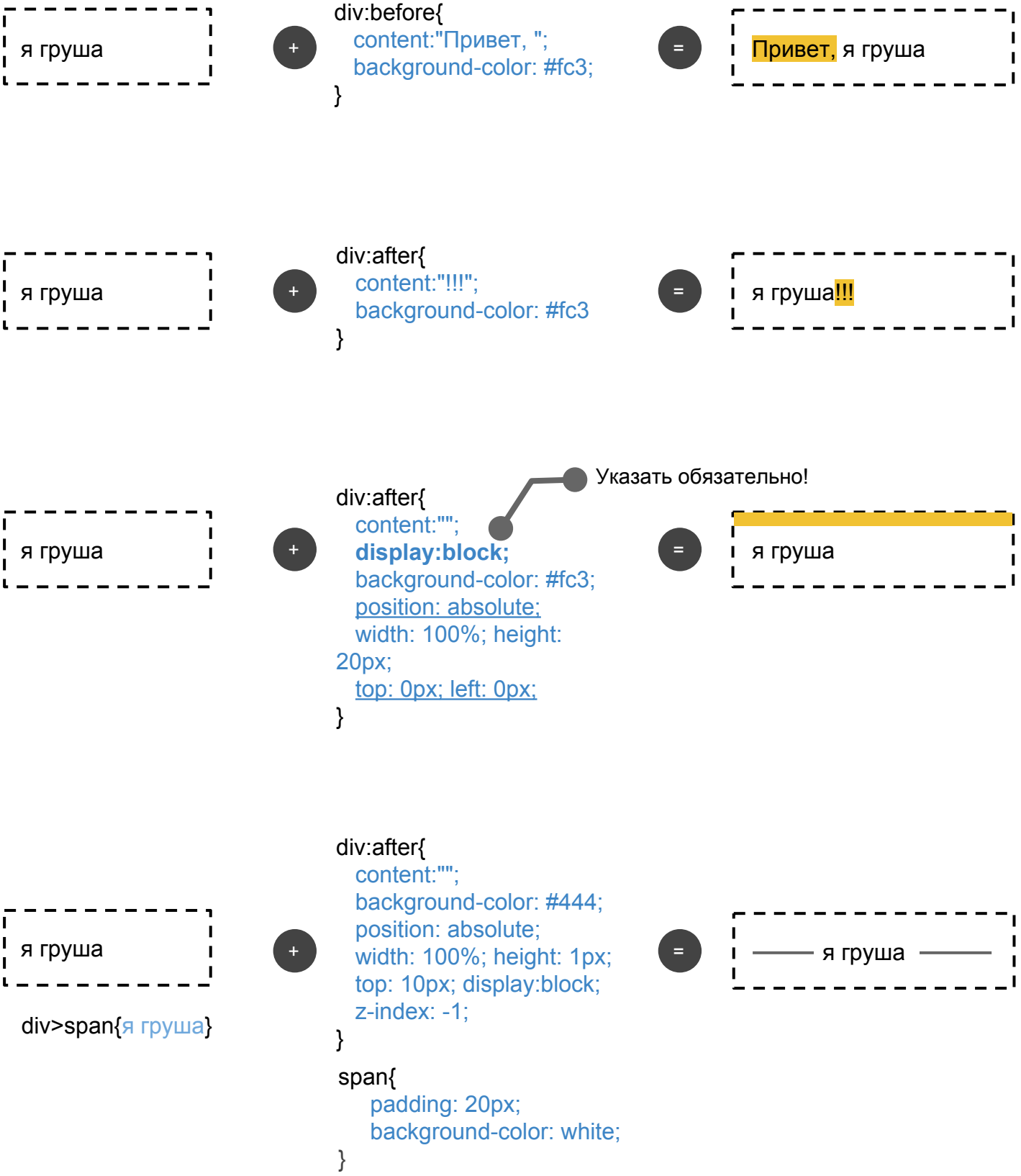
```
.icon.stop{
    background-position: -45px 0px;
}
```

Показать остальные иконки можно передвигая фон

Преимущества спрайтов:

- однократная загрузка сервером сразу всех элементов в одном файле
- использование единого спрайта позволяет уменьшить вес графики и повысить производительность

Псевдоэлементы



POSITION:

STATIC;

float: left;
position: static;
margin: 0px auto;

Значение по умолчанию

FIXED;

position: fixed;

Приклеивается к экрану

ABSOLUTE;

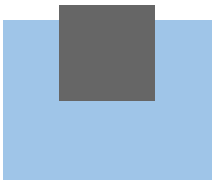
position: absolute;
bottom: 30px;
right: 10px;

Относительно вьюпорта

RELATIVE;

position: relative;
top: -30px;

Относительно текущей позиции



RELATIVE родителю

position: relative;

ABSOLUTE элементу

position: absolute;
top: 0px; left: 0px;

Относительно родителя

Трансформация CSS

Синтаксис **transition: <property> <duration> <timing-function> <delay>;**

transition-property	Свойство, к которому применяем анимацию	color, background, width, font-size и т.д.
transition-duration	Длительность анимации	0.5s, 1s, 1/5s и т.д
transition-timing-function	Временная функция, используемая для анимации	ease, linear, ease-in-out, ease-in, ease-out
transition-delay	Задержка анимации	0.5s - полсекунды, 1s - одна секунда и т.д.

Примеры

1

```
<div class="block"></div>
```

```
.block {
    width: 100px; height: 100px;
    background-color: #e74a2a;
    -moz-transition: all 3s linear;
    -o-transition: all 3s linear;
    -webkit-transition: all 3s linear;
}
.block:hover {
    background-color: #38761d;
    width: 200px;
}
```

При наведении на блок меняется цвет и ширина

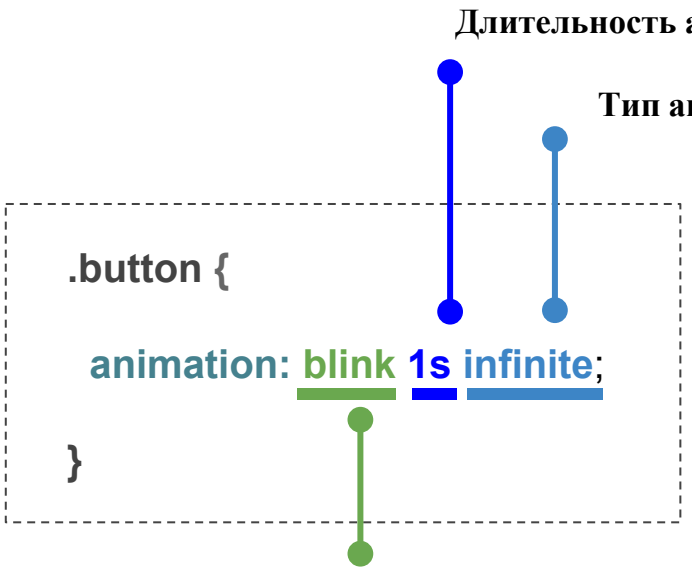
2

```
<div class="block"></div>
```

```
.block {
    width: 200px;  height: 50px;
    background-color: #0B5394;
    -moz-transition: all 1s 1s ease-in;
    -o-transition: all 1s 1s ease-in;
    -webkit-transition: all 1s 1s ease-in;
}
```

```
.block:hover {
    background-color: #38761d;
    -webkit-transform: rotate(-360deg);
    -moz-transform: rotate(-360deg);
    -o-transform: rotate(-360deg);
}
```

Анимация на CSS



ease - плавная анимация

linear - линейная анимация

steps(4) - дерганная анимация

infinite - кольцовая анимация

Название анимации

```
@keyframes blink{

    from {
       opacity: 1; color: #fff;
     }

    50% {
       opacity: 0; color: #bbb;
     }

    to {
       opacity: 1; color: #fff;
     }

}
```

Полный синтаксис

```
Название анимации
animation-name: sunrise;
Длительность анимации
animation-duration: 10s;
Тип анимации
animation-timing-function: ease;
Количество повторений
animation-iteration-count: 1;
Направление анимации
animation-direction: normal;
Отсрочка анимации
animation-delay: 5s;
```

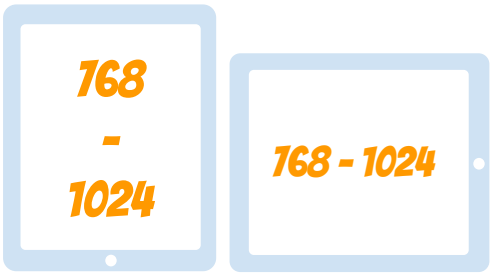
@media queries

И их использование в Адаптивной верстке

Разрешение экрана сейчас колеблется от 320px (iPhone) до 2560px (большие мониторы) или даже выше. Пользователи больше не просматривают сайты только на настольных компьютерах. Теперь пользователи используют мобильные телефоны, небольшие ноутбуки, планшетные устройства, такие как iPad или Playbook для доступа в интернет. Поэтому, традиционный дизайн с фиксированной шириной больше не работает. Дизайн должен быть адаптивным.

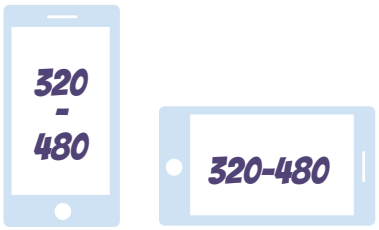
Экран от 1024px и больше

```
@media (min-width: 1024px) {
    .my{  color: blue;  }
}
```



Экран от 1024px до 768px

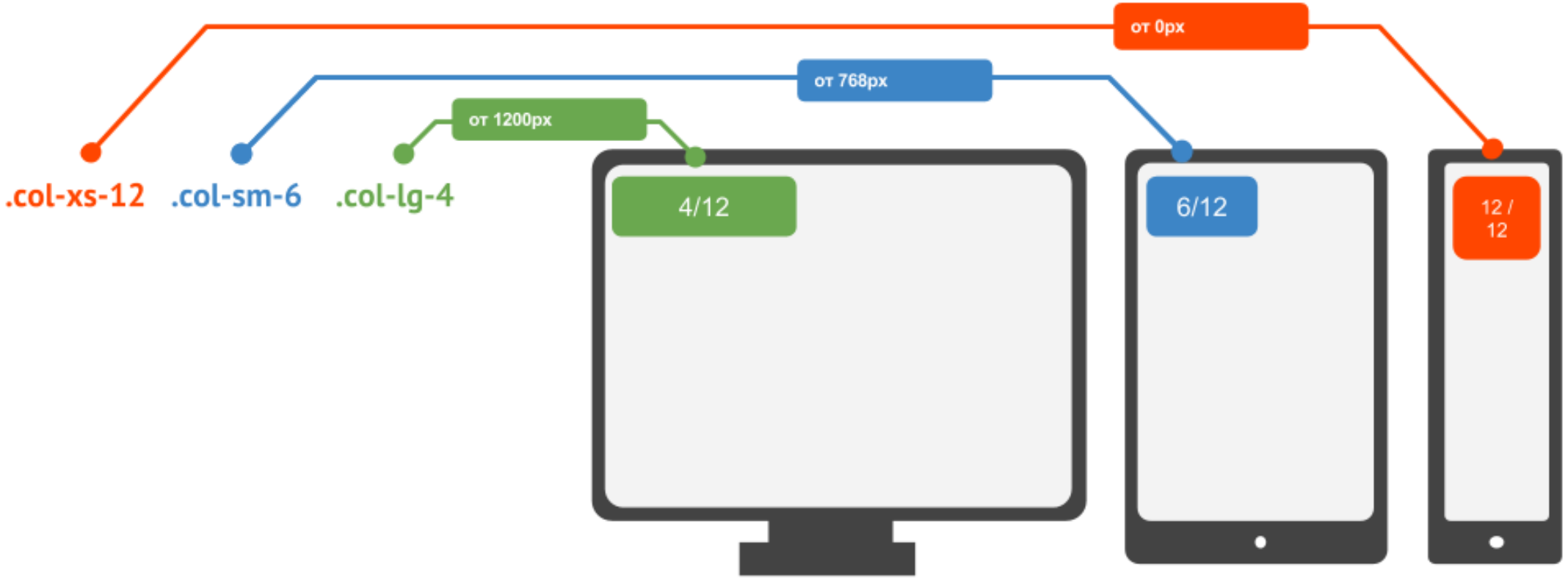
```
@media (min-width: 768px) and (max-width: 1024px) {
    .my{  color: orange;  }
}
```



Экран от 480px до 320px

```
@media (min-width: 320px) and (max-width: 480px) {
    .my{  color: purple;  }
}
```

Bootstrap адаптивность



XS 0 +

SM 768 +

MD 992 +

LG 1200 +

jQuery CSS

Получение свойства

```
var color = $("a").css("background-color");
```

Установка свойства

```
$("a").css("background-color","green");
```

Добавление класса

```
$(a).addClass("fancylink");
```

Удаление класса

```
$(a).removeClass("fancylink");
```

jQuery HTML

Получение html

```
var content = $("a").html();;
```

Редактирование html

```
$(".message").html("<em>hey!</em>");
```

Заворачивание в html

```
$("a").wrap("<div></div>");
```

Клонирование

```
var content =$(".inputbox").clone();
```

Добавление элемента

```
$(".log").append("<em>Миу</em>");
$(".log").prepend("<em>Миу</em>");
```

Удаление элемента

```
$(a).remove();
```

Перенести элемент

```
$(a).appendTo("header");
$(a).prependTo("header");
```

jQuery - события

1 Создаем обработчик → 2 Привязываем к событию

```
var swipe = function(){
    $(this).css("background-color","#fc0");
}
```

```
$("#right").on("click",
swipe);
```

Создавая обработку события, мы на самом деле используем метод, которому передаем объект-функцию. Это как если бы мы передавали пакет с инструкцией на определенный случай. При наступлени события инструкция выполняется

Обработка нажатия

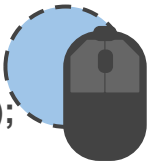
```
$("#target").on("click",действие);
```

Двойной клик

```
$("#target").on("dblclick",
действие);
```

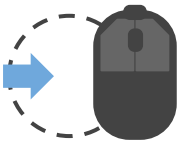
Наведение на объект

```
$("#target").on("mouseover",действие);
```



Фокус на объекте

```
$("#target").on("focus",действие);
```



Отведение курсора

```
$("#target").on("mouseout",
действие);
```

1+2 Можно использовать анонимную функцию для обработки событий

```
$("#right").on("click",function(){
    $(this).animate({"left": "+=50px"}, "slow");
});
```

В этом случае инструкция "что делать, если" передается напрямую объекту без объявления функции

правильно Совсем правильно использоать метод **on** для создания обработчика

```
$("#members li a").on("click", swipe);
```

```
$("#members li a").on("click", function(){
    $(this).animate({"left": "+=50px"}, "slow");
});
```

переменная метод параметр http://jquery-docs.ru/manipulation/

HTML+CSS - Типичные ошибки

Кофе-брейк

От мелких неисправимых ошибок легко перейти к крупным порокам.

Сенека, информационный архитектор, инфографер

Инлайновая запись стилей **<h1 style="color: blue".....>**

Причина ошибки

Когда очень лень писать в style.css хочется написать стили внутри аттрибута style. Не надо так.

Последствия и вред

HTML-код превращается в нечитаемый документ.
Сложно отследивать, почему после изменений правила не применились.
При изменении придется править ВСЕ инлайновые стили
Нельзя просто так заменить таблицу стилей

Справка

Инлайновые стили имеют приоритет выше, чем любые селекторы и уничтожаются только с помощью !important.
Инлайновые стили противоречат взгляду на оформление как на задание единых правил для отображения контента.

Верстка декоративных элементов с использованием <img>

Причина ошибки

Мы быстро запоминаем, что картинка это img, хотя это не всегда правильно.

Последствия и вред

1. Если владелец сайта решит поменять дизайн кнопочек, иконок или каких-либо других декоративных элементов, кому-то придется потратить много времени на поиск и замену url в каждой html странице. А он точно захочет.
2. Вред: увеличение кода страницы, затруднение редактирования.

Справка

Декоративные элементы (иконки, кнопки и пр.) следует **добавлять с помощью стилей** для какого-нибудь элемента, либо **заключать в блочный элемент** (или псевдоэлемент) и указывать свойства внешнего вида в стилевом файле: например, задать ему background-image.

Не сброшены стили по умолчанию (нет css-reset)

Причина ошибки

Разное отображение верстки в разных браузерах
Отступы у параграфов, заголовков и списков, которые приходится уменьшать

Неправильно

Сбрасывать стили вручную или не сбрасывать
Подключать файл для сброса после основного css и терять все стили :)
Подключать css reset с ненадежных ресурсов

Правильно

Использовать Modernizr
Начинать верстку с подключения reset
Не верстать больше 8 часов подряд :)

Классы у колонок, идущих друг за другом, называются **.first, .second, .third**

Причина ошибки

Конечно, удобнее задать отдельные классы, чем придумывать универсальный список стилей

Последствия и вред

Для программиста при работе с циклом вывод дополнительных аттрибутов зависимых от номера нежелателен, придется переверстывать или дописывать условия. Програмист вас возненавидит :)
Вред: потеря времени на переверстку, насилие в команде :)

Правильно

Верстать так, чтобы не зависеть от атрибутов, к первому и последнему элементу всегда можно обратиться с помощью псевдоклассов :first-child или :last-child, а используя порядковые номера элементов, можно задать :nth-child.

NB: используя псевдоклассы :first-child, :last-child и :nth-child помним про кроссбраузерность.

Запись стилей внутри html документа: <style>....</style>

Причина ошибки

Когда очень лень писать в style.css или стили только для одной страницы хочется добавить код прямо на страницу

Последствия и вред

Увеличение веса страницы
Путаница что написано в коде страницы, а что в стилях
Насилие над программистами, которым сложно работать с лишним кодом

Правильно

Собираем свою волю в кулак и добавляем новые правила в конец основного файла style.css. Кстати, добавленные позже правила имеют более высокий приоритет.

Перенос строки с помощью нескольких тегов

Причина ошибки

Написать
 видимо проще, чем добавить параграф

Последствия и вред

Отступы, созданные
 невозможно стилизовать с помощью CSS.
Правила типографики для абзацев не работают.
Жесткие точки для переноса ухудшают читаемость, да и выглядит это так себе.

Правильно

Использовать
 только там, где необходим перенос строки, а не начало нового параграфа.

Имена классов и id написаны не по правилам

Причина ошибки

Кажется, что имена классов и идентификаторов могут быть любыми. На самом деле это не так.

Последствия и вред

Правило просто не будут работать

Правильно

В именах классов могут использоваться только строчные латинские буквы, цифры, знак подеркивание и тире. Название класса и id не может начинаться с цифры.

Заголовки **h1 h2 h3** определены по размеру в макете, а не по семантической значимости

Причина ошибки

Привычно, что чем больше шрифт у заголовка - тем он важнее. Однако это не всегда так

Последствия и вред

Вред для SEO
Конфликты при составлении общей типографики

Справка

Заголовки начинаются с H1, следующий за ним **обязательно должен быть H2.**
При выборе тега для заголовков следует руководствоваться уровнем значимости и отношениями с другими заголовками, а не размером. H1-это главный заголовок страницы, а не самый большой.

Подключение css-reset и прочих стилевых файлов после собственного style.css

Причина ошибки

Мы добавляем стили, не всегда понимая в каком порядке мы их подключаем.

Последствия и вред

В результате, файл reset или bootstrap затирает наши новые стили, а мы не понимаем в чем дело

В файле html не указана кодировка utf-8

Симптомы

Браузер отображает неизвестные ёС‡РµСЃРєРѕРіРѕ на только что сверстаной страничке.

Причина ошибки

Браузер неверно распознал кодировку html-документа

Правильно

Добавляем в шапку html-файла, после открывающего тега **<head>** строчку **<meta encoding="utf-8">** и сохраняем страничку в **utf-8** кодировке **без BOM.**

Не указано свойство vertical-align у идущих подряд инлайн-блоков

Причина ошибки

Мы привыкли выравнивать блоки по горизонтали, но не по вертикали

Последствия и вред

Бывает, что некоторые инлайн-блоки выпадают по высоте из стройного ряда остальных.

Правильно

Чтобы выровнять все инлайн-объекты на один уровень, нужно задать ему css-свойство vertical-align:top/middle/bottom/...

Злоупотребление !Important

Причина ошибки

Мне лень думать над универсальной и логичной таблицей стилей, поэтому, если свойство не работает, я просто присваиваю ему !important

Последствия и вред

Слишком частая расстановка !important нарушает расстановку приоритетности правил CSS, что может привести к ситуации, когда вы не понимаете, почему тот или иной кусок кода не работает :(

Правильно

Используем !important только в тех случаях, когда без него не обойтись. То есть, желательно, никогда. Профессионалы и супергерои спокойно обходятся без него.

Не указан DOCTYPE в начале документа

Причина ошибки

Непонятно зачем нужен этот doctype, у меня и без него все прекрасно работает.

Последствия и вред

Возможно некорректное отображение html-страницы в Internet Explorer.

Справка

Запись <!DOCTYPE html> говорит браузеру о том, что веб-страницу следует отображать согласно стандарту html5

Злоупотребление селекторами по id при описании стилей элемента

Причина ошибки

Задать одинаковым элементам классы и выделить отдельные с помощью id кажется логичным способом записи стилей

Последствия и вред

При программировании нельзя задать еще один id
Возможен конфликт имен (несколько элементов называются одинаково)
Сложнее переопределять стили

Правильно

Если хочется выделить отдельный элемент, ему можно задать несколько классов через пробел, например <img class="thumb left" ...>.

Использование margin или padding, чтобы прижать элементы к разным краям

Причина ошибки

Иногда увеличение отступа слева кажется удачным решением, чтобы поставить элемент справа. Однако при изменении контента или размера верстка разламывается

Последствия и вред

Необходимость переверстывать элементы
Неправильное поведение при переносе на новую строку

Правильно

Использовать float:right + float:left
Или абсолютное позиционирование.

Указание inline-элементам значений высоты, ширины и вертикальных margin

Причина ошибки

Иногда, нам кажется, что мы можем управлять всеми размерами элементов у себя на странице.

Последствия и вред

Ваши правила не будут работать

Справка

К строчным элементам относятся теги **<img>**, ****, **<a>** и др. Свойства, связанные с размерами, вроде **height** и **width** для них не работают, также как и вертикальные внешние отступы **margin**. Ширина строчных элементов складывается из ширины содержимого и значений границ, отступов и полей.

Сохранение всех фрагментов макета в формате .png-24

Причина ошибки

.png - отличный формат! Качество изображений - супер! Прозрачность - есть! Всегда буду использовать только его!

Последствия и вред

Ваш сайт будет дольше загружаться.

Правильно

При сохранении фрагментов макета, проверять возможность использования и других форматов изображений (png-8, jpg, gif) и применять png-24, когда это действительно необходимо. **Области применения .png-24**: изображения с прозрачными и полупрозрачными участками, изображения с большим количеством цветов.

Не сброшены float у родителя

Справка

Только float недостаточно, чтобы реализовать надежное разделение на колонки или приклеить элемент справа. Поскольку размер родителя считается исходя из суммы размеров дочерних элементов **в нормальном потоке,** необходимо использовать один из способов сброса float

Последствия и вред

Полное разрушение всей верстки. Сложность при поиске ошибки.

Правильно

Не забываем сбрасывать float одним из перечисленных ниже способом, даже если, на первый взгляд, все выглядит хорошо.

Способы сброса

1. **overflow:hidden;**
   для родителя
2. ** **
   внутри родителя
3. **:after{ clear:both; content:” ”; display:block; }**
   для родителя

По имени класса или id невозможно понять к чему они относятся

Причина ошибки

Название класса, как кажется, может быть любым, потому берется первое, что приходит в голову. На практике это приводит с созданию плохих селекторов

Последствия и вред

В важем коде гораздо сложнее разобраться
Вы запутаетесь в собственных селекторах
Больше вероятность ошибок и усложнение их поиска

Правильно

Выучить стандартные названия классов, используемые по всему миру, незнакомые слова искать в словаре. **НЕ использовать транслит никогда !!**

Центрирование элементов с помощью фиксированных margin или padding

Причина ошибки

Чтобы задать расположение по центру кажется разумным задать одинаковые отсутпы с 2-х сторон. Однако при изменении контента это сразу перестает работать.

Последствия и вред

Неустойчивая к изменениям верстка - очень плохая верстка

Правильно

Если нужно выровнять элемент по центру лучше использовать margin: 0px auto и фиксированную ширину. Чтобы выровнять по центру элемент, ширина которого неизвестна - использовать для него display: inline-block, задав text-align: center для родителя

Элемент input не заключен в тэг form

Причина ошибки

Форма сама по себе не отображается, поэтому ее иногда забывают. На самом деле отправка запроса без формы невозможна

Последствия и вред

Ваши правила не будут работать

Правильно

Объявите форму и проставьте action

Вертикальный список через
 а не соответствующие <ul> или

Причина ошибки

Использование
 для перевода строки создает впечаление, что этот тег обеспечивает разделение контента по строкам. На самом деле лишние html-теги не должны использоваться для оформления

Последствия и вред

Нарушение семантики, неправильное копирование, сложности со стилизацией (в частности с заданием отступов)

Правильно

Использовать <ul>, <ol> или <dl> для создания списка

Использование отрицательного значения margin, чтобы убрать отступы

Причина ошибки

Простой и грязный способ который используется если по хорошему с версткой договориться не удалось. Использовать только в ситуации безысходности.

Последствия и вред

Нарушение логики блочной модели, сложности при изменении верстки, адаптивности.

Правильно

Сбрасывать отступы элементов до 0.
Для инлайн-блоков использовать хак с font-size: 0 у родительского элемента.

Забыты закрывающие теги в файле html, фигурные скобки и ; в файле css

Симптомы

Контейнер вдруг оказался совсем не там, где вы ожидали, часть кода не работает, а текст, внезапно стилизован абсолютно не по вашим правилам.

Причина ошибки

Мы ленимся или попросту забываем закрывать все элементы сразу.

Совет

Если закрывающие теги и скобочки все равное забываются, редакторы кода такие как, например, Sublime 3 или Notepad++ подставят их за вас :)

Стили для однотипных элементов продублированы для каждого в css файле

Пример

```
.top-stripe{
    height: 40px;
    margin:20px auto;
    width: 80%;
    background-color: #444;
    color: #eee;
}
```

```
.bottom-stripe{
    height: 40px;
    margin:20px auto;
    width: 80%;
    background-color: #fc3;
    color: #eee;
}
```

Последствия и вред

Избыточность кода, сложность обновления, сложность прочтения, ненависть коллег

Правильно

Писать меньше стилей, используя больше классов и более сложные селекторы.

Не подключен указанный в стилях нестандартный шрифт

Причина ошибки

Если у нас отображается нужный шрифт, почему он не будет отображаться у пользователя?

Последствия и вред

Браузер будет отображать страничку с использованием стандартных шрифтов типа Times New Roman.

Способы подключения

1. **Google Web Fonts**

   Выбираем из коллекции

   или

   Загружаем шрифты

2. Добавляем в начало css-файла правило:

```
@font-face {
    font-family: 'Ubuntu';
    src:local('Ubuntu');
    url:('http://.....');
}
```

3. Указываем имя шрифта элементу:

```
div {
    font-family: 'Ubuntu',
    sans-serif;
}
```

Текст оформлен тегами <b> и <i>

Причина ошибки

*Если спросить у поисковика “как оформить текст жирным”, первые же ссылки в выдаче предлагают использовать тег **<b>**.*

Теги **<b>** и **<i>** всем известны, понятны и отлично работают, зачем использовать другие?

Последствия и вред

Любое оформление в html-коде это ошибка. Использовать **<b>**, чтобы просто сделать текст жирным - это неправильно. Дело принципа.

Правильно

Другое дело, если вы хотите выделить тот или иной фрагмент текста, потому что он важный, тогда можно использовать теги **** и ****. Это правильно, и, это учитывается поисковиками, для всего остального есть css-стили.

Изображениям <img> не прописан alt

Справка

Обязательный Атрибут **alt** определяет альтернативный текст для случаев, когда пользователь не может видеть изображение.

Последствия и вред

Нарушение логики блочной модели, сложности при изменении верстки, адаптивности.

Правильно

Всегда указывать атрибут **alt**, для верстальщика допустимо оставлять его пустым: alt=””

Неверно указано имя изображения: bg.jpeg вместо bg.jpg

Симптомы

Мы на 100% верно указали адрес к картинке, и, она точно находится в указанной папке, но на экране все равно не отображается

Причина ошибки

Путаница в форматах JPG, jpeg, jpg

Совет

Обращайте внимание, каком формате сохранен нужный файл и будет вам счастье :)

Элемент оказался не активен по наведению из-за неуказанного z-index

Причина ошибки

Позиционируя элементы, мы не всегда обращаем внимание на то, что некоторые элементы могут перекрывать друг-друга.

Последствия и вред

Перекрытые элементы могут отображаться как и должны, но при этом не функционировать.

Правильно

Если есть необходимость в наслоении элементов, необходимо указать им правило **position: absolute, fixed** или **relative**, и задать значение **z-index**: наименьшее элементу, который должен оказаться на заднем плане и набольшее - элементу на переднем плане. А после этого проверить - все ли работает так, как нужно :)

Хранение всех файлов верстки в одной папке

Причина ошибки

“Мне удобно, когда все лежит в одном месте!”

Последствия и вред

Хранение изображений, стилевых файлов, файлов со скриптами и прочих в одной папке грозит тотальной путаницой и ненавистью тех, кому дальше с этим придется работать.

Правильно

Раскладываем файлы по папкам: стилевые изображения в *img/*, контентные изображения в *pic/*, таблицы стилей в *css/*, шрифты в *fonts/*, скрипты в *js/* и прочие используемые файлы должны храниться в разных директориях.

Забыл проверить, как работает в IE :(

Причина ошибки

Мы радуемся тому, что в *нормальном* браузере наша верстка выглядит отлично, и забываем проверить кроссбраузерность.

Причина ошибки

Реальность такова: разные браузеры могут по-разному отображать вашу верстку, а в некоторых она рушится вовсе. Недоволен клиент, недовольны вы.

Правильно

Проверяем во всех браузерах:

Очумелые ручки, или набор юного верстальщика

Генераторы

Изображения

1. Градиенты CSS http://www.colorzilla.com/gradient-editor/
2. Генератор спрайтов http://apps.stupid-studio.com/
3. Еще один генератор спрайтов http://spritepad.wearekiss.com/
4. Генератор фавиконок (в т.ч. для iOS/Android/Windows 8) http://realfavicongenerator.net/
5. Полосатый фон http://www.stripegenerator.com/
6. Картинки-прелоадеры http://preloaders.net/en/circular и тут http://www.loadinfo.net/
   и даже тут http://www.webscriptlab.com/

Тени

1. Box-shadow generator
   https://developer.mozilla.org/en-US/docs/Web/CSS/Tools/Box-shadow_generator
2. Длинные тени http://sandbox.juan-i.com/longshadows/

Кнопки

1. Генератор кнопок http://css3buttongenerator.com/
2. Ещё один (большая коллекция внутри) http://www.bestcssbuttongenerator.com/
3. Сервис для создания социальных кнопок http://www.addthis.com/
4. Кнопки для email шаблонов http://buttons.cm/

Геометрические фигуры

1. Треугольники на CSS http://apps.eky.hk/css-triangle-generator/
2. Статья, примеры. Фигуры на CSS http://css-tricks.com/examples/ShapesOfCSS/
3. Генератор лент и бантиков http://livetools.uiparade.com/ribbon-builder.html
4. Продвинутый border-radius
   https://developer.mozilla.org/en-US/docs/Web/CSS/Tools/Border-radius_generator
5. Разместить изображение в border-e
   https://developer.mozilla.org/en-US/docs/Web/CSS/Tools/Border-image_generator

Разные генераторы

1. Генератор CSS http://www.createcss3.com/
2. Ещё один генератор CSS http://css3generator.com/
3. 4 в 1: градиенты, границы, текстуры, тени http://www.cssmatic.com/
4. Быстро создать сетку-шаблон http://csstemplater.com/

Очумелые ручки, или набор юного верстальщика

Шрифты и иконки

Шрифты

1. 20 вопросов про веб-шрифт. Статья http://habrahabr.ru/company/adv/blog/184864/
2. Таблица веб-безопасных шрифтов http://www.w3schools.com/cssref/css_websafe_fonts.asp
3. Безопасные шрифты http://cssfontstack.com/
4. Google Fonts http://www.google.com/fonts
5. Webfont - каталог шрифтов http://webfont.ru/
6. FontSquirrel - бесплатные шрифты и генератор шрифтов http://www.fontsquirrel.com/
7. PXtoEM - калькулятор перевода в em http://pxtoem.com/
8. Удобный предпросмотрщик шрифтов http://wordmark.it/

Иконки

1. FontAwesome. Очень адаптируемый. http://fortawesome.github.io/Font-Awesome/
2. Иконки, использующиеся в Bootstrap http://www.webhostinghub.com/glyphs/
3. Пул ссылок на иконочные ресурсы https://github.com/fontello/fontello/wiki
4. Лучший генератор иконочных шрифтов http://fontello.com/
5. Ещё генераторы, с другими наборами иконок http://fontastic.me/ и https://pictonic.co/
6. Поиск по иконкам https://www.iconfinder.com/ и http://findicons.com/
7. Иконки проектировщикам (работают в axure) http://www.justbenice.ru/studio/websymbols/

CSS-анимации

1. Animate.css - набор анимаций http://daneden.github.io/animate.css/
2. Еще эффекты http://www.justinaguilar.com/animations/index.html
3. Effeckt.css http://h5bp.github.io/Effeckt.css/dist/
4. Hover эффекты http://ianlunn.github.io/Hover/
5. Генератор переходов(transitions) http://matthewlein.com/ceaser/

Jquery плагины

1. Обширная галерея различных плагинов http://www.unheap.com/
2. Похожий ресурс http://jquerylist.com/
3. Коллекция минималистичных плагинов с кратким описанием http://microjs.com/
4. Удобная библиотека компонент http://www.uibox.in/

Разные ссылки

Чистота кода

1. Автоформатирование кода http://www.dirtymarkup.com/
2. Сортировка CSS http://csscomb.ru/
3. Анализировать код и вытащить все классы и id http://extractcss.com/
4. Проверка иерархии заголовков h1, h2… http://gsnedders.html5.org/outliner/
5. Статьи по принципам написания CSS-кода
   https://github.com/matmuchrapna/CSS-Guidelines/blob/master/README%20Russian.md
   https://github.com/necolas/idiomatic-css/tree/master/translations/ru-RU

Адаптивность, кроссбраузерность

1. Генератор адаптивных модульных сеток http://app.responsify.it/
2. Тестирование на адаптивность http://www.dimensionsapp.com/a/
3. Другой тест на адаптивность http://viewlike.us/index.php
4. Изменение размеров viewport, можно делать закладкой браузера
   http://lab.maltewassermann.com/viewport-resizer/
5. Поддержка html5 и css3 браузерами http://html5please.com/
6. Can i use… справочник по поддержке тегов и свойств http://caniuse.com/

lorem ipsum

1. Просто lorem http://ru.lipsum.com/
2. Яндекс.Рефераты http://referats.yandex.ru/
3. Контейнер для изображений http://placehold.it/

Полезные сервисы

1. Справочник по вёрстке №1 http://htmlbook.ru/
2. Генератор ASCII рисунков (используй для комментариев) http://ascii-arts.org.ua/gen.html
3. Подбор типографики http://type-scale.com/
3. Уменьшаем размер .png https://tinypng.com/
4. Быстро показать набор цветов http://colorpeek.com/
5. Быстро показать дизайн и прокомментировать https://redpen.io/
6. Быстрое прототипирование https://gomockingbird.com/mockingbird/
8. Визуальный редактор Google карт http://googlemapbuilder.mynameisdonald.com/
9. Mind maps online http://my-mind.github.io/