



Конспект «Ссылки и изображения»

Ссылки

Что такое ссылка

Типичная ссылка представляет собой участок текста, щёлкая на который, вы переходите на другую страницу, открываете изображение, начинаете скачивать файл или перемещаетесь к какому-то месту на текущей странице.

Ссылки создаются с помощью тега `<a>`. Например:

```
<a href="https://htmlacademy.ru">HTML Academy</a>
```

Тег `<a>` можно использовать вообще без адреса, то есть без атрибута `href`. Такой тег обозначает «ссылку-заглушку», которая в других условиях может стать обычной ссылкой. Часто ссылки-заглушки используют, чтобы показать, что мы находимся на текущей странице:

```
<ul>
  <li><a>1 страница</a></li>
  <li><a href="2">2 страница</a></li>
  <li><a href="3">3 страница</a></li>
</ul>
```

Когда мы удаляем атрибут `href` у ссылки, то лучше оставить подсказку о том, почему мы это сделали. Подсказку можно добавить с помощью атрибута `title`. Подсказка появится, когда курсор задержится над ссылкой некоторое время.

- 1 страница
- [2 страница](#)
- [3 страница](#)

Абсолютные ссылки

Если нужно сделать ссылку на другой сайт в интернете, то необходимо использовать «обычный» адрес. Этот «обычный» или полный адрес называется абсолютным. Выглядит он, например, так:

`https://site.ru/blog/index.html`

Абсолютные адреса содержат минимум три части: протокол, имя сервера и путь.

Относительные ссылки

Когда файл по ссылке должен открываться локально на компьютере, используются относительные адреса. В отличие от «обычных» адресов, в нём нет адреса сайта. Например:

`day-2.html`

Относительные адреса работают не только для файлов на компьютере, но и для страниц в сети. Если выложить два каких-то файла в интернет (не меняя их взаимное расположение), то их ссылка друг на друга всё равно будет работать.

Если в адресе нет имени сервера или протокола, то это относительный адрес.

Ссылки на файл

По ссылкам можно не только переходить, но и скачивать файлы. Для этого необходимо просто в атрибуте `href` прописать ссылку на этот файл. А для того чтобы предотвратить открытие файлов прямо в браузере, у тега `<a>` существует атрибут `download`.

```
<a href="file.pdf" download>Браузер скачает меня, а не будет читать</a>
```

Ссылки-якоря

Ссылка-якорь — это обычная ссылка, в адресе которой используется символ `#`, после которого следует идентификатор элемента. Идентификатор создаётся с помощью атрибута `id` у того тега, к которому надо перейти при щелчке по ссылке.

```
<a href="#part1">Глава 1</a>
```

Ссылка-якорь используется для прокрутки к заданной части страницы, в том числе используется и в абсолютных адресах.

Изображения

Подключение изображений

Для подключения изображений существует тег ``, для него не требуется закрывающего тега, а путь к картинке задаётся в атрибуте `src`. Например:

```

```

Форматы изображений

Существует несколько основных форматов изображений: `JPEG`, `PNG`, `SVG` и `GIF`.

Формат **SVG** переводится как масштабируемая векторная графика. Качество таких изображений не меняется при изменении размеров, да и вес у них небольшой. Отлично подходит для малоцветных схем, логотипов и иконок. Чаще всего используется в случаях, когда необходимо масштабировать изображение без потерь, изменять цвет элементов изображения, анимировать части изображения.

Формат **JPEG** подходит для фотографий, рисунков с большим количеством разноцветных деталей, изображений с плавным переходом яркости и контраста. При сжатии изображения ухудшается его качество.

Формат **PNG** позволяет сохранять изображения, в которых требуется особенная чёткость. Главная особенность этого формата — поддержка прозрачности. Подходит для изображений с прозрачностью и полупрозрачностью, когда необходима повышенная точность полноцветных изображений и для изображений с резкими переходами цветов.

Формат **GIF** используется для простейших анимаций. В последнее время GIF-изображения становятся всё менее используемыми и заменяются на другие, более оптимальные форматы.

Размеры изображения

Для управления шириной или высотой изображения, используются атрибуты **width** и **height**. Размеры в этих атрибутах задаются без единиц измерения.

```

```

Если задать только один из размеров, ширину или высоту, то вторую размерность браузер вычислит самостоятельно исходя из пропорций изображения. Если задать картинке одновременно и высоту, и ширину, то браузер может нарушить пропорции исходного изображения.

Атрибут alt

В случаях, когда картинка не может отобразиться, для того, чтобы было понятно, что на ней изображено, существует альтернативный текст. Также альтернативный текст помогает сайтам оставаться доступными, например, для категории пользователей, которая не имеет возможности видеть картинки.

Альтернативный текст изображения задаётся с помощью атрибута `alt`. Например:

```

```

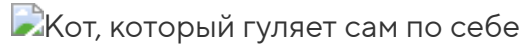


Figure и figcaption

Тег `<figure>` подходит для любого иллюстративного или демонстрационного материала, которым можно дополнить содержание документа: схемы, графики, примеры кода, таблицы и так далее. При удалении такого материала основное содержание не должно пострадать, иначе это не дополнительный материал и тег `<figure>` не подходит для его разметки.

Для разъясняющего комментария к такому иллюстративному материалу существует тег `<figcaption>`, который размещается первым или последним элементом внутри `<figure>`. Например:

```
<figure>
  
  <figcaption>Подпись к содержимому</figcaption>
</figure>
```



Подпись к содержимому

Изображение-ссылка

Ссылки можно делать не только с помощью текста, но и с помощью изображений. Для этого нужно обернуть тег `` в тег `<a>`. Например:

```
<a href="http://keksby.ru">  
    
</a>
```

Продолжить

Практикум

Тренажёры

Подписка

Курсы

HTML и CSS. Профессиональная вёрстка сайтов

HTML и CSS. Адаптивная вёрстка и автоматизация



Для команд и компаний
Учебник по PHP

Профессии

Фронтенд-разработчик
React-разработчик
Фулстек-разработчик
Бэкенд-разработчик

Услуги

Работа наставником
Для учителей
Стать автором

Блог

С чего начать
Шпаргалки для разработчиков
Отчеты о курсах

JavaScript. Профессиональная разработка веб-интерфейсов
JavaScript. Архитектура клиентских приложений
React. Разработка сложных клиентских приложений
PHP. Профессиональная веб-разработка
PHP и Yii. Архитектура сложных веб-сервисов
Node.js. Разработка серверов приложений и API
Анимация для фронтендеров
Вёрстка email-рассылок
Vue.js для опытных разработчиков
Регулярные выражения для фронтендеров
Шаблонизаторы HTML
Алгоритмы и структуры данных
Анатомия CSS-каскада

Информация

Об Академии
О центре карьеры

Остальное

Написать нам
Мероприятия
Форум

[Соглашение](#) [Конфиденциальность](#) [Сведения об образовательной организации](#) [Лицензия № 3026](#)