

Лабораторная работа 5. Введение в CSS

- CSS (англ. Cascading Style Sheets «каскадные таблицы стилей») — описание внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML).
 - CSS используется создателями веб-страниц для задания цветов, шрифтов, стилей, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц.
 - Основной целью разработки CSS является ограждение и отделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS).
-

Существуют разные способы добавить стили на страницу. Можно написать их прямо в HTML в теге [<style>](#). Можно подключить внешний CSS-файл. Можно задать стили с помощью JavaScript.

В программе VS Code создайте файл index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
</head>
<body>
  Мороз и солнце, день чудесный.
</body>
</html>
```

Создайте структуру проекта:

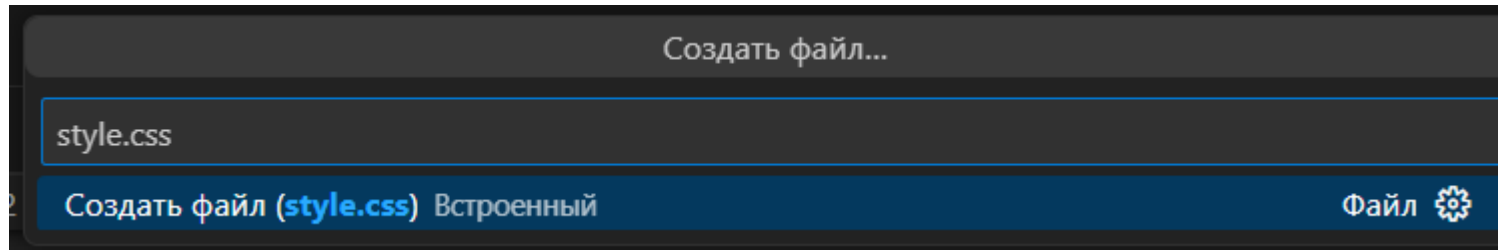


Внешний файл со стилями

При этом подходе CSS пишется в отдельном файле с расширением .css. Такие стили называют связанными.

Для связывания страницы с файлом стилей используется тег <link> внутри <head>

В программе VS Code создайте файл style.css и сохраните его в папке CSS.



В файл CSS вставьте код и сохраните:

```
1  body {  
2      background-color: #fafafa;  
3      color: navajowhite;  
4      font-size: 30px;  
5  }
```

Теперь подключим файл стилей к нашему файлу index.html с помощью тега link.

Атрибут rel="stylesheet" указывает, что этот файл является таблицей стилей. В href указывается относительный или абсолютный путь к CSS-файлу.

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <link rel="stylesheet" href="css/style.css">
9 </head>
10 <body>
11   Мороз и солнце, день чудесный.
12 </body>
13 </html>
```

Результат:

Мороз и солнце, день чудесный.

Мы можем подключать последовательно несколько файлов стилей, а затем делать сброс стилей. Приоритет будет иметь последний файл.

Преимущества

- Можно использовать один CSS-файл для нескольких страниц. Изменение стилей в таком файле автоматически применится ко всем страницам, к которым он подключен.
- При первой загрузке страницы файл со стилями кэшируется, и в следующие разы она открывается быстрее.
- Во внешних стилях можно свободно использовать псевдоклассы и псевдоэлементы. Например, задавать интерактивные состояния отдельно выбранным кнопкам, стилизовать каждый чётный элемент списка и тому подобное.

Недостатки

- Чтение стилей приостанавливает чтение кода страницы. Если файл, к которому вы обращаетесь, находится на отдалённом и недоступном сервере, это может повлиять на работоспособность остальных элементов страницы.

Встроенные стили

Вариант, когда CSS-свойства описываются в самом HTML-документе внутри тега `<style>`.

В том случае, когда нам нужны минимальные изменения, стили можно прописать непосредственно в файле `index`. Это также позволит не обращаться лишний раз к серверу.

- Закомментируйте подключение файла стилей и добавьте в код фрагмент кода файла `css`. Измените цвета.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <!-- <link rel="stylesheet" href="css/style.css" -->
9  </head>
10 <body>
11     <style>
12         body {
13             background-color: #fafafa;
14             color: green;
15             font-size: 30px;
16         }
17     </style>
18     Мороз и солнце, день чудесный.
19 </body>
20 </html>
```

Результат:

Мороз и солнце, день чудесный.

Преимущества

- Поскольку код находится непосредственно в HTML-файле, браузер не загружает сторонние файлы. Это позволяет отрисовать страницу быстрее.
- Встроенные стили работают изолированно и применяются непосредственно к странице, на которой прописаны.
- Можно использовать псевдоклассы и псевдоэлементы.

Недостатки

- С каждым новым правилом вес HTML-файла будет увеличиваться и страница будет загружаться медленнее.
- Со временем такие же стили могут понадобиться на других страницах, и CSS придётся дублировать.

Селекторы CSS

Наша задача: стилизовать все заголовки H1 на сайте и написать их жирным шрифтом.

Перед тем, как стилизовать, мы должны найти эти заголовки. CSS селектор позволяет выбрать тег по разным признакам.

Самый простой CSS селектор - это название самого тега.

Откройте файл CSS и удалите всё содержимое. Стилизуем заголовки H1 так, чтобы они были размером 30 пикселей и наклонные.

```
1  h1 {  
2      font-size: 30px;  
3      font-style: italic;  
4  }
```

Файл index оставьте таким:

```
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4      <meta charset="UTF-8">  
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">  
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7      <title>Document</title>  
8      <link rel="stylesheet" href="css/style.css">  
9  </head>  
10 <body>  
11     <h1>  
12         Мороз и солнце, день чудесный.  
13     </h1>  
14 </body>  
15 </html>
```

Посмотрите результат:

Мороз и солнце, день чудесный.

В процессе написания редактор подсказывает нам многие параметры. Ознакомьтесь с ними внимательно.

Добавим ещё один заголовок H1:

```
10 <body>
11   <h1>
12     Мороз и солнце, день чудесный.
13   </h1>
14
15   <h1>
16     Люблю грозу в начале мая
17   </h1>
18
19 </body>
20 </html>
```

Сохраните и убедитесь, что все заголовки изменили свой стиль.

Мороз и солнце, день чудесный.

Люблю грозу в начале мая

Далее добавьте тег h2. Он, естественно, не принимает никаких дополнительных стилей.

```
<body>
  <h1>
    Мороз и солнце, день чудесный.
  </h1>

  <h1>
    Люблю грозу в начале мая
  </h1>
  <h2>
    У Лукоморья дуб зеленый
  </h2>
</body>
```

Однако у них есть общая черта: они заголовки. И если мы хотим, чтобы все заголовки были, например, красные, то можем указать их через запятые.

```
1  h1 {  
2      font-size: 30px;  
3      font-style: italic;  
4  }  
5  h1, h2 {  
6      color: red  
7  }
```

Убедитесь, что все заголовки стали красными.

Использование атрибута Class

Наиболее предпочтительный способ объединения тегов является создание классов.

Отнесем все заголовки к классу heading:

```
9   </head>
10  <body>
11    <h1 class="heading">
12      Мороз и солнце, день чудесный.
13    </h1>
14
15    <h1 class="heading">
16      Люблю грозу в начале мая
17    </h1>
18    <h2 class="heading">
19      У Лукоморья дуб зеленый
20    </h2>
21  </body>
22  </html>
```

Для того, чтобы обратиться в CSS по этому атрибуту, поставим точку. Когда браузер читает стили и видит точку, он понимает, что дальше будет идти название класса.

```
1  h1 {
2    font-size: 30px;
3    font-style: italic;
4  }
5  .heading {
6    color: green
7  }
```

Результат:

Мороз и солнце, день чудесный.

Люблю грозу в начале мая

У Лукоморья дуб зеленый

Селектор выбора по ID

Мы можем указать в требуемых тегах атрибут id и его название.

```
10 <body>
11   <h1 class="heading">
12     Мороз и солнце, день чудесный.
13   </h1>
14
15   <h1 id="ID" class="heading">
16     Люблю грозу в начале мая
17   </h1>
18     <h2 class="heading">
19       У Лукоморья дуб зеленый
20     </h2>
21 </body>
22 </html>
```

В файле CSS укажем решетку, а за ней имя атрибута id (мы его назвали ID):

```
1  h1 {  
2    font-size: 30px;  
3    font-style: italic;  
4  }  
5  .heading {  
6    color: green  
7  }  
8  
9  #ID {  
10   color: blue;  
11 }
```

Теперь все элементы класса ID будут иметь указанный стиль. Проверьте результат.

Мороз и солнце, день чудесный.

Люблю грозу в начале мая

У Лукоморья дуб зеленый

Единицы измерения CSS

Каждое CSS-правило состоит из одной или нескольких пар «свойство-значение». В зависимости от свойства значение может быть, например, числом, или ключевым словом, или несколькими числами с единицами измерения (например, 45deg или 100px).

Абсолютные величины

Абсолютные величины ни от чего не зависят и привязаны к физическим единицам измерения: дюймам или сантиметрам. Этот факт имеет исторические предпосылки. Когда появились первые программы предпечатной подготовки, возникла необходимость понимать, как картинка на экране будет соответствовать напечатанному варианту.

Экраны мониторов в то время были примерно одинаковые, и оказалось, что одному дюйму на экране соответствуют 72 экранных точки. Именно это соотношение зафиксировано в единице измерения pt. С развитием технологий улучшались экраны и их разрешение, в один дюйм стало уместиться 96 точек, и это было зафиксировано в px.

- px соответствуют пикселям на экране. 1px интерпретируется как 1/96 дюйма.
- cm — сантиметры. $1\text{cm} = 96\text{px} / 2.54$.
- in — дюймы. $1\text{in} = 96\text{px} = 2.54\text{cm}$.
- mm — миллиметры. $1\text{mm} = 1/10\text{cm}$.
- Q — четверть миллиметра. $1Q = 1/40\text{cm}$.
- pc — пики. $1\text{pc} = 1/16\text{in}$.

- pt — пункты. 1pt = 1/72in.

Абсолютные величины, отличные от пикселей, удобно использовать для вывода на печать.

На сегодняшний день существуют экраны с разной плотностью пикселей и разным разрешением, поэтому соотношение единиц измерения к точкам экрана утратило прежний смысл. Сантиметры и дюймы при выводе на экран не будут соответствовать своим реальным размерам.

Создайте, соответственно, html и css файлы:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <meta name="viewport" content="width=device-width, initial-scale=0.5">
6          <title>Изучаем CSS</title>
7          <link rel="stylesheet" href="css/style.css">
8      </head>
9      <body>
10         <div class="parent">
11             <div class="child"></div>
12             <div class="child"></div>
13             <div class="child"></div>
14         </div>
15     </body>
16 </html>
```

```
1  .parent {
2      max-width: 1000px;
3      margin: 0 auto;
4      display: flex;
5  }
6
7  .child {
8      min-height: 200px;
9      width: 300px;
10     background-color: rgb(17, 221, 34);
11     border: 2px solid black;
12 }
```

Результат:



Начнем с контейнера, который называется parent.

Максимальная ширина контейнера 1000 пикселей.

Контейнер используется там, где есть верстка по сетке, и он ограничивает его размеры, чтобы пользователю было удобно читать материал даже на больших экранах. Если мы укажем на больших экранах размер в процентах, то изображение будет большим и не комфортным для глаз.

Абсолютные величины используются, когда нам нужно точно попадать в макет.

Относительные величины

Используются для задания размера или расстояния относительно чего-либо.

Иногда используется термин **вьюпорт** - область просмотра страницы в окне браузера.

- `em` задаёт величину относительно вычисленного размера шрифта элемента. Если для элемента `font-size: 20px`, то `2em` будут эквивалентны `40px`. При задании размера шрифта в `em` он считается относительно размера шрифта родительского блока.
- `rem` рассчитывается относительно размера шрифта корневого элемента. Для веб-страницы это `<html>`. По умолчанию размер шрифта для `<html>` равен `16px`. При таких условиях `2rem` эквивалентны `32px`.
- `vh` задаёт размер в процентах от высоты вьюпорта. `1vh` составляет 1% от высоты.
- `vw` задаёт размер в процентах от ширины вьюпорта. `1vw` составляет 1% от ширины.
- `vmin` считается относительно меньшей стороны вьюпорта. Если ширина вьюпорта больше высоты, то `vmin` будет считаться относительно высоты. И наоборот. `1vmin` составляет 1% от меньшей стороны вьюпорта.
- `vmax` считается относительно большей стороны вьюпорта. `1vmax` составляет 1% от большей стороны.

Изменим ширину на относительную в классе child. **Ширина - 33,33% относительно родительского класса parent.**

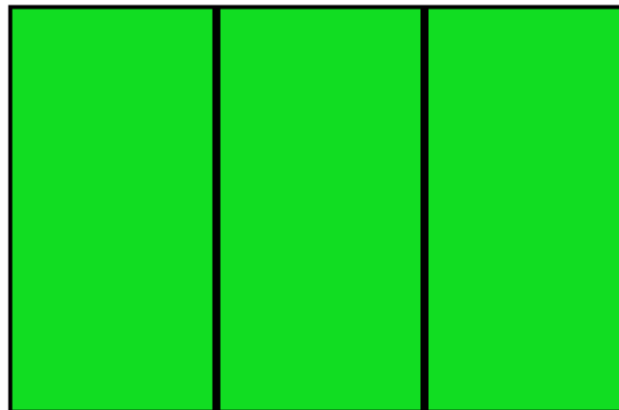
```
1  .parent {  
2    max-width: 1000px;  
3    margin: 0 auto;  
4    display: flex;  
5  }  
6  
7  .child {  
8    min-height: 200px;  
9    width: 33.33%;  
10   background-color: rgb(17, 221, 34);  
11   border: 2px solid black;  
12 }
```



А теперь измените значения, например, так:

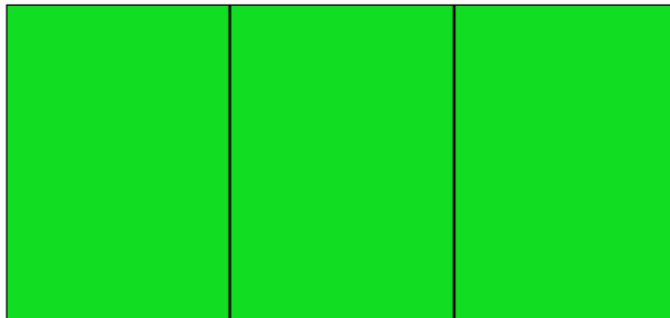
```
.child {  
  min-height: 200px;  
  width: 10%;  
  background-color: ■ rgb(17, 221, 34);  
  border: 2px solid □ #000;  
}
```

Результат:





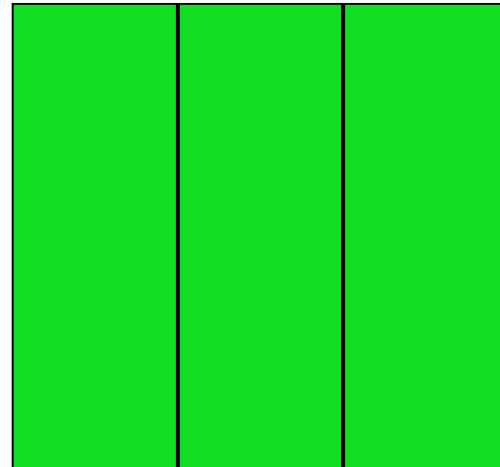
Размеры через величину выюпорта:

```
.child {  
  min-height: 50vh;  
  width: 33.33%;  
  background-color: rgb(17, 221, 34);  
  border: 2px solid black;  
}
```



Использование относительных величин очень удобно, так как позволяет менять разные величины, изменив только одно значение родительского класса. Например:

```
1  .parent {  
2    max-width: 500px;  
3    margin: 0 auto;  
4    display: flex;  
5  }  
6  
7  .child {  
8    min-height: 50vh;  
9    width: 33.33%;  
10   background-color:  rgb(17, 221, 34);  
11   border: 2px solid  #000;  
12 }
```



Работа со шрифтами

Допишите коды:


```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=0.5">
    <title>Изучаем CSS</title>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>
    <div class="parent">
      <div class="child"></div>
      <div class="child"></div>
      <div class="child"></div>
    </div>

    <div class="block">
      <p>Дай, Джим, на счастье лапу мне</p>
    </div>
  </body>
</html>
```

```
.parent {  
  max-width: 500px;  
  margin: 0 auto;  
  display: flex;  
}  
  
.child {  
  min-height: 50vh;  
  width: 33.33%;  
  background-color: #17222d;  
  border: 2px solid black;  
}  
  
.block {  
  font-size: 50px;  
}  
  
.block p {  
  font-size: 1em;  
  margin-bottom: 1em;  
}
```

- **em** задаёт величину относительно размера шрифта родительского блока. Если для элемента `font-size: 20px`, то `2em` будут эквивалентны `40px`.

У нас есть класс `block`, родительский по отношению к классу `block p` (параграф).

Мы задали размер шрифта и отступы размером 1em, то есть одинарный относительно размера шрифта.

Результат:

Дай, Джим, на счастье лапу мне

Дай, Джим, на счастье лапу мне

Дай, Джим, на счастье лапу мне

Изменим размер шрифта на 10 px и увидим, что и межстрочный интервал изменился.

Дай, Джим, на счастье лапу мне

Дай, Джим, на счастье лапу мне

Дай, Джим, на счастье лапу мне

- **rem** рассчитывается относительно размера шрифта корневого элемента. Для веб-страницы это `<html>`. По умолчанию размер шрифта для `<html>` равен 16px. При таких условиях `2rem` эквивалентны 32px.

```
.parent {  
  max-width: 500px;  
  margin: 0 auto;  
  display: flex;  
}  
  
.child {  
  min-height: 50vh;  
  width: 33.33%;  
  background-color: rgb(17, 221, 34);  
  border: 2px solid black;  
}  
  
html {  
  font-size: 10px;  
}  
  
.block {  
  font-size: 10px;  
}  
  
.block p {  
  font-size: 5rem;  
}
```

Запустите. Мы видим, что увеличился и шрифт, и интервал.

Свойства CSS

Свойство Display

Начнем изучение свойств CSS.

Ранее мы узнали, что бывают строчные и блочные элементы.

Блочные элементы (такие как `<p>`, ``, `<div>`) начинаются с новой строки и занимают всё доступное им пространство по ширине, независимо от содержимого, строчные (такие как `<a>`, `<var>`, ``) являются частью строки и размеры элементов определяются их содержимым.

`<div>` — это блочный элемент, предназначенный для создания структурных блоков на всю ширину страницы, тогда как `` — это строчный элемент, использующийся для стилизации и выделения определенных фрагментов текста, не влияющий на его расположение в строке.

Создайте файлы `index` и `css`:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <meta name="viewport" content="width=device-width, initial-scale=0.5">
6          <title>Изучаем CSS</title>
7          <link rel="stylesheet" href="css/style.css">
8      </head>
9      <body>
10         <div>Блочный</div>
11         <span>Строчный</span>
12     </body>
13 </html>
```

CSS:

```
1  div, span {
2      background-color: aqua;
3  }
```

Результат:

Блочный
Строчный

Блочный элемент всегда занимает всю доступную ширину и всегда растягивается на 100%, строчный стремится занять как можно меньше места только для того, чтобы отобразить контент внутри него.

Отступы

Margin - отступ снаружи элемента

CSS свойство `margin` определяет внешний отступ на всех четырёх сторонах элемента. Это сокращение, которое устанавливает все отдельные поля одновременно: `margin-top`, `margin-right`, `margin-bottom` и `margin-left`.

Одновременные задания отступов сверху и далее по часовой стрелке (справа, снизу, слева).

Правило отсчета по часовой стрелке в будущем будет часто применяться.

В файле `index` наберите:


```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-
6     <title>Изучаем CSS</title>
7     <link rel="stylesheet" href="css/style.css">
8   </head>
9   <body>
10    <div class="block"> Дай, Джим, на счастье лапу мне</div>
11    <br>
12  </body>
13 </html>
```

В CSS :

```
1 .block {
2   display: inline-block;
3   background-color: aqua;
4   margin: 30px; 50px; 70px; 90px;
5 }
```

Результат:

Дай, Джим, на счастье лапу мне


Закомментируем строку (в CSS комментарии ставятся /*.....*/

```
1  .block {  
2      display: inline-block;  
3      background-color: aqua;  
4      /* margin: 30px; 50px; 70px; 90px; */  
5  }
```

Результат:


Дай, Джим, на счастье лапу мне

Вторая сокращенная запись: отступы сверху и снизу 10 пикселей, слева и справа 30 пикселей. Можно сказать так: отступы по вертикали 10, по горизонтали 30.


```
1  .block {  
2      display: inline-block;  
3      background-color:  aqua;  
4      /* margin: 30px; 50px; 70px; 90px; */  
5      margin: 10px; 30px;  
6  }
```

Проверьте результат.


Отступы со всех сторон одинаковые:

```
1  .block {  
2      display: inline-block;  
3      background-color:  aqua;  
4      /* margin: 30px; 50px; 70px; 90px; */  
5      /* margin: 10px; 30px; */  
6      margin: 10px;  
7  }
```

Отступы сверху и снизу 0, по горизонтали - 10.

```
1  .block {  
2      display: inline-block;  
3      background-color:  aqua;  
4      /* margin: 30px; 50px; 70px; 90px; */  
5      /* margin: 10px; 30px; */  
6      /*margin: 10px;*/  
7      margin: 0; 10px 0;  
8  }
```

Задание отступа для каждой стороны

```
1  .block {  
2      display: inline-block;  
3      background-color:  aqua;  
4      /* margin: 30px; 50px; 70px; 90px; */  
5      /* margin: 10px; 30px; */  
6      margin: 10px;  
7      margin-top: 50px;  
8      margin-right: 50px;  
9      margin-bottom: 50px;  
10     margin-left: 50px;  
11 }
```

Проверьте результат.

Padding - отступ внутри элемента.

Свойство padding устанавливает внутренние отступы/поля со всех сторон элемента. Область отступов это пространство между содержанием элемента и его границей.

Все правила аналогичны Margin.

Введите:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=0.5">
6      <title>Изучаем CSS</title>
7      <link rel="stylesheet" href="css/style.css">
8    </head>
9    <body>
10     <div class="block"> Дай, Джим, на счастье лапу мне</div>
11     <br>
12     <div class="padding-block"> Дай, Джим, на счастье лапу мне</div>
13   </body>
14 </html>
```

```
1  .block {
2      display: inline-block;
3      background-color: ■ aqua;
4      /* margin: 30px; 50px; 70px; 90px; */
5      /* margin: 10px; 30px; */
6      margin: 10px;
7      margin-top: 50px;
8      margin-right: 50px;
9      margin-bottom: 50px;
10     margin-left: 50px;
11 }
12 .padding-block {
13     display: inline-block;
14     background-color: ■ crimson;
15     color: ■ #fff;
16     padding: 10px 20px 30px 40px;
17 }
```

Результат: обратите внимание на текст внутри красного блока и то, насколько он отстоит от границ внутри.

Дай, Джим, на счастье лапу мне

Дай, Джим, на счастье лапу мне

Примените другие формы записи по аналогии с Margin.

Свойство Border

Свойство CSS border - это универсальное свойство для указания всех свойств границ за раз: border-width, border-style, и border-color.

Введите:

```
<body>
  <div class="block">
    Однажды, в студеную зимнюю пору, я из лесу вышел, был сильный мороз.
  </div>
</body>
```


Мы видим задание ширины границы и свойства solid (сплошная).

```
1 .block {  
2     border-width: 2px;  
3     border-style: solid;  
4     border-color: #ff0;  
5 }
```

Однажды, в студеную зимнюю пору, я из лесу вышел, был сильный мороз.

Попробуйте изменить цвет границы и её стиль.

Можно записывать так:

```
1 .block {  
2     border-top: 10px solid #ff0;  
3     border-bottom: 3px dashed black;  
4     border-right: 1px dotted red;  
5     border-left: 2px solid gray;  
6 }  
7
```

1 пиксел	3 пиксела	5 пикселей	7 пикселей
dotted	dotted	dotted	dotted
dashed	dashed	dashed	dashed
solid	solid	solid	solid
double	double	double	double
groove	groove	groove	groove
ridge	ridge	ridge	ridge
inset	inset	inset	inset
outset	outset	outset	outset

Высота и ширина блока - height и width (с блочными элементами)

Width и height - устанавливает ширину и высоту блочных или заменяемых элементов.

Если мы укажем абсолютную величину, то при изменении экрана размер будет неизменным.

Введите следующие данные. Текст:

У лукоморья дуб зелёный;
Златая цепь на дубе том:
И днём и ночью кот учёный
Всё ходит по цепи кругом;
Идёт направо — песнь заводит,
Налево — сказку говорит.
Там чудеса: там леший бродит,
Русалка на ветвях сидит;
Там на неведомых дорожках
Следы невиданных зверей;
Избушка там на курьих ножках
Стоит без окон, без дверей;

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <meta name="viewport" content="width=device-width, initial-scale=0.5">
6          <title>Изучаем CSS</title>
7          <link rel="stylesheet" href="css/style.css">
8      </head>
9      <body>
10         <div class="block">
11             У лукоморья дуб зелёный;
12             Златая цепь на дубе том:
13             И днём и ночью кот учёный
14             Всё ходит по цепи кругом;
15             Идёт направо — песнь заводит,
16             Налево — сказку говорит.
17             Там чудеса: там леший бродит,
18             Русалка на ветвях сидит;
19             Там на неведомых дорожках
20             Следы невиданных зверей;
21             Избушка там на курьих ножках
22             Стоит без окон, без дверей;
23         </div>
24
25
26
27     </body>
28 </html>
```

CSS:

```
1  .block {  
2      border: 1px solid black #000;  
3      width: 150px;  
4      height: 500px;  
5  }
```

Результат:

У лукоморья дуб
зелёный; Златая цепь
на дубе том: И днём и
ночью кот учёный
Всё ходит по цепи
кругом; Идёт направо
— песнь заводит,
Налево — сказку
говорит. Там чудеса:
там леший бродит,
Русалка на ветвях
сидит; Там на
неведомых дорожках
Следы невиданных
зверей; Избушка там
на курьих ножках
Стоит без окон, без
дверей;

Измените **height: 100px;**

Результат:

У лукоморья дуб
зелёный; Златая цепь
на дубе том: И днём и
ночью кот учёный
Всё ходит по цепи
крутом; Идёт направо
— песнь заводит,
Налево — сказку
говорит. Там чудеса:
там леший бродит,
Русалка на ветвях
сидит; Там на
неведомых дорожках
Следы невиданных
зверей; Избушка там
на курьих ножках
Стоит без окон, без
дверей;

Для того, чтобы избежать таких ситуаций, которые могут возникнуть при добавлении или удалении текста, мы можем задать минимальную и максимальную ширину и высоту.

Введите:

У лукоморья дуб зелёный;
Златая цепь на дубе том:
И днём, и ночью кот учёный
Всё ходит по цепи кругом;
Идёт направо — песнь заводит,
Налево — сказку говорит.
Там чудеса: там леший бродит,
Русалка на ветвях сидит;
Там на неведомых дорожках
Следы невиданных зверей;
Избушка там на курьих ножках
Стоит без окон, без дверей;
Там лес и дол видений полны;
Там о заре прихлынут волны
На брег песчаный и пустой,
И тридцать витязей прекрасных
Чредой из вод выходят ясных,
И с ними дядька их морской;
Там королевич мимоходом
Пленяет грозного царя;
Там в облаках перед народом
Через леса, через моря
Колдун несёт богатыря;

В темнице там царица тужит,
А бурый волк ей верно служит;
Там ступа с Бабою Ягой
Идёт, бредёт сама собой,
Там царь Кощей над златом чахнет;
Там русский дух... там Русью пахнет!
И там я был, и мёд я пил;
У моря видел дуб зелёный;
Под ним сидел, и кот учёный
Свои мне сказки говорил.

```
9      <body>
10      |      <div class="max-width">
11      |      У лукоморья дуб зелёный;
12      |      Златая цепь на дубе том:
13      |      И днём, и ночью кот учёный
14      |      Всё ходит по цепи кругом;
15      |      Идёт направо – песнь заводит,
16      |      Налево – сказку говорит.
17      |      Там чудеса: там леший бродит,
18      |      Русалка на ветвях сидит;
19      |      Там на неведомых дорожках
20      |      Следы невиданных зверей;
21      |      Избушка там на курьих ножках
22      |      Стоит без окон, без дверей;
23      |      Там лес и дол видений полны;
24      |      Там о заре прихлынут волны
25      |      На брег песчаный и пустой,
26      |      И тридцать витязей прекрасных
27      |      Чредой из вод выходят ясных,
28      |      И с ними дядька их морской;
29      |      Там королевич мимоходом
30      |      Пленяет грозного царя;
31      |      Там в облаках перед народом
32      |      Через леса, через моря
33      |      Колдун несёт богатыря;
34      |      В темнице там царевна тужит,
35      |      А бурый волк ей верно служит;
36      |      Там ступа с Бабою Ягой
37      |      Идёт, бредёт сама собой,
```

```
1  .max-width {  
2      background-color: aqua;  
3      border-color: blueviolet;  
4      max-width: 1000px;  
5      min-width: 320px;  
6      max-height: 500px;  
7      min-height: 200px;  
8  }
```

Результат:

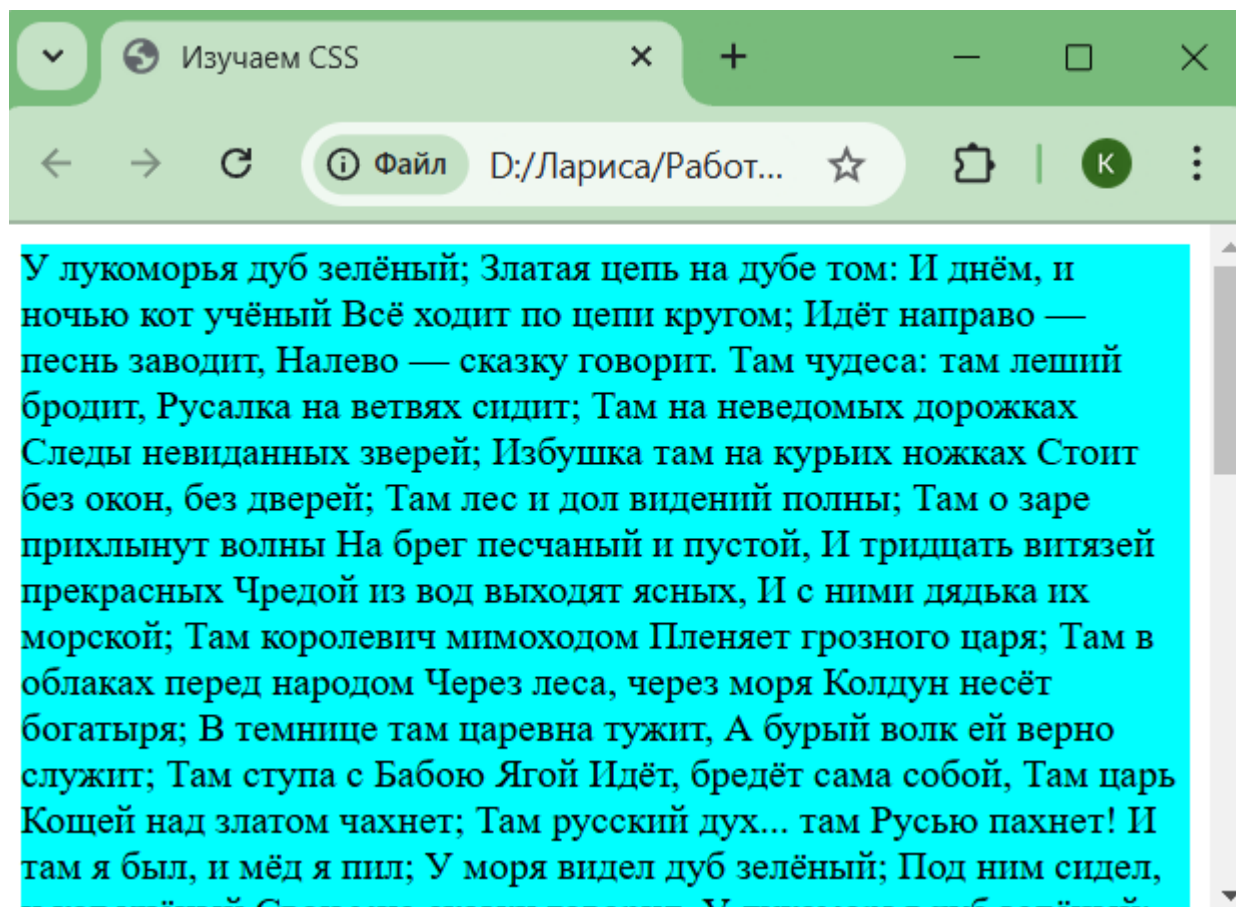
У лукоморья дуб зелёный; Златая цепь на дубе том: И днём, и ночью кот учёный Всё ходит по цепи кругом; Идёт направо — песнь заводит, Налево — сказку говорит. Там чудеса: там леший бродит, Русалка на ветвях сидит; Там на неведомых дорожках Следы невиданных зверей; Избушка там на курьих ножках Стоит без окон, без дверей; Там лес и дол видений полны; Там о заре прихлынут волны На брег песчаный и пустой, И тридцать витязей прекрасных Чредой из вод выходят ясных, И с ними дядька их морской; Там королевич мимоходом Пленяет грозного царя; Там в облаках перед народом Через леса, через моря Колдун несёт богатыря; В темнице там царевна тужит, А бурый волк ей верно служит; Там ступа с Бабою Ягой Идёт, бредёт сама собой, Там царь Кощей над златом чахнет; Там русский дух... там Русью пахнет! И там я был, и мёд я пил; У моря видел дуб зелёный; Под ним сидел, и кот учёный Свои мне сказки говорил.

А теперь ещё два раза вставьте текст:

У лукоморья дуб зелёный; Златая цепь на дубе том: И днём, и ночью кот учёный Всё ходит по цепи кругом; Идёт направо — песнь заводит, Налево — сказку говорит. Там чудеса: там леший бродит, Русалка на ветвях сидит; Там на неведомых дорожках Следы невиданных зверей; Избушка там на курьих ножках Стоит без окон, без дверей; Там лес и дол видений полны; Там о заре прихлынут волны На брег песчаный и пустой, И тридцать витязей прекрасных Чредой из вод выходят ясных, И с ними дядька их морской; Там королевич мимоходом Пленяет грозного царя; Там в облаках перед народом Через леса, через моря Колдун несёт богатыря; В темнице там царевна тужит, А бурый волк ей верно служит; Там ступа с Бабою Ягой Идёт, бредёт сама собой, Там царь Кощей над златом чахнет; Там русский дух... там Русью пахнет! И там я был, и мёд я пил; У моря видел дуб зелёный; Под ним сидел, и кот учёный Свои мне сказки говорил. У лукоморья дуб зелёный; Златая цепь на дубе том: И днём, и ночью кот учёный Всё ходит по цепи кругом; Идёт направо — песнь заводит, Налево — сказку говорит. Там чудеса: там леший бродит, Русалка на ветвях сидит; Там на неведомых дорожках Следы невиданных зверей; Избушка там на курьих ножках Стоит без окон, без дверей; Там лес и дол видений полны; Там о заре прихлынут волны На брег песчаный и пустой, И тридцать витязей прекрасных Чредой из вод выходят ясных, И с ними дядька их морской; Там королевич мимоходом Пленяет грозного царя; Там в облаках перед народом Через леса, через моря Колдун несёт богатыря; В темнице там царевна тужит, А бурый волк ей верно служит; Там ступа с Бабою Ягой Идёт, бредёт сама собой, Там царь Кощей над златом чахнет; Там русский дух... там Русью пахнет! И там я был, и мёд я пил; У моря видел дуб зелёный; Под ним сидел, и кот учёный Свои мне сказки говорил.

Проанализируйте результат, задавая разные значения.

Уменьшите размер окна. Что вы заметили?



Задание для самостоятельной работы

Стилизируйте созданную ранее страницу про животных (по вариантам)