|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | 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 ПРАКТИЧЕСКИМ РАБОТАМ** | | | |
| **по дисциплине** | | | |
| **«Разработка клиентских частей интернет-ресурсов»** | | | |
| Выполнил студент группы ИКБО-02-19 | | | Миронов А.Д. |
|  | | |  |
| Принял  *Ассистент* | | | Меркулов Е.С. |
| Практические работы выполнены | «\_\_\_»\_\_\_\_\_\_\_2020 г. | (подпись студента) | | |
| «Зачтено» | «\_\_\_»\_\_\_\_\_\_\_2020 г. | (подпись руководителя) | | |
|  |  |  | | |

Москва 2020

**Оглавление**

[Практическая работа №4: «Работа с внешними таблицами стилей (CSS)» 3](#_Toc56295664)

[Практическое задание №1: Внешние таблицы стилей (CSS) 3](#_Toc56295665)

[Практическое задание №2: CSS3-селекторы](#_Toc56295666) 31

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

## Практическое задание №1: Внешние таблицы стилей (CSS)

**Задание 1:**

1. Используя CSS-коды спецсимволов вывести.
2. ‘Левая одиночная и правая одиночная кавычки’
3. ‚Нижняя одиночная кавычка‚
4. “Левая и правая двойные кавычки”
5. „Нижняя двойная кавычка„
6. «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. Используем коды CSS и разные стили в CSS-файле (Листинг 1.1).

Листинг 1.1 – коды CSS и разные стили в CSS-файле

<!DOCTYPE html>

<head>

<title>

Задание 1

</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="1.css">

</head>

<body>

<ol>

<li id="i1">&#8216Левая одиночная и правая одиночная кавычки&#8217</li>

<li id="i2">&#8218Нижняя одиночная кавычка&#8218</li>

<li id="i3">&#10077Левая и правая двойные кавычки&#10078</li>

<li id="i4">&bdquo;Нижняя двойная кавычка&bdquo;</li>

<li id="i5">&laquo;Левая и правая двойные угловые кавычки&raquo;</li>

</ol>

</body>

</html>

#i1{

    color:rgb(30, 179, 11);

}

#i2{

    text-decoration:underline;

}

#i3{

    color:rgb(179, 78, 11);

}

#i4{

    font-size:20px;

}

#i5{

    font-style:italic;

}

Результат выполнения кода на странице (Рисунок 1.1):

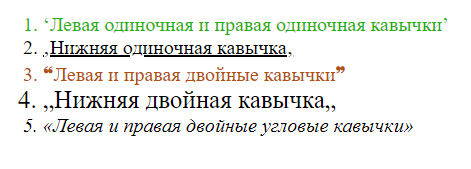


Рисунок 1.1 – Результат выполнения кода с кодами CSS и разными стили в CSS-файле

**Задание 2:**

1. 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).

**Описание выполнения задания**

1. Используем теги <ol>, <il> и оформление в CSS (Листинг 1.2).

Листинг 1.2 – теги <ol>, <il> и оформление в CSS

<!DOCTYPE html>

<head>

<title>

Задание 2

</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="2.css">

</head>

<body>

<ol>

<li id="s1">Окунь</li>

<li id="s2">Сапог</li>

<li id="s3">криволинейный</li>

<li id="s4">рефрижератор</li>

<li id="s5">конец</li>

</ol>

</body>

</html>

#s1{

    text-decoration:line-through;

    text-decoration-style:dotted;

    text-decoration-color: rgb(4, 236, 54);

    font-family:Comic Sans MS;

    font-style:bold;

}

#s2{

    text-decoration: underline overline;

}

#s3{

    text-decoration:underline;

    text-decoration-style:wavy;

}

#s4{

    text-decoration:overline;

    text-decoration-style:dashed;

    text-decoration-color: rgb(128, 0, 100);

    font-family:Comic Sans MS;

}

#s5{

    text-decoration:line-through;

    text-decoration-style:double;

    text-decoration-color: rgb(17, 154, 172);

}

Результат выполнения кода на странице (Рисунок 1.2):

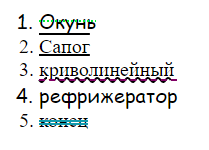


Рисунок 1.2 – Результат выполнения кода с тегами <ol>, <il> и оформлением в CSS

**Задание 3:**

1. 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.

**Описание выполнения задания**

1. Используем теги <tr>, <td>, <table> и оформление в CSS (Листинг 1.3).

Листинг 1.3 – теги <ol>, <il> и оформление в CSS для первого и второго примеров

<!DOCTYPE html>

<head>

<title>

Задание 3.1

</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="3\_1.css">

</head>

<body>

<table id="tab1">

 <caption>Таблица "вырви глаз"</caption>

   <tr>

    <th>Столбец 1</th>

    <th>Столбец 2</th>

    <th>Столбец 3</th>

   </tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

   <tr><td>ЗНАЧЕНИЕ</td><td>ЗНАЧЕНИЕ</td><td>ZNACHENIE</td></tr>

  </table>

</body>

</html>

Первое задание

th{

    background-color:rgb(199, 94, 94);

    border-collapse: collapse;

    border: 1px solid white;

}

#tab1{

    width:600px;

    background-image: url("https://memepedia.ru/wp-content/uploads/2016/08/fdKZ-ZpN7iw.jpg");

    border-collapse: collapse;

    border: 1px solid rgb(62, 190, 79);

}

td{

    color:rgb(151, 22, 130);

}

tr:hover {

    background: #b4179f;

}

Второе задание

th{

    background-color:white;

    border-collapse: collapse;

    border-bottom: 3px solid rgb(255, 51, 0);

    color:rgb(0, 253, 0);

    text-align:left;

}

table{

    width:600px;

    background:rgb(255, 255, 255);

    border: 0;

    border-spacing: 10px 2px;

}

td{

    color:rgb(241, 3, 241);

}

Результат выполнения кода на странице (Рисунок 1.3-1.4):

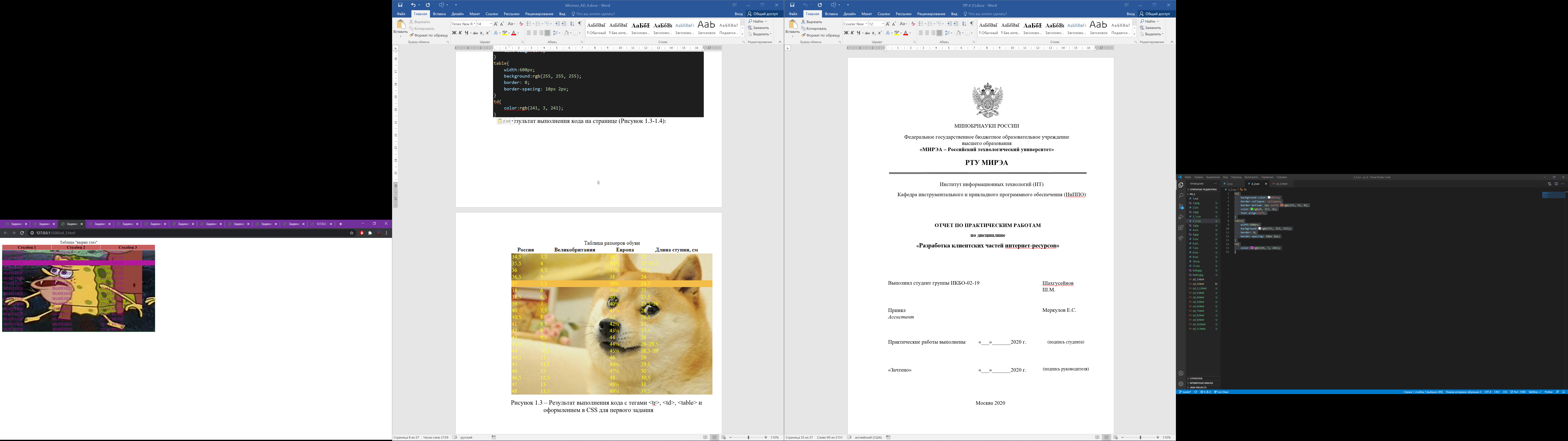


Рисунок 1.3 – Результат выполнения кода с тегами <tr>, <td>, <table> и оформлением в CSSдля первогозадания

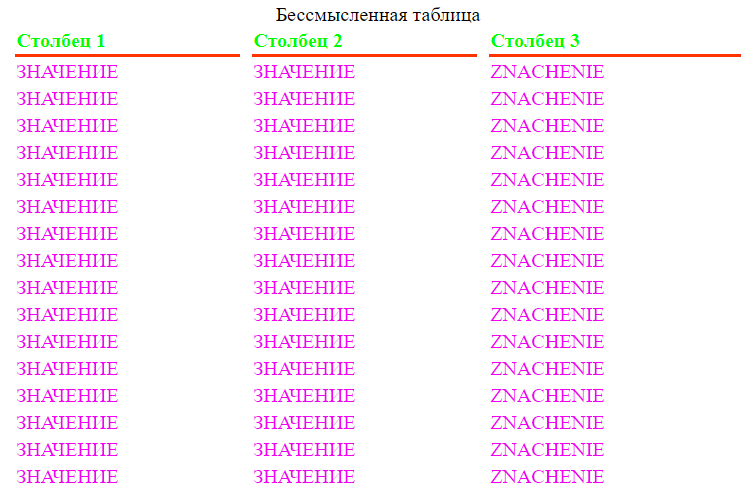


Рисунок 1.4 – Результат выполнения кода с тегами <tr>, <td>, <table> и оформлением в CSSдля второгозадания

**Задание 4:**

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

**Описание выполнения задания**

1. Используем тег <div> и оформляем его в CSS (Листинг 1.4).

Листинг 1.4 – теги <div> и оформление в CSS

<!DOCTYPE html>

<head>

<title>

Задание 4

</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="4.css">

</head>

<body>

<div id="red">

</div>

<br>

<div id="yel">

</div>

<br>

<div id="green">

</div>

</body>

</html>

div{

    border:5px solid;

    width:200px;

    height:80px;

    position: relative;

}

#red:before{

    content:"";

    display:block;

    background:red;

    width:33%;

    height:100%;

}

#yel:before{

    content:"";

    display:block;

    background:yellow;

    width:66%;

    height:100%;

}

#green:before{

    content:"";

    display:block;

    background:green;

    width:100%;

    height:100%;

}

div:after {

    content:"";

    width: 10px;

    height: 20px;

    background: #000;

    right: -10px; top: 30px;

    position: absolute;

}

Результат выполнения кода на странице (Рисунок 1.5):

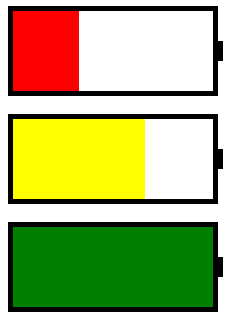


Рисунок 1.5 – Результат выполнения кода с тегом <div> и оформлением в CSS

**Задание 5:**

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

**Описание выполнения задания**

1. Используем атрибут text-decoration-line со значением overline (Листинг 1.5).

Листинг 1.5 – атрибут text-decoration-line со значением overline

<!DOCTYPE html>

<head>

<title>

Задание 5

</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="5.css">

</head>

<body>

<span>&radic;</span><span id="number">1</span><br>

<span>&radic;</span><span id="number">12</span><br>

<span>&radic;</span><span id="number">123</span><br>

<span>&radic;</span><span id="number">1234</span><br>

<span>&radic;</span><span id="number">12345</span><br>

<span>&radic;</span><span id="number">123456</span><br>

<span>&radic;</span><span id="number">1234567</span><br>

<span>&radic;</span><span id="number">12345678</span><br>

<span>&radic;</span><span id="number">123456789</span><br>

<span>&radic;</span><span id="number">AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA</span>

</body>

</html>

#number {

    text-decoration-line: overline;

    text-decoration-color: black;

}

Результат выполнения кода на странице (Рисунок 1.6):

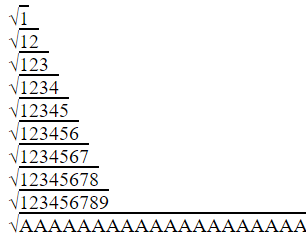


Рисунок 1.6 – Результат выполнения кода с атрибутом text-decoration-line со значением overline

**Задание 6:**

1. Создать столбчатую диаграмму.

**Описание выполнения задания**

1. Используем тег <div> и оформляем его в CSS (Листинг 1.6).

Листинг 1.6 – тег <div> и оформление его в CSS

<!DOCTYPE html>

<!DOCTYPE html>

<head>

<title>

Задание 6

</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="6.css">

</head>

<body>

<div id="main">

<div id="red">

</div>

<div id="orange">

</div>

<div id="yellow">

</div>

<div id="green">

</div>

<div id="dgreen">

</div>

</div>

</body>

</html>

#red{

    width:40px;

    height:40px;

    background:red;

    position:absolute;

    bottom:1px;

    left:1px;

}

#orange{

    width:40px;

    height:100px;

    background:orange;

    position:absolute;

    bottom:1px;

    left:45px;

}

#yellow{

    width:40px;

    height:200px;

    background:yellow;

    position:absolute;

    bottom:1px;

    left:90px;

}

#green{

    width:40px;

    height:300px;

    background:YellowGreen;

    position:absolute;

    bottom:1px;

    left:135px;

}

#dgreen{

    width:40px;

    height:400px;

    background:green;

    position:absolute;

    bottom:1px;

    left:180px;

}

#red:after{

    content:"10%";

    position: absolute;

    top:-15px;

    left:5px;

}

#orange:after{

    content:"25%";

    position: absolute;

    top:-15px;

    left:5px;

}

#yellow:after{

    content:"50%";

    position: absolute;

    top:-15px;

    left:5px;

}

#green:after{

    content:"75%";

    position: absolute;

    top:-15px;

    left:5px;

}

#dgreen:after{

    content:"100%";

    position: absolute;

    top:-15px;

}

Результат выполнения кода на странице (Рисунок 1.7):

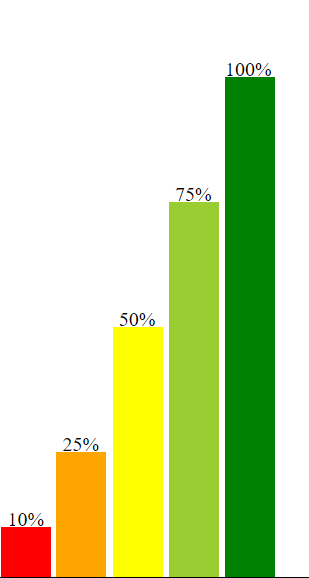
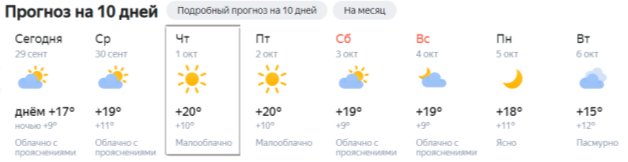


Рисунок 1.7 – Результат выполнения кода с тегом <div> и оформляем его в CSS

**Задание 7:**

1. 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. Используем теги <div> и оформляем его в CSS (Листинг 1.7).

Листинг 1.7 – теги <div> и оформление его в CSS

<!DOCTYPE html>

<head>

<title>

Задание 7

</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="7.css">

</head>

<body>

<span class="title">Прогноз на 10 дней</span>

<a class="button">Подробный прогноз на 10 дней</a>

<a class="button">На месяц</a>

<div class="w\_row">

    <div class="w\_day">

        <div class="name">Сегодня</div>

        <div class="date">10 нояб</div>

        <div><img src="1.png" width="50%"></div>

        <div class="tday">+20°</div>

        <div class="tnight">+16°</div>

        <div class="condition">Ясно</div>

    </div>

    <div class="w\_day">

        <div class="name">Ср</div>

        <div class="date">11 нояб</div>

        <div><img src="1.png" width="50%"></div>

        <div class="tday">+22°</div>

        <div class="tnight">+17°</div>

        <div class="condition">Ясно</div>

    </div>

    <div class="w\_day">

        <div class="name">Чт</div>

        <div class="date">12 нояб</div>

        <div><img src="2.jpg" width="50%"></div>

        <div class="tday">+19°</div>

        <div class="tnight">+15°</div>

        <div class="condition">Облачно с прояснениями</div>

    </div>

    <div class="w\_day">

        <div class="name">Пт</div>

        <div class="date">13 нояб</div>

        <div><img src="3.jpg" width="50%"></div>

        <div class="tday">+999°</div>

        <div class="tnight">+999°</div>

        <div class="condition">Армагедон</div>

    </div>

    <div class="w\_day">

        <div class="name" style="color: red">Сб</div>

        <div class="date">14 нояб</div>

        <div><img src="4.jpg" width="50%"></div>

        <div class="tday">-96°</div>

        <div class="tnight">−104°</div>

        <div class="condition">Ледниковый период</div>

    </div>

    <div class="w\_day">

        <div class="name" style="color: red">Вс</div>

        <div class="date">15 нояб</div>

        <div><img src="1.png" width="50%"></div>

        <div class="tday">+20°</div>

        <div class="tnight">+17°</div>

        <div class="condition">Ясно</div>

    </div>

</div>

</body>

</html>

.w\_row

{

    display: flex;

    margin-top: 10px;

}

.w\_day

{

    width: 150px;

    margin-right: 2px;

}

.w\_day:hover

{

    box-shadow: 0 0 5px 1px black;

}

.name

{

    font-size: 18px;

    margin-left: 6px;

}

.date

{

    display: block;

    color: grey;

    font-size: 14px;

    margin: 6px;

}

.title

{

    font-size: 24px;

    font-weight: bold

}

.button

{

    padding: 8px 16px;

    border-radius: 20px;

    background: #eceef2;

}

.condition

{

    color: grey;

    line-height: 20px;

    margin: 6px;

    font-size: 18px;

}

.tnight

{

    color: grey;

    line-height: 14px;

    font-size: 16px;

    margin-bottom: 18px;

    margin-left: 6px;

}

.tday

{

    font-size: 21px;

    margin: 6px;

    font-weight: bold;

}

Результат выполнения кода на странице (Рисунок 1.8):



Рисунок 1.8 – Результат выполнения кода с тегами <div> и оформление его в CSS

**Задание 8:**

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

**Описание выполнения задания**

1. Используем теги <div> и оформляем его в CSS свойствами transform (Листинг 1.8).

Листинг 1.8 – теги <div> и оформление его в CSS

<!DOCTYPE html>

<head>

<title>

Задание 8

</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="8.css">

</head>

<body>

<div id="i1"></div>

<div id="i2"></div>

<div id="i3"></div>

<div id="i4"></div>

<div id="i5"></div>

</body>

</html>

div{

    height:100px;

    width:200px;

    border:2px solid black;

}

#i1{

    transform:translate(140px, 40%);

}

#i2{

    transform:rotate(45deg) scale(2, 2) skew(30deg, 30deg) translate(120px, 50%);

}

#i3{

    transform:scale(3, 2);

}

#i4{

    transform:skew(30deg, 30deg);

}

#i5{

    transform:rotate(70deg);

}

Результат выполнения кода на странице (Рисунок 1.9):

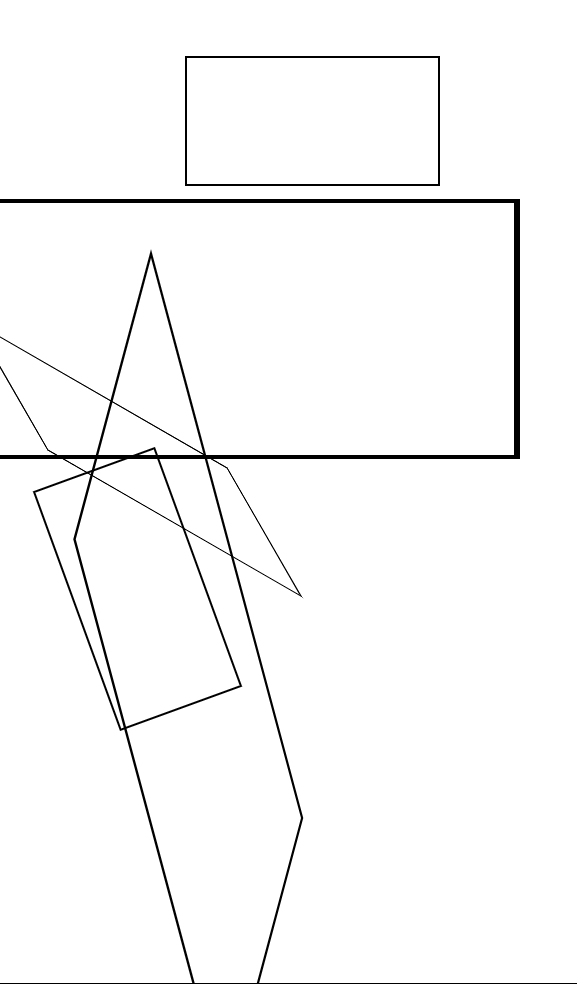


Рисунок 1.9 – Результат выполнения кода

**Задание 9:**

1. 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. Используем теги <div> и оформляем его в CSS свойствами transform (Листинг 1.9).

Листинг 1.9 – теги <div> и оформление его в CSS

<!DOCTYPE html>

<head>

<title>

Задание 9

</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="9.css">

</head>

<body>

<div id="rot">Поворот на 90°</div>

<div id="but1">Кнопка 1</div>

<div id="but2">Кнопка 2</div>

</body>

</html>

div{

    height:100px;

    width:200px;

    border:2px solid black;

    position:absolute;

}

#rot{

    top:30%;

}

#rot:hover{

    box-shadow: 0 0 20px 12px Red,

                15px 15px 30px Orange;

    transform:rotate(90deg);

}

#but1{

    transition-duration: 5s;

    transform: scale(1.5);

    top:30%;

    left:30%;

    transition-timing-function: linear;

    transition-delay: .5s;

    background: linear-gradient(to top, #fefcea, #f1da36);

    box-shadow: 0 0 20px 12px Green,

                15px 15px 30px Yellow;

}

#but2{

    transition-duration: 5s;

    transform: scale(1.2);

    top:30%;

    left:60%;

    transition-timing-function: ease;

    transition-delay: 1.3s;

    background: linear-gradient(to top, #DC143C, #FFC0CB);

    box-shadow: 0 0 20px 12px Blue,

                15px 15px 30px Purple;

}

Результат выполнения кода на странице (Рисунок 1.10):

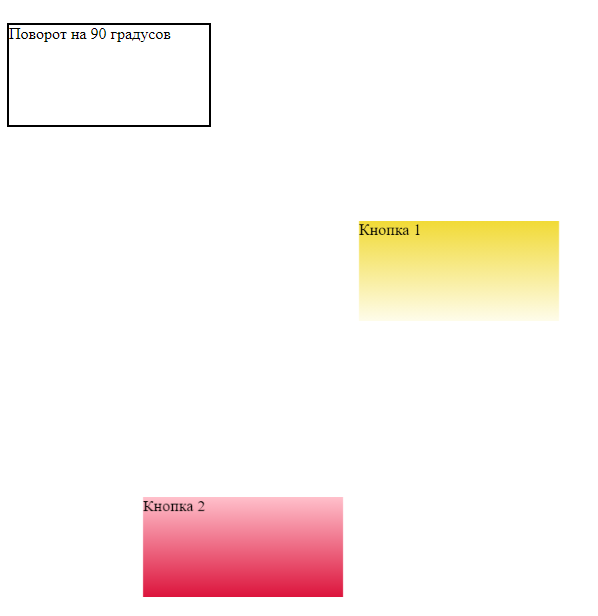


Рисунок 1.10 – Результат выполнения кода с тегами <div> и оформление его в CSS

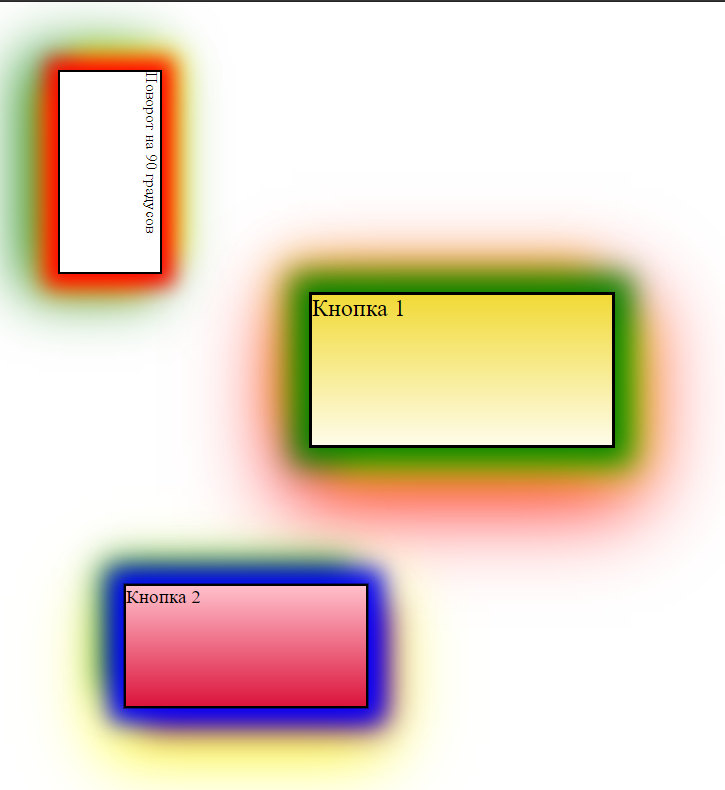


Рисунок 1.11 – Результат выполнения кода с тегами <div> и оформление его в CSS

**Задание 10:**

1. 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. Используем теги <ul>, <ol> и описываем их в CSS-файле (Листинг 1.10).

Листинг 1.10 – теги <ul>, <ol> и их описание в CSS-файле

<!DOCTYPE html>

<head>

<title>

Задание 10

</title>

<meta charset="UTF-8">

<link rel="stylesheet" type="text/css" href="10.css">

</head>

<body>

<ul>

<li><a href="#">Home</a></li>

<li><a href="#">About</a>

    <ul id="submenu">

    <li><a href="#">what?</a></li>

    <li><a href="#">what??</a></li>

    </ul>

</li>

<li><a href="#">Portfolio</a></li>

</ul>

</body>

</html>

ul{

    list-style: none;

    margin: 0;

    padding: 0;

    background: #0b5e03;

    width:300px;

  }

  li{

    display: inline-block;

    margin-right: 20px;

    position: relative;

  }

  #submenu{

    position: absolute;

    top: 100%;

    left: 0;

    width: 100%;

    z-index: 10;

    display: none;

    background: #0000ff;

  }

  a{

    display: block;

    padding: 10px 15px;

    color: #000000;

    outline: none;

    text-decoration: none;

  }

  ul li:hover #submenu {

    display: block;

  }

  a:hover{

      background:#0000ff;

  }

Результат выполнения кода на странице (Рисунок 1.12):

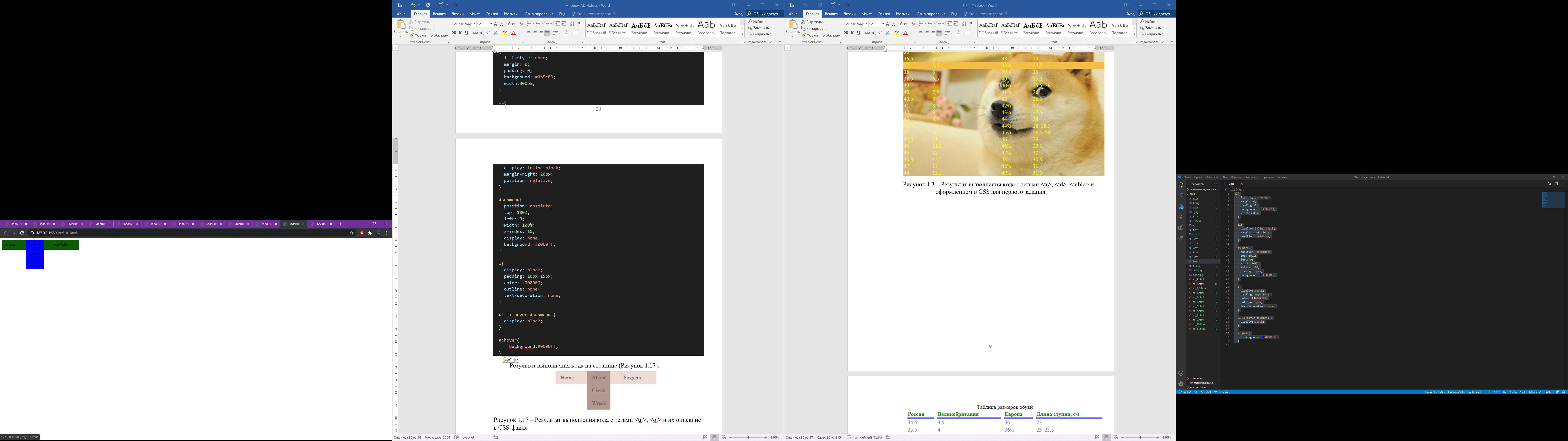


Рисунок 1.12 – Результат выполнения кода с тегами <ul>, <ol> и их описание в CSS-файле

## Практическое задание №2: CSS3-селекторы

**Задание 11:**

1. 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. Используем селекторы в CSS-файле (Листинг 1.11).

Листинг 1.11 – селекторы в CSS-файле

<!DOCTYPE HTML>

<html>

<head>

  <meta charset="utf-8">

  <link rel="stylesheet" href="11.css">

</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>

</body>

</html>

input[type="checkbox"] {

    box-shadow: 0 0 10px red;

    }

    input[type="checkbox"]:not(:checked) {

        box-shadow: 0 0 5px orange;

    }

    [id|="message"] {

        box-shadow: 0 0 5px rgb(224, 248, 3);

    }

    [id^="message-"] {

        color: rgb(48, 3, 248);

    }

    a[href$=".zip"] {

        color: green;

    }

    [data-action~="delete"] {

        color: blue;

    }

    [data-action]:not([data-action~="delete"]) {

        box-shadow: inset 0 0 15px black;

    }

    #messages li:nth-child(2n):before {

        content: "2n -> | ";

    }

    h3#widget-title + \* {

        background: rgb(255, 255, 255);

    }

    h3#widget-title ~ a:before {

        content: "link -> | ";

    }

    h3#widget-title ~ a:before {

        content: "link -> | ";

    }

    #messages li:last-child {

        background: rgb(207, 67, 184);

    }

Результат выполнения кода на странице (Рисунок 1.13):

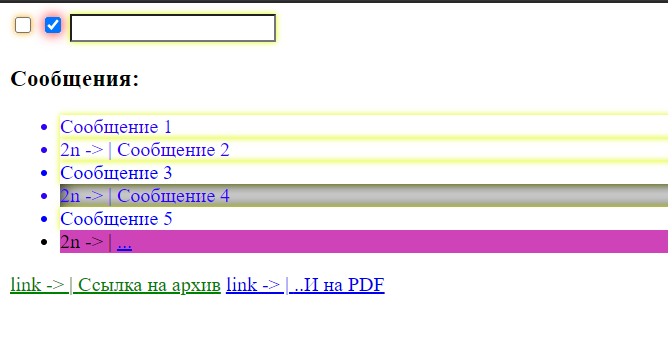


Рисунок 1.13 – Результат выполнения кода с селекторами в CSS-файле

**Вывод:** в данной практической работебыли изучены новые способы работы с CSS-файлами: добавление разных стилей к строкам, использование спецсимволов, использование оформление с помощью text-decoration со значениями underline, line-through, overline, text-decoration-style с разными значениями, text-decoration-color, описание таблиц, использование атрибута transform, использование псведоэлемента ::after, создание адаптивного корня, столбчатой диаграммы, прогноза погоды, создание выпадающего меню и использование селекторов