

ОФОРМЛЕНИЕ ТЕКСТОВЫХ БЛОКОВ С ПОМОЩЬЮ CSS



АНТОН ВАРНАВСКИЙ / BRAINIT



АНТОН ВАРНАВСКИЙ

co-основатель и frontend developer в BrainIT

ПЛАН ЗАНЯТИЯ

1. Селектор класса

- Как записать селектор класса
- Комбинация тег + класс
- Решение задач

2. Оформление текстовых блоков

- Основные свойства для стилизации шрифта
- Трансформации шрифта
- Свойства списков

3. Цвет в CSS

- Цветовая модель RGB
- Задание цвета в hex-формате
- Задание цвета в rgb-формате
- Задание цвета в rgba-формате

4. Свойства фоновых изображений



СЕЛЕКТОР КЛАССА

РАЗНЫЕ СТИЛИ ДЛЯ ОДИНАКОВЫХ ТЕГОВ

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

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

ИСПОЛЬЗУЕМ ДОПОЛНИТЕЛЬНЫЕ ТЕГИ

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

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

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

```
1 .main-header {
2   color: red;
3   font-size: 50px;
4 }
5 .simple-header {
6   color: blue;
7 }
```

СЕЛЕКТОР КЛАССА

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

```
1  .имя-класса {  
2      свойство: значение;  
3  }
```


ПОЛНОЕ СОВПАДЕНИЕ

Важно: имя класса в CSS-стилях должно в точности совпадать с именем атрибута класса, записаного 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:

```
1  .menu .item {  
2      font-size: 10px;  
3  }  
4  .menu > .item {  
5      font-size: 20px;  
6  }
```

КОМБИНАЦИЯ ТЕГ + КЛАСС

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

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

```
1 h2.header {
2     color: green;
3 }
```



РЕЗУЛЬТАТ РАБОТЫ КОМБИНИРОВАННОГО СЕЛЕКТОРА



ЗАГОЛОВОК 1

ЗАГОЛОВОК 2
ЗАГОЛОВОК 3

РЕШЕНИЕ ЗАДАЧ

```
1 <h1 class="header">Заголовок</h1>
2 <p class="lead">Lorem ipsum...</p>
3 <p>Text</p>
4 <p>Text</p>
5 <h2 class="header">Заголовок списков</h2>
6 <ul class="big">
7   <li class="first-level">item 1</li>
8   <li class="first-level">item 2</li>
9   <li class="first-level">item 3
10     <ul>
11       <li>item 1</li>
12       <li>item 2</li>
13       <li>item 3</li>
14     </ul>
15   </li>
16 </ul>
17 <ol>
18   <li class="first-level">item 1</li>
19   <li class="first-level">item 2</li>
20   <li class="first-level">item 3</li>
21 </ol>
```

ПЕРЕКРАШИВАЕМ ПАРАГРАФ

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

```
1 | .lead {  
2 |     color: yellow;  
3 | }
```

УВЕЛИЧИВАЕМ ШРИФТ ЗАГОЛОВКА

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

```
1 | h1 {  
2 |     font-size: 40px;  
3 | }
```

КУРСИВНЫЕ ПУНКТЫ

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

```
1 | ol .first-level {  
2 |     font-style: italic;  
3 | }
```


МЕНЯЕМ ЖИРНОСТЬ ЗАГОЛОВКОВ

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

```
1 | .header {  
2 |     font-weight: 200;  
3 | }
```



ОФОРМЛЕНИЕ ТЕКСТОВЫХ БЛОКОВ С ПОМОЩЬЮ CSS

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;
- `text-align` – позволяет задавать выравнивание текста `left`, `right` – по левому и правому краю соответственно, `center` – по центру, `justify` – по ширине.

ЭКСПЕРИМЕНТ С СЕМЕЙСТВОМ


```
1  .sans-serif {  
2    font-family: sans-serif;  
3  }  
4  .serif {  
5    font-family: serif;  
6  }
```

Этот шрифт принадлежит к семейству sans-serif.

А этот — к семейству serif.

МЕЖСТРОЧНЫЙ ИНТЕРВАЛ

```
1 | .line-height-big {  
2 |     line-height: 50px;  
3 | }
```



Здесь величина междустрочного интервала увеличена до 50px, поэтому можно заметить, что строки находятся далеко друг от друга.

А здесь величина междустрочного интервала равна значению по умолчанию (примерно 1.3 величины шрифта).

ПЕРЕЧЕРКИВАЕМ ТЕКСТ

```
1  .underline {  
2    text-decoration: underline;  
3  }  
4  .overline {  
5    text-decoration: overline;  
6  }  
7  .line-through {  
8    text-decoration: line-through;  
9  }
```

МЕНЯЕМ РЕГИСТР

```
1  .uppercase {  
2    text-transform: uppercase;  
3  }  
4  .lowercase {  
5    text-transform: lowercase;  
6  }  
7  .capitalize {  
8    text-transform: capitalize;  
9  }
```


ВЫРАВНИВАЕМ ТЕКСТ

```
1  .align-right {  
2    text-align: right;  
3  }  
4  .align-center {  
5    text-align: center;  
6  }  
7  .align-justify {  
8    text-align: justify;  
9  }
```



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.
- III. `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

```
1  .list {  
2      list-style-image: url("i/hand.png");  
3  }  
4  .list {  
5      list-style-image: url("http://somewebsite.ru/images/hand.png");  
6  }
```

УПРАВЛЯЕМ ПОЛОЖЕНИЕМ МАРКЕРА

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

Обычный текст

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

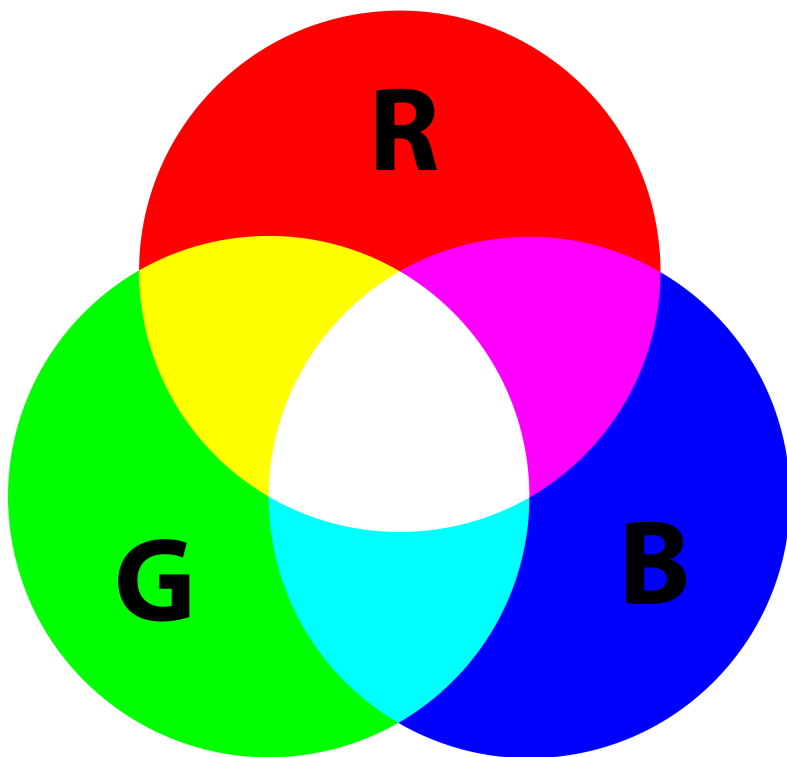
Обычный текст

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



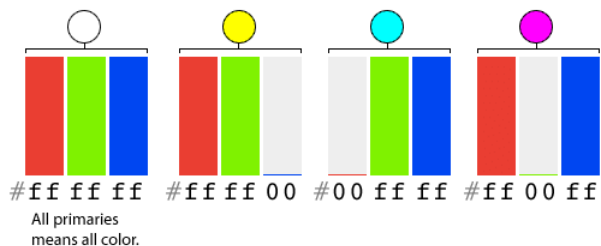
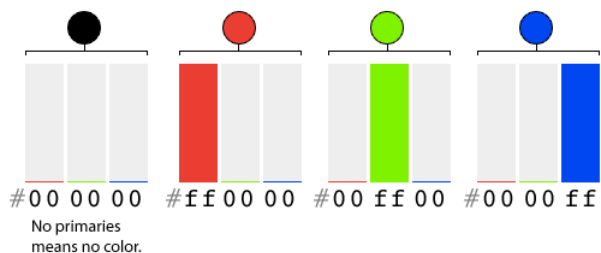
ЦВЕТ В CSS

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.

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.

СВОЙСТВА ФОНОВЫХ ИЗОБРАЖЕНИЙ

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:

```
1 | .block{  
2 |     background-color: blue;  
3 | }
```

или так:

```
1 | .block{  
2 |     background-color: #0000ff;  
3 | }
```

ЦВЕТНОЙ ФОН

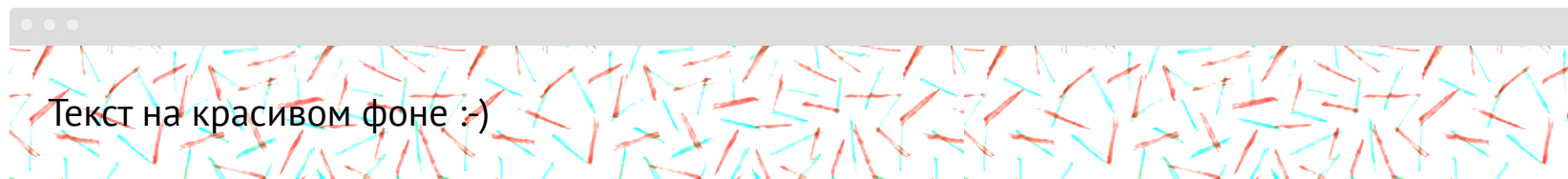


Текст на синем фоне

КАРТИНКА НА ФОНЕ

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

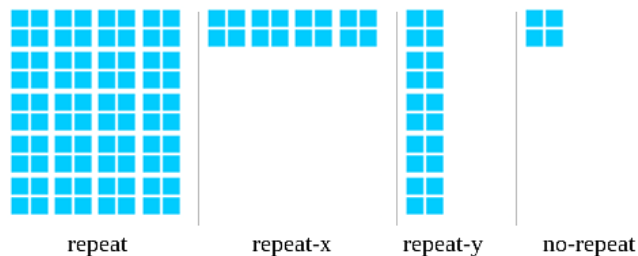
```
1  section {  
2      background-image: url("i/bg.jpg");  
3  }  
4  section{  
5      background-image: url("http://somewebsite.ru/images/bg.jpg");  
6  }
```



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.



нетология
университет интернет-профессий

Задавайте вопросы и напишите отзыв о лекции!

АНТОН ВАРНАВСКИЙ