Оформление текстовых блоков с помощью CSS

Владимир Языков
Основатель UsefulWeb

Владимир Языков

Основатель UsefulWeb

Аккаунты в соц.сетях

t.me/neizerth

План занятия

1. Селектор класса
2. Оформление текстовых блоков
3. Цвет в CSS
4. Свойства фоновых изображений

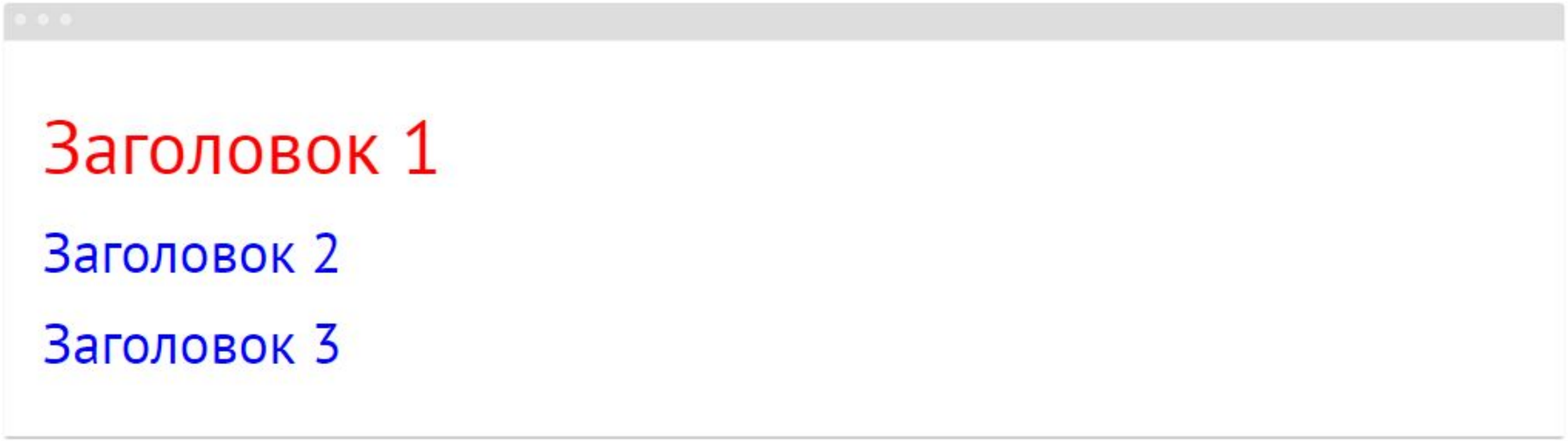
Селектор класса

1

Разные стили для одинаковых тегов

Как поступить в случае, когда нужны особые стили для одинаковых тегов?

```
<h2>Заголовок 1</h2>
<h2>Заголовок 2</h2>
<h2>Заголовок 3</h2>
```



Используем дополнительные теги

Конечно, можно окружить некоторые теги другими тегами и написать правила на основе этой вложенности:

```
<h2>Заголовок 1</h2>
<section><h2>Заголовок 2</h2></section>
<div><h2>Заголовок 3</h2></div>
```

Но это очень неудобно :-(

Селектор класса

Удобная возможность сделать тег или группу тегов особенными, что позволит применить к ним те стили, которые не нужны для прочих одноименных тегов.

```
<h2 class="main-header">Заголовок 1</h2>
<h2 class="simple-header">Заголовок 2</h2>
<h2 class="simple-header">Заголовок 3</h2>
```

```
.main-header {
  color: red;
  font-size: 50px;
}
.simple-header {
  color: blue;
}
```

Live Demo

Селектор класса

Чтобы обращаться к тегу по имени класса, нужно записать это имя с точкой перед началом:

```
.имя-класса {
  свойство: значение;
}
```

Полное совпадение

Важно: имя класса в CSS-стилях должно в точности совпадать с именем атрибута класса, записанного в разметке, например

```
<h2 class="main-head">Заголовок</h2>
```

и `.main-header` не будут совпадать, наше правило не сработает.

Комбинации для селектора класса

Для классов также действуют комбинации селектора потомков и дочернего селектора:

```
.menu .item {
  font-size: 10px;
}
.menu > .item {
  font-size: 20px;
}
```

Комбинация тег + класс

Также мы можем комбинировать тег с классом для получения более точной выборки элементов:

```
<h2 class="header">Заголовок 1</h2>
<h3 class="header">Заголовок 2</h3>
<h3 class="header">Заголовок 3</h3>
```

```
h2.header {
  color: green;
}
```

Live Demo

Результат работы комбинированного селектора

Решение задач

```
<h1 class="header">Заголовок</h1>
<p class="lead">Lorem ipsum…</p>
<p>Text</p>
<p>Text</p>
<h2 class="header">Заголовок списков</h2>
<ul class="big">
  <li class="first-level">item 1</li>
  <li class="first-level">item 2</li>
  <li class="first-level">item 3
    <ul>
      <li>item 1</li>
      <li>item 2</li>
      <li>item 3</li>
    </ul>
    </li>
</ul>
<ol>
  <li class="first-level">item 1</li>
  <li class="first-level">item 2</li>
  <li class="first-level">item 3</li>
</ol>
```

Live Demo

Перекрашиваем параграф

С помощью какого правила можно обратиться к первому параграфу и сделать его желтым?

```
.lead {
  color: yellow;
}
```

Live Demo

Увеличиваем шрифт заголовка

С помощью какого правила можно обратиться к заголовку текста первого уровня и увеличить его размер до 40px?

```
h1 {
  font-size: 40px;
}
```

Live Demo

Курсивные пункты

С помощью какого правила можно сделать текст в элементах первого уровня нумерованного списка курсивным?

```
ol .first-level {
  font-style: italic;
}
```

Live Demo

Меняем жирность заголовков

С помощью какого правила можно уменьшить жирность всех заголовков до значения 200?

```
.header {
  font-weight: 200;
}
```

Live Demo

Оформление текстовых блоков с помощью CSS

2

Основные свойства шрифта

На прошлом занятии мы уже успели познакомиться с некоторыми свойствами, при помощи которых мы можем управлять внешним видом шрифта. Давайте вспомним, какие это были свойства:

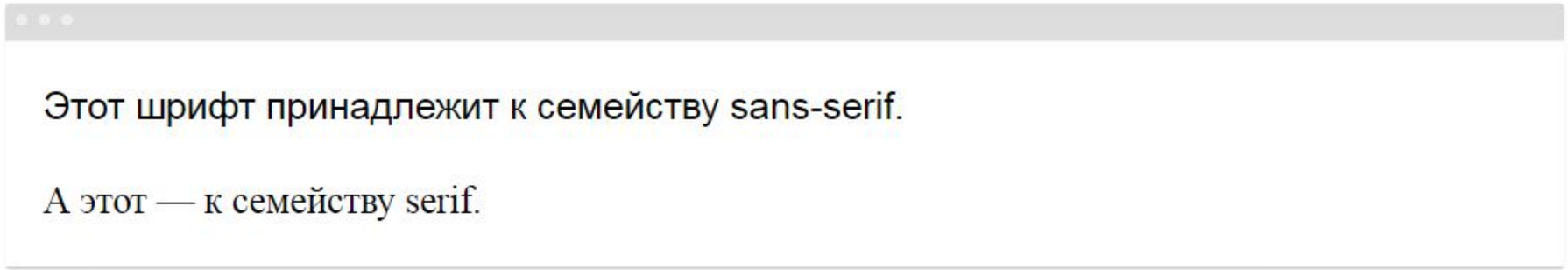
- `font-size` — задает размер шрифта;
- `font-weight` — задает жирность шрифта;
- `font-style` — задает стиль начертание шрифта;
- `color` — задает цвет шрифта.

Еще свойства шрифта

- `font-family` — гарнитура шрифта — serif, sans-serif, Arial;
- `line-height` — междустрочный интервал;
- `text-decoration` — underline сделает надпись подчеркнутой, overline добавит линию выше надписи, line-through — для перечеркнутой надписи;
- `text-transform` — uppercase сделает все символы заглавными, lowercase сделает все символы строчными; Capitalize сделает первую букву каждого слова заглавной.
- `text-align` — позволяет задавать выравнивание текста left, right — по левому и правому краю соответственно, center — по центру, justify — по ширине.

Эксперимент с семейством

```
.sans-serif {
  font-family: sans-serif;
}
.serif {
  font-family: serif;
}
```



Live Demo

Межстрочный интервал

```
.line-height-big {
  line-height: 50px;
}
```

Здесь величина межстрочного интервала равна 50px, поэтому можно заметить, что строчки находятся далеко друг от друга

Live Demo

Перечеркиваем текст

```
.underline {
  text-decoration: underline;
}
.overline {
  text-decoration: overline;
}
.line-through {
  text-decoration: line-through;
}
```

Live Demo

Меняем регистр

```
.uppercase {
  text-transform: uppercase;
}
.lowercase {
  text-transform: lowercase;
}
.capitalize {
  text-transform: capitalize;
}
```

ЭМО ТЕКСТ

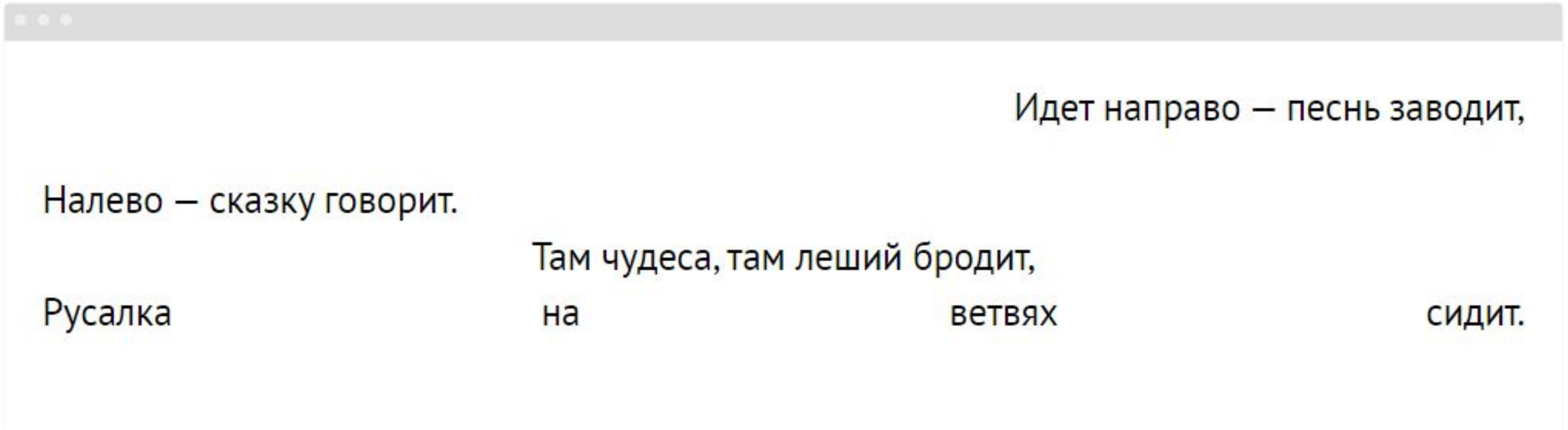
эмо текст

Пример Текста

Live Demo

Выравниваем текст

```
.align-right {
  text-align: right;
}
.align-center {
  text-align: center;
}
.align-justify {
  text-align: justify;
}
```



Live Demo

Свойства списков

Свойства списков

Для маркированных и нумерованных списков у нас есть несколько специальных свойств, которые не имеет смысла использовать для прочих тегов.

Задаем тип маркера

`list-style-type` — позволяет задать тип маркера.

Для маркированных списков доступны следующие значения:

- `disc` – Заполненный кружок (значение по умолчанию).
- `circle` – Пустой кружок.
- `square` – Квадрат.

Маркеры нумерованных списков

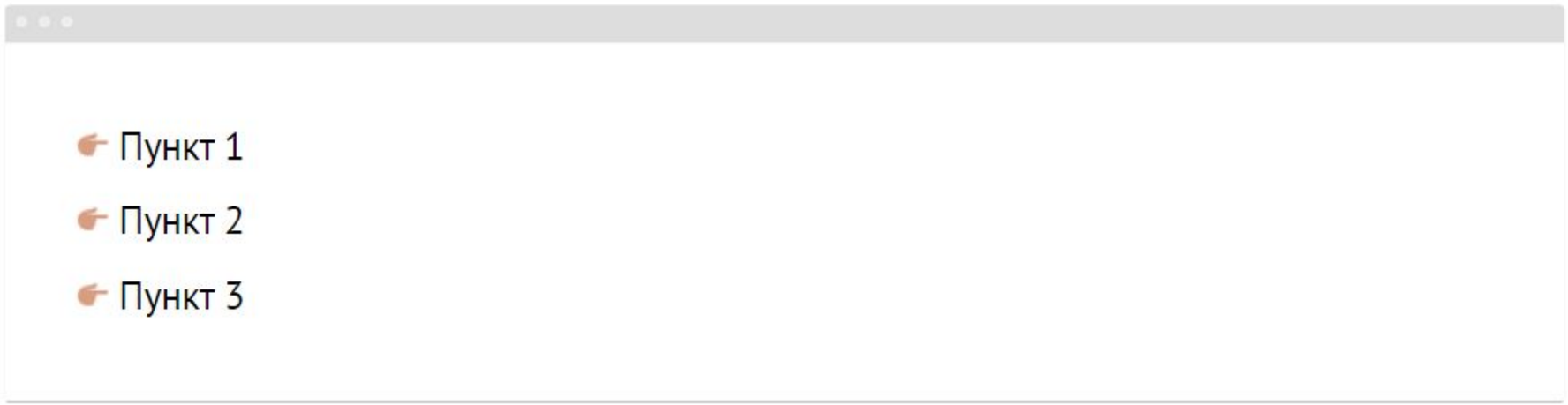
Для нумерованных списков доступны следующие значения:

1. `decimal` арабские цифры (значение по умолчанию).
ii. `lower-roman` – строчные римские цифры, например i , ii , iii.
II. `upper-roman` – заглавные римские цифры, I, II , III.
d. `lower-alpha` – строчные буквы латинского алфавита, a, b, c.
E. `upper-alpha` – заглавные буквы латинского алфавита A, B, C.

Свой маркер

`list-style-image` — свойство позволяет установить любую картинку в качестве символа маркера

```
.list {
  list-style-image: url("i/hand.png");
}
.list {
  list-style-image: url("http://somewebsite.ru/images/hand.png");
}
```



Управляем положением маркера

`list-style-position` — позволяет указать положение маркера списка. `outside` для положения снаружи (является значением по умолчанию), `inside` — внутри.

Live Demo

Обычный текст

- Список с значением list-style-position: outside
- Список с значением list-style-position: outside
- Список с значением list-style-position: outside

Обычный текст

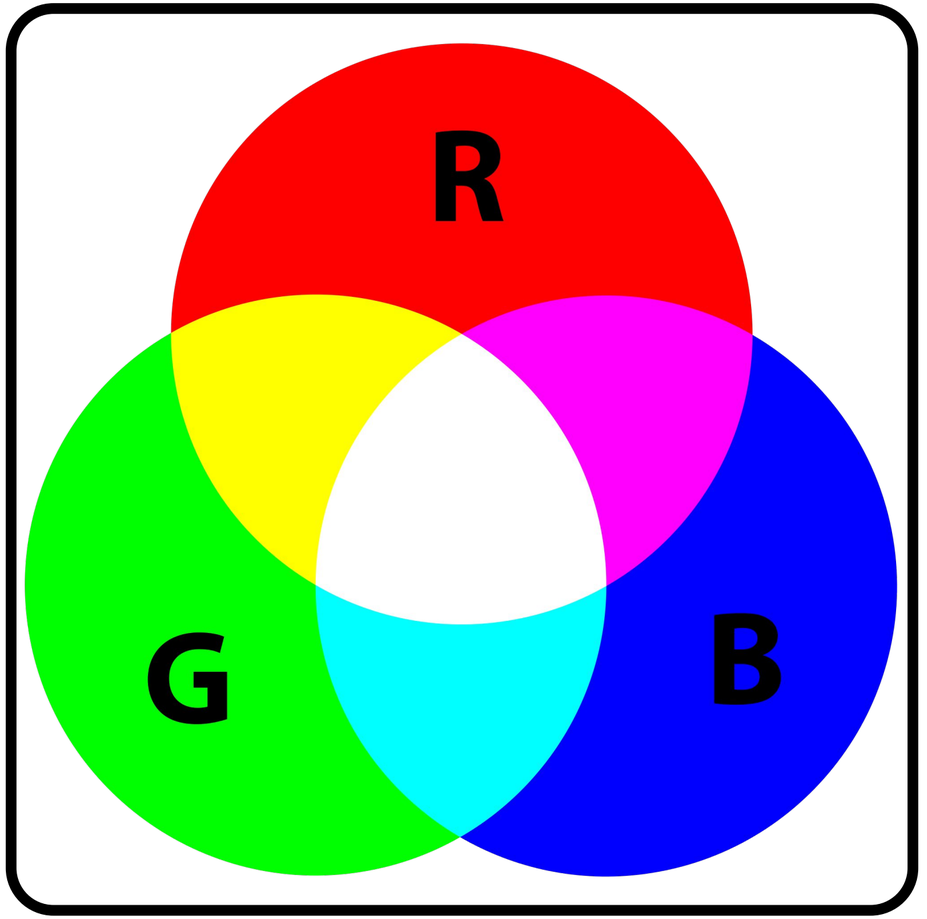
- Список с значением list-style-position: inside
- Список с значением list-style-position: inside
- Список с значением list-style-position: inside

Цвет в CSS

3

Цветовая модель RGB

Для браузеров используется RGB модель цветовоспроизведения, где R,G,B — первые буквы основных цветов(красный, зеленый, синий) — и выбор именно этих цветов обусловлен особенностями цветовосприятия человеческого глаза.
Согласно этой модели, любой цвет является результатом смешения основных цветов в разных пропорциях.

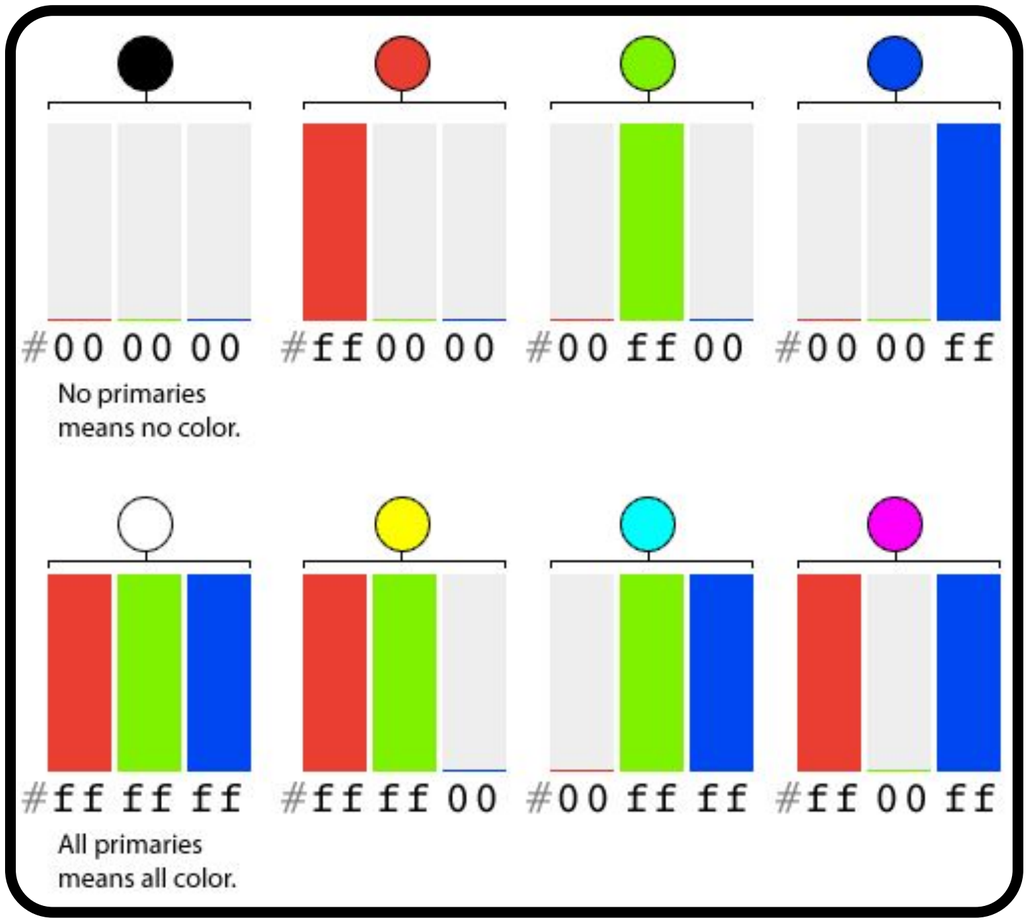


Hex-цвета

Цвет можно задать в виде 16-ричного кода.

В этом случае цвет представляется в виде 6-ти значного кода #RRGGBB, где RR означает интенсивность красного, GG — интенсивность зеленого, BB — интенсивность синего цвета, согласно модели RGB. Каждая из трех интенсивностей может принимать значения от 0 до 255, и записывается при помощи 2х значного 16-ричного кода.

Например: `color: #000000;` — черный цвет, `color: #ff00ff;` — цвет фуксии.



Автор, ссылка на источник

RGB формат

Также можно задать цвет в rgb-формате: воспользовавшись функцией rgb(r, g, b) где соответственно r, g и b — соответствующие интенсивности красного, зеленого и синего цветов.

Например: `color: rgb(0,0,0);` — черный цвет, `color: rgb(255, 0, 255);` — цвет фуксии.

RGBA формат

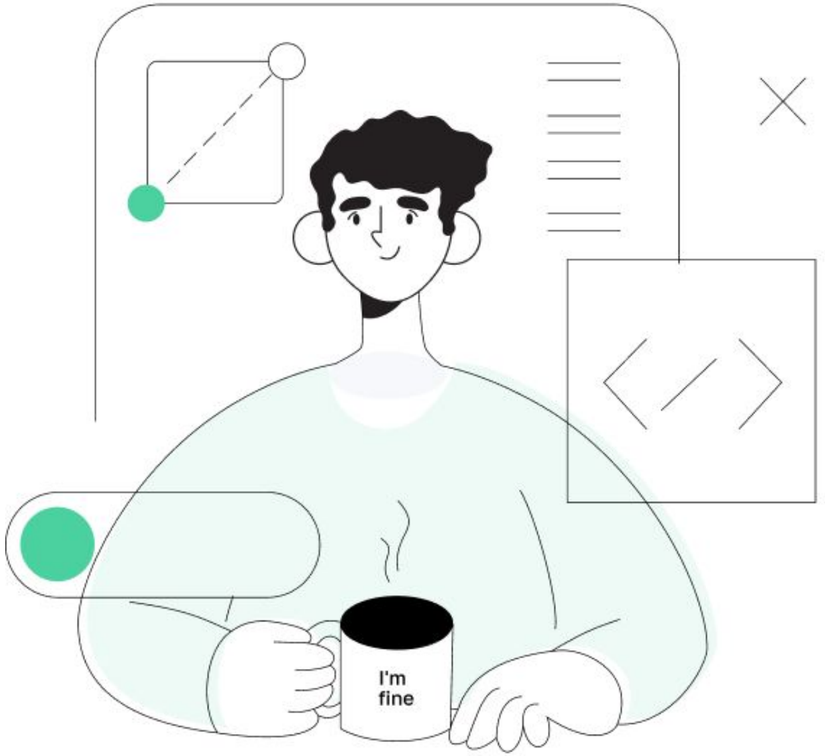
Но это еще не все: мы можем задать полупрозрачный цвет, воспользовавшись функцией rgba(r,g,b,a), где параметры r,g и b имеют точно такой же смысл, как и в функции rgb, а последний, четвертый параметр «a» (alpha) может принимать значения от 0 до 1 и означает степень непрозрачности цвета, где 0 — полная прозрачность, 1 — полная непрозрачность.

`color: rgba(0,0,0,0.5)` — черный цвет, прозрачный наполовину
`color: rgba(255,0,255,0.2)` — цвет фуксии с 20%-ной непрозрачностью.

Консультация по выбору направления

- Перейдите по ссылке в чате
- Ответьте на вопросы анкеты
- Получите консультацию специалиста по обучению и скидку 45% на курсы

[Анкета участника (ссылка)]



Свойства фоновых изображений

4

Свойства фоновых изображений

Используя все доступные нам цветовые форматы, мы можем менять не только цвет текста, но также и менять параметры фона. Например, мы можем задать синий фон у блока так:

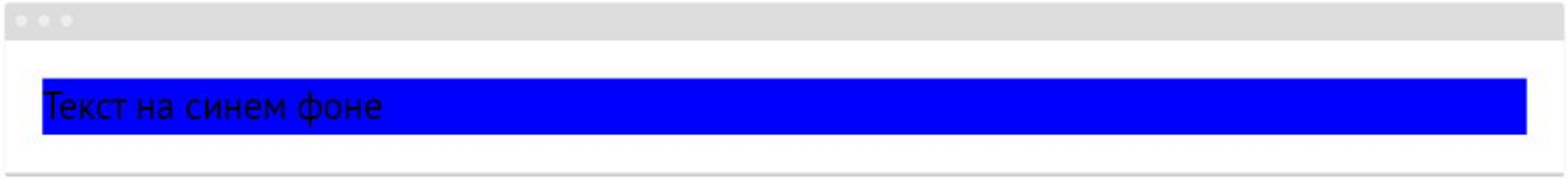
```
.block{
  background-color: blue;
}
```

или так:

```
.block{
  background-color: #0000ff;
}
```

Live Demo

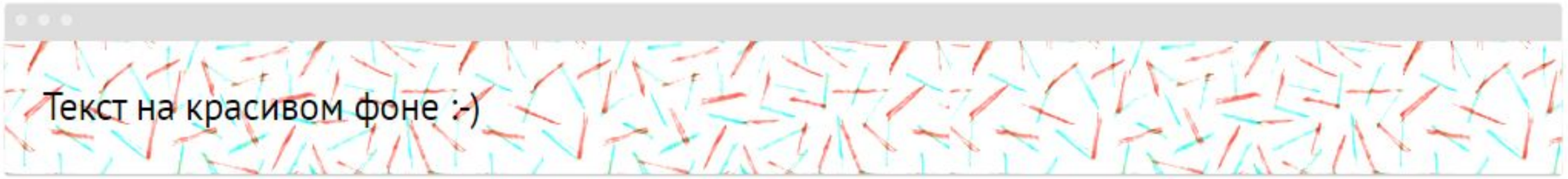
Цветной фон



Картинка на фоне

`background-image` — задает фоновое изображение. Нужно указать путь до изображения в формате: `url("путь/до/picture.jpg")`

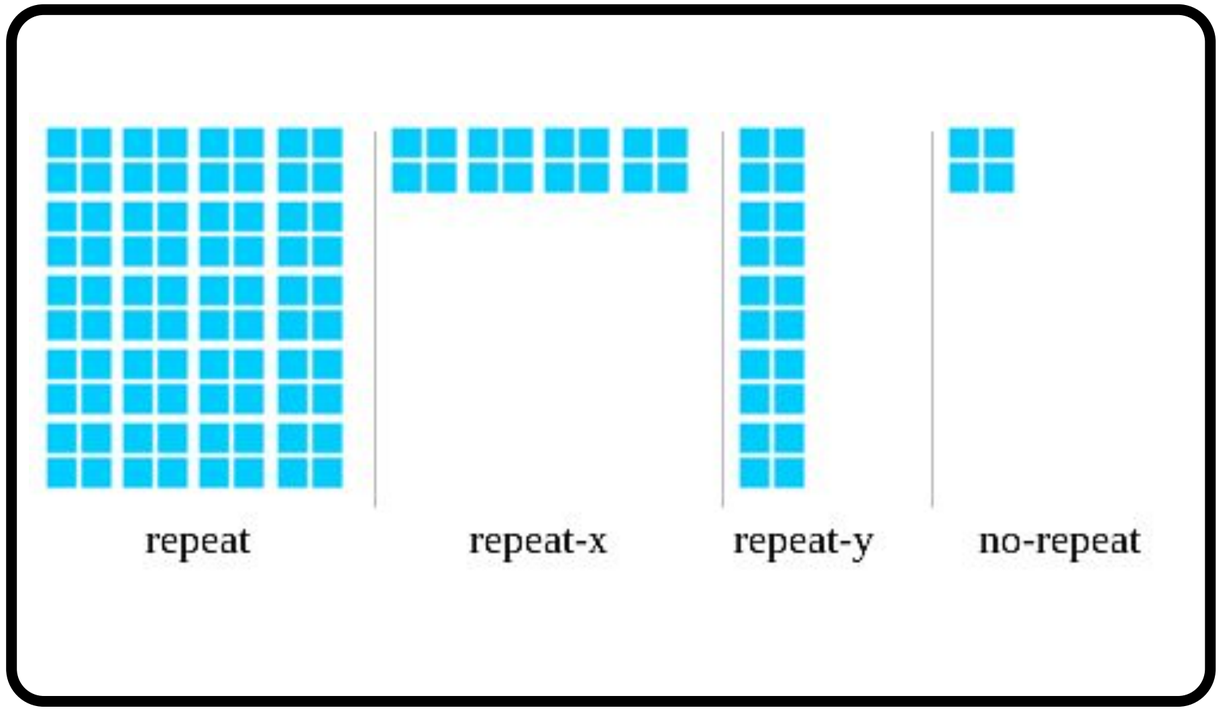
```
section {
  background-image: url("i/bg.jpg");
}
section{
  background-image: url("http://somewebsite.ru/images/bg.jpg");
}
```



Live Demo

Повторение фона

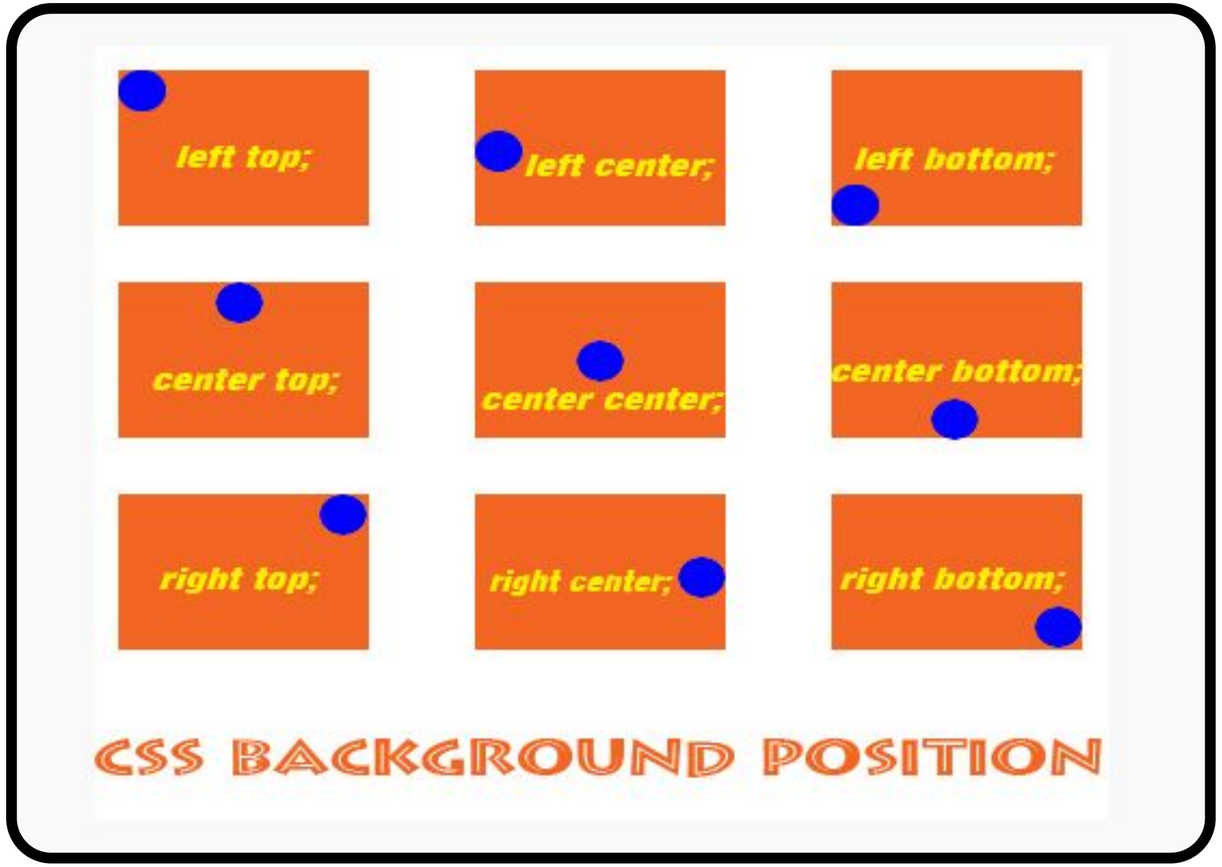
`background-repeat` — позволяет указать, хотим ли мы, чтобы заданное нами фоновое изображение повторялось.
Возможные значения: no-repeat — изображение повторяться не будет, отобразится только 1 раз: repeat-x — изображение будет повторяться только по оси x, repeat-y — изображение будет повторяться только по оси y, repeat (по умолчанию) — изображение будет повторяться по обеим осям — и x, и y.



Позиция фона

`background-position` — позволяет задать положение фонового изображения.

Можно указывать координаты относительно верхнего левого угла в px, можно указать ключевые слова (для каждой из 2х осей) — `top`, `bottom`, `left`, `right` или `center`.



Подведем итоги

Мы знаем:

1. Как работает селектор по классу;
2. Основные характеристики шрифта;
3. Преобразования текста;
4. Цвета в hex, rgb, rgba;
5. Свойства фоновых изображений.

Подведем итоги

Мы умеем:

1. Управлять свойствами шрифта;
2. Менять оформление текста;
3. Менять стили списка;
4. Управлять фоном.

Задавайте вопросы и напишите отзыв о лекции!

Владимир Языков
Основатель UsefulWeb

t.me/neizerth