HTML / CSS Best Practices

Exadel®

О том, как важно писать хороший код

- Код пишется для людей.
- Хороший код ясно выражает намерения.
- Не откладывать исправление «до лучших времен». Эти «лучшие времена» никогда так и не наступают.
- Начните с себя! Стремитесь писать качественный и эффективный код сразу.

О том, как важно писать хороший код

- Архитектура, именование — ключевые моменты.
- Не создавайте хаков.
- Не пишите наугад. Полное осознание каждой строчки кода. Не понял почему работает — не комить.

Теория разбитых окон

Общие правила

- Не смешивайте символы табуляции и пробелы.

- Пишите весь код в нижнем регистре

```
<!-- Not recommended -->
<A HREF="/">Home</A>
```

```
<!-- Recommended -->
<img src="google.png" alt="Google">
```

```
/* Not recommended */
color: #E5E5E5;
```

```
/* Recommended */
color: #e5e5e5;
```

Общие правила

- Расставляйте верно скобки, пробелы и табуляцию

```
/* Not recommended:
missing space */
#video{
   margin-top: 1em;
}
```

```
/* Not recommended:
unnecessary line break
*/
#video
 {
   margin-top: 1em;
 }

/* Recommended */
#video {
   margin-top: 1em;
}
```

```
/* Not recommended */
h3 {
   font-weight:bold;
}

/* Recommended */
h3 {
   font-weight: bold;
}
```

Общие правила

- Избавляйтесь от ненужных html элементов.
- Не пишите сложные CSS селекторы.
- Будьте **последовательны**.

HTML

- HTML5.

```
<!DOCTYPE html>
```

- Пишите валидный код. Проверить на валидность можно здесь.

```
<!-- Not recommended -->
<title>Test</title>
<article>This is only a test.

<!-- Recommended -->
<!DOCTYPE html>
<meta charset="utf-8">
<title>Test</title>
<article>This is only a test.</article>
```

HTML

- Предоставляйте альтернативную версию мультимедиа контенту.

```
<!-- Not recommended -->
<img src="spreadsheet.png">

<!-- Recommended -->
<img src="spreadsheet.png" alt="Spreadsheet screenshot.”>
```

- Опускайте необязательный атрибут type у стилей и скриптов.

```
<!-- Not recommended -->
<link rel="stylesheet" href=“main.css" type="text/css">

<!-- Recommended -->
<link rel="stylesheet" href="main.css">

<!-- Not recommended -->
<script src=“main.js" type="text/javascript"></script>

<!-- Recommended -->
<script src="main.js"></script>
```

HTML

- Не смешивайте разметку и стили.

```
<!-- Not recommended -->
<!DOCTYPE html>
<title>HTML sucks</title>
<link rel="stylesheet" href="base.css" media="screen">
<link rel="stylesheet" href="grid.css" media="screen">
<link rel="stylesheet" href="print.css" media="print">
<h1 style="font-size: 1em;">HTML sucks</h1>
<p>I’ve read about this on a few sites but now I’m sure:   <u>HTML is
stupid!!1</u> <center>I can’t believe there’s no way to control the
styling of   my website without doing everything all over again!</
center>

<!-- Recommended -->
<!DOCTYPE html>
<title>My first CSS-only redesign</title>
<link rel="stylesheet" href="default.css">
<h1 class=“main-heading”>My first CSS-only redesign</h1>
<p class=“text-content”>I’ve read about this on a few sites but today
I’m actually   doing it: separating concerns and avoiding anything in
the HTML of   my website that is presentational.
<p>It’s awesome!
```

Семантика

- Пишите семантически верный код.

Пример:

```
<!-- Not recommended -->
<div onclick="goToRecommendations();">All
recommendations</div>

<!-- Recommended -->
<a href="recommendations/">All recommendations</a>
```

Семантика

- HTML5 предоставляет большой набор семантических элементов

До HTML5:

, ,

Теперь:

<header>	<footer>
<nav>	<details>
<section>	<summary>
<article>	<mark>
<aside>	<time>
<figure>	<figcaption>

Семантика

<SECTION>

- Структурный элемент самого высокого уровня.
- Обычно содержит заголовок.
- Ближе к структуре, нежели к главному содержимому.
- Для иных целей — элемент .
- Пример: секция новостей, секция контактной информации на главной странице сайта

Семантика

<ARTICLE>

- **Независимое** содержимое.
- Часто содержит содержит шапку и подвал.
- Более универсальный, чем <section>, может использоваться на разных уровнях.
- Примеры: пост, комментарий к посту, отдельный модуль.

Семантика

<ASIDE>

- Элемент, непосредственно относящийся к содержимому вокруг него.
- Используется как в контексте <article>, так и в контексте всего документа.
- Примеры: цитаты, реклама, навигация и т.п.

Семантика

<HEADER>

- Универсальная шапка для содержимого.
- Содержит заголовки (H1-H6), группы заголовков <hgroup>, форму поиска, логотип и т.п.
- Используется как в контексте отдельного блока, так и в контексте всей страницы.

Семантика

<FOOTER>

- Относится к ближайшей родительской секции.
- Содержит информацию о секции (кто написал, дата, копирайт и т.п.).
- Не обязательно в конце секции.

CSS

CSS

- Пишите валидный CSS. Используйте CSS валидатор
- Давайте понятные названия ID и классам

```
/* Not recommended: meaningless */
#yee-1901 {}

/* Not recommended: presentational */
.button-green {} .clear {}

/* Recommended: specific */
#gallery {} #login {} .video {}

/* Recommended: generic */
.aux {} .alt {}
```

CSS

- Пользуйтесь shorthand-свойствами

```
/* Not recommended */
border-top-style: none;
font-family: palatino, georgia, serif;
font-size: 100%;
line-height: 1.6;
padding-bottom: 2em;
padding-left: 1em;
padding-right: 1em;
padding-top: 0;

/* Recommended */
border-top: 0;
font: 100%/1.6 palatino, georgia, serif;
padding: 0 1em 2em;
```

CSS

- Используйте единый способ именования для ID и классов (например, через дефис для классов и camelCase для ID)

```
/* Not recommended */
.demoimage {}
.error_status {}
.ok-status {}
#video-status {}
#breakingNews {}

/* Recommended */
#videoId {}
#navBar {}
.video-sample {}
.ads-sample {}
```

CSS

- Называйте ID и классы как можно более коротко, но в то же время понятно

```
/* Not recommended */
#navigation {}
.atr {}

/* Recommended */
#nav {}
.author {}
```

- Не пишите хаки

```
*background: green;
*:first-child+html .hack {
  background: green;
}
```

Принципы хорошей архитектуры CSS

- Предсказуемость
- Повторное использование
- Поддержка
- Масштабируемость

Распространенные ошибки в CSS

- Изменение компонентов в зависимости от родителя

```
.widget {
    background:yellow;
    border:1px solid black;
    color:black;
    width:50%;
}
#sidebar .widget {
    width:200px;
}
body.homepage .widget {
    background:white;
}
```

- Слишком сложные селекторы

```
#main-nav ul li ul li div { }
#content article h1:first-child { }
#sidebar > div > h3 + p { }
```

Распространенные ошибки в CSS

- Слишком общие имена селекторов

```
.widget {}
.widget .title {}
.widget .contents {}
.widget .action {}
```

- Правило делает слишком много

```
.widget {
    position:absolute;
    top:20px;
    left:20px;
    background-color:red;
    font-size:1.5em;
    text-transform:uppercase;
}
```

Принципы хорошей архитектуры CSS

CSS должен как можно меньше «знать» про структуру HTML.

- Точное описание

```
/* Граната */                    /* Снайперская винтовка */
#main-nav ul li ul { }           .subnav { }
```

- Разделение ответственности

`background`, `color` и `font` в одном правиле с `position`, `width`, `height` и `margin`

- Использование пространства имен

```
/* Not recommended */ .widget { } .widget .title { }
/* Recommended */ .widget { } .widget-title { }
```

- Расширение компонентов модификаторами классов

```
/* Not recommended */ .widget { } #sidebar .widget { }
/* Recommended */ .widget { } .widget-sidebar { }
```

БЭМ

БЭМ (Блок, Элемент, Модификатор) — методология, в основе которой лежат следующие принципы:

- Дизайн сайта может измениться в любой момент, надо быть к этому готовым.
- HTML/CSS-разметка развивается вместе с дизайном и готова к его изменению.
- Программист вместе с верстальщиком работают над кодом сайта, дополняя код друг друга.

БЭМ

Блоки

- Блок — независимая сущность, самостоятельная часть интерфейса.

Пример: заголовок, кнопка, навигационная панель.

BLOCK text-input

CSS класс: .b-text-input

БЭМ

Элементы

- Элемент является частью блока, связан с блоком функционально и семантически.
- Не существует вне блока.
- Не все блоки имеют элементы.

Пример: пункты меню на навигационной панели, иконки на кнопках

BLOCK text-input

ELEMENT text-field

ELEMENT label

CSS классы:

.b-text-input__text-field

.b-text-input__label

Модификаторы

- Модификаторы — флаги для блоков или элементов.
- Определяют свойство или состояние.
- Один блок или элемент может иметь несколько модификаторов.

Пример: размер кнопки, состояние кнопки, скрыт или отображается.

BLOCK text-input
MOD multiline
MOD disabled
ELEMENT text-field
ELEMENT label

CSS класс:
.b-text-input_disabled

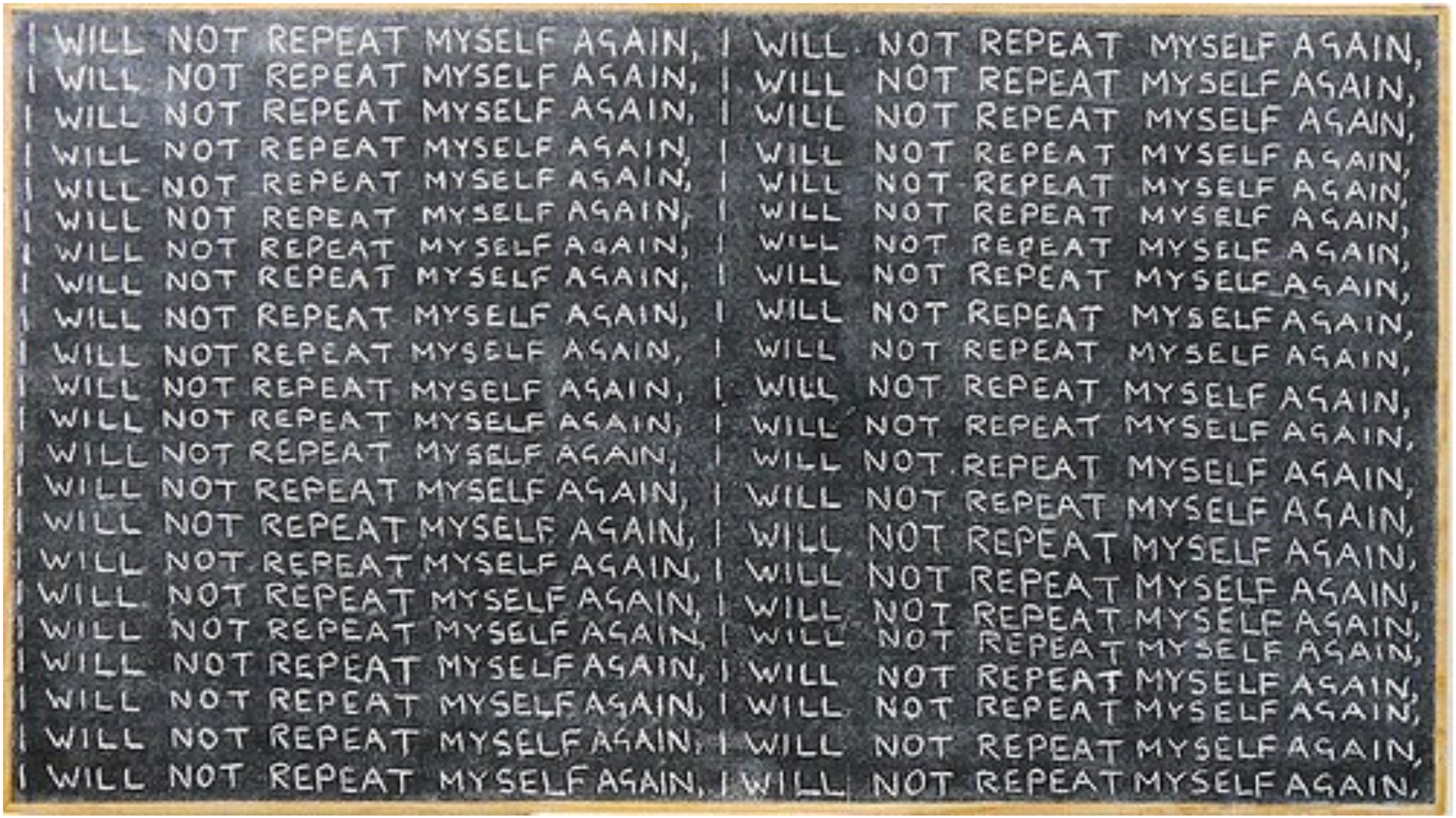
БЭМ

CSS

```
b-popup
  _hidden
  _size _big
        _medium
       _large
  _dir _left
       _right
      _top
      _bottom
_color-scheme _dark
              _light

  __anchor-node
  __popup-box
  __close-btn
  __controls
  __ok
  __cancel
```

Препроцессоры



Препроцессоры

Без них:

```
.header { background-color: #333; }
.col-left { background-color: #333; }
.content { background-color: #333; }
.footer { background-color: #333; }
```

Препроцессоры

С ними:

```
$bg-color: #333;

.header { background-color: $bg-color; }
.col-left { background-color: $bg-color; }
.content { background-color: $bg-color; }
.footer { background-color: $bg-color; }
```

Препроцессоры

- Помогают писать CSS **быстрее**
- Помогают писать **плохой** CSS **быстрее** ;)

SASS
LESS
Stylus
Closure Stylesheets

Препроцессоры

- Переменные

```
$primary-color: #333;
body { color: $primary-color; }
```

- Вложенности

```
nav {
   color: #000;
   ul { margin: 0; }
}
=>
nav { color: #000; }
nav ul { margin: 0; } // результат
```

Препроцессоры

- Примеси

```
@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
  -moz-border-radius: $radius;
  -ms-border-radius: $radius;
  border-radius: $radius;
}
.box {
  @include border-radius(10px);
}
```

Препроцессоры

- Наследование

```
.message {
    border: 1px solid #ccc;
    padding: 10px;
    color: #333;
}
.success {
    @extend .message;
    border-color: green;
}
.error {
    @extend .message;
    border-color: red;
}
```

Препроцессоры

- Операторы

```
.container { width: 100%; }

article[role="main"] {
    float: left;
    width: 600px / 960px * 100%;
}

aside[role="complimentary"] {
    float: right;
    width: 300px / 960px * 100%;
}
```

Анализаторы кода

HTML Inspector

- написан на JavaScript и запускается в браузере
- правила можно изменять, добавлять новые
- интегрируется с системой сборки

Проверяет:

- валидность:
  - правильное использование элементов, атрибутов
  - отсутствие повторяющихся ID
- хорошие практики:
  - отсутствие обработчиков событий, установленных как атрибуты
    `<input id="b1" value="Win gold medal" onclick="alert('Phelps');" type="button"/>`
  - нет не используемых классов, элементов

и др.

```
⚠ The 'bgcolor' attribute is no longer valid on the <body> element and should not be used. ▶ body
⚠ <style> elements outside of <head> must declare the 'scoped' attribute. ▶ style
⚠ The <hgroup> element is obsolete and should not be used. ▶ hgroup
⚠ The class 'post' is used in the HTML but not found in any stylesheet. ▶ article.post
⚠ The 'alt' attribute is required for <img> elements. ▶ img
⚠ The <il> element is not a valid HTML element. ▶ il
⚠ The 'action' attribute is required for <form> elements. ▶ form
```

Анализаторы кода

- CSS Lint — интегрируется с системой сборки

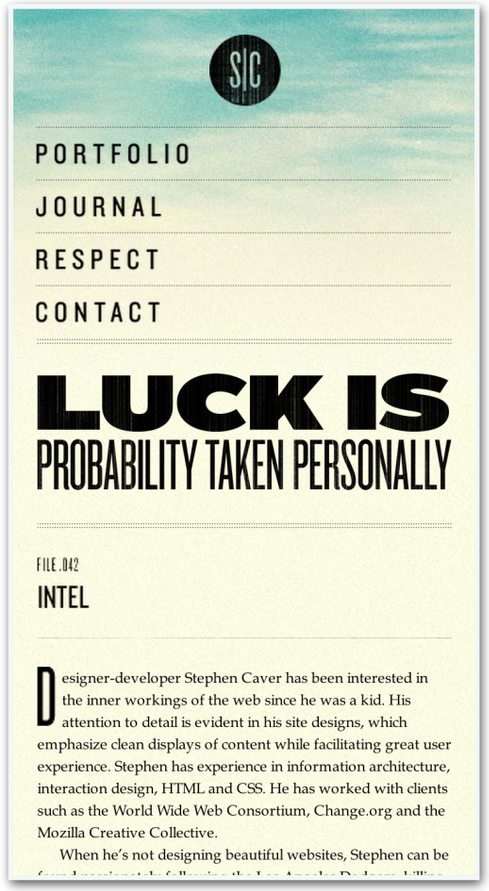
  Примеры возможных правил:
  - запретить использование !important;
  - запретить использование ID в селекторах `#menu { ... }`
  - запретить пустые правила .sidebar { }
  - запретить единицы измерения для нулевых значений .sidebar { width: 0px; }

- csscss — анализирует повторяющиеся правила

```
{.contact .content .primary} and {article,
#comments} share 5 rules
```

Responsive design

Responsive design — дизайн, адаптирующийся под конкретное устройство, его размеры, ориентацию.



Responsive design

Основные принципы

- «Гибкая» разметка
- «Гибкие» изображения
- Использование media queries

  ```
  @media screen and (min-width:500px) { ... }
  @media tv and (min-width: 700px) and (orientation: landscape) { ... }
  ```

- Прогрессивное улучшение

Доступность

- Используйте альтернативный вариант для любого нетекстового контента
  Пример: alt тэг для изображений, описание видео или аудио и т.д.
- Используйте HTML тэги по назначению
  Пример: не используйте тэги H1, H2, … только для форматирования/увеличения текста
- Указывайте язык содержимого
- Используйте достаточно контрастные цвета (фон и текст или изображения на нем)
- Разделяйте содержимое и представление
- Не разрабатывайте под конкретное устройство
- Предсказуемое поведение, интуитивно понятная навигация
- Следуйте рекомендациям W3C

Сжатие ресурсов

YUI Compressor

До:

```
/*****   Multi-line comment before a new class name *****/
.classname {
     /* comment in declaration block */
     font-weight: normal;
}
.empty { ;}
.nonempty {border: 0;}
.classname {
     border: none;
     background: none;
     outline: none;
}
```

После:

```
.classname{font-weight:normal}
.nonempty{border:0}
.classname{border:0;background:0;outline:0}
```

Сжатие ресурсов

CSSO (CSS-optimizer)

- Минимизация без изменения структуры
  - Удаление пробелов
  - Удаление концевых ‘;’
  - Удаление комментариев
  - Минимизация margin и padding
  - Минимизация font-weight

    и прочее

- Минимизация с изменением структуры
  - Слияние блоков с одинаковыми селекторами
  - Удаление перекрываемых свойств

    и прочее

Пример:

```
.test { color: red } .test { color: green }
=> .test{color:green}
```

Полезные ссылки

- Robert C. Martin «Clean Code»

- Accessibility:
  - HTML Techniques for Web Content Accessibility Guidelines
  - http://accessibility.psu.edu/

- Семантический HTML

- CSS:
  - http://philipwalton.com/articles/css-architecture/

- Responsive web design

- Материалы, используемые в подготовке лекции:
  - Доклад «Вёрстка со смыслом. Новая семантика HTML5»
  - Google guidelines

The End

Exadel®