

## Лекция 2. Гиперссылки и навигация

### Оглавление

Лекция 2. Гиперссылки и навигация.....	1
2.1. Что такое ссылка.....	1
2.2. Относительные ссылки .....	2
2.3. Абсолютные ссылки .....	3
2.4. Ссылки на файл.....	3
2.5. Внутренние ссылки (якоря) .....	3
2.6. Атрибуты target и title .....	4
2.7. Ссылки на адрес электронной почты .....	4
2.8. Ссылка-картинка.....	4
Задание к лабораторной работе 2:.....	5

Ссылки являются основой гипертекстовых документов и позволяют переходить с одной web-страницы (*или закладки в документе*) на другую, а также выполнять действия, связанные с некоторыми прикладными протоколами Web (например, вызывать почтовый клиент по умолчанию, менеджер закачек и т.п.) Ссылка может вести не только на HTML-файл, но и на файл любого другого типа, причем этот файл может размещаться на другом сайте. Как конкретно будет обрабатываться тот или иной тип файлов, зависит от настроек операционной системы и браузера.

### 2.1. 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>`. Полный список атрибутов тега ссылки можно посмотреть здесь: <http://htmlbook.ru/html/a>

Основной атрибут - **href**, он определяет URL для перехода по ссылке. Текст, расположенный между тегами `<a>` и `</a>`, по умолчанию становится синего цвета и подчеркивается.

Например:

```
<a href="http://htmlbook.ru">HTML Book</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. Пример:

```
<a href="http://sibstrin.ru" title="НГАГУ">университет</a>
```

## 2.2. Относительные ссылки

Над сайтом мы в основном работаем на своём компьютере, то есть *локально*. Ссылка на другую страницу сайта должна открывать файл с нашего компьютера, а не откуда-то из интернета. Для таких ссылок нужно использовать относительный путь к файлу, например:

```
<a href="inner.html">Введение</a>
```

В относительном пути **нет** адреса сайта! Чтобы перейти по относительному адресу, браузер должен его "расшифровать". Для этого он обычно использует положение текущей страницы. Например, в папке c:/blog есть два файла:

```
c:/blog/
|-index.html // в браузере открыта эта страница
|-inner.html
```

В браузере открыта страница c:/blog/index.html, и в ней есть ссылка с относительным адресом inner.html. Чтобы перейти по этой ссылке, браузер смотрит на расположение открытой страницы и заменяет в нём последнюю часть:

```
c:/blog/index.html + inner.html // заменяем последнюю часть
c:/blog/inner.html // открываем этот файл
```

Относительные адреса работают не только для файлов на компьютере, но и для страниц в сети. Если выложить два файла из примера в интернет (не меняя их взаимное расположение), то ссылка всё равно будет работать. Относительный адрес inner.html на странице https://site.ru/blog/index.html расшифруется так:

```
https://site.ru/blog/index.html + inner.html
// заменяем последнюю часть
https://site.ru/blog/inner.html
// открываем этот адрес
```

Обратите внимание, что в относительных адресах также можно использовать имена вложенных папок, ссылку на родительскую папку . . (два символа точки), ссылку на текущую папку . (один символ точки) и т.п., в общем, все виды путей к файлу из курса Информатики. При составлении путей к файлам нужно использовать прямой слэш "/", а не обратный "\", как в папках Windows.

## 2.3. Абсолютные ссылки

Относительные адреса отлично подходят, если вы делаете сайт на своём компьютере или создаёте навигацию по страницам *своего* же сайта. Но если нужно сделать ссылку на *другой* сайт в интернете, то необходимо использовать уже "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.

https: — протокол

//site.ru — имя сервера

/blog/index.html — путь от корневой папки сервера к файлу

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

https://site.ru/blog/index.html — абсолютный адрес

/blog/index.html — относительный адрес

index.html — относительный адрес

## 2.4. Ссылки на файл

По ссылкам можно не только переходить, но и скачивать файлы. Для этого необходимо 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>
```

Если существует стандартная обработка для типа файла, адресуемого ссылкой, браузер может игнорировать указание этого атрибута.

## 2.5. Внутренние ссылки (якоря)

Ссылка-якорь — это обычная ссылка, в адресе которой используется символ #, после которого следует идентификатор элемента. Идентификатор создаётся с помощью тега <a name> или 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.

**Пример.** Создадим в конце web-страницы ссылку, которая введет в начало документа. Для этого в начале документа, задаем метку:

```
<body>
```

```
<a name="ttt"></a>
```

Затем в конце страницы задаем ссылку:

```
<a href="#ttt">В начало</a>
```

```
</body>
```

Тем не менее, этот классический способ устаревает, лучше использовать атрибут `id` элемента:

```
<a id="ttt"></a>
```

Вот так выглядит адрес, состоящий из одного якоря:

```
<a href="#part1">Глава 1</a>
```

При щелчке по такой ссылке браузер найдёт на странице элемент с соответствующим атрибутом `id` и прокрутит окно страницы к нему.

```
<section id="part1">Содержание первой главы</section>
```

Также можно сделать ссылку на метку другого документа. Пример:

```
<a href="text.html#ttt">Как написано на сайте...</a>
```

## 2.6. Атрибуты *target* и *title*

С тегом `<a>` можно использовать атрибуты `target` и `title`.

Атрибут `target` указывает окно, в котором нужно открыть ссылку. По умолчанию, при переходе по ссылке документ открывается в текущем окне. Синтаксис следующий:

```
<a target="имя окна">...</a>
```

В качестве значения используется имя окна или фрейма, заданное атрибутом `name`. Если установлено несуществующее имя, то будет открыто новое окно. В качестве зарезервированных имен применяются следующие:

- `_blank` — загружает страницу по ссылке в новое окно (вкладку) браузера.
- `_self` — загружает страницу по ссылке в текущее окно.
- `_parent` — загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как `_self` (*устарело*).
- `_top` — отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как `_self` (*устарело*).

## 2.7. Ссылки на адрес электронной почты

Для создания **ссылки на адрес электронной почты** в атрибуте `href` надо указать `mailto:адрес электронной почты`. Например:

```
<a href="mailto:admin@site.ru">администратор</a>
```

## 2.8. Ссылка-картинка

Поместив внутрь элемента `<a>` элемент `<img>`, можно сделать ссылку-изображение:

```
<a href="index.html">
```

```
  
```

```
</a>
```

## Задание к лабораторной работе 2:

Создать шаблон страницы сайта, включающий его логические части и подготовить несколько страниц, связанных ссылками, например:

- Главная
- О себе
- Мои работы (для последующей публикации заданий курса)
- Файлы (с возможностью скачать какие-либо файлы на странице)
- Контакты (контактная информация, включающая адрес E-mail)
- Ссылки (страница с внешними ссылками)

В коде использовать разметку HTML5 и относительные ссылки, ссылки-якоря, ссылки на адрес E-mail.

Проверить работу сайта и его валидность.