**Практическая работа 16**

**Объекты Image в JavaScript**

**Цель:** Научить создавать объекты типаImage

**Задания:**

1. Ознакомиться с теоретическими материалами темы.
2. Создайте два рисунка Image1.jpeg и Image2.jpeg. Создайте документ, в котором содержится первый рисунок. При нажатии на кнопку первый рисунок должен быть заменен вторым, имеющим те же размеры, что и первый.
3. Создайте документ, в котором загружается определенная картинка, при наведении на которую происходит смена картинку на другую.
4. Создайте документ, содержащий три рисунка (image1.jpeg, image2.jpeg, image3.jpeg), служащих ссылками на документы (link1.htm, link2.htm, link3.htm), при наведении на определенный рисунок, рисунок должен изменяться, при нажатии, должна запускаться соответствующая страничка.

**Необходимые приборы:** ПК, текстовый редактор Блокнот, браузер

**Методические рекомендации к выполнению лабораторной работы:**

**Методические рекомендации к выполнению задания 1**

**Изображения на web-странице**

Объект Image стал доступен начиная с версии с 1.1 языка JavaScript (то есть с Netscape Navigator 3.0). С помощью объекта Image можно вносить изменения в графические образы, присутствующие на web-странице. В частности, это позволяет нам создавать мультипликацию.

Заметим, что пользователи браузеров более старых версий (таких как Netscape Navigator 2.0 или Microsoft Internet Explorer 3.0 - т.е. использующих версию 1.0 языка JavaScript) не смогут запускать скрипты, приведенные в этой части описания. Или, в лучшем случае, на них нельзя будет получить полный эффект.

В JavaScript все изображения предстают в виде массива. Массив этот называется *images* и является свойством объекта document. Каждое изображение на web-странице получает порядковый номер: первое изображение получает номер 0, второе - номер 1 и т.д. Таким образом, к первому изображению мы можем адресоваться записав document.images[0].   
Каждое изображение в HTML-документе рассматривается в качестве объекта Image. Объект Image имеет определенные свойства, к которым и можно обращаться из языка JavaScript. Например, можно определить, какой размер имеет изображение, обратившись к его свойствам *width* и *height*. То есть по записи document.images[0].width Вы можете определить ширину первого изображения на web-странице (в пикселах).

К сожалению, отслеживать индекс всех изображений может оказаться затруднительным, особенно если на одной странице их содержится довольно много. Эта проблема решается назначением изображениям своих собственных имен. Так, если заводить изображение с помощью тэга

<img src="img.gif" name="myImage" width=100 height=100>

то можно обращаться к нему, написав document.myImage или document.images["myImage"].

**Загрузка новых изображений**

Часто приходится смену изображений на web-странице и для этого требуется использование атрибута *src*. Как и в случае тэга <img>, атрибут *src* содержит адрес представленного изображения. Теперь - в языке JavaScript 1.1 - имеется возможность назначать новый адрес изображению, уже загруженному в web-страницу. И в результате, изображение будет загружено с этого нового адреса, заменив на web-странице старое. Рассмотрим к примеру запись:

<img src="img1.gif" name="myImage" width=100 height=100>

Здесь загружается изображение *img1.gif* и получает имя *myImage*. В следующей строке прежнее изображение *img1.gif* заменяется уже на новое - *img2.gif*:

document.myImage.src= "img2.src";

При этом новое изображение всегда получает тот же размер, что был у старого. И уже нельзя изменить размер поля, в котором это изображение размещается.

**Начало формы**

**Конец формы**

**Упреждающая загрузка изображения**

Один из недостатков такого подхода может заключаться в том, что *после* записи в *src* нового адреса начинает процесс загрузки соответствующего изображения. И поскольку этого не было сделано заранее, то еще пройдет некоторое время, прежде чем новое изображение будет передано через Интернет и встанет на свое место. В некоторых ситуациях это допустимо, однако часто подобные задержки неприемлемы. Решением данной проблемы была бы упреждающая загрузка изображения. Для этого мы должны создать новый объект Image. Рассмотрим следующие строки:

hiddenImg= newImage();

hiddenImg.src= "img3.gif";

В первой строке создается новый объект Image. Во второй строке указывается адрес изображения, которое в дальнейшем будет представлено с помощью объекта *hiddenImg*. Как мы уже видели, запись нового адреса в атрибуте *src* заставляет браузер загружать изображение с указанного адреса. Поэтому, когда выполняется вторая строка нашего примера, начинает загружаться изображение *img2.gif*. Но как подразумевается самим названием *hiddenImg* ("скрытая картинка"), после того, как браузер закончит загрузку, изображение на экране не появится. Оно будет лишь будет сохранено в памяти компьютера (или точнее в кэше) для последующего использования. Чтобы вызвать изображение на экран, мы можем воспользоваться строкой:

document.myImage.src= hiddenImg.src;

Но теперь изображение уже немедленно извлекается из кэша и показывается на экране. Таким образом, сейчас мы управляли упреждающей загрузкой изображения.

Браузер должен к моменту запроса закончить упреждающую загрузку, чтобы необходимое изображение было показано без задержки. Поэтому, если Нужно предварительно загрузить большое количество изображений, то может иметь место задержка, поскольку браузер будет занят загрузкой всех картинок. Вы всегда должны учитывать скорость связи с Интернет - загрузка изображений не станет быстрее, если пользоваться показанными командами. Мы лишь пытаемся чуть раньше загрузить изображение - поэтому и пользователь может увидеть их раньше. В результате и весь процесс пройдет более гладко. Можно создавать красивые эффекты, используя смену изображений в качестве реакции на определенные события. Например, можно изменять изображения в тот момент, когда курсор мыши попадает на определенную часть страницы.

Объект Image дает возможность создавать действительно сложные эффекты. Однако заметим, что *не всякое* изображение или программа JavaScript способны улучшить страницу. Не количество изображений делает web-страницу привлекательной, а их качество. Сама загрузка 50 килобайт плохой графики способна вызвать раздражение. При создании специальных эффектов с изображениями с помощью JavaScript следует помнить об этом.

**Методические рекомендации к выполнению задания 2**

1. Создайте два рисунка Image1.jpeg и Image2.jpeg. Создайте документ, в котором содержится первый рисунок. При нажатии на кнопку первый рисунок должен быть заменен вторым, имеющим те же размеры, что и первый. Для создания такого документа используйте следующие команды:

<html>

<head>

</head>

<body bgcolor=#925142>

<center>

<img src="image1.jpeg" name="myImage" width=400 height=400>

</center>

<p>

<form name="myForm">

<input type="button" value="Нажми" name="myButton" onClick= 'document.myImage.src= "image2.jpeg";' >

</form>

</center>

</body>

</html>

**Методические рекомендации к выполнению задания 3**

1 Создайте документ, в котором загружается определенная картинка, при наведении на которую происходит смена картинку на другую.

Исходный код этого примера выглядит следующим образом:

<a href="#"

onMouseOver="document.myImage2.src='img2.gif'"

onMouseOut="document.myImage2.src='img1.gif'">

<img src="img1.gif" name="myImage2" width=160 height=50 border=0></a>

**Методические рекомендации к выполнению задания 4**

1. Заранее подготовьте рисунки: image1.jpeg, image2.jpeg, image3.jpeg.
2. Создайте три документа с произвольным содержанием: link1.htm, link2.htm, link3.htm.
3. Создайте документ, содержащий три рисунка (image1.jpeg, image2.jpeg, image3.jpeg), служащих ссылками на документы (link1.htm, link2.htm, link3.htm), при наведении на определенный рисунок, рисунок должен изменяться, при нажатии, должна запускаться соответствующая страничка.

**Вопросы для самоконтроля:**

1. Для чего предназначен объект Image?
2. Какие действия можно производить с использованием объекта Image?
3. В виде чего представляются все изображения в JavaScript?
4. Какие команды используются при загрузке новых изображений?
5. Каким образом производится замена рисунков при нажатии?