

Лекция 6

HTML – ссылки

Тег **<a>** является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия атрибутов **name** или **href** тег **<a>** устанавливает ссылку или якорь. Якорем называется закладка внутри страницы, которую можно указать в качестве цели ссылки. При использовании ссылки, которая указывает на якорь, происходит переход к закладке внутри веб-страницы.

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. В качестве значения атрибута **href** используется адрес документа (URL, Universal Resource Locator, универсальный указатель ресурсов), на который происходит переход. Адрес ссылки может быть абсолютным и относительным. Абсолютные адреса работают везде и всюду независимо от имени сайта или веб-страницы, где прописана ссылка. Относительные ссылки, как следует из их названия, построены относительно текущего документа или корня сайта.

Синтаксис

```
<a href="URL">...</a>
<a name="идентификатор">...</a>
```

Атрибуты

accesskey - Активация ссылки с помощью комбинации клавиш.

coords - Устанавливает координаты активной области.

download - Предлагает скачать указанный по ссылке файл.

href - Задаёт адрес документа, на который следует перейти.

hreflang - Идентифицирует язык текста по ссылке.

name - Устанавливает имя якоря внутри документа.

rel - Отношения между ссылаемым и текущим документами. **type**

HTML – текстовые ссылки. Веб-страница может содержать различные ссылки, которые ведут вас непосредственно на другие страницы и даже определенные части данной страницы. Эти ссылки известны как гиперссылки.

Гиперссылки позволяют посетителям перемещаться между веб-сайтами, нажимая на слова, фразы и изображения. Таким образом, вы можете создавать гиперссылки, используя текст или изображения, доступные на веб-странице.

Связывание документов. Ссылка указывается с использованием HTML-тега **<a>**. Этот тег называется **тегом привязки**, и все, что находится между открывающим тегом **<a>** и закрывающим тегом ****, становится частью ссылки, и пользователь может щелкнуть эту часть, чтобы перейти к связанному документу. Ниже приведен простой синтаксис для использования тега **<a>**.

```
<a href = "Document URL" ... attributes-list>Link Text</a>
```

Давайте попробуем следующий пример, который ссылается <http://www.tutorialspoint.com> на вашей странице:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hyperlink Example</title>
  </head>
  <body>
    <p>Click following link</p>
    <a href = "https://www.tutorialspoint.com" target = "_self">Tutorials Point</a>
  </body>
```

</html>

Это приведет к следующему результату, где вы можете нажать на сгенерированную ссылку, чтобы перейти на домашнюю страницу Tutorials Point (в этом примере).

Атрибут цели. Мы использовали **целевой** атрибут в нашем предыдущем примере. Этот атрибут используется для указания места, где открывается связанный документ. Ниже приведены возможные варианты

No	Вариант и описание
1	_blank Открывает связанный документ в новом окне или вкладке.
2	_self Открывает связанный документ в том же фрейме.
3	_parent Открывает связанный документ в родительском фрейме.
4	_Top Открывает связанный документ в полном теле окна.
5	targetframe Открывает связанный документ в именованном <i>целевом кадре</i> .

пример

Попробуйте следующий пример, чтобы понять принципиальную разницу в нескольких опциях, заданных для целевого атрибута.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hyperlink Example</title>
    <base href = "https://www.tutorialspoint.com/">
  </head>
  <body>
    <p>Click any of the following links</p>
    <a href = "/html/index.htm" target = "_blank">Opens in New</a> |
    <a href = "/html/index.htm" target = "_self">Opens in Self</a> |
    <a href = "/html/index.htm" target = "_parent">Opens in Parent</a> |
    <a href = "/html/index.htm" target = "_top">Opens in Body</a>
  </body>
</html>
```

Это приведет к следующему результату, где вы можете нажать на разные ссылки, чтобы понять разницу между различными опциями, заданными для целевого атрибута.

Использование базового пути. Когда вы размещаете ссылки на документы HTML, относящиеся к одному и тому же веб-сайту, необязательно указывать полный URL-адрес для каждой ссылки. Вы можете избавиться от него, если используете тег **<base>** в заголовке HTML-документа. Этот тег используется для определения базового пути для всех ссылок. Таким образом, ваш браузер объединит заданный относительный путь с этим базовым путем и создаст полный URL.

Ссылка на раздел страницы. Вы можете создать ссылку на определенный раздел данной веб-страницы, используя атрибут **name** . Это двухступенчатый процесс.

Сначала создайте ссылку на место, куда вы хотите попасть с веб-страницы, и назовите ее с помощью тега <a ...> следующим образом:

```
<h1>HTML Text Links <a name = "top"></a></h1>
```

Второй шаг — создать гиперссылку, связывающую документ и место, куда вы хотите попасть:

```
<a href = "/html/html_text_links.htm#top">Go to the Top</a>
```

Это приведет к следующей ссылке, где вы можете нажать на сгенерированную ссылку. **Перейти к началу**, чтобы добраться до верхней части учебника HTML Text Link.

[Go to the Top](#)

Установка цветов ссылок. Вы можете установить цвета ваших ссылок, активных ссылок и посещенных ссылок, используя атрибуты **link** , **alink** и **vlink** тега <body>.

пример

Сохраните следующее в test.htm и откройте его в любом веб-браузере, чтобы увидеть, как **работают** атрибуты **link** , **alink** и **vlink** .

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hyperlink Example</title>
    <base href = "https://www.tutorialspoint.com/">
  </head>
  <body alink = "#54A250" link = "#040404" vlink = "#F40633">
    <p>Click following link</p>
    <a href = "/html/index.htm" target = "_blank" >HTML Tutorial</a>
  </body>
</html>
```

Просто проверьте цвет ссылки перед тем, как щелкнуть по ней, затем проверьте ее цвет при активации и при посещении ссылки.

Ссылки для скачивания

Вы можете создать текстовую ссылку, чтобы сделать ваши PDF, DOC или ZIP файлы загружаемыми. Это очень просто; вам просто нужно дать полный URL-адрес загружаемого файла следующим образом:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hyperlink Example</title>
  </head>
  <body>
    <a href = "https://www.tutorialspoint.com/page.pdf">Download PDF File</a>
  </body>
</html>
```

Это даст следующую ссылку и будет использовано для загрузки файла.

Диалоговое окно загрузки файла

Иногда желательно, чтобы вы указали опцию, при которой пользователь щелкает ссылку, и вместо всплывающего окна отображается всплывающее окно «Загрузка файла». Это очень просто и может быть достигнуто с помощью HTTP-заголовка в вашем HTTP-ответе.

Например, если вы хотите, чтобы файл **имени** файла загружался по заданной ссылке, его синтаксис будет следующим.

```
#!/usr/bin/perl

# Additional HTTP Header
print "Content-Type:application/octet-stream; name = \"FileName\"\\r\\n";
print "Content-Disposition:attachment; filename = \"FileName\"\\r\\n\\n";

# Open the target file and list down its content as follows
open( FILE, "<FileName" );

while(read(FILE, $buffer, 100)){
    print("$buffer");
}
```

HTML – Ссылки на изображения

Мы увидели, как создавать гипертекстовые ссылки, используя текст, а также научились использовать изображения на наших веб-страницах. Теперь мы узнаем, как использовать изображения для создания гиперссылок.

Просто использовать изображение в качестве гиперссылки. Нам просто нужно использовать изображение внутри гиперссылки в месте текста, как показано ниже:

```
<!DOCTYPE html>
<html>

  <head>
    <title>Image Hyperlink Example</title>
  </head>

  <body>
    <p>Click following link</p>
    <a href = "https://www.tutorialspoint.com" target = "_self">
      <img src = "/images/logo.png" alt = "Tutorials Point" border = "0"/>
    </a>
  </body>

</html>
```

Это приведет к следующему результату, где вы можете нажать на изображения, чтобы перейти на домашнюю страницу Tutorials Point.

Это был самый простой способ создания гиперссылок с использованием изображений. Далее мы увидим, как мы можем создавать чувствительные к мышке ссылки на изображения.

Чувствительные к мышке изображения

Стандарты HTML и XHTML предоставляют функцию, которая позволяет встраивать множество различных ссылок в одно изображение. Вы можете создавать разные ссылки на одно изображение на основе различных координат, доступных на изображении. Как только разные ссылки прикреплены к разным координатам, мы можем щелкнуть по разным частям изображения, чтобы открыть целевые документы. Такие чувствительные к мышке изображения известны как карты изображений.

Есть два способа создания карт изображений —

- **Карты изображений на стороне сервера** — это включено атрибутом **ismap** тега `` и требует доступа к серверу и связанным приложениям обработки карт изображений.

- **Карты изображений на стороне клиента** — создаются с помощью атрибута **usemap** тега `` вместе с соответствующими тегами `<map>` и `<area>`.

Карты изображений на стороне сервера — это включено атрибутом **ismap** тега `` и требует доступа к серверу и связанным приложениям обработки карт изображений.

Карты изображений на стороне клиента — создаются с помощью атрибута **usemap** тега `` вместе с соответствующими тегами `<map>` и `<area>`.

Серверные карты изображений

Здесь вы просто помещаете свое изображение в **гиперссылку** и используете атрибут **ismap**, который делает его специальным изображением, а когда пользователь щелкает какое-то место на изображении, браузер передает координаты указателя мыши вместе с URL-адресом, указанным в теге `<a>`, на веб-сервер. Сервер использует координаты указателя мыши, чтобы определить, какой документ следует доставить обратно в браузер.

Когда используется **ismap**, атрибут **href** содержащего тега `<a>` должен содержать URL-адрес серверного приложения, такого как сценарий `cgi` или `PHP` и т. Д., Для обработки входящего запроса на основе переданных координат.

Координаты положения мыши — это пиксели экрана, отсчитываемые от верхнего левого угла изображения, начиная с (0,0). Координаты, перед которыми стоит знак вопроса, добавляются в конец URL.

Например, если пользователь щелкает 20 пикселей вверх и 30 пикселей вниз от верхнего левого угла следующего изображения —

Который был сгенерирован следующим фрагментом кода —

```
<!DOCTYPE html>
<html>
  <head>
    <title>ISMAP Hyperlink Example</title>
  </head>
  <body>
    <p>Click following link</p>
    <a href = "/cgi-bin/ismap.cgi" target = "_self">
      <img ismap src = "/images/logo.png" alt = "Tutorials Point" border = "0"/>
    </a>
  </body>
</html>
```

Затем браузер отправляет следующие параметры поиска на веб-сервер, которые могут быть обработаны сценарием **ismap.cgi** или **файлом карты**, и вы можете связать любые документы, которые вам нравятся, с этими координатами:

```
/cgi-bin/ismap.cgi?20,30
```

Таким образом, вы можете назначать разные ссылки на разные координаты изображения, и когда вы щелкаете по этим координатам, вы можете открыть соответствующий связанный документ. Чтобы узнать больше об **атрибуте ismap**, вы можете проверить, как использовать изображение [ismap?](#)

Примечание. Вы изучите программирование на CGI, когда будете изучать программирование на Perl. Вы можете написать свой скрипт для обработки этих

переданных координат, используя PHP или любой другой скрипт. А сейчас давайте сосредоточимся на изучении HTML, а позже вы можете вернуться к этому разделу.

Примечание. Вы изучите программирование на CGI, когда будете изучать программирование на Perl. Вы можете написать свой скрипт для обработки этих переданных координат, используя PHP или любой другой скрипт. А сейчас давайте сосредоточимся на изучении HTML, а позже вы можете вернуться к этому разделу.

Клиентские карты изображений

Карты изображений на стороне клиента **активируются** атрибутом **usemap** тега `` и определяются специальными тегами расширения `<map>` и `<area>`.

Изображение, которое будет формировать карту, вставляется на страницу с использованием тега `` в качестве обычного изображения, за исключением того, что оно содержит дополнительный атрибут, называемый **usemap**. Значением атрибута `usemap` является значение, которое будет использоваться в теге `<map>` для связывания тегов карты и изображения. `<Map>` вместе с тегами `<area>` определяют все координаты изображения и соответствующие ссылки.

Тег `<area>` внутри тега карты определяет форму и координаты для определения границ каждой активной точки, доступной на изображении, доступной на изображении. Вот пример из карты изображений:

```
<!DOCTYPE html>
<html>
  <head>
    <title>USEMAP Hyperlink Example</title>
  </head>
  <body>
    <p>Search and click the hotspot</p>
    <img src = /images/html.gif alt = "HTML Map" border = "0" usemap = "#html"/>
    <!-- Create Mappings -->
    <map name = "html">
      <area shape = "circle" coords = "80,80,20"
        href = "/css/index.htm" alt = "CSS Link" target = "_self" />
      <area shape = "rect" coords = "5,5,40,40" alt = "jQuery Link"
        href = "/jquery/index.htm" target = "_self" />
    </map>
  </body>
</html>
```

Контрольные вопросы

1. Как вставлять текстовые ссылки?
2. Какой тег используется для ссылки на изображения?