



Frontender[1.0] Работа с изображениями. IMG, Picture, CSS Background

🔗 YouTube	https://youtu.be/SXszhgEmMW8 и https://youtu.be/qRvNGlAo_LE
🔗 Telegram	https://t.me/Dmitry_Kolotilshikov
# Номер урока	37

Tag IMG

Тег `` позволяет вставлять изображения на веб-страницу. Обычно этот тег используется с атрибутом `src`, который указывает на источник изображения. Также можно использовать атрибут `alt` для предоставления альтернативного текста, который будет отображаться вместо изображения, если оно не может быть загружено. Атрибут `width` и `height` позволяют указать ширину и высоту изображения (в пикселях), но лучше использовать **CSS** для стилизации.

Например:

```

```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body style="margin: 0">
  <!-- -->

  <div class="container" style="max-width: 600px; margin: 0 auto;">
    <img style="width: 100%" srcset="
      https://placeholder.co/500x200 500w,
      https://placeholder.co/800x300 700w,
      https://placeholder.co/1200x800 800w
    "
      sizes="(max-width: 600px) 100vw, 800px"
    >
  </div>

  <!--
  Атрибут (от англ. "sizes" - «размеры») указывает список допустимых относительных размеров источника.
```

```
100vw для экранов с шириной до 600px;  
800px для экранов с остальным разрешением.
```

С помощью `sizes` мы управляем браузером, указывая ему какую иллюстрацию подгружать в каком случае. Без атрибута `sizes` браузер может подтянуть картинки из `srcset` некорректно. При первичной калькуляции браузер не знает ширину изображения в верстке страницы. Чтобы понять размер картинки в CSS-пикселях, браузер и зучает значение атрибута `sizes`.

```
https://serpstat.com/ru/blog/kak-ispolzovat-atribut-srcset-tega-img/  
-->  
</body>  
</html>
```



Tag picture

Тег `<picture>` используется для определения нескольких источников изображений для одного элемента ``. Это позволяет браузеру выбирать изображение, которое наилучшим образом подходит для текущего разрешения экрана или размера элемента.

Например, мы можем использовать тег `<picture>` для загрузки разных изображений в зависимости от **разрешения экрана или размера элемента**. Также мы можем использовать этот тег для загрузки специальных версий изображений для устройств с высокой плотностью пикселей (например, Retina-дисплеи).

Синтаксис тега `<picture>` выглядит следующим образом:

```
<picture>  
  <source media="(min-width:650px)" srcset="img_pink_flowers.jpg">  
  <source media="(min-width:465px)" srcset="img_white_flower.jpg">  
    
</picture>
```

```
<picture>  
  <source srcset="image-1.webp" type="image/webp">  
  <source srcset="image-1.jpg" type="image/jpeg">  
    
</picture>
```

Здесь мы указываем два источника изображений - в формате **WebP** и **JPEG**. Браузер будет загружать изображение в том формате, который он поддерживает. Браузер будет использовать первый формат, который он распознает, и проигнорирует любой из следующих элементов. Если браузер не поддерживает ни один из форматов, он загрузит изображение, указанное в атрибуте `src` элемента ``.



CSS свойство background

CSS свойство background позволяет устанавливать фоновое изображение для элемента, а также задавать цвет фона и другие свойства фона (такие как повторение изображения, позиционирование и размер).

<https://css-tricks.com/almanac/properties/b/background/>

Синтаксис (схема):

```
background:  
  url(sweettexture.jpg) /* image */  
  top center / 200px 200px /* position / size */  
  no-repeat /* repeat */  
  fixed /* attachment */  
  padding-box /* origin */
```

```
content-box      /* clip */
red;             /* color */
```

- `background-image` задает изображение, которое будет использоваться в качестве фонового. Можно использовать URL-адрес, линейный градиент, радиальный градиент, SVG-файл и т.д.
- `background-position` определяет позицию изображения на фоне. Можно использовать ключевые слова (например, `left`, `top`, `center`) или указать расстояние от левого и верхнего края элемента в пикселях.
- `background-size` позволяет задавать размер фонового изображения.

```
/* Ключевые слова */
background-size: cover;
background-size: contain;

/* Указано одно значение - ширина изображения, */
/* высота в таком случае устанавливается в auto */
background-size: 50%;
background-size: 3em;
background-size: 12px;
background-size: auto;

/* Указаны два значения - */
/* ширина и высота соответственно */
background-size: 50% auto;
background-size: 3em 25%;
background-size: auto 6px;
background-size: auto auto;
```

- `background-repeat` определяет, будет ли изображение повторяться или нет. Возможные значения: `repeat`, `repeat-x`, `repeat-y` или `no-repeat`.
- `background-attachment` определяет, будет ли изображение закреплено к окну просмотра или к элементу. Возможные значения: `scroll` или `fixed`.
- `background-origin` устанавливает область расположения фона. Значения **border-box**, **padding-box**, **content-box**.
- `background-clip` определяет как цвет фона или фоновое изображение будут выводиться под границами блока.

```
/* Значения ключевых слов */
background-clip: border-box;
background-clip: padding-box;
background-clip: content-box;
background-clip: text;

/* Глобальные значения */
background-clip: inherit;
background-clip: initial;
background-clip: unset;
```

- `background-color` задает цвет фона элемента. Можно использовать ключевые слова (например, `transparent`, `inherit`) или указать цвет в формате RGB, HEX, HSL и других.

Пример:

```
<!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>CSS BACKGROUND</title>
  <style>
    div {
      border: 50px solid olive;
      padding: 3rem;
      width: 500px;
      height: 400px;
    }
  </style>
</head>
<body>
  <div></div>
</body>
</html>
```

```

        /* background: url("space.jpeg") center/cover no-repeat scroll border-box; */
        /* background: url("space.jpeg") center/cover no-repeat scroll border-box; */
        /* background: url("space.jpeg") center/cover no-repeat scroll padding-box; */
        background: url("space.jpeg") center/cover no-repeat scroll content-box;
    }

    p {
        margin: 0;
        color: rgb(102, 232, 117);
    }
</style>
</head>

<body>
    <div><p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam, eos?</p></div>
</body>

</html>

```



CSS: object fit

CSS-свойство `object-fit` определяет, как содержимое заменяемого элемента, такого как тег `` или `<video>`, будет масштабироваться для подгонки его размеров к контейнеру. Возможные значения для `object-fit` - это `fill`, `contain`, `cover`, `none` и `scale-down`.

`object-fit` - это очень полезное **CSS-свойство**, которое позволяет управлять тем, как изображение масштабируется для подгонки его размеров к контейнеру.

- `fill`: Это значение растягивает изображение на всю ширину контейнера, что может исказить соотношение сторон.
- `contain`: Это значение масштабирует изображение для подгонки его размеров к контейнеру с сохранением соотношения сторон.
- `cover`: Это значение масштабирует изображение **для покрытия всего** контейнера с сохранением соотношения сторон. Это может обрезать часть изображения.
- `none`: Это значение не изменяет размер изображения и может привести к его переполнению контейнера.
- `scale-down`: Контент изменяет размер, сравнивая разницу между `none` и `contain`, для того, чтобы найти наименьший конкретный размер объекта. Если изображение меньше контейнера, оно отображается в своем исходном размере.

```

<!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>CSS Object-fit</title>
    <style>
        img {
            border: 5px solid olive;
            width: 800px;
            height: 400px;
            object-fit: fill;
            /* object-fit: none; */
            /* object-fit: contain; */
            /* object-fit: scale-down; */
            /* object-fit: cover; */
        }

        p {
            margin: 0;
        }
    </style>
</head>

<body>
    <div>

```

```

  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Veniam, eos?</p>
</div>
</body>

</html>
```