

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании (КСУП)

ОТЧЁТ

К Лабораторной Работе №2 (CSS – каскадные таблицы стилей. Приемы  
верстки страниц)

по дисциплине

"Научно-учебная работа"

Студент группы 589-2

Кабаев В.А.

Приняли

К.т.н, доцент кафедры КСУП

Хабибулина Н.Ю.

К.т.н, доцент кафедры КСУП

Черкашин М.В.

Томск 2021

## Содержание

|     |                                     |    |
|-----|-------------------------------------|----|
| 1   | Введение.....                       | 3  |
| 2   | Задания .....                       | 4  |
| 2.1 | Задание 1 .....                     | 4  |
| 2.2 | Задание 2 .....                     | 13 |
| 2.3 | Задание 3 .....                     | 14 |
| 2.4 | Задание 4 .....                     | 18 |
| 2.5 | Задание 5 .....                     | 20 |
| 3   | Таблица изученных свойств.....      | 22 |
| 4   | Заключение .....                    | 24 |
| 5   | Ответы на контрольные вопросы ..... | 25 |

## **1 Введение**

Цель лабораторной работы: знакомство с языком разметки CSS, изучение основных приемов верстки страниц.

Задачи работы:

- знакомство с CSS - каскадными таблицами стилей.
- изучение основных параметров CSS.
- знакомство с приемами табличной и блочной верстки web-страницы.

## 2 Задания

### 2.1 Задание 1

Разработайте минимум три каскадные таблицы стилей для оформления странички. Предусмотрите внешние, глобальные и внутренние таблицы стилей, стилевые классы (относящиеся к конкретному элементу и к любому элементу html-документа). Файлы с внешними CSS сохраните в специальной папке;

1) Внешние таблицы стилей:

```
/* Nav */
.nav {
  font-size: 0.35em;
  text-transform: uppercase;
}

.nav__link {
  display: inline-block;
  vertical-align: top;
  margin: 0 15px;
  position: relative;
  font-size: 1.05em;

  color: #fff;
  text-decoration: none;
}

.nav__link::before{
  content: '';
  position: absolute;
  left: 0;
  bottom: -1.5px;
  height: 1.5px;
  width: 100%;
  background: #000;
  transform: scale(0);
  -webkit-transition: all 0.3s;
  -o-transition: all 0.3s;
  transition: all 0.3s;
}

.nav__link:hover::before{
  transform: scale(1);
}
```

```
.nav__link:hover {
  color: #ed45ac;
  -webkit-transition: all 0.3s;
  -o-transition: all 0.3s;
  transition: all 0.3s;
}
```

```
main {
  background-image: url("../img/background_image.png");
  height: 100vh;
  padding: 0;
  margin: 0;
  background-size: cover;
}
```

```
.main_title {
  font-weight: 600;
  font-size: 50px;
  color: #fff;
  position: absolute;
  bottom: 60px;
  left: 35%;
}
```

```
.basic {
  width: 90%;
  margin: 0 auto;
  position: relative;
  height: 100%;
}
```

```
.container {
  color: #fff;
  width: 90%;
  margin: 0 auto;
  position: relative;
  height: 100%;
  align-items: center;
}
```

```
.page2{
  color: #ffffff;
}
```

```
p {
  display: block;
  margin-inline-start: 0px;
  margin-inline-end: 0px;
  margin-block-start: 1em;
  margin-block-end: 1em;
}
```

```

.footer {
  padding: 45px;
}

table.table{
  width: 100%;
  border-collapse: collapse;
  height: auto;
}
table.table, table.table td, table.table th{
  border: 2px solid;
  text-align: center;;
  border-color: #fff;
}
table.table td,table.table th {
  padding: 3px;
  width: 25%;
  height: 20%;
}
table.table th {
  background: #347c99;
  color: #fff;
  font-weight: normal;
}

.content {
  margin: 0 auto;
  width: 80%;
}
.block_container {
  font-size: 1.45rem;
  width: 100%;
  display: -webkit-flex;
  display: -ms-flex;
  display: flex;
  justify-content: space-between;
  height: 350px;
  margin-bottom: 30px;
}

.block_header {
  width: 100%;
  background: #7B68EE;
  color: #000;
  font-size: 1.45rem;
  height: 150px;
  display: -webkit-flex;
  display: -ms-flex;
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 20px 0;
}

```

```

.left_info {
    background: #E9967A;
    color: #000;
    display: -webkit-flex;
    display: -ms-flex;
    display: flex;
    align-items: center;
    justify-content: center;
    width: 100%;
}

.center_info {
    background: #00FFFF;
    color: #000;
    display: -webkit-flex;
    display: -ms-flex;
    display: flex;
    align-items: center;
    justify-content: center;
    width: 100%;
    margin: 0 15px;
}

.right_info {
    background: #ADFF2F;
    color: #000;
    display: -webkit-flex;
    display: -ms-flex;
    display: flex;
    align-items: center;
    justify-content: center;
    width: 100%;
}

.example__mogo {
    color: #fff;
}

.example__mogo:hover {
    color: #7B68EE;
    text-decoration: none;
    text-transform: uppercase;
}

h3 {
    display: block;
    font-size: 40px;
    margin-block-start: 1em;
    margin-block-end: 1em;
    margin-inline-start: 0px;
    margin-inline-end: 0px;
    font-weight: bold;
}

```

```

}

.position {
  position: relative;
  height: 200px;
  width: 400px;
  margin-bottom: 30px;
  font-size: 20px;
}

.position.over img{
  position: absolute;
  top: 0;
  left: 0;
}

.position.down span {
  position: absolute;
  top: 0;
  left: 0;
  color: #fff;
}

footer {
  width: 90%;
  margin: 0 auto;
}

.footer__text {
  font-size: 2em;
}

.footer__references a {
  font-size: 1.5em;
  color: #fff;
  margin-left: 15px;
  text-decoration: underline;
  display: inline-block;
  padding-right: 20px;
}

.footer__container {
  display: -webkit-flex;
  display: -ms-flex;
  display: flex;
  -ms-align-items: center;
  align-items: center;
  justify-content: space-between;
}

```



Результат применения внешних каскадных таблиц можно увидеть на рисунке 2.1.1

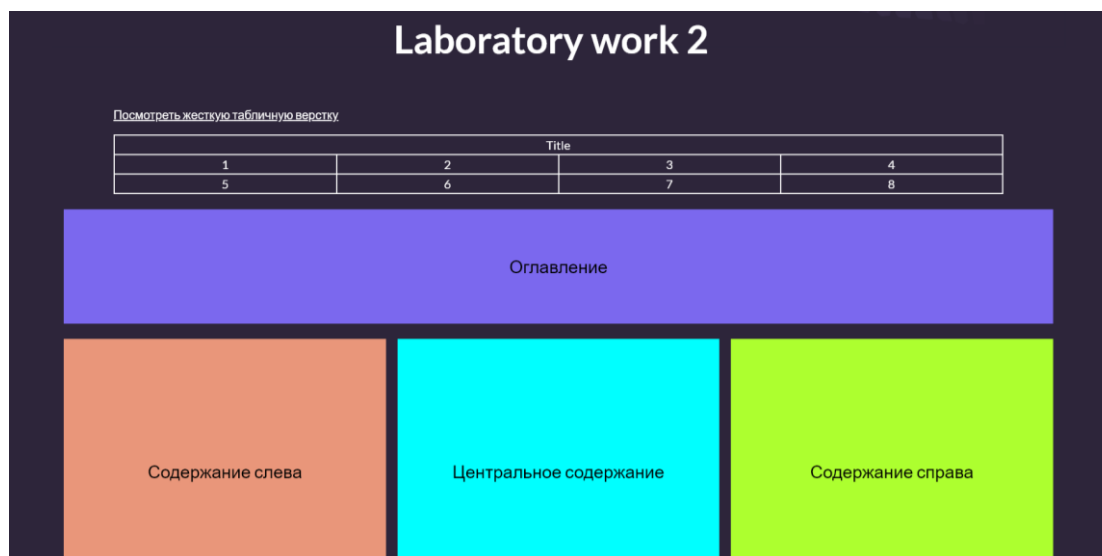


Рисунок 2.1.1 – результат применения внешних каскадных таблицстилей

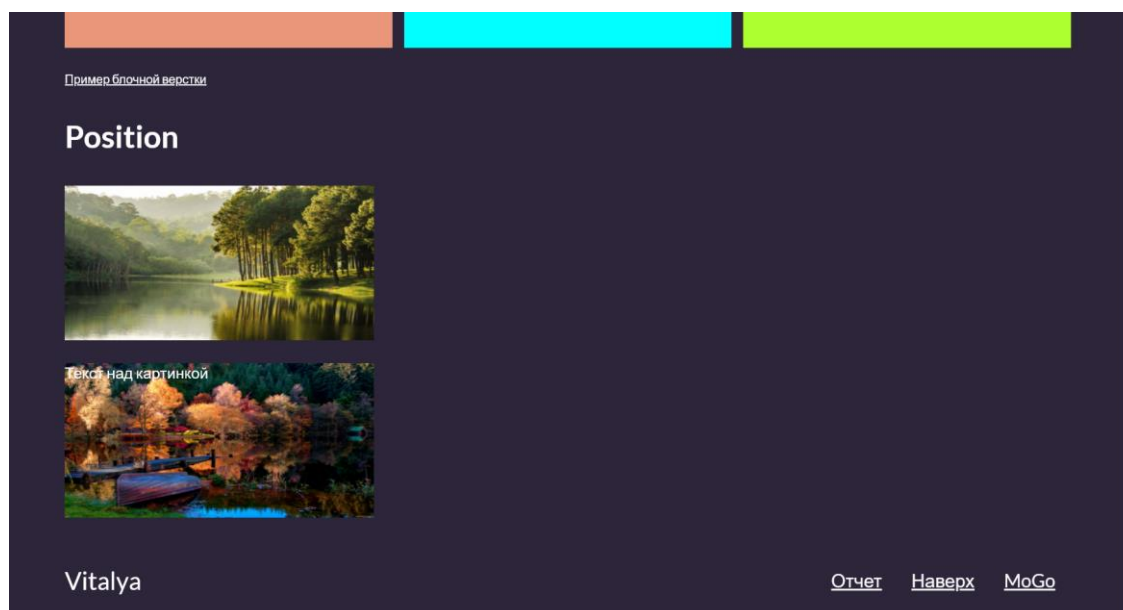


Рисунок 2.1.2 – результат применения внешних каскадных таблицстилей

## 2) Глобальные таблицы стилей:

```

<style type="text/css">
  * {
    box-sizing: border-box;
    padding: 0;
    margin: 0;
  }
  body {
    color: white;
    background: #2d253a !important;
    background: #fff;
    font-family: 'Lato', sans-serif;
  }

  .header__container {
    width: 100%;
    max-width: 1200px;
    margin: 0 auto;
  }

  .header {
    font-size: 50px;
    padding-top: 30px;
    width: 100%;

    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    z-index: 1000;
  }

  .header__logo {
    font-weight: 600;
    color: #fff;
  }

  .header__inner {
    display: flex;
    justify-content: space-between;
    align-items: center;
  }

  .header__titile {
    font-weight: 600;
    color: #fff;
  }
</style>

```

### 3) Внутренние таблицы стилей:

```
<a class="page2" href="page2.html" target= "_blank"><font color  
="#E52B50">Посмотреть жесткую табличную верстку</font></a>
```

Добавил фоновый цвет телу страницы. Наглядно можно посмотреть на рисунке 2.1.3

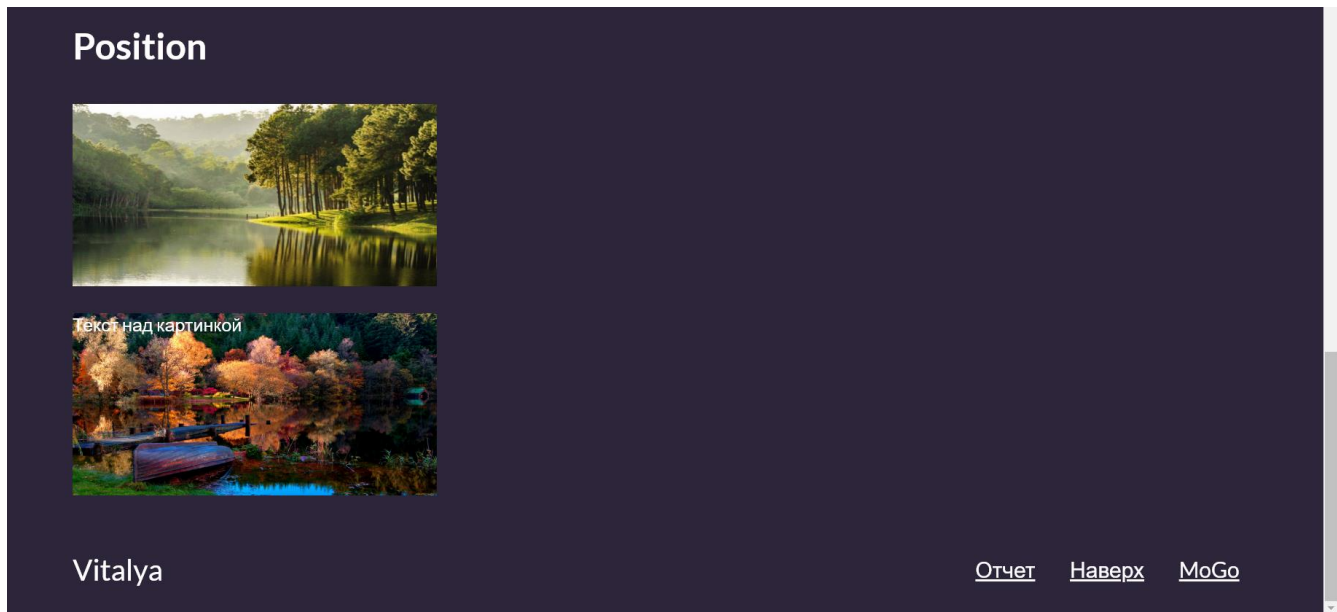


Рисунок 2.1.3 – результат применения внутренних каскадных таблиц стилей

- 4) Стилиевые классы, относящиеся ко всем элементам:

```
* {
    box-sizing: border-box;
    padding: 0;
    margin: 0;
}
```

У всех элементов убираются внутренние и внешние отступы, также свойства высоты и ширины задавали не размер контента, а размеры блока.

- 5) Стилиевые классы, относящиеся к конкретному элементу:

```
.example__mogo:hover {
    color: #7B68EE;
    text-decoration: none;
    text-transform: uppercase;
}
```

При наведении на текст меняется текст, убирается линия под текстом и буквы становятся заглавными, этот эффект можно увидеть на рисунке 2.1.5

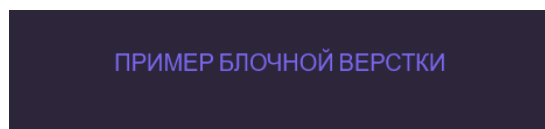


Рисунок 2.1.5 – результат применения стилиевых классов, относящихся к конкретному элементу

## 2.2 Задание 2

Реализуйте пример использования модификатора `!important`.

```
body {  
    color: white;  
    background: #2d253a !important;  
    background: #fff;  
    font-family: 'Lato', sans-serif;  
}
```

Результат показан на рисунке 2.2.1

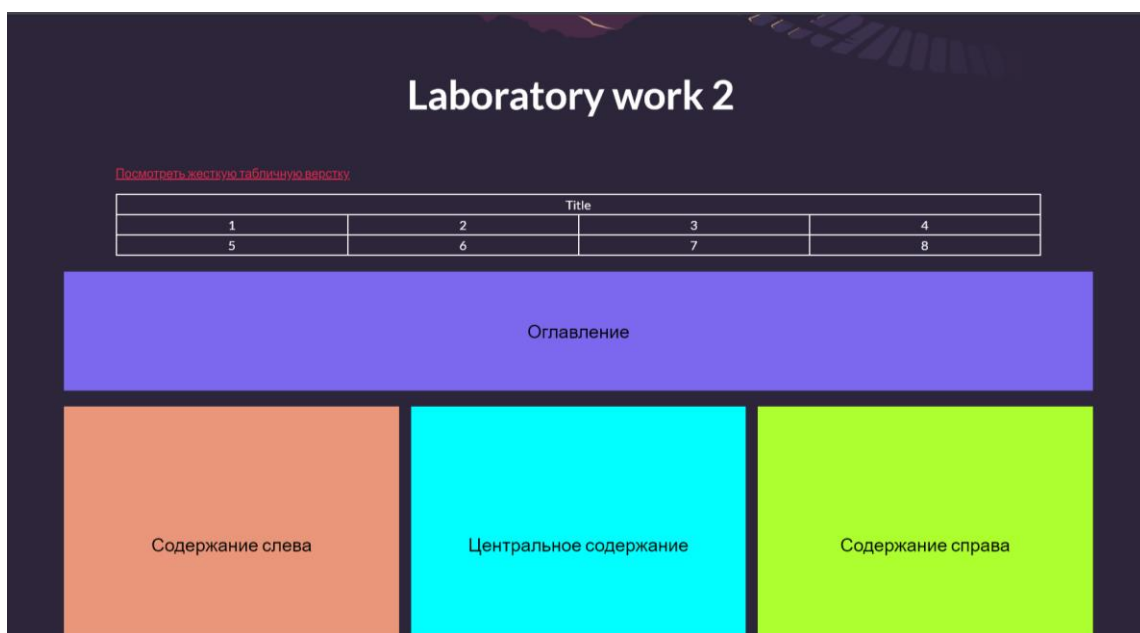


Рисунок 2.2.1 – результат применения `!important`

Как видно на рисунке 2.2.1, фон не стал белым, хоть строка с инструкцией заливки фона белым цветом шёл позже.

### 2.3 Задание 3

Реализуйте жесткую и резиновую табличную верстку любой странички. Сравните результаты поведения информации странички при изменении размеров окна и загрузке в разных браузерах. Опишите полученные результаты в отчете.

1) Жёсткая табличная вёрстка:

```
<html>
  <head>
    <title>Жесткая табличная верстка сайта</title>
    <style type="text/css">
      .header {width:100%; height:60px; background-color:#ffff00;
        padding:20px; text-align:center}
      .left.title {width:120px; height:100px; background-color:#dddddd;
        padding:15px; vertical-align:top}
      .center.title {width:500px; background-color:#ffffff;
        padding:15px; vertical-align:top}
      .right.title {width:120px; background-color:#dddddd; padding:15px;
        vertical-align:top}
      .left_col {width:120px; height:460px; background-color:#dddddd;
        padding:15px; vertical-align:top}
      .center_col {width:500px; background-color:#ffffff;
        padding:15px; vertical-align:top}
      .right_col {width:120px; background-color:#dddddd; padding:15px;
        vertical-align:top}
      footer {width:100%; padding:10px 0 10px 20px;
        backgroundcolor:#717dc9; font-size:13px}
    </style>
  </head>
  <body>
    <table cellpadding="0" cellspacing="0" width="860"
      align="center">
      <tr>
        <td colspan="3" class="header">История Египта</td>
      </tr>
      <tr>
        <td class="left title">Период</td>
        <td class="center title">События</td>
        <td class="right title">Культура</td>
      </tr>
      <tr>
        <td class="left_col">Доисторическое время,
        V–VI тысячелетия</td>
        <td class="center_col">5000–4000 гг., новый каменный век.
```

4000–3000 гг., медный век.

Взаимные переходы верхнеегипетских кочевых племён и  
нижнеегипетских крестьян. Господствующее положение городов Буто,  
Иераконполя и Абидоса.</td>

<td class="right\_col">Тотемные изображения. Местные божества  
в образе животных и растений. Поклонения матери-богине.  
Геометрическая орнамента нового каменного века. Первые письмена на  
памятниках Иераконполя.</td>

</tr>

<tr>

<td colspan="3" class="footer">&copy; Все права защищены</td>

</tr>

</table>

</body>

</html>

Как это выглядит на странице, показано на рисунке 2.3.1

| История Египта                          |   |   |
|---|---|---|
| Период                                  | События   | Культура  |
| Дописторическое время, V–VI тысячелетия | 5000–4000 гг., новый каменный век. 4000–3000 гг., медный век. Взаимные переходы верхнеегипетских кочевых племён и нижнеегипетских крестьян. Господствующее положение городов Буто, Пераконполя и Абидоса. | Тотемные изображения. Местные божества в образе животных и растений. Поклонения матери-богине. Геометрическая орнамента нового каменного века. Первые письмена на памятниках Пераконполя. |

© Все права защищены

Рисунок 2.3.1 – результат жёсткой табличной вёрстки

Если уменьшить окно браузера, то размеры элементов не изменятся, как показано на рисунке 2.3.2.

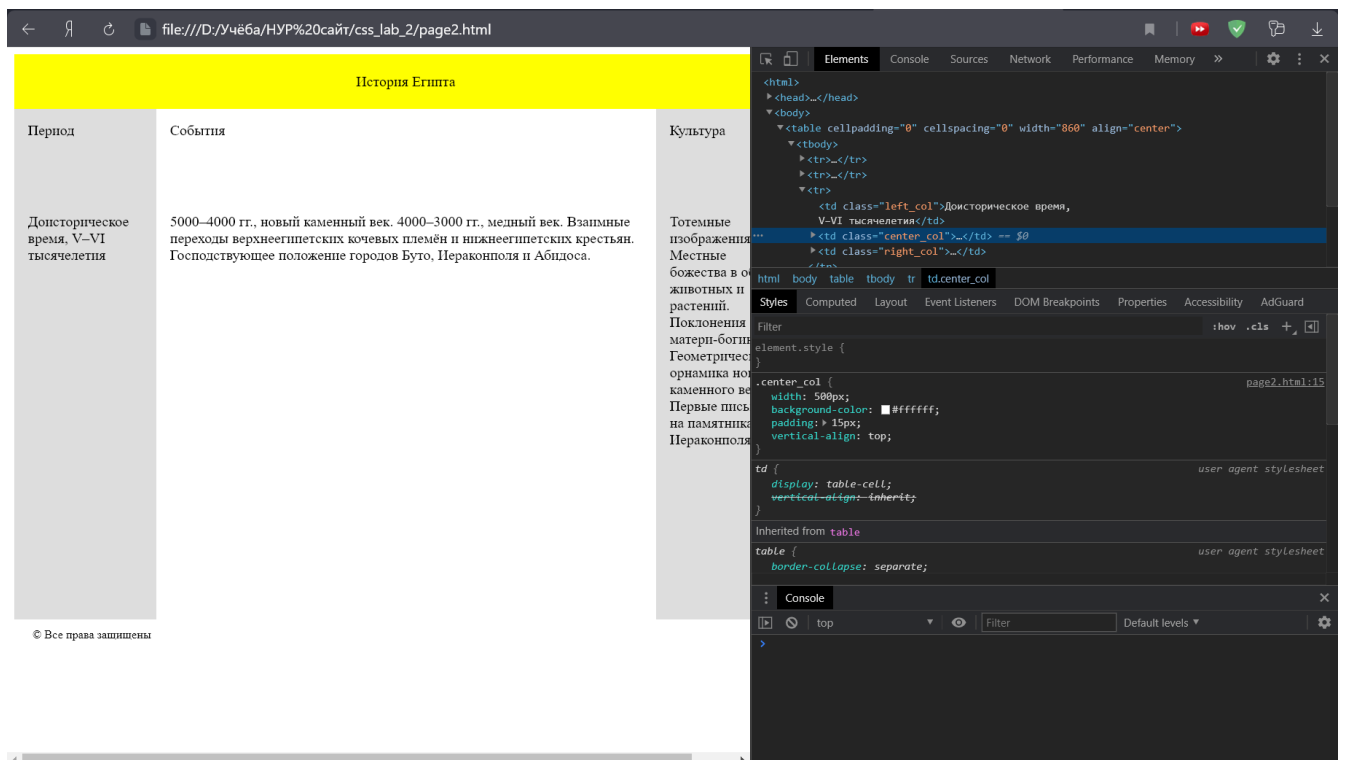


Рисунок 2.3.2 – результат жёсткой табличной вёрстки в уменьшенном окне браузера



## Резиновая табличная вёрстка:

Таблица, где атрибут width задан относительно и равен 100%. Результаты резиновой табличной вёрстки показаны на рисунках 2.3.3 и 2.3.4.

[Посмотреть жесткую табличную вёрстку](#)

| Title |   |   |   |
|-------|---|---|---|
| 1     | 2 | 3 | 4 |
| 5     | 6 | 7 | 8 |

Рисунок 2.3.3 – результат резиновой табличной вёрстки

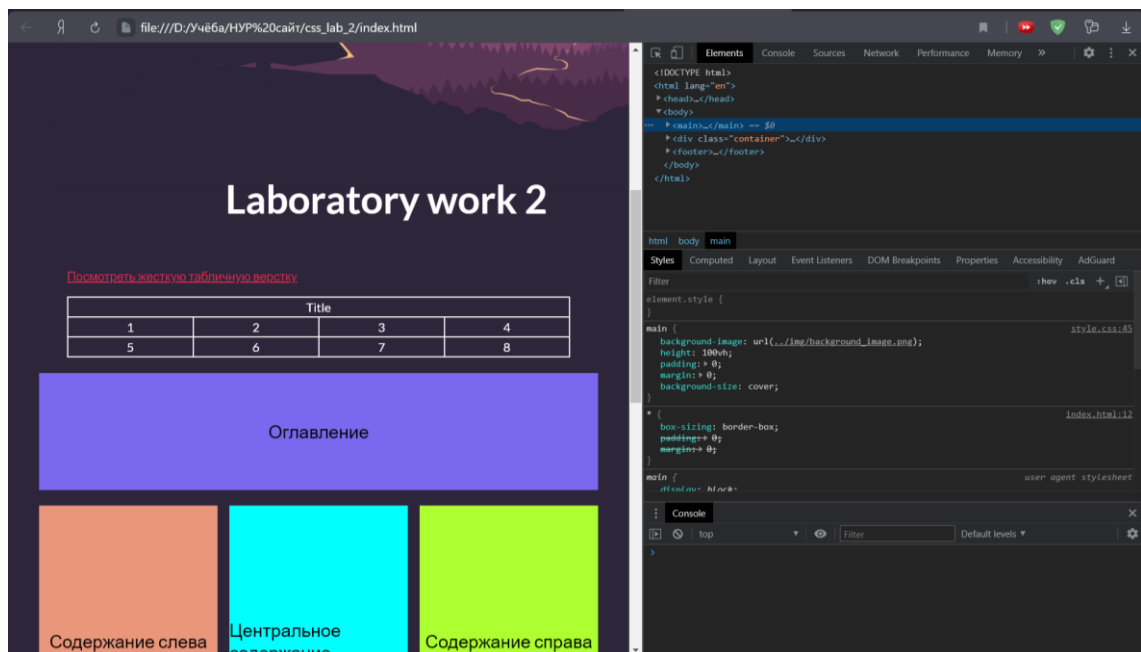


Рисунок 2.3.4 – результат резиновой табличной вёрстки

в уменьшенном окне браузера

## 2.4 Задание 4

Реализуйте блочную верстку странички.

1) Код HTML:

```
<div class="block_header">Оглавление</div>
  <div class="block_container">
    <div class="left_info">Содержание слева</div>
    <div class="center_info">Центральное содержание</div>
    <div class="right_info">Содержание справа</div>
  </div>
```

2) Код CSS:

```
.block_container {
  font-size: 1.45rem;
  width: 100%;
  display: -webkit-flex;
  display: -ms-flex;
  display: flex;
  justify-content: space-between;
  height: 350px;
  margin-bottom: 30px;
}
```

```
.block_header {
  width: 100%;
  background: #7B68EE;
  color: #000;
  font-size: 1.45rem;
  height: 150px;
  display: -webkit-flex;
  display: -ms-flex;
  display: flex;
  align-items: center;
  justify-content: center;
  margin: 20px 0;
}
```

```
.left_info {
  background: #E9967A;
  color: #000;
  display: -webkit-flex;
  display: -ms-flex;
  display: flex;
  align-items: center;
  justify-content: center;
  width: 100%;
}
```

```
.center_info {
  background: #00FFFF;
  color: #000;
  display: -webkit-flex;
```

```
display: -ms-flex;  
display: flex;  
align-items: center;  
justify-content: center;  
width: 100%;  
margin: 0 15px;  
}
```

```
.right_info {  
  background: #ADFF2F;  
  color: #000;  
  display: -webkit-flex;  
  display: -ms-flex;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  width: 100%;  
}
```

Результат блочной вёрстки показан на рисунке 2.4.1

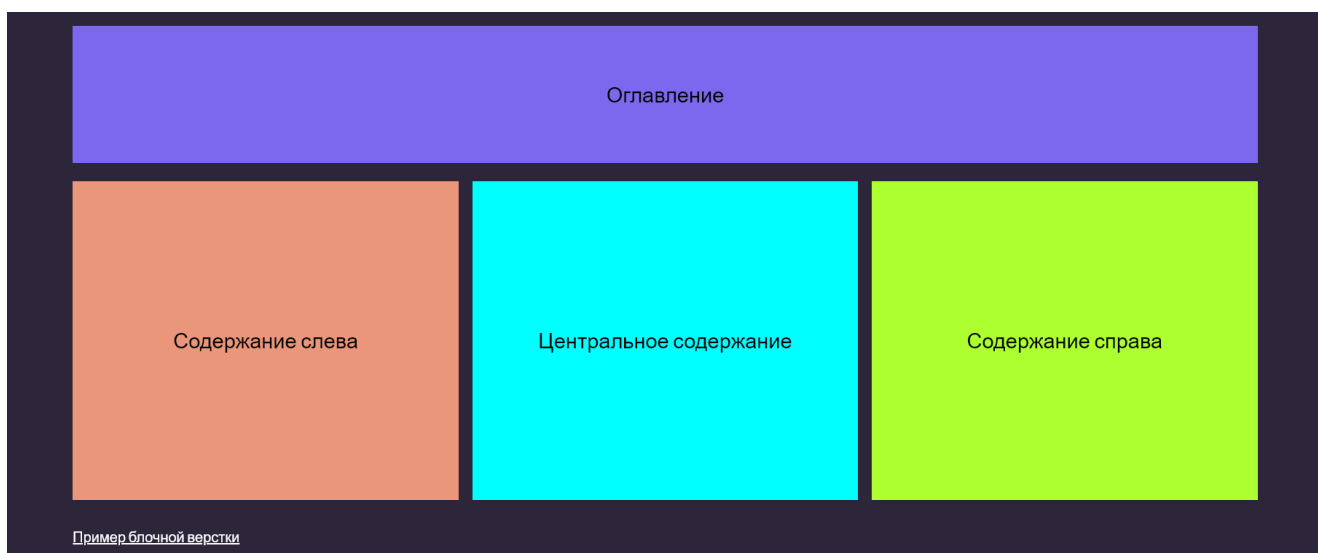


Рисунок 2.4.1 – результат блочной вёрстки

## 2.5 Задание 5

Реализуйте свободное позиционирование нескольких элементов странички, в том числе выполните расположение текста поверх рисунков и наоборот – расположение рисунков поверх текста.

Код HTML:

```
<div class="position over">
  <span>Текст</span>
  
</div>
<div class="position down">
  <span>Текст над картинкой</span>
  
</div>
```

Код CSS:

```
.position {
  position: relative;
  height: 200px;
  width: 400px;
  margin-bottom: 30px;
  font-size: 20px;
}

.position.over img {
  position: absolute;
  top: 0;
  left: 0;
}

.position.down span {
  position: absolute;
  top: 0;
  left: 0;
  color: #fff;
}
```

Результат показан на рисунке 2.5.1.



Рисунок 2.5.1 – результат применения абсолютного позиционирования

### 3 Таблица изученных свойств

|                       |   |
|-----------------------|---|
| color                 | задаёт цвет текста  |
| background-attachment | устанавливает поведение фонового изображения во время прокрутки страницы  |
| background-size       | устанавливает размеры фонового изображения  |
| text-decoration       | добавляет декорирование текста (подчеркивание, перечеркивание, линия над текстом)   |
| margin-left           | устанавливает величину внешнего отступа от левого края элемента   |
| margin-right          | устанавливает величину внешнего отступа от правого края элемента  |
| margin-top            | устанавливает величину внешнего отступа от верхнего края элемента   |
| margin-bottom         | устанавливает величину внешнего отступа от нижнего края элемента  |
| width                 | устанавливает ширину элемента   |
| padding               | устанавливает величину внутренних отступов элемента в одном объявлении  |
| float                 | определяет будет ли элемент плавающим   |
| position              | указывает какой тип позиционирования используется для элемента (статический, относительный, абсолютный или фиксированный) |
| background            | позволяет установить необходимые свойства фона в одном объявлении (универсальное свойство)                                |
| visibility            | определяет, является ли элемент видимым   |
| opacity               | определяет уровень прозрачности для элемента  |
| filter                | устанавливает визуальный эффект (фильтр) для элемента (например, размытие или изменение цвета)                            |

|                |  |
|----------------|--|
| border-radius  | позволяет определить форму границ элемента   |
| z-index        | определяет порядок расположения позиционированных элементов по оси Z (порядок наложения элементов друг на друга) |
| line-height    | отвечает за величину межстрочного интервала  |
| vertical-align | определяет вертикальное позиционирование встроенных строчных (inline) элементов или ячеек таблицы (table-cell)   |
| padding-left   | устанавливает левый внутренний отступ элемента   |
| padding-right  | устанавливает правый внутренний отступ элемента  |
| border-left    | устанавливает все свойства левой границы в одном объявлении  |
| border-right   | устанавливает все свойства правой границы в одном объявлении   |
| border-top     | устанавливает все свойства верхней границы в одном объявлении  |
| padding-top    | устанавливает верхний внутренний отступ элемента   |
| padding-bottom | устанавливает нижний внутренний отступ элемента  |
| border-bottom  | устанавливает все свойства нижней границы в одном объявлении   |
| display        | определяет как должен отображаться определенный элемент HTML   |
| flex-direction | позволяет указать направление, в соответствии с которым располагаются флекс элементы внутри флекс контейнера.    |

#### **4 Заключение**

В ходе данной лабораторной работы я познакомился с CSS, изучил основные приёмы верстки страниц.



## 5 Ответы на контрольные вопросы

1) CSS (Cascading Style Sheets – каскадные таблицы стилей) – формальный язык описания внешнего вида документа, написанного с использованием языка разметки. Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и XHTML, но может также применяться к любым XML-документам, например к SVG или XUL.

2) При использовании внешних таблиц стилей описание селекторов и их значений располагается в отдельном файле.

3) Встроенные таблицы стилей применяются к одиночному тегу как расширение. К ним нельзя применить классы, так как мы непосредственно влияем на элемент, который необходимо стилизовать.

4) `<style>`

5) Создается отдельный CSS-файл, где описываются все селекторы и их значения. Можно связать как с помощью тега `link`, так и с помощью директивы `@important`.

6) `<style>`

`p { color: green; }`

`</style>`

В теге `<style>` описываются селектор, его свойство и значение. В данном случае селектором выступает `p`, свойством `color`, а значением `green`.

7) Стилиевой класс задается через точку. После чего пишем пользовательское имя и описываем значения свойств класса. Например:

`.header {width:100%; height:60px;}`

8) В CSS существует два типа чисел: целые (integer) и вещественные (real). Процентное значение (percentage value) – это вычисляемое вещественное число, за которым следует знак процента (%). Процентные значения практически всегда выражены относительно другого значения, которым может быть все что угодно, включая значение другого свойства того же элемента, значение, унаследованное от родительского элемента, или значение элемента предка. Цвета в CSS можно задавать различными способами. Существует два варианта задания цвета, основанных на функциональном формате записи RGB (functional RGB notation). Обобщенный синтаксис кодировки цвета – `rgb(color)`, где `color` представляет собой комбинацию трех процентных значений или целых чисел. Модель RGBA – вариант модели RGB, но с дополнительным параметром (альфа-канал), который задает прозрачность цвета. CSS позволяет определять цвет с помощью шестнадцатеричной записи. При такой форме записи цвет задается посредством объединения трех шестнадцатеричных чисел в диапазоне от 00 до FF. Обобщенный синтаксис для этой формы записи – `#RRGGBB`. Для измерения длины существует множество способов. Различают два типа единиц измерения длины: абсолютные единицы измерения (absolute length units) и относительные (relative length units). К абсолютным единицам измерения относятся пункты (pt) и пики (pc). Относительные единицы измерения получили такое название потому, что они измеряются относительно других единиц измерения. Измеряемое ими фактическое (или абсолютное) расстояние может меняться под действием внешних факторов, таких как разрешение экрана, ширина

области просмотра, предпочтительные настройки пользователя и массы других параметров. Кроме того, для некоторых относительных единиц измерения их размер практически всегда измеряется относительно использующего их элемента и соответственно будет меняться от элемента к элементу:

- px – пиксел – это точка на экране.
- em – в CSS один «em» – это значение свойства font-size заданного шрифта.
- rem – это единица измерения em, отсчет которой ведется относительно значения font-size элемента body.

9) CSS использует тот же синтаксис «блока комментариев», что и языки семейства C — вы начинаете комментарий с / \* , и заканчиваете его \* /.

10) Под каскадированием понимается одновременное применение разных стилевых правил к элементам документа

— с помощью подключения нескольких стилевых файлов, наследования свойств и других методов. Чтобы не возникало конфликтов существуют некоторые приоритеты. Если в таблице есть селекторы с одинаковой специфичностью, то для элемента устанавливается значение нижнего селектора.

11) Использование модификатора important

12) Свободное позиционирование позволяет задать произвольное положение и размер элемента страницы. Задается с помощью свойства position и его значений absolute или relative.

- absolute – абсолютное позиционирование, элемент выпадает из потока и располагается относительно ближайшего родительского элемента со значением свойства position

отличным от static (по умолчанию это элемент body).

- relative – относительное позиционирование – промежуточный вариант между static и absolute: элементы можно сдвигать с помощью свойств top, left, right и bottom (при этом образующееся пустое место не заполняется), а также менять порядок наложения этого элемента с помощью свойства z-index (переместить на задний план или, наоборот, ближе к зрителю).

13) Вёрстка веб-страниц - процесс формирования веб-страниц в текстовом либо WYSIWYG-редакторе, а также результат этого процесса, то есть собственно веб-страницы.

14) Жесткая и резиновая табличные верстки. Блочная вёрстка или верстка div'ами.

15) Табличная вёрстка выполняется с помощью таблиц, которые вкладываются друг в друга. Жесткость такой верстки заключается в том, что размеры таблицы заданы в пикселях, а не в процентах, то есть они не зависят от размера монитора или окна браузера.

16) В случае резиновой табличной верстки размеры таблицы задаются относительных величинах.

17) В случае блочной верстки каждый логический элемент (текст, картинка, таблица) рассматриваются в виде отдельного блока, и таким же способом размещаются на странице. HTML- элемент <DIV> ... </DIV> позволяет создавать блочные элементы. При этом можно объединять несколько элементов (в том числе и блочных) в один блочный элемент.

18) Рамка представляет собой границы, которые окружают содержимое элемента.

19) Поле называется расстояние от внутреннего края рамки

элемента до воображаемого прямоугольника, ограничивающего его содержимое.

20) Отступом является пространство от границы текущего элемента до внутренней границы его родительского элемента.

21) `<div class="block1">`

Lorem ipsum dolor sit amet, consectetur

adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

`</div>`

22) Обозначение селектора идентификатора

23) Обозначение универсального селектора

24) «Шапка»-верхний блок страницы. «Контент»- любой вид информации, которая находится на странице. Может быть графическим, мультимедийным или текстовым. «Подвал»- размещается в нижней части страницы, противоположность

«шапки».