|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИиППО) | |

|  |  |  |  |
| --- | --- | --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №4** | | | |
| **по дисциплине** | | | |
| **«Разработка клиентских частей интернет-ресурсов»** | | | |
| Выполнил студент группы ИКБО-13-19 | | Стоянова А.Г. | |
|  | |  | |
| Принял  *Ассистент* | | Братусь Н.В. | |
| Практическая работа выполнена | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | (подпись студента) | |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_2020 г. | | (подпись руководителя) | |
|  |  | |  | |

Москва 2020

**Оглавление**

[Практическая работа №4: «Работа с внешними таблицами стилей (CSS)» 3](#_Toc55219597)

[Цель работы 3](#_Toc55219598)

[Практическое задание №1: Работа с внешними таблицами стилей CSS 3](#_Toc55219599)

[Практическое задание №2: Работа с внешними таблицами стилей CSS 7](#_Toc55219600)

[Практическое задание №3: Работа с внешними таблицами стилей CSS 11](#_Toc55219601)

[Практическое задание №4: Работа с внешними таблицами стилей CSS 16](#_Toc55219602)

[Практическое задание №5: Работа с внешними таблицами стилей CSS 19](#_Toc55219603)

[Практическое задание №6: Работа с внешними таблицами стилей CSS 21](#_Toc55219604)

[Практическое задание №7: Работа с внешними таблицами стилей CSS 24](#_Toc55219605)

[Практическое задание №8: Работа с внешними таблицами стилей CSS 27](#_Toc55219606)

[Практическое задание №9: Работа с внешними таблицами стилей CSS 31](#_Toc55219607)

[Практическое задание №10: Работа с внешними таблицами стилей CSS 37](#_Toc55219608)

[Практическое задание №11: CSS3-селекторы 41](#_Toc55219609)

[Вывод 44](#_Toc55219610)

# Практическая работа №4: «Работа с внешними таблицами стилей (CSS)»

Цель работы**:** ознакомиться со структурой языка HTML, научиться создавать HTML-документы, оформлять комментарии в коде, работать с основными тегами языка, добавлять изображения и ссылки на страницы, подключать внешние CSS-файлы и придавать простейшие стили HTML-документам.

## Практическое задание №1: Работа с внешними таблицами стилей CSS

**Задание:**

Используя CSS-коды спецсимволов вывести:

1. ‘Левая одиночная и правая одиночная кавычки’
2. ‚Нижняя одиночная кавычка‚
3. “Левая и правая двойные кавычки”
4. „Нижняя двойная кавычка„
5. «Левая и правая двойные угловые кавычки»

Добавить комментарии к кодам с описанием самих символов, разные стили для каждой из вышеперечисленных строк.

**Описание выполнения задания**

1. Создание HTML-страницы в редакторе Notepad++, которая содержит CSS-коды спецсимволов. (Листинг 1.1).

Листинг 1.1 – Создание HTML-страницы с CSS-кодами спецсимволов

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

<style>

.one {

text-shadow: 2px 4px 3px rgba(0, 0, 0, 0.3);

color: darkblue;

}

.two {

color: rgba(0, 0, 0, 0.6);

text-shadow: 2px 2px 3px rgba(255, 255, 255, 0.1);

}

.three {

color: darkred;

text-shadow: 6px 6px 0px rgba(0, 0, 0, 0.2);

}

.four {

color: darkslateblue;

text-shadow: 4px 3px 0px #fff, 9px 8px 0px rgba(0, 0, 0, 0.15);

}

.five {

color: forestgreen;

text-shadow: 0px 3px 0px #b2a98f,

0px 14px 10px rgba(0, 0, 0, 0.15),

0px 24px 2px rgba(0, 0, 0, 0.1),

0px 34px 30px rgba(0, 0, 0, 0.1);

}

p.one:after {

content: '\2019';

}

p.one::before {

content: '\2018';

}

p.two:after {

content: '\201A';

}

p.two::before {

content: '\201A';

}

p.three:after {

content: '\201C';

}

p.three::before {

content: '\201D';

}

p.four:after {

content: '\201E';

}

p.four::before {

content: '\201E';

}

p.five:after {

content: '\BB';

}

p.five::before {

content: '\AB';

}

</style>

</head>

<body>

<p class="one">Левая одиночная и правая одиночная кавычки</p>

<!--Левая одиночная и правая одиночная кавычки -->

<p class="two">Нижняя одиночная кавычка</p> <!-- Нижняя одиночная кавычка-->

<p class="three">Левая и правая двойные кавычки</p>

<!--Левая и правая двойные кавычки -->

<p class="four">Нижняя двойная кавычка</p>

<!--Нижняя двойная кавычка -->

<p class="five">Левая и правая двойные угловые кавычки</p>

<!--Левая и правая двойные угловые кавычки -->

</body>

</html>

Результат выполнения кода на странице (Рисунок 1.1):

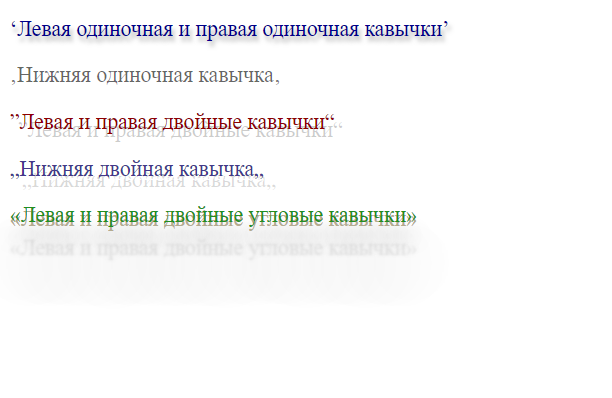


Рисунок 1.1 – Результат выполнения кода с созданием HTML-страницы с CSS-кодами спецсимволов

## Практическое задание №2: Работа с внешними таблицами стилей CSS

**Задание:**

Создать список ключевых слов с оформлением (использовать свойства *text-decoration* со значениями underline, line-through, overline (к одному из ключевых слов одновременно применить два эффекта), *text-decoration-style* с разными значениями, *text-decoration-color* с разными цветами, задать шрифты и выделение тексту).

Пример списка (нужно создать свой):

1. **А**

* Авангард

1. **Б**

* Большие данные

1. *В*

* *Винчестер*

1. ***Г***

* *Гигабайт*

**Описание выполнения задания**

1. Создание HMTL-документа со списком ключевых слов с оформлением (Листинг 2.1).

Листинг 2.1 – Создание HTML-документа со списком ключевых слов

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

<style>

.i {

color: fuchsia;

}

.g {

color: gold;

}

.a {

color: green;

}

.m {

color: indigo;

}

.ul {

text-align: left;

}

.one {

color: indigo;

text-decoration: line-through underline;

text-decoration-style: solid;

text-decoration-color: fuchsia;

}

.two {

color: fuchsia;

text-decoration: line-through;

text-decoration-style: double;

text-decoration-color: gold;

}

.three {

color: gold;

text-decoration: overline;

text-decoration-style: dotted;

text-decoration-color: green;

}

.four {

color: green;

text-decoration: underline;

text-decoration-style: wavy;

text-decoration-color: indigo;

}

</style>

</head>

<body>

<ol type="I">

<li class="a">A</li>

<ul>

<li class="one">Apple</li>

</ul>

<li class="g">G</li>

<ul>

<li class="two">Google</li>

</ul>

<li class="i">I</li>

<ul>

<li class="three">IBM</li>

</ul>

<li class="m">M</li>

<ul>

<li class="four">Microsoft</li>

</ul>

</ol>

</body>

</html>

Результат выполнения кода на странице (Рисунок 2.1):



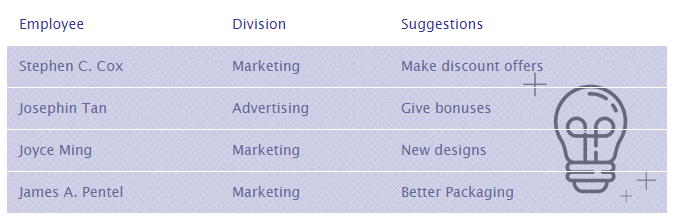
Рисунок 2.1– Результат выполнения кода с созданием списка ключевых слов

## Практическое задание №3: Работа с внешними таблицами стилей CSS

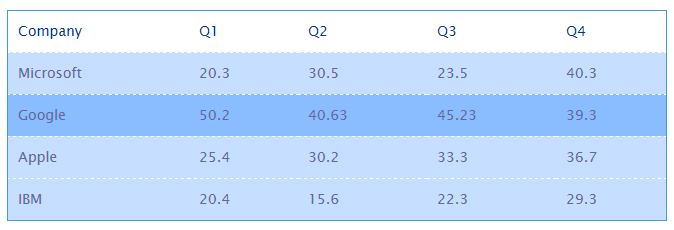
**Задание:**

Выбрать два шаблона таблиц (один из Примеров 1.1. и 1.2, второй – из Примеров 2.1 и 2.2) и создать по ним свои таблицы.

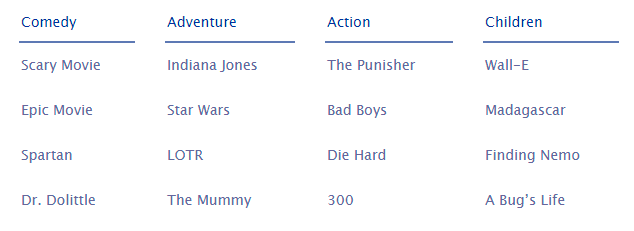
Пример 1.1 (Таблица имеет фоновое изображение, и строка выделяется при наведении курсором мыши на нее):



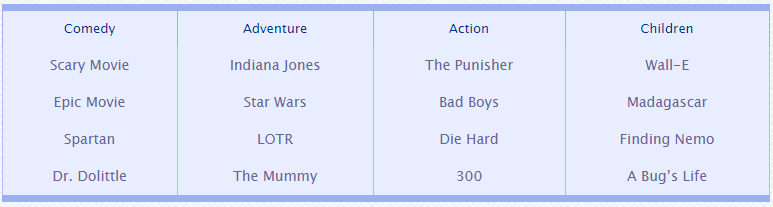
Пример 1.2 (Строка выделяется при наведении курсором мыши на нее):



Пример 2.1:



Пример 2.2:



**Описание выполнения задания**

1. Создание HMTL-документа с таблицами (Листинг 3.1).

Листинг 3.1 – Создание HTML-документа с таблицами

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

<style>

table {

width: 60%;

/\* Ширина таблицы \*/

border-collapse: collapse;

/\* Убираем двойные линии между ячейками \*/

}

td,

th {

padding: 3px;

/\* Поля вокруг содержимого таблицы \*/

border: 1px solid rgb(255, 255, 255);

/\* Параметры рамки \*/

}

th {

background: #f6fafa;

/\* Цвет фона \*/

color: rgb(72, 73, 150);

/\* Цвет текста \*/

}

tbody tr {

background-image: url(fon.PNG);

color: rgb(72, 73, 150);

}

tbody tr:hover {

background: #488cf3;

/\* Цвет фона при наведении \*/

color: rgb(25, 26, 68);

/\* Цвет текста при наведении \*/

}

.table\_col {

font-family: "Lucida Sans Unicode", "Lucida Grande", Sans-Serif;

font-size: 14px;

width: 660px;

background: white;

text-align: left;

border-collapse: collapse;

color: #3c5569;

}

.table\_col th {

font-weight: normal;

border-bottom: 2px solid #a6dff5;

border-right: 20px solid white;

border-left: 20px solid white;

padding: 8px 10px;

}

.table\_col td {

border-right: 20px solid white;

border-left: 20px solid white;

padding: 12px 10px;

color: #6788a8;

}

</style>

</head>

<body>

<table>

<tr>

<th></th>

<th>2004</th>

<th>2005</th>

<th>2006</th>

</tr>

<tbody>

<tr>

<td>Рубины</td>

<td>43</td>

<td>51</td>

<td>79</td>

</tr>

<tr>

<td>Изумруды</td>

<td>28</td>

<td>34</td>

<td>48</td>

</tr>

<tr>

<td>Сапфиры</td>

<td>29</td>

<td>57</td>

<td>36</td>

</tr>

</tbody>

</table>

<table class="table\_col">

<tr>

<th>Comedy</th>

<th>Adventure</th>

<th>Action</th>

<th>Children</th>

</tr>

<tr>

<td>Scary Movie</td>

<td>Indiana Jones</td>

<td>The Punisher</td>

<td>Wall-E</td>

</tr>

<tr>

<td>Epic Movie</td>

<td>Star Wars</td>

<td>Bad Boys</td>

<td>Madagascar</td>

</tr>

<tr>

<td>Spartan</td>

<td>LOTR</td>

<td>Die Hard</td>

<td>Finding Nemo</td>

</tr>

<tr>

<td>Dr. Dolittle</td>

<td>The Mummy</td>

<td>300</td>

<td>A Bug's Life</td>

</tr>

</table>

</body>

</html>

Результат выполнения кода на странице (Рисунок 3.1):

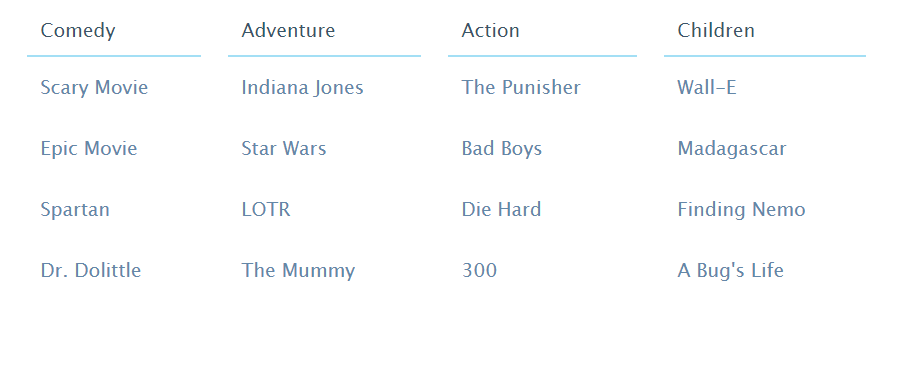
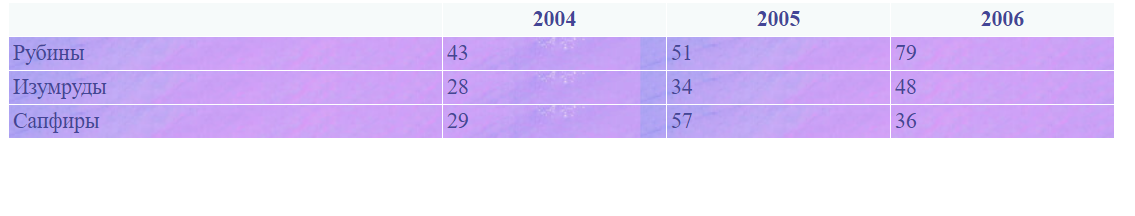
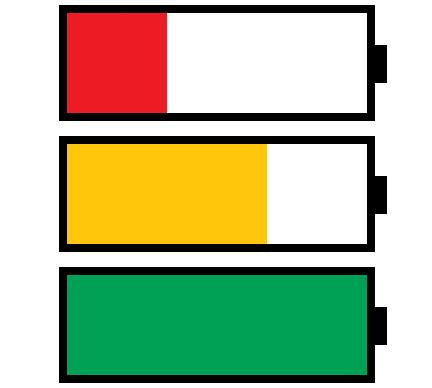


Рисунок 3.1 – Результат выполнения кода с созданием таблиц

## Практическое задание №4: Работа с внешними таблицами стилей CSS

**Задание:**

Создать батарейки как представлено на рисунке. Размеры заданы в пикселах и не масштабируются. Использовать псевдоэлемент **::after**.



**Описание выполнения задания**

1. Создание HMTL-документа с батарейками (Листинг 4.1).

Листинг 4.1 – Создание HTML-документа с батарейками

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

<style>

.b {

width: 150px;

/\* Ширина \*/

height: 50px;

/\* Высота \*/

border: 4px solid #000;

/\* Параметры рамки \*/

margin-bottom: 10px;

/\* Отступ снизу \*/

position: relative;

/\* Относительное позиционирование \*/

}

.b::before,

.b::after {

content: '';

position: absolute;

}

.b::after {

/\* Добавляем контакт справа \*/

width: 10px;

height: 20px;

background: #000;

right: -10px;

top: 15px;

}

.b::before {

height: 100%;

}

.low::before {

/\* Красная батарейка \*/

width: 33%;

background: #ED1C24;

}

.medium::before {

/\* Жёлтая батарейка \*/

width: 66%;

background: #FFC60B;

}

.high::before {

/\* Зелёная батарейка \*/

width: 100%;

background: #00A055;

} </style>

</head>

<body>

<div class="b low"></div>

<div class="b medium"></div>

<div class="b high"></div>

</body>

</html>

Результат выполнения кода на странице (Рисунок 4.1):

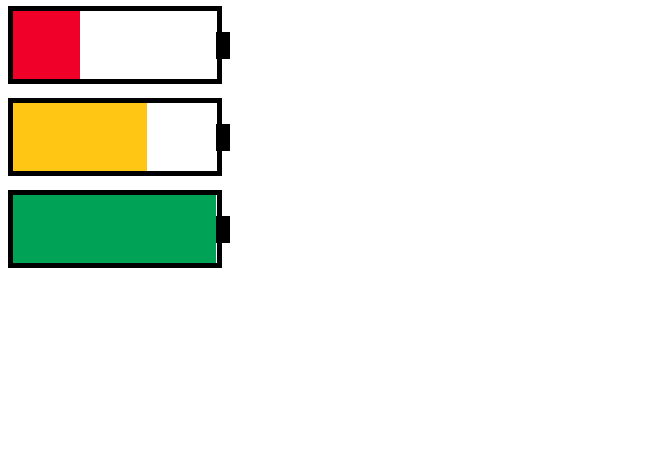


Рисунок 4.1 – Результат выполнения кода с созданием батареек

## Практическое задание №5: Работа с внешними таблицами стилей CSS

**Задание:**

Создать адаптивный квадратный корень на CSS (знак квадратного корня должен отображаться корректно независимо от используемого числа).

**Описание выполнения задания**

1. Создание HMTL-документа с адаптивным квадратным корнем на CSS (Листинг 5.1).

Листинг 5.1 – Создание HTML-документа с адаптивным квадратным корнем на CSS

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии --> <style>

body {

width: 800px;

padding: 50px;

font-size: 50px;

}

.wrapper {

font-style: italic;

}

.root {

display: inline-block;

vertical-align: middle;

border-top: 2px solid;

border-left: 2px solid;

transform: skew(-15deg);

transform-origin: bottom left;

margin: 0 10px;

position: relative;

}

.root:before {

content: "";

position: absolute;

bottom: 0;

height: 40%;

width: 9px;

left: -11px;

border-top: 2px solid;

border-right: 2px solid;

transform: skew(30deg);

transform-origin: bottom right;

}

.radicand {

display: inline-block;

padding-left: 3px;

transform: skew(15deg);

}

</style>

</head>

<body>

<div class="wrapper">

<span class="root">

<span class="radicand">

<var>1586

</var>

</span>

</span>

= ?

</div>

</body>

</html>

Результат выполнения кода на странице (Рисунок 5.1):

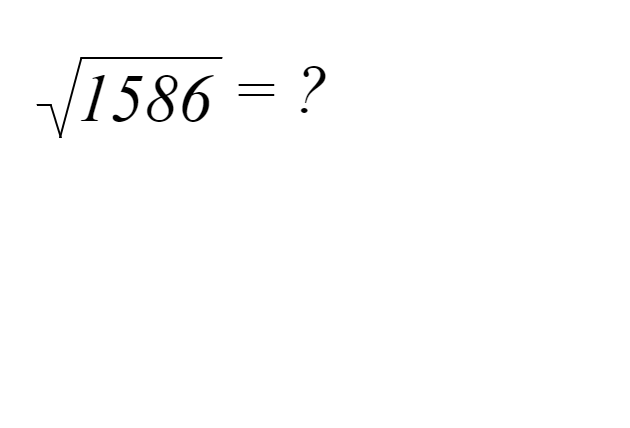
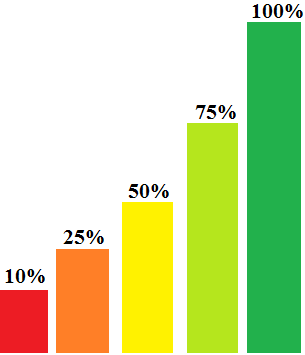


Рисунок 5.1 – Результат выполнения кода с созданием адаптивного квадратного корня на CSS

## Практическое задание №6: Работа с внешними таблицами стилей CSS

**Задание:**

Создать столбчатую диаграмму.

****

**Описание выполнения задания**

1. Создание HMTL-документа со столбчатой диаграммой (Листинг 6.1).

Листинг 6.1 – Создание HTML-документа со столбчатой диаграммой

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

<style>

#graph.graph {

border: 1px dashed #f9fcf8;

padding: 5px;

}

#graph .item {

margin: 2px;

color: #fff;

text-shadow: 1px 1px 1px #000;

padding: 3px;

border-radius: 3px;

background-color: #00508C;

box-shadow: 0 0 1px #2280AC;

}

#graph .item:hover {

background-color: #2280AC;

box-shadow: 0 0 2px #2280AC;

}

#graph.horizontal {

width: 200px;

}

#graph.horizontal .item {

height: 20px;

text-align: right;

}

#graph.vertical {

height: 120px;

width: 200px;

padding-bottom: 10px;

}

#graph.vertical .item {

width: 20px;

float: left;

display: inline-block;

}

</style>

</head>

<body>

<div id="graph" class="graph vertical">

<div class="item" style="height:20px;margin-top:90px">10%</div>

<div class="item" style="height:45px;margin-top:65px">25%</div>

<div class="item" style="height:60px;margin-top:50px">30%</div>

<div class="item" style="height:100px;margin-top:10px">50%</div>

</div>

</body>

</html>

Результат выполнения кода на странице (Рисунок 6.1):

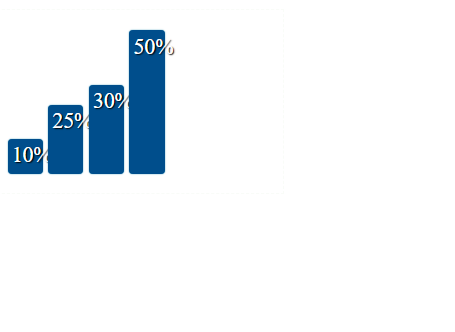
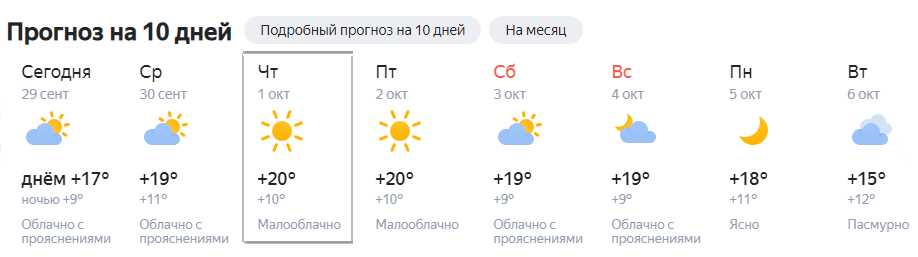


Рисунок 6.1 – Результат выполнения кода с созданием столбчатой диаграммы

## Практическое задание №7: Работа с внешними таблицами стилей CSS

**Задание:**

Создать страницу со следующим содержимым:



Добавить отображение рамки вокруг каждой из колонок погоды при наведении на них курсором мыши (как показано на рисунке выше).

**Описание выполнения задания**

1. Создание HMTL-документа с виджетом погоды (Листинг 7.1).

Листинг 7.1 – Создание HTML-документа с виджетом погоды

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

<style>

.weather\_\_row {

display: -webkit-flex;

display: -moz-flex;

display: -ms-flex;

display: -o-flex;

display: flex;

}

.weather\_\_day {

width: 150px;

font-size: 20px;

margin-right: 2px;

}

.weather\_\_day:hover {

box-shadow: 0 0 5px 1px #000000;

}

.weather\_\_image {

padding: 15px;

}

.weather\_\_top {

background: #0097B9;

color: #FFFFFF;

text-align: center;

padding: 2px;

}

.weather\_\_bottom {

text-align: center;

font-size: 1.3em;

}

</style>

</head>

<body>

<h3>Прогноз на 3 дня</h3>

<div class="weather">

<div class="weather\_\_row">

<div class="weather\_\_day">

<div class="weather\_\_top">31.08</div>

<div class="weather\_\_image"><img src="http://htmlbook.ru/files/weather-sun.jpg" alt=""></div>

<div class="weather\_\_bottom">+20&deg;</div>

</div>

<div class="weather\_\_day">

<div class="weather\_\_top">01.09</div>

<div class="weather\_\_image"><img src="http://htmlbook.ru/files/weather-snow.jpg" alt=""></div>

<div class="weather\_\_bottom">-3&deg;</div>

</div>

<div class="weather\_\_day">

<div class="weather\_\_top">02.09</div>

<div class="weather\_\_image"><img src="http://htmlbook.ru/files/weather-blizzard.jpg" alt=""></div>

<div class="weather\_\_bottom">-10&deg;</div>

</div>

</div>

</div>

</body>

</html>

Результат выполнения кода на странице (Рисунок 7.1):

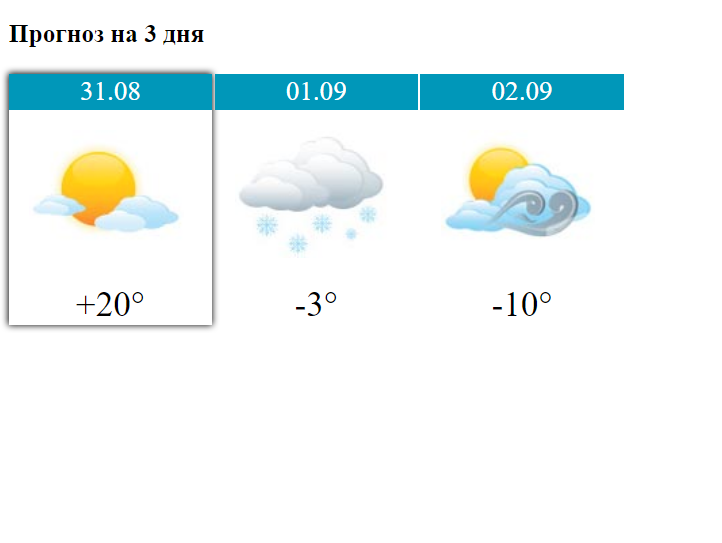


Рисунок 7.1 – Результат выполнения кода с созданием виджета погоды

## Практическое задание №8: Работа с внешними таблицами стилей CSS

**Задание:**

Создать пять блоков *div* и задать им параметры поворота, масштабируемости, наклона и сдвига при помощи CSS-свойства *transform*.

**Описание выполнения задания**

1. Создание HMTL-документа с пятью блоками *div* (Листинг 8.1).

Листинг 8.1 – Создание HTML-документа с пятью блоками *div*

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

<style>

/\*1\*/

div {

width: 170px;

height: 100px;

line-height: 100px;

margin: 0 auto;

-o-transition: all 0.5s ease-in-out;

-moz-transition: all 0.5s ease-in-out;

-webkit-transition: all 0.5s ease-in-out;

transition: all 0.5s ease-in-out;

}

kbd {

font-size: 0.9em;

display: inline-block;

line-height: 1.1;

}

.wrap {

display: inline-block;

margin: 0 40px 2em 0;

background: rgba(253, 252, 253, 0.5);

}

.seven {

border: 1px solid #000;

background-image: url(kitten.jpg);

}

.seven:hover {

-o-transform: rotate(720deg);

-ms-transform: rotate(720deg);

-moz-transform: rotate(720deg);

-webkit-transform: rotate(720deg);

transform: rotate(720deg);

}

/\*2\*/

.target {

width: 170px;

height: 100px;

border: 1px solid #000;

margin: 10px;

background-image: url(kitten.jpg);

}

.container {

float: left;

}

.container:hover .target {

transform: scale(0.9);

transition-timing-function: ease-in;

transition: 0.2s;

}

.container .target:hover {

transform: scale(1.1);

transition-timing-function: ease-in;

transition: 0.2s;

}

/\*3\*/

.ten {

border: 1px solid #000;

background-image: url(kitten.jpg);

}

.ten:hover {

-o-transform: skew(-20deg, 20deg);

-ms-transform: skew(-20deg, 20deg);

-moz-transform: skew(-20deg, 20deg);

-webkit-transform: skew(-20deg, 20deg);

transform: skew(-20deg, 20deg);

}

/\*4\*/

.three {

border: 1px solid #000;

background-image: url(kitten.jpg);

}

.three:hover {

-o-transform: translate(20px, 20px);

-ms-transform: translate(20px, 20px);

-moz-transform: translate(20px, 20px);

-webkit-transform: translate(20px, 20px);

transform: translate(20px, 20px);

}

</style>

</head>

<body>

1

<div class="wrap">

<div class="seven"><kbd>Number 1</kbd></div>

</div> <br>

2 <br>

<div class="container">

<div class="target">Numder 2</div>

</div> <br><br>

<br>

<br>

<br>

<br>

<br>

<br>

3 <br>

<div class="wrap">

<div class="ten"><kbd>Numder 3</kbd></div>

</div>

<br>

4 <br>

<div class="wrap">

<div class="three"><kbd>Number 4</kbd></div>

</div>

</body>

</html>

Результат выполнения кода на странице (Рисунок 8.1):



Рисунок 8.1 – Результат выполнения кода с созданием пяти блоков *div*

## Практическое задание №9: Работа с внешними таблицами стилей CSS

**Задание:**

Создать 3 анимационных кнопки: с поворотом кнопки на 90 градусов (использовать CSS-свойство *transform* с функцией *rotateX),* две кнопки с разными плавными изменениями свойств кнопки (использовать CSS-свойство перехода *transition*).

Задать названия всем кнопкам, тени (использовать ключевое слово *inset*), цветовое градиентное оформление.

**Описание выполнения задания**

1. Создание HMTL-документа с анимационными кнопками (Листинг 9.1).

Листинг 9.1 – Создание HTML-документа с анимационными кнопками

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

<style>

.c-hamburger {

display: block;

position: relative;

overflow: hidden;

margin: 0;

padding: 0;

width: 96px;

height: 96px;

font-size: 0;

text-indent: -9999px;

appearance: none;

box-shadow: none;

border-radius: none;

border: none;

cursor: pointer;

transition: background 0.3s;

}

.c-hamburger:focus {

outline: none;

}

.c-hamburger span {

display: block;

position: absolute;

top: 44px;

left: 18px;

right: 18px;

height: 8px;

background: white;

}

.c-hamburger span::before,

.c-hamburger span::after {

position: absolute;

display: block;

left: 0;

width: 100%;

height: 8px;

background-color: #fff;

content: "";

}

.c-hamburger span::before {

top: -20px;

}

.c-hamburger span::after {

bottom: -20px;

}

.c-hamburger--rot {

box-shadow: inset 0 0 6px;

background: linear-gradient(90deg, #cfecd0, #a0cea7, #9ec0db);

}

.c-hamburger--rot span {

transition: transform 0.3s;

}

/\* состояние active при открытом меню \*/

.c-hamburger--rot.is-active {

background-color: #166888;

}

.c-hamburger--rot.is-active span {

transform: rotateX(90deg);

}

.c-hamburger--htx {

background-color: #ff3264;

box-shadow: inset 0 0 20px;

}

.c-hamburger--htx span {

transition: background 0s 0.3s;

}

.c-hamburger--htx span::before,

.c-hamburger--htx span::after {

transition-duration: 0.3s, 0.3s;

transition-delay: 0.3s, 0s;

}

.c-hamburger--htx span::before {

transition-property: top, transform;

}

.c-hamburger--htx span::after {

transition-property: bottom, transform;

}

/\* состояние active при открытом меню \*/

.c-hamburger--htx.is-active {

background: linear-gradient(90deg, #ee5c87, #f1a4b5, #d587b3);

}

.c-hamburger--htx.is-active span {

background: none;

}

.c-hamburger--htx.is-active span::before {

top: 0;

transform: rotate(45deg);

}

.c-hamburger--htx.is-active span::after {

bottom: 0;

transform: rotate(-45deg);

}

.c-hamburger--htx.is-active span::before,

.c-hamburger--htx.is-active span::after {

transition-delay: 0s, 0.3s;

}

.c-hamburger--htla {

background-color: #32dc64;

box-shadow: inset 0 0 15px;

}

.c-hamburger--htla span {

transition: transform 0.3s;

}

.c-hamburger--htla span::before {

transform-origin: top right;

transition: transform 0.3s, width 0.3s, top 0.3s;

}

.c-hamburger--htla span::after {

transform-origin: bottom right;

transition: transform 0.3s, width 0.3s, bottom 0.3s;

}

/\* состояние active при открытом меню \*/

.c-hamburger--htla.is-active {

background: linear-gradient(66deg, #e38010, #1535bf);

}

.c-hamburger--htla.is-active span {

transform: rotate(180deg);

}

.c-hamburger--htla.is-active span::before,

.c-hamburger--htla.is-active span::after {

width: 50%;

}

.c-hamburger--htla.is-active span::before {

top: 0;

transform: translateX(38px) translateY(4px) rotate(45deg);

}

.c-hamburger--htla.is-active span::after {

bottom: 0;

transform: translateX(38px) translateY(-4px) rotate(-45deg);

}

</style>

</head>

<body>

Поворот на 90 градусов

<button class="c-hamburger c-hamburger--rot">

<span>Поворот на 90 градусов</span>

</button>

Крест

<button class="c-hamburger c-hamburger--htx">

<span>Крест</span>

</button>

Стрелка

<button class="c-hamburger c-hamburger--htla">

<span>Стрелка</span>

</button>

<script>

(function () {

"use strict";

var toggles = document.querySelectorAll(".c-hamburger");

for (var i = toggles.length - 1; i >= 0; i--) {

var toggle = toggles[i];

toggleHandler(toggle);

};

function toggleHandler(toggle) {

toggle.addEventListener("click", function (e) {

e.preventDefault();

(this.classList.contains("is-active") === true) ? this.classList.remove("is-active") : this.classList.add("is-active");

});

}

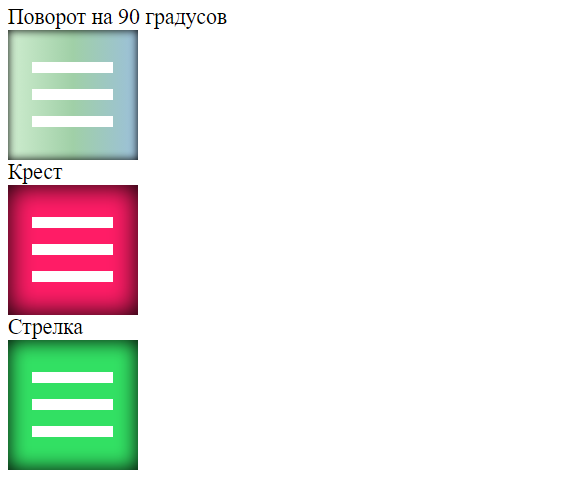
})();

</script>

</body>

</html>

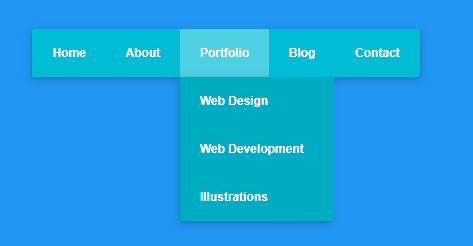
Результат выполнения кода на странице (Рисунок 9.1):

 Рисунок 9.1 – Результат выполнения кода с созданием анимационных кнопок

## Практическое задание №10: Работа с внешними таблицами стилей CSS

**Задание:**

Создать выпадающее меню (с собственным оформлением и содержанием) на примере, представленном на рисунке ниже, с изменением вида курсора при наведении на пункты меню.



**Описание выполнения задания**

1. Создание HMTL-документа с выпадающим меню (Листинг 10.1).

Листинг 10.1 – Создание HTML-документа с выпадающим меню

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

<style>

\* {

box-sizing: border-box;

}

body {

margin: 0;

background: radial-gradient(#BFD6E2 1px, rgba(255, 255, 255, 0) 2px);

background-size: 10px 10px;

}

nav ul {

list-style: none;

margin: 0;

padding: 0;

}

nav a {

display: block;

text-decoration: none;

outline: none;

transition: .4s ease-in-out;

}

.topmenu {

backface-visibility: hidden;

background: rgba(255, 255, 255, .8);

}

.topmenu>li {

display: inline-block;

position: relative;

}

.topmenu>li>a {

font-family: 'Exo 2', sans-serif;

height: 70px;

line-height: 70px;

padding: 0 30px;

font-weight: bold;

color: #003559;

text-transform: uppercase;

}

.down:after {

content: "\f107";

margin-left: 8px;

font-family: FontAwesome;

}

.topmenu li a:hover {

color: #E6855F;

}

.submenu {

background: white;

border: 2px solid #003559;

position: absolute;

left: 0;

visibility: hidden;

opacity: 0;

z-index: 5;

width: 150px;

transform: perspective(600px) rotateX(-90deg);

transform-origin: 0% 0%;

transition: .6s ease-in-out;

}

.topmenu>li:hover .submenu {

visibility: visible;

opacity: 1;

transform: perspective(600px) rotateX(0deg);

}

.submenu li a {

color: #7f7f7f;

font-size: 13px;

line-height: 36px;

padding: 0 25px;

font-family: 'Kurale', serif;

}

.movelink {

cursor: move;

}

.down {

cursor: move;

}

</style>

</head>

<body>

<nav>

<ul class="topmenu">

<li><a href="" class="movelink">Home</a></li>

<li><a href="" class="movelink">Shop</a></li>

<li><a href="" class="down">Blog</a>

<ul class="submenu">

<li><a href="" class="movelink">Category</a></li>

<li><a href="" class="movelink">Author</a></li>

<li><a href="" class="movelink">Archive</a></li>

<li><a href="" class="movelink">Tags</a></li>

</ul>

</li>

<li><a href="" class="down">Portfolio</a>

<ul class="submenu">

<li><a href="" class="movelink">Category1</a></li>

<li><a href="" class="movelink">Author2</a></li>

<li><a href="" class="movelink">Archive3</a></li>

<li><a href="" class="movelink">Tags4</a></li>

</ul>

</li>

<li><a href="" class="movelink">Contact</a></li>

</ul>

</nav>

</body>

</html>

Результат выполнения кода на странице (Рисунок 10.1):

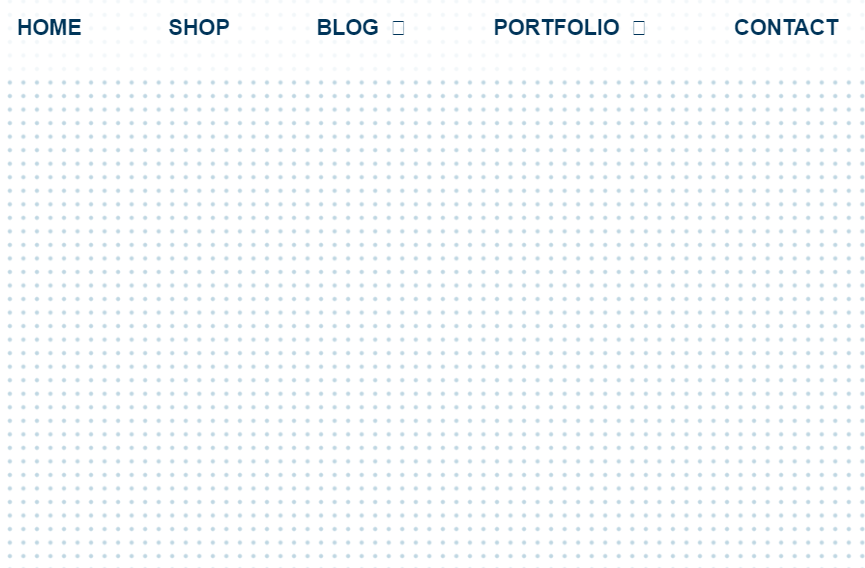


Рисунок 10.1 – Результат выполнения кода с созданием выпадающего меню

## Практическое задание №11: CSS3-селекторы

**Задание:**

1. Выбрать *input* типа *checkbox*.
2. Выбрать *input* типа *checkbox*, НЕ отмеченный.
3. Найти все элементы с *id=message* или *message-\**.
4. Найти все элементы с *id=message-\**.
5. Найти все ссылки с расширением *href="...zip"*.
6. Найти все элементы с атрибутом *data-action*, содержащим *delete* в списке (через пробел).
7. Найти все элементы, у которых ЕСТЬ атрибут *data-action*, но он НЕ содержит *delete* в списке (через пробел).
8. Выбрать все чётные элементы списка *#messages*.
9. Выбрать один элемент сразу за заголовком *h3#widget-title* на том же уровне вложенности.
10. Выбрать все ссылки, следующие за заголовком *h3#widget-title* на том же уровне вложенности.
11. Выбрать ссылку внутри последнего элемента списка *#messages*.

**Описание выполнения задания**

1. Создание HMTL-документа с требуемыми элементами (Листинг 11.1).

Листинг 11.1 – Создание HTML-документа с требуемыми элементами

<!DOCTYPE html> <!-- Объявление формата документа -->

<html>

<head>

<!-- Техническая информация о документе -->

<meta charset="UTF-8"> <!-- Определяем кодировку символов документа -->

<title>Главная</title> <!-- Задаем заголовок документа -->

<link rel="stylesheet" type="text/css" href="style.css"> <!-- Подключаем внешнюю таблицу стилей -->

<script src="script.js"></script> <!-- Подключаем сценарии -->

<style>

</style>

</head>

<body>

<input type="checkbox">

<input type="checkbox" checked>

<input type="text" id="message">

<h3 id="widget-title">Сообщения:</h3>

<ul id="messages">

<li id="message-1">Сообщение 1</li>

<li id="message-2">Сообщение 2</li>

<li id="message-3" data-action="delete">Сообщение 3</li>

<li id="message-4" data-action="edit do-not-delete">Сообщение 4</li>

<li id="message-5" data-action="edit delete">Сообщение 5</li>

<li><a href="#">...</a></li>

</ul>

<a href="http://site.com/list.zip">Ссылка на архив</a>

<a href="http://site.com/list.pdf">..И на PDF</a>

<script>

// тестовая функция для селекторов

// проверяет, чтобы элементов по селектору selector было ровно count

function test(selector, count) {

var elems = document.querySelectorAll(selector);

var ok = (elems.length == count);

if (!ok) alert(selector + ": " + elems.length + " != " + count);

}

// ------------- селекторы --------------

// Выбрать input типа checkbox

test('input[type="checkbox"]', 2);

// Выбрать input типа checkbox, НЕ отмеченный

test('input[type="checkbox"]:not(:checked)', 1);

// Найти все элементы с id=message или message-\*

test('[id|="message"]', 6);

// Найти все элементы с id=message-\*

test('[id^="message-"]', 5);

// Найти все ссылки с расширением href="...zip"

test('a[href$=".zip"]', 1);

// Найти все элементы с data-action, содержащим delete в списке (через пробел)

test('[data-action~="delete"]', 2);

// Найти все элементы, у которых ЕСТЬ атрибут data-action,

// но он НЕ содержащит delete в списке (через пробел)

test('[data-action]:not([data-action~="delete"])', 1);

// Выбрать все чётные элементы списка #messages

test('#messages li:nth-child(2n)', 3);

// Выбрать один элемент сразу за заголовком h3#widget-title

// на том же уровне вложенности

test('h3#widget-title + \*', 1);

// Выбрать все ссылки, следующие за заголовком h3#widget-title

// на том же уровне вложенности

test('h3#widget-title ~ a', 2);

// Выбрать ссылку внутри последнего элемента списка #messages

test('#messages li:last-child a', 1);

</script>

</body>

</html>

Результат выполнения кода на странице (Рисунок 11.1):



Рисунок 11.1 – Результат выполнения кода с созданием требуемых элементов

Вывод**:** в результате выполнения данной практической работы было выполнено ознакомление со структурой языка HTML, пройдено обучение по созданию HTML-документов, оформлению комментариев в коде, работе с основными тегами языка, добавлению изображений и ссылок на страницы, подключению внешних CSS-файлов и созданию простейших стилей HTML-документам.