

HTML/ CSS Base

Ильин Антон

DIGITAL
DESIGN

План

- Структура HTML документа
- HTML-элементы и их типы
- HTML 5
- Подключение CSS стилей к документу
- Синтаксис CSS
- CSS селекторы
- CSS Flexbox

КЛЮЧЕВЫЕ ПОНЯТИЯ

- Элемент — это компонент страницы, отдающий браузеру какую-либо команду (например, создать абзац или вставить изображение);
- Элемент состоит из пары тегов и содержимого. Некоторые элементы не имеют содержимого в принципе, они записываются одиночным самозакрывающимся тегом;
- Атрибут — модификатор элемента. Записывается в открывающем теге. Позволяет уточнить поведение элемента (например, указать выравнивание в абзаце или ссылку на конкретное изображение). Может быть как



ОБЩАЯ СТРУКТУРА

- DOCTYPE — тип документа (HTML/HTML5/XHTML);
- html — корневой элемент;
- head — заголовочная секция;
- body — основная секция (тело страницы);

```
<!DOCTYPE html>

<html>
  |
  | <head>
  |   |
  |   | <title>
  |   |   |
  |   |   | </title>
  |   |   |
  |   |   | </head>
  |   |   |
  |   |   | <body>
  |   |   |   |
  |   |   |   | </body>
  |   |   |   |
  |   |   |   | </html>
```

Doctype

- Указание doctype должно быть в самом верху любого HTML документа, перед тегом <html>.
- Doctype - это не HTML тег. Это инструкция браузеру про

версию языка разметки страницы.

Doctype

Code

HTML 4.01 Frameset

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD  
HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.  
dtd">
```

HTML 4.01 Strict

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD  
HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd"  
>
```

HTML 4.01 Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD  
HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd"  
>
```

HTML 5

```
<!DOCTYPE html">
```

HTML-элементы и их типы

В HTML практически все элементы относятся к двум типам, различающихся своим поведением и свойствами — это блочные и строчные элементы.

Блочные элементы предназначены для **структурирования** основных частей вашей страницы, путём разделения содержимого на логически связанные блоки.

Примеры блочных элементов: `<div>`, `<p>`, ``, ``, `<h1>` и т. д.

Строчные элементы предназначены, чтобы разграничить часть текста и придать ему определённую функцию или смысл. Строчные элементы, как правило, содержат одно или несколько слов.

Примеры строчных элементов: `<a>`, ``, ``, ``, `` и т. д.

display: inline-block

Если для формирования секций использовать блочный элемент, то он будет каждый раз начинаться с новой строки. Для строчных элементов нельзя задавать размеры всей секции. Одним из решений является применение строчно-блочных элементов, которые сочетают преимущества строчных и блочных элементов.

Строчно-блочный элемент можно определить, задав ему в стилях свойство `display` со значением `inline-block`.

Кроме того, в HTML существует ряд элементов, для которых значение `inline-block` задано по умолчанию: `<button>`, `<input>`, `<textarea>`, `<select>`.

Характеристики этих элементов следующие.

- Высота и ширина элемента вычисляется браузером автоматически, исходя из содержимого блока.
- Размеры содержимого можно устанавливать через свойства `width` и `height`.
- Ширина блока получается сложением значений `width`, `margin`, `border` и `padding`.
- Высота блока получается сложением значений `height`, `margin`, `border` и `padding`.
- Несколько элементов идущих подряд располагаются на одной строке и переносятся на другую строку при необходимости.
- Элементы можно выравнивать по вертикали с помощью свойства `vertical-align`.
- Перенос текста считается за пробел.

Table

Таблица состоит из строк и столбцов ячеек, которые могут содержать текст и рисунки. Обычно таблицы используются для упорядочения и представления данных, однако возможности таблиц этим не ограничиваются.

| | |
|-----------------|-----------------|
| Ячейка 1 | Ячейка 2 |
| Ячейка 3 | Ячейка 4 |

Table

Для добавления таблицы на веб-страницу используется тег **<table>**. Этот элемент служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются соответственно с помощью тегов **<tr>** и **<td>**. Таблица должна содержать хотя бы одну ячейку. Допускается вместо тега **<td>** использовать тег **<th>**. Текст в ячейке, оформленной с помощью тега **<th>**, отображается браузером шрифтом жирного начертания и выравнивается по центру ячейки. В остальном, различия между ячейками, созданными через теги **<td>** и **<th>** нет.

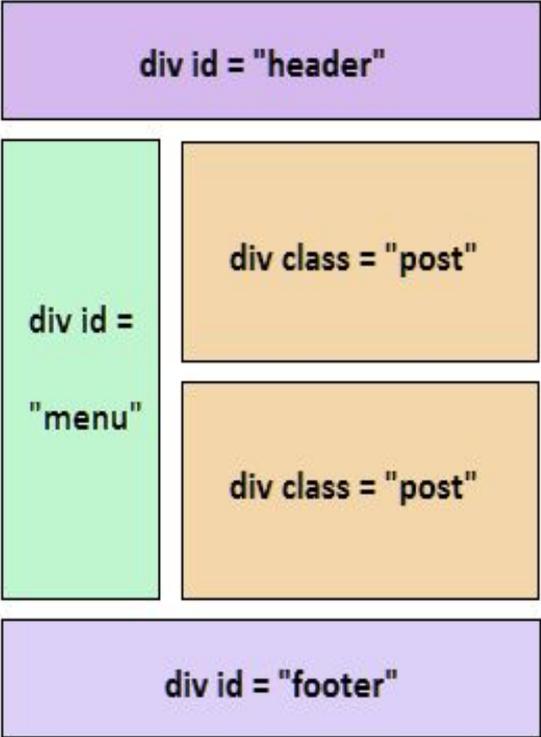
```
<table border="1" width="100%" cellpadding="5">
  <tr>
    <th>Ячейка 1</th>
    <th>Ячейка 2</th>
  </tr>
  <tr>
    <td>Ячейка 3</td>
    <td>Ячейка 4</td>
  </tr>
</table>
```

HTML



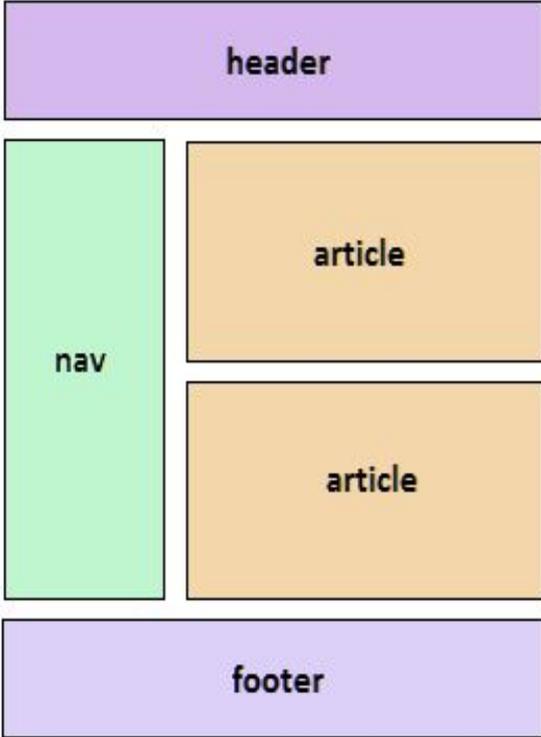
СЕМАНТИКА В HTML 5

HTML4



VS

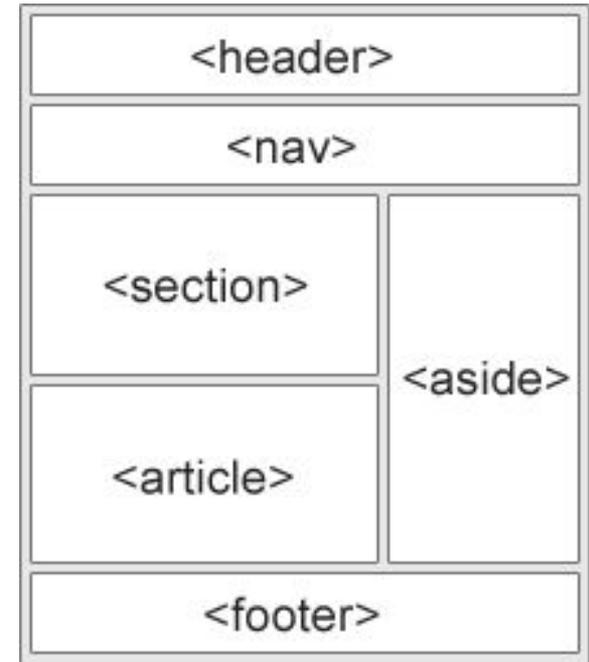
HTML5



СЕМАНТИКА В HTML 5

HTML5 вводит ряд новых семантических элементов, предназначение которых определять блоки различных частей веб-страницы:

- `<article>`
- `<aside>`
- `<details>`
- `<figure>`
- `<footer>`
- `<header>`
- `<main>`
- `<mark>`
- `<nav>`
- `<section>`



ПОДДЕРЖКА БРАУЗЕРАМИ HTML5

| | Chrome | Opera | Firefox | Edge | Safari |
|----------|--------|-----------|---------|-------------------|----------|
| Upcoming | 68 528 | | 60 497 | 18 496 | 11.2 477 |
| Current | 66 528 | 45 518 | 59 491 | 17 492 | 11.1 471 |
| Older | 65 528 | 37 489 | 58 486 | 16 476 | 11 452 |
| | 64 528 | 30 479 | 57 486 | 15 473 | 10.1 406 |
| | 63 528 | 12.10 309 | 56 478 | 14 460 | 10.0 383 |
| | 62 528 | | 55 478 | 13 433 | 9.1 370 |
| | 61 526 | | 54 474 | 12 377 | 9.0 360 |
| | 60 523 | | 53 474 | Internet Explorer | 8.0 354 |
| | | | | 11 312 | |

<http://html5test.com/results/desktop.html>

CSS



ПРЕИМУЩЕСТВА ИСПОЛЬЗОВАНИЯ CSS

- Разделение структуры документа от стиля отображения. CSS можно написать независимо от HTML.
- Поддержка большим количеством устройств (в том числе телефоны и принтеры).
- Оптимизация производительности. CSS файлы кэшируются браузерами.

ПОДКЛЮЧЕНИЕ CSS СТИЛЕЙ К ДОКУМЕНТУ

- Inline (атрибут `style` в HTML-элементе);
- Тег `<style>` в HTML-документе
- Подключение внешнего CSS файла при помощи тега `<link>`
- Подключение внутри CSS файла (использование `@import`)

БАЗОВЫЙ СИНТАКСИС CSS

- Любой стиль (кроме inline) состоит из селектора и набора правил, заданных в формате свойство: значение;
- Селектор описывает параметры элементов, к которым будут применяться описанные в его теле правила;
- В качестве селекторов могут использоваться теги, классы и другие характеристики.

```
селектор  
{  
  свойство1: значение1;  
  свойство2: значение2;  
  ...  
  свойствоN: значениеN;  
}
```

```
body {  
  font-family: 'Times New Roman';  
  color: yellow;  
  background: green;  
}  
  
.active-link {  
  color: red;  
}
```

КОММЕНТАРИИ В CSS

- Добавление пояснений по поводу использования того или иного стилевого свойства;
- Разделение на блоки;
- Собственные заметки;
- Позволяют легко вспомнить логику и структуру селекторов;
- Повышают разборчивость кода.

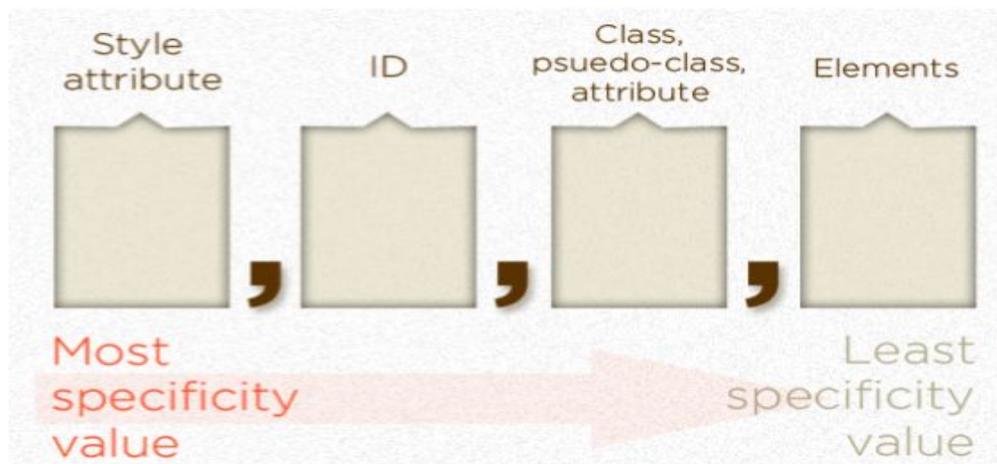
```
/*  
    комментарий  
*/  
body {  
    font-family: 'Times New Roman';  
    color: yellow;  
    background: green;  
}  
  
.active-link {  
    color: /* комментарий */ red;  
}
```

СЕЛЕКТОРЫ CSS

| Описание | Пример |
|------------------------|---|
| Универсальный селектор | <code>* { color: red; }</code> |
| Элемент | <code>p { font-family: Garamond, serif; }</code> |
| Класс | <code>.note { color: red; background: yellow; }</code> |
| Идентификатор | <code>#paragraph1 { margin: 0; }</code> |
| Атрибут | <code>[href="http://google.com"]{ font-weight: bold; }</code> |
| Потомок | <code>div p { color: red; }</code> |
| Дочерний элемент | <code>.note > b { color: green; }</code> |
| Родственный элемент | <code>h1 ~ p { font-size: 24pt; }</code> |
| Соседний элемент | <code>h1 + p { font-size: 24pt; }</code> |
| Псевдо-класс | <code>a:active { color: yellow; }</code> |
| Псевдо-элемент | <code>.new::before { content: '!'; color: red; }</code> |

КАСКАДИРОВАНИЕ

- Аббревиатура CSS расшифровывается как Cascading Style Sheets (каскадные таблицы стилей);
- Под каскадом понимается одновременное применение разных стилевых правил к элементам документа.



- <http://htmlbook.ru/samcss/kaskadirovanie>

ИСПОЛЬЗОВАНИЕ !IMPORTANT

- Играет роль в том случае, когда пользователи подключают свою собственную таблицу стилей;
- Если возникает противоречие, когда стиль автора страницы и пользователя для одного и того же элемента не совпадает, то !important позволяет повысить приоритет стиля.

```
<p id="earth"> Lorem ipsum <p>  
<p style="color: green;" > Dolor sit amet <p>
```

```
#earth {  
  color: blue;  
}
```

Lorem ipsum

Dolor sit amet

```
#earth {  
  color: blue;  
}  
  
p {  
  color: red !important;  
}
```

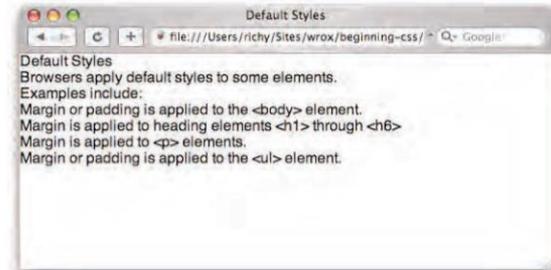
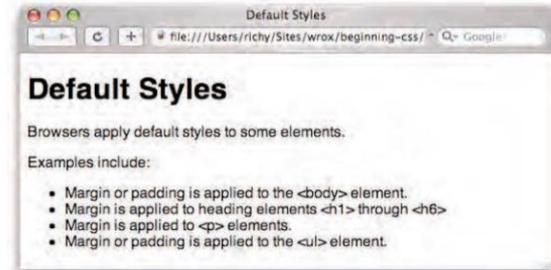
Lorem ipsum

Dolor sit amet

Best Practice#1: Используйте CSS Reset

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

```
body{  
    margin:0;  
    padding:0;  
}  
h1{  
    margin:0;  
    font-weight:normal;  
    font-size:16px;  
}  
p{  
    margin:0;  
}  
ul{  
    margin:0;  
    padding:0;  
    list-style:none  
}
```



Best Practice#2: Создавайте переиспользуе мые стили

Так **не надо** делать

```
#column1 {  
  float: left;  
  width: 500px;  
}  
#column2 {  
  float: right;  
  width: 500px;  
}
```

Так **надо** делать

```
.column {  
  width: 500px;  
}  
.float-left {  
  float: left;  
}  
.float-right {  
  float: right;  
}  
<div class="row">  
  <div class="column float-left">  
    here's a left column!  
  </div>  
</div>
```

Best Practice #3:
Не
повторяйтесь

Так **не надо** делать

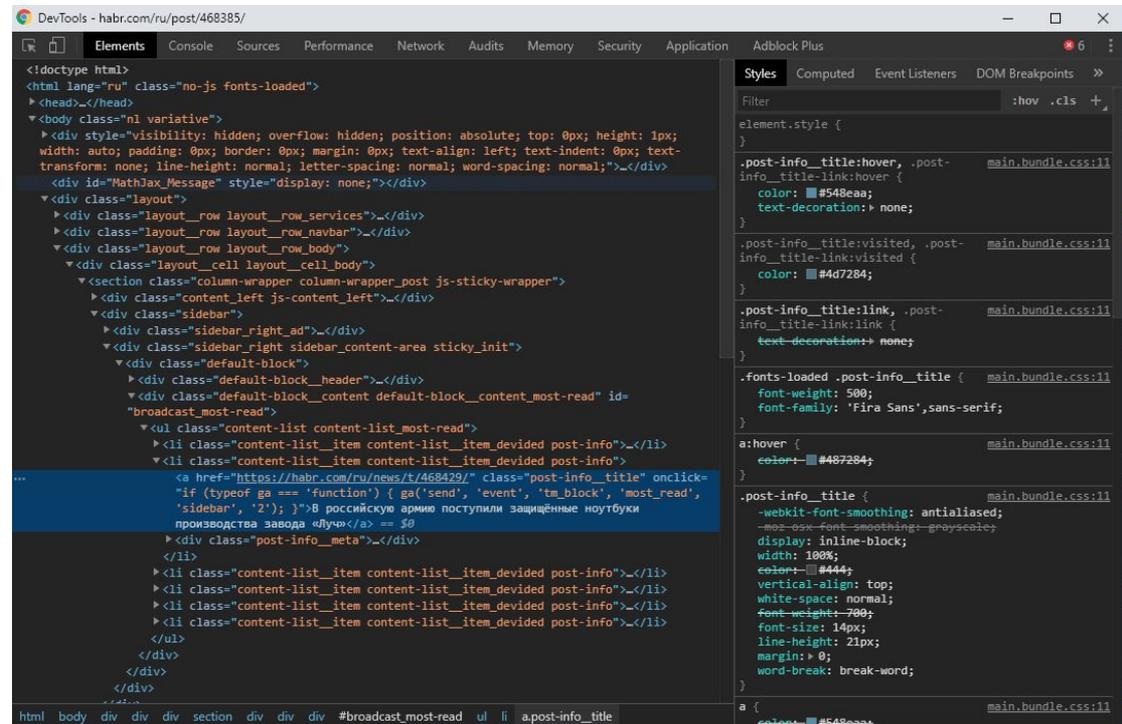
```
h1 {  
  color: #dadada;  
  margin: 10px 10px;  
  font-family: sans-serif;  
}  
h2 {  
  color: #dadada;  
  margin: 10px 10px;  
  font-family: sans-serif;  
}  
h3 {  
  color: #dadada;  
  margin: 10px 10px;  
  font-family: sans-serif;  
}
```

Так **надо** делать

```
h1, h2, h3 {  
  color: #dadada;  
  margin: 10px 10px;  
  font-family: sans-serif;  
}
```

DEVELOPER TOOLS

Developer Tools — удобный инструмент вебразработки. Он содержит в себе необходимый функционал для создания и отладки полноценных веб-приложений.

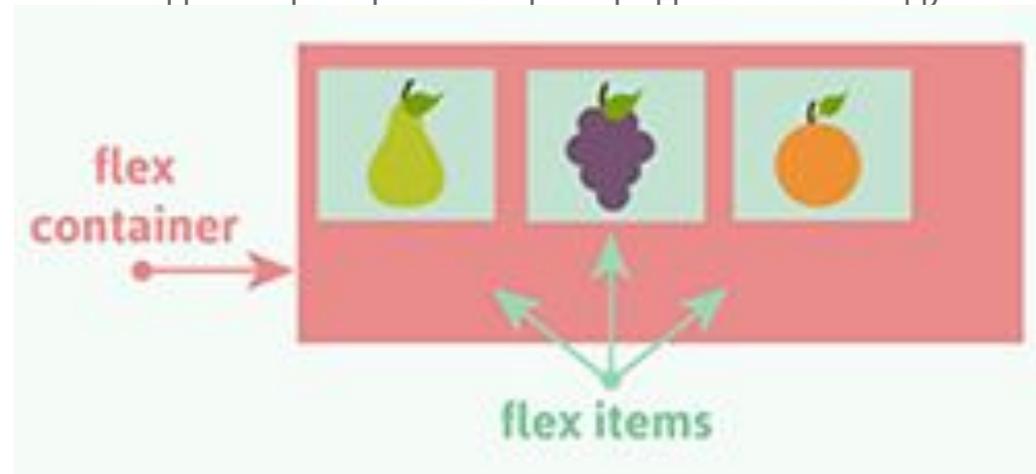


Flexbox



Flexbox

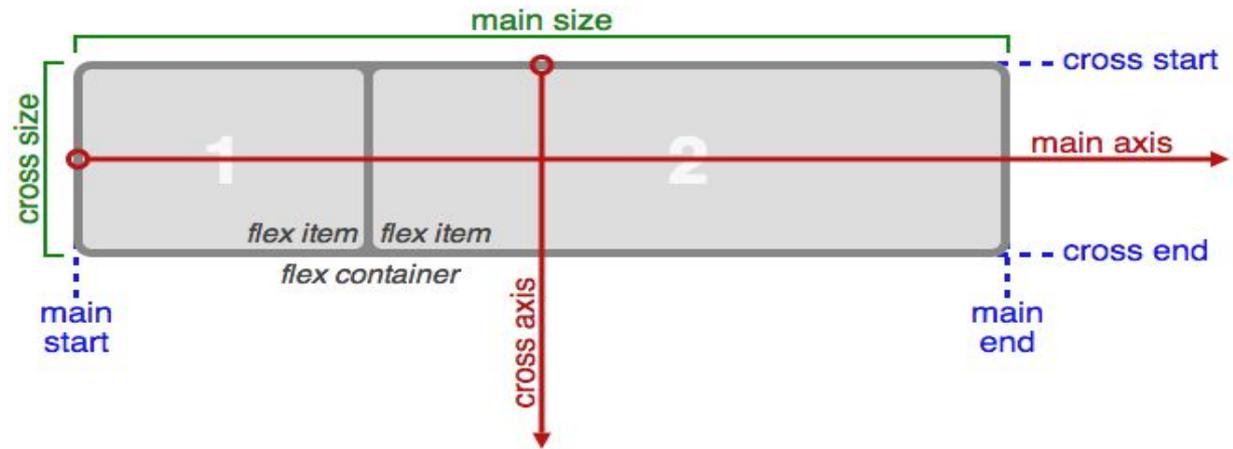
- **CSS flexbox** (*Flexible Box Layout Module*) — модуль макета гибкого контейнера — представляет собой способ компоновки элементов, в основе лежит идея оси.
- Flexbox состоит из **гибкого контейнера (flex container)** и **гибких элементов (flex items)**. Гибкие элементы могут выстраиваться в строку или столбик, а оставшееся свободное пространство распределяется между ними



- <https://frontender.info/a-guide-to-flexbox/>

Flexbox

Элементы можно располагать по строкам или столбцам в прямом или обратном порядке с контролем над направлением переноса. Нельзя использовать: `columns`, `float`, `clear` и `vertical-align`



ПОЛЕЗНЫЕ ССЫЛКИ

- <http://htmlbook.ru/>
- <http://www.webremeslo.ru/html/glava0.html>
- http://professorweb.ru/my/html/html5/level1/html5_index.php
- <https://validator.w3.org/>
- CSS справочник: <http://htmlbook.ru/css>
- Поддержка CSS стилей различными браузерами: <http://caniuse.com/>
- <https://code.tutsplus.com/tutorials/30-css-best-practices-for-beginners--net-6741>
- <https://flexboxfroggy.com/#ru>

Контакты



www.digdes.ru

info@digdes.ru

Санкт-Петербург

наб. реки Смоленки, д. 33

телефон: +7 812 346 58 33

Москва

Варшавское шоссе, д. 36, стр. 8

телефон: +7 499 788 74 94

