

Лабораторная работа 3. Создание ссылок

Цель работы: Освоение приемов создания ссылок при помощи гипертекстовой разметки.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. Оба действия выполняются с помощью тега `<a>`. Общий синтаксис создания ссылок следующий.

```
<a href="URL">текст ссылки</a>
```

Атрибут `href` определяет URL (Universal Resource Locator, универсальный указатель ресурса), иными словами, адрес документа, на который следует перейти, а содержимое контейнера `<a>` является ссылкой. Текст, расположенный между тегами `<a>` и ``, по умолчанию становится синего цвета и подчеркивается. В примере 8.1 показано создание нескольких ссылок на разные веб-страницы.

Пример 1. Добавление ссылок

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>

  <head>

<meta http-equiv="content-type" content="text/html; charset=utf-8">

    <title>Ссылки на странице</title>

  </head>

  <body>

    <p><a href="dog.html">Собаки</a></p>

    <p><a href="cat.html">Кошки</a></p>

  </body>

</html>
```

В данном примере создаются две ссылки с разными текстами. При щелчке по тексту «Собаки» в окне браузера откроется документ `dog.html`, а при щелчке на «Кошки» — файл `cat.html`.

Результат примера показан на рис. 8.1. Обратите внимание, что при наведении курсора мыши на ссылку, в строке состояния браузера отображается полный путь к ссылаемому файлу.



Если указана ссылка на файл, которого не существует, например, его имя в атрибуте *href* набрано с ошибкой, то такая ссылка называется битая. Битых ссылок следует категорически избегать, поскольку они вводят посетителей сайта в заблуждение. Так, при щелчке по ссылке из примера 1 в браузере Safari откроется не сам документ, а окно с предупреждением (рис. 2).

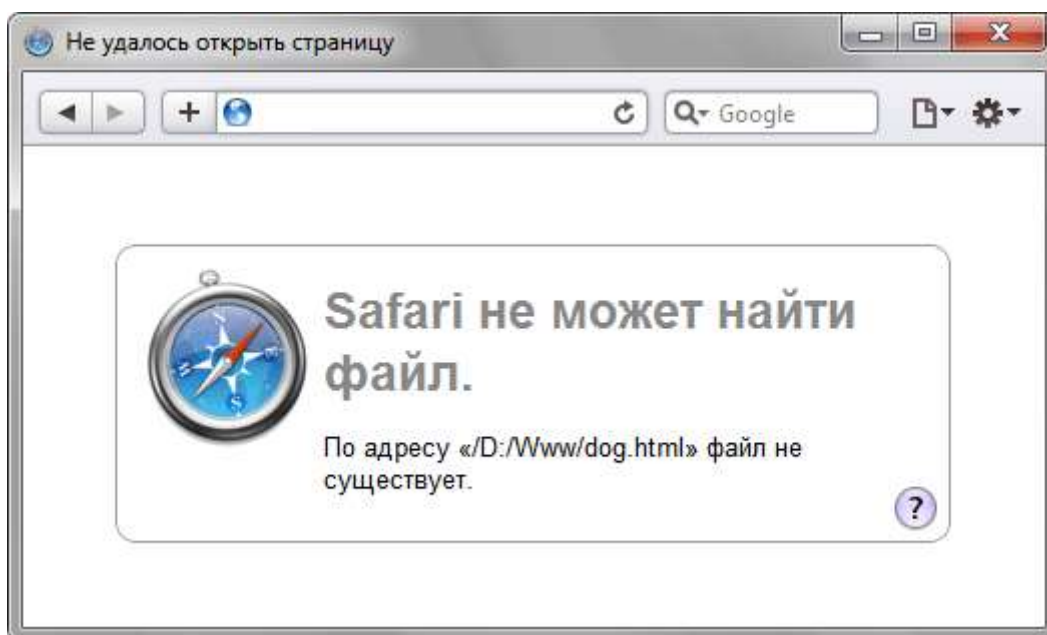
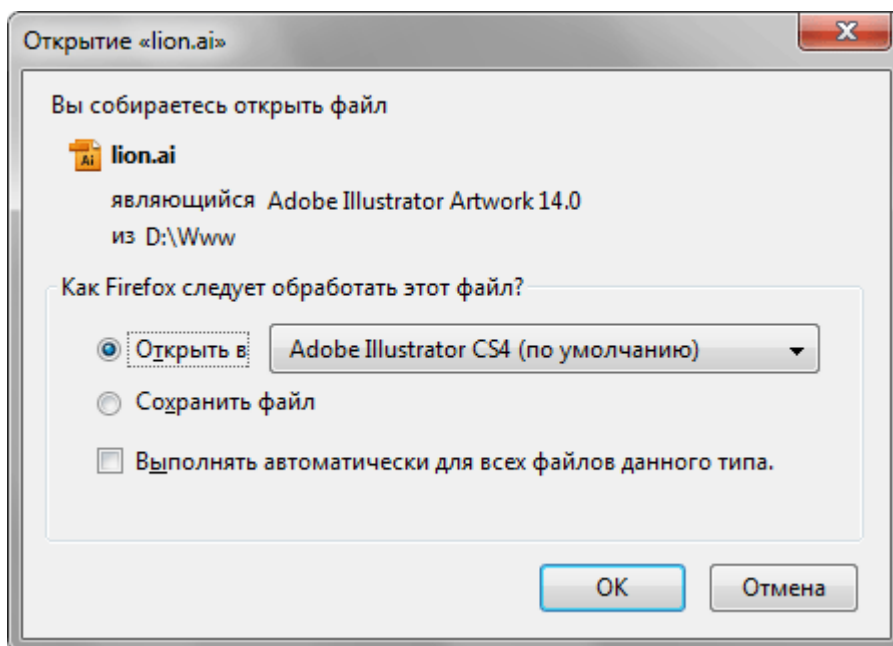


Рис. 2. Результат при открытии битой ссылки

Естественно, подобное сообщение будет различаться в браузерах, но смысл остается один — документ, на который ведет ссылка, не может быть открыт. Чтобы не возникало подобных ошибок, тестируйте все ссылки на их работоспособность и сразу же устраняйте имеющиеся погрешности.

Файл по ссылке открывается в окне браузера только в тех случаях, когда браузер знает тип документа. Но поскольку ссылку можно сделать на файл любого типа, то браузер не всегда может отобразить документ. При этом выводится сообщение, как следует

обработать файл — открыть его или сохранить в указанную папку. Например, в браузере Firefox выводится следующее окно (рис. 3).



Адрес ссылки может быть как абсолютным, так и относительным.

Абсолютные адреса должны начинаться с указания протокола (обычно http://) и содержать имя сайта.

Относительные ссылки ведут отсчет от корня сайта или текущего документа.

В примере 2 показано создание абсолютной ссылки на другой сайт.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>

<head>

<meta http-equiv="content-type" content="text/html; charset=utf-8">

<title>Абсолютный адрес</title>

</head>

<body>

<p><a href="http://htmlbook.ru">Изучение HTML</a></p>

</body>

</html>
```

В данном примере ссылка вида `Изучение HTML` является абсолютной и ведет на главную страницу сайта htmlbook.ru. При указании в качестве ссылки каталога сайта (например, `http://htmlbook.ru/css/`),

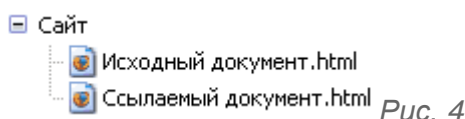
отображается *индексный файл*. Это файл, который загружается по умолчанию при обращении к каталогу без явного указания имени файла. Обычно в качестве индексного файла выступает документ с именем *index.html*. Абсолютные ссылки обычно применяются для указания документа на другом сетевом ресурсе, впрочем, допустимо делать абсолютные ссылки и внутри текущего сайта. Однако подобное практикуется нечасто, поскольку такие ссылки достаточно длинные и громоздкие. Поэтому внутри сайта преимущественно используются относительные ссылки.

Ссылки относительно текущего документа

При создании относительных ссылок надо понимать, какое значение для атрибута **href** следует указывать, поскольку оно зависит от исходного расположения файлов.

Рассмотрим несколько типичных вариантов.

1. Файлы располагаются в одной папке (рис. 4).

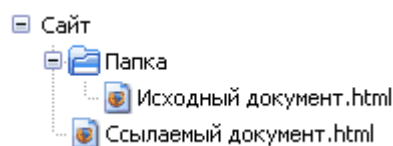


Необходимо сделать ссылку из исходного документа на ссылаемый. В таком случае код будет следующий.

```
<a href="Ссылаемый документ.html">Ссылка</a>
```

Подобное имя файла взято только для образца, на сайте в именах файлов не следует использовать русские символы с пробелами, да еще и в разном регистре.

2. Файлы размещаются в разных папках (рис. 5).

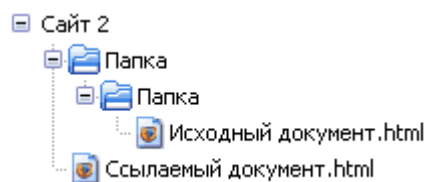


Когда исходный документ хранится в одной папке, а ссылаемый в корне сайта, то перед именем файла в адресе ссылки следует поставить две точки и слэш (/), как показано ниже.

```
<a href="../Ссылаемый документ.html">Ссылка</a>
```

Две точки в данном случае означают выйти из текущей папки на уровень выше.

3. Файлы размещаются в разных папках (рис. 6).



Теперь исходный файл находится в двух вложенных папках, и чтобы сослаться на документ в корне сайта, требуется повторить написание предыдущего примера два раза.

```
<a href="../../Ссылаемый документ.html">Ссылка</a>
```

Аналогично обстоит дело с любым числом вложенных папок.

4. Файлы размещаются в разных папках (рис. 7).

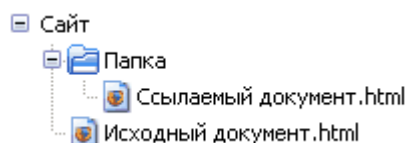


Рис. 7

Теперь ситуация меняется, исходный файл располагается в корне сайта, а файл, на который необходимо сделать ссылку - в папке. В этом случае путь к файлу будет следующий.

```
<a href="Папка/Ссылаемый документ.html">Ссылка</a>
```

Заметьте, что никаких дополнительных точек и слэшей перед именем папки нет. Если файл находится внутри не одной, а двух папок, то путь к нему записывается так.

```
<a href="Папка 1/Папка 2/Ссылаемый документ.html">Ссылка</a>
```

Ссылки относительно корня сайта

Иногда можно встретить путь к файлу относительно корня сайта, он выглядит как **"/Папка/Имя файла"** со слэшем вначале. Так, запись `Курсы` означает, что ссылка ведет в папку с именем course, которая располагается в корне сайта, а в ней необходимо загрузить индексный файл.

Любая ссылка на веб-странице может находиться в одном из следующих состояний.

Непосещенная ссылка. Такое состояние характеризуется для ссылок, которые еще не открывали. По умолчанию непосещенные текстовые ссылки изображаются синего цвета и с подчеркиванием.

Активная ссылка. Ссылка помечается как активная в момент ее открытия. Поскольку время между нажатием на ссылку и началом загрузки нового документа достаточно мало, подобное состояние ссылки весьма кратковременно. Активной ссылка становится также, при ее выделении с помощью клавиатуры. Цвет такой ссылки по умолчанию красный.

Посещенная ссылка. Как только пользователь открывает документ, на который указывает ссылка, она помечается как посещенная и меняет свой цвет на фиолетовый, установленный по умолчанию.

Любая ссылка является встроенным элементом, поэтому для нее действуют те же правила, что и для встроенных элементов. А именно, нельзя размещать внутри тега `<a>` блочные элементы, но допустимо делать наоборот, и вкладывать ссылку в блочный контейнер. В примере 3 показано ошибочное и правильное использование тегов.

Пример 3. Вложение тегов

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>

  <head>

    <meta http-equiv="content-type" content="text/html; charset=utf-8">

    <title>Ошибки при использовании ссылок</title>

  </head>

  <body>

    <a href="lion.html"><h1>Охота на льва</h1></a>

    <h1><a href="lion.html">Как поймать льва в пустыне</a></h1>

  </body>

</html>
```

В строке 8 данного примера содержится типичная ошибка — тег `<h1>` располагается внутри контейнера `<a>`. Поскольку `<h1>` это блочный элемент, то его недопустимо вкладывать внутрь ссылки. В строке 9 этого же примера показан корректный вариант.

Атрибуты ссылок

Основной атрибут `href` тега `<a>` мы уже освоили, рассмотрим еще несколько полезных, но необязательных атрибутов этого тега.

target

По умолчанию, при переходе по ссылке документ открывается в текущем окне или фрейме. При необходимости, это условие может быть изменено атрибутом `target` тега `<a>`. Синтаксис следующий.

```
<a target="имя окна">...</a><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 в новое окно браузера.
- `_self` — загружает страницу в текущее окно (это значение задается по умолчанию).
- `_parent` — загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как `_self`.
- `_top` — отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как `_self`.

В примере 8.4 показано, как сделать, чтобы ссылка открывалась в новом окне.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

  <meta http-equiv="content-type" content="text/html; charset=utf-8">

  <title>Ссылка в новом окне</title>

</head>

<body>

  <p><a href="new.html" target="_blank">Открыть

      в новом окне</a></p>

</body>

</html>
```

Атрибут `target` корректно использовать только при переходном `<!DOCTYPE>`, при строгом `<!DOCTYPE>` будет сообщение об ошибке, поскольку в этой версии HTML `target` уже не поддерживается.

Учтите также, что пользователи не любят, когда ссылки открываются в новых окнах, поэтому используйте подобную возможность осмотрительно и при крайней необходимости.

title

Добавляет поясняющий текст к ссылке в виде всплывающей подсказки. Такая подсказка отображается, когда курсор мыши задерживается на ссылке, после чего подсказка через некоторое время пропадает. Синтаксис следующий.

```
<a title="текст">...</a><a title="текст">...</a>
```

В качестве значения указывается любая текстовая строка. Строка должна заключаться в двойные или одинарные кавычки. В примере 5 показано, как использовать атрибут **title** для ссылок.

Пример 5. Создание всплывающей подсказки

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>

<head>

  <meta http-equiv="content-type" content="text/html; charset=utf-8">

  <title>Подсказка к ссылке</title>

</head>

<body>

  <p><a href="zoo.html" title="Рисунки различных животных и не
только...">Рисунки</a></p>

</body>

</html>
```

Результат данного примера показан на рис. 8.

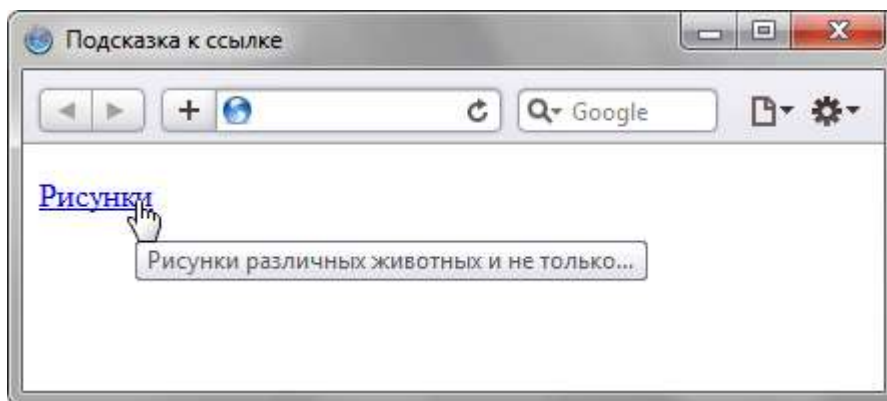


Рис. 8. Вид всплывающей подсказки в браузере

Цвета и оформления всплывающей подсказки зависят от настроек операционной системы и браузера, и меняться разработчиком не могут.

Создание ссылки на адрес электронной почты делается почти также как и ссылка на веб-страницу. Только вместо URL указывается **mailto:адрес электронной почты** (пример 6).

Пример 6. Ссылка на адрес электронной почты


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>

<head>

  <meta http-equiv="content-type" content="text/html; charset=utf-8">

  <title>Адрес почты</title>

</head>

<body>

  <p><a href="mailto:vlad@htmlbook.ru">Задавайте вопросы по электронной
почте</a></p>

</body>

</html>
```

В атрибуте **href** тега **<a>** вначале пишется ключевое слово **mailto**, затем через двоеточие желаемый почтовый адрес. Подобная ссылка ничем не отличается от ссылки на веб-страницу, но при нажатии на нее запускается почтовая программа, установленная по умолчанию. Поэтому в названии ссылки следует указывать, что она имеет отношение к электронной почте, чтобы читатели понимали, к чему приведет нажатие на нее.

Можно также автоматически добавить тему сообщения, присоединив к адресу электронной почты через символ вопроса (?) параметр **subject=тема сообщения**, как показано в примере 7.

Пример 7. Задание темы сообщения

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>

<head>

  <meta http-equiv="content-type" content="text/html; charset=utf-8">

  <title>Тема письма</title>

</head>

<body>

  <p><a href="mailto:vlad@htmlbook.ru?subject=Вопрос по HTML">Задавайте
  вопросы по электронной почте</a></p>

</body>
```

```
</html>
```

При запуске почтовой программы поле Тема (Subject) будет заполнено автоматически. Якорем называется закладка с уникальным именем на определенном месте веб-страницы, предназначенная для создания перехода к ней по ссылке. Якоря удобно применять в документах большого объема, чтобы можно было быстро переходить к нужному разделу.

Для создания якоря следует вначале сделать закладку в соответствующем месте и дать ей имя при помощи атрибута **name** тега `<a>` (пример 1). В качестве значения **href** для перехода к этому якорю используется имя закладки с символом решетки (#) впереди.

Пример 1. Создание якоря

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>

<head>

  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

  <title>Быстрый переход внутри документа</title>

</head>

<body>

  <p><a name="top"></a></p>

  <p>...</p>

  <p><a href="#top">Наверх</a></p>

</body>

</html>
```

Между тегами `` и `` текст не обязателен, так как требуется лишь указать местоположение перехода по ссылке, находящейся внизу страницы. Имя ссылки на якорь начинается с символа #, после чего идет имя якоря, оно выбирается любое, соответствующее тематике. Главное, чтобы значения атрибутов **name** и **href** совпадали (символ решетки не в счет).

С якорями связана одна особенность работы браузера. После перехода к указанному якорю нажатие на кнопку «Назад» возвращает не на предыдущую просмотренную страницу, а к ссылке, с которой был сделан переход к якорю. Получается, что для перехода к предыдущему документу надо нажать кнопку «Назад» два раза. Ссылку можно

также сделать на закладку, находящуюся в другой веб-странице и даже другом сайте. Для этого в атрибуте **href** тега `<a>` надо указать адрес документа и в конце добавить символ решетки `#` и имя закладки (пример 2).

Пример 2. Ссылка на закладку из другой веб-страницы

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>

  <head>

    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">

    <title>Якорь в другом документе</title>

  </head>

  <body>

    <p><a href="text.html#bottom">Перейти к нижней части текста</a></p>

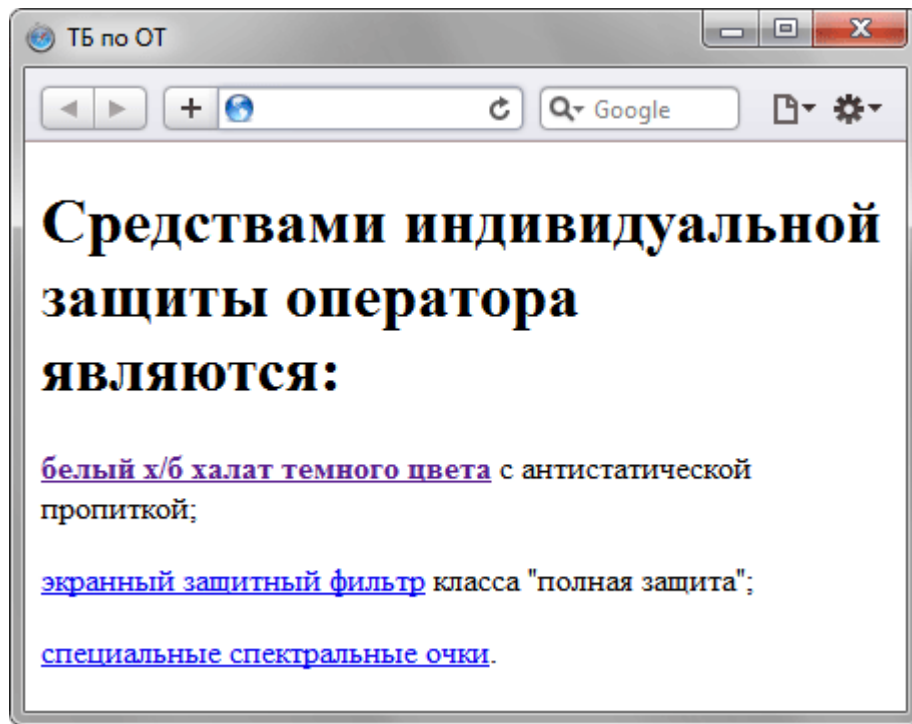
  </body>

</html>
```

В данном примере показано создание ссылки на файл `text.html`, при открытии этого файла происходит переход на закладку с именем `bottom`.

Если на веб-странице содержится ссылка на якорь, а самого якоря нет, то никакой ошибки не возникнет

Задание 1. Создайте веб-страницу, результат которой показан на рис. 9. Адреса ссылок произвольны. Ссылка должна открывать существующую страницу. Создайте всплывающую подсказку на ваше усмотрение.



Задание 2. Создайте три файла с именами 1.html, 2.html и 3.html, в каждом из них должен быть заголовок вида «Страница 1» и ссылка с текстом «Перейти». Причем ссылки должны быть замкнуты по схеме кольца, т.е. ссылка с документа 1.html должна вести на 2.html, с файла 2.html на 3.html, а файл 3.html опять ссылается на 1.html.