JavaScript и объекты браузера

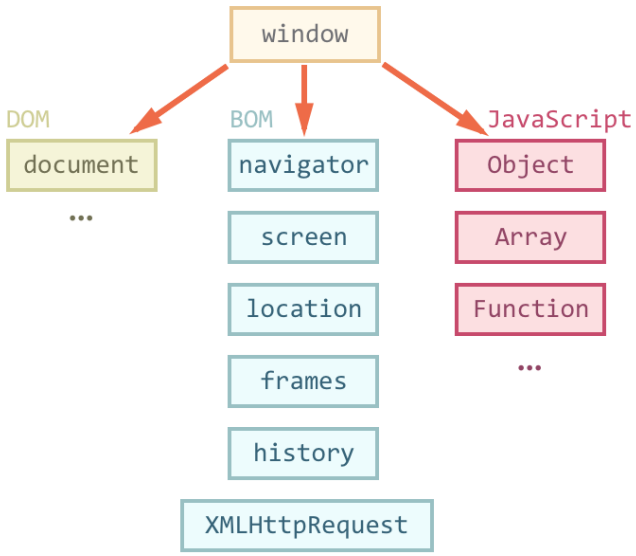
План

1. Объекты браузера
2. Объекты BOM
   2.1. Объект window
   2.2. Объект navigator
   2.3. Объект Screen
   2.4. Объект document
3. Элементы документа
   3.1. Объект Anchor
   3.2. Объект Link
   3.3. Объект Location
   3.4. Объект Image

Объекты браузера

Сам по себе язык JavaScript не предусматривает работы с браузером, документом и вообще не знает про HTML. Но позволяет легко расширять себя новыми функциями и объектами. Каждая из реализаций JavaScript имеет набор доступных объектов и объектных типов. Мы с Вами изучаем JavaScript применительно к Front-End разработке. Т.е. используем его внутри браузера. И для этого языку доступны некоторые глобальные объекты для управления окном браузера, загруженным в него html документом и т.д.

На рисунке ниже схематически отображена структура, которая получается если посмотреть на совокупность браузерных объектов с «высоты птичьего полёта».



Как видно из рисунка, на вершине стоит `window`.

У этого объекта двоякая позиция – он с одной стороны является глобальным объектом в JavaScript, с другой – содержит свойства и методы для управления окном браузера, открытия новых окон, например:

```
// открыть новое окно/вкладку с URL http://ya.ru
window.open('http://ya.ru');
```

Объектная модель документа (DOM)

Глобальный объект document даёт возможность взаимодействовать с содержимым страницы.

Пример использования:

```
document.body.style.background = 'red';
alert( 'Элемент BODY стал красным, а сейчас обратно вернётся' );
document.body.style.background = '';
```

Он и громадное количество его свойств и методов описаны в стандарте W3C DOM.

По историческим причинам когда-то появилась первая версия стандарта DOM Level 1, затем придумали ещё свойства и методы, и появился DOM Level 2, на текущий момент поверх них добавили ещё DOM Level 3 и готовится DOM 4.

Современные браузеры также поддерживают некоторые возможности, которые не вошли в стандарты, но де-факто существуют давным-давно и отказываться от них никто не хочет. Их условно называют «DOM Level 0».

Также информацию по работе с элементами страницы можно найти в стандарте HTML 5.

Мы подробно ознакомимся с DOM далее в этой части учебника.

Объектная модель браузера (BOM)

BOM – это объекты для работы с чем угодно, кроме документа.

Объект navigator содержит общую информацию о браузере и операционной системе. Особенно примечательны два свойства: navigator.userAgent – содержит информацию о браузере и navigator.platform – содержит информацию о платформе, позволяет различать Windows/Linux/Mac и т.п.

Объект location содержит информацию о текущем URL страницы и позволяет перенаправить посетителя на новый URL.

Функции alert/confirm/prompt – тоже входят в BOM.

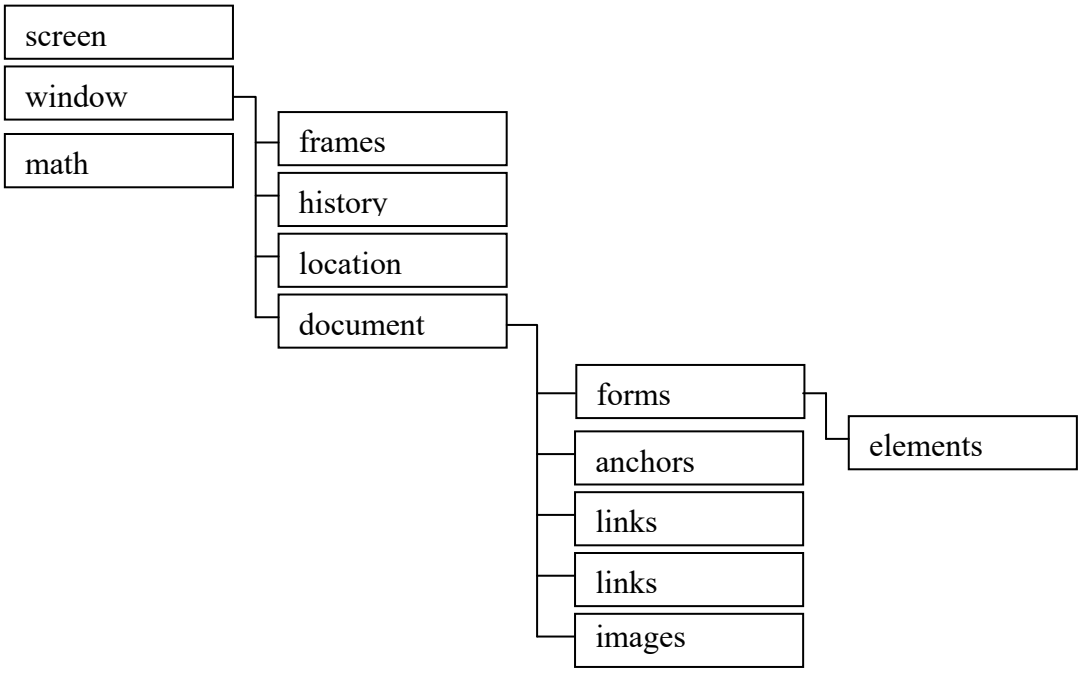
Пример использования:

```
alert( location.href ); // выведет текущий адрес
```

Большинство возможностей BOM стандартизированы в HTML 5, хотя различные браузеры и предоставляют зачастую что-то своё, в дополнение к стандарту.

Объекты BOM

Модель документа, которую мы рассмотрим сейчас несколько вышла из моды, но у нее есть несомненное преимущество «Она работает на всех браузерах», т.к. была предложена еще компанией Netscape, которая собственно и придумала JavaScript.



Объект window

Объект верхнего уровня для групп объектов document, location и history.

Синтаксис:

Для определения окна используется метод open:

```
windowVar = window.open("URL", "windowName" [,"windowFeatures"])
```

windowVar имя нового окна. Эта переменная используется при ссылках на свойства, методы и контейнеры окна.

windowName имя окна, используемое в атрибуте TARGET тегов <FORM> и <A>.

Параметры открытия окна приведены в следующей таблице.

Параметр	Значения	Альт. значения
Toolbar	yes | no	1 | 0
Location	yes | no	1 | 0
Directoties	yes | no	1 | 0
Status	yes | no	1 | 0
Menubar	yes | no	1 | 0
Scrollbars	yes | no	1 | 0
Resizable	yes | no	1 | 0
Width	*pixels*	
Height	*pixels*	

Использование свойств и методов window:

window.*propertyName*
window.*methodName(parameters)*
self.*propertyName*
self.*methodName(parameters)*

top.*propertyName*
top.*methodName(parameters)*
parent.*propertyName*
parent.*methodName(parameters)*
windowVar.propertyName
windowVar.methodName(parameters)
propertyName
methodName(parameters)

windowVar переменная, ссылающаяся на обект window. Смотрите синтаксис определения окна.

propertyName одно из свойств, описаныых ниже.

methodName один из методов, описанных ниже.

Для определения событий onLoad и onUnload для объекта window используются теги <BODY> и <FRAMESET>:

Описание:

Объект window является объектом верхнего уровня в клиентской иерархии JavaScript. Объекты frame также являются окнами.
Свойства self и window являются синонимами для текущего окна, и вы можете использовать их для ссылки на текущее окно. Например, вы можете закрыть текущее окно, используя window.close() или self.close(). Вы можете использовать эти свойства для однозначного определения свойства self.status из формы, называемой status.
Свойства top и parent также являются синонимами и могут быть использованы вместо имени окна. top ссылается на самое верхнее окно Navigator-а, а parent ссылается на окно, содержащее frameset. Смотрите свойства top и parent. Поскольку допускается существование текущего окна, вам не нужно ссылаться на имя окна, когда вы объявляете его методы или назначаете свойства. Например, status="Jump to a new location" является действительным назначением свойства и close() является действительным вызовом метода. Однако, когда вы открываете или закрываете окно внутри события, вы должны определить window.open() или window.close() вместо того, чтобы использовать просто open() или close(). Благодаря to the scoping статических объектов в JavaScript, объявление close() без определения имени объекта равносильно document.close().

Вы можете ссылаться на объекты frame окна, используя массив frames. Массив frames содержит запись для каждого фрейма в окне с тегом <FRAMESET>.
У окон отсутствуют события пока в них не загружен некоторый HTML-документ, содержащий теги <BODY> или <FRAMESET>.

Свойства:

- defaultStatus отражает сообщение по умолчанию, отображаемое в строке состояния окна
- frames массив, отражающий все фреймы окна
- length отражает количество фреймов в родительском окне
- name отражает аргумент *windowName*
- parent является синонимом аргумента *windowName* и ссылается на окно, содержащее frameset
- self является синонимом аргумента *windowName* и ссылается на текущее окно

- status определяет текущее сообщение строки состояния окна
- top является синонимом аргумента *windowName* и ссылается на самое верхнее окно Navigator-a
- window является синонимом аргумента *windowName* и ссылается на текущее окно
- document Ссылка на объект документа, загруженного в окно
- location Ссылка на объект, location, по которому определяет адрес документа, загруженного в окне. Этот адрес отображается в адресной строке броузера.
- history Массив, содержащий адреса всех посещенных страниц.
- opener Ссылка на объект window, который открыл данное окно.

К оъекту History можно обращаться по номеру history[0], history[1], history[2].
У каждого этого объекта имеются следующие свойства

- ♦ current Содержит URL, данного элемента истории.
- ♦ length Содержит длину списка истории.
- ♦ next Содержит адрес следующей страницы
- ♦ previous Содержит адрес предыдущей страницы

Методы:

alert Выдает текстовое сообщение
close Закрывает окно
confirm Выдает сообщение с кнопками "Ok" и "Cancel"
open Открывает окно
showModalDialog() Показывает создаваемое окно в модальном режиме, т.е. пользователь должен закрыть его, чтобы продолжить работать с родительским окном
navigate() Перенаправляет окно браузера на другую страницу.
prompt Открывает диалоговое окно с строкой ввода
setTimeout Выполняет выражение по истечении установленного количества миллисекунд.
setInterval() Серия вызовов через указанный интервал
clearTimeout Окончание задержки, установленной методом setTimeout.
clearInterval() Окончание работы setInterval()

События

onblur() Окно потеряло фокус ввода.
onfocus() Окно получило фокус ввода (стало активным).
onhelp() Пользователь нажал кнопку F1
onresize() Изменился размер окна
onscroll() Пользователь прокрутил окно
onerror() Возникла ошибка при передаче
onbeforeunload() Окно будет выгружено (закрыто)
onunload() Окно закрывается
onload() Окно полностью загружено

Объект navigator

Этот объект содержит информацию о параметрах браузера, таких как: производитель, версия, поддержка cookie и других.

Свойства	
appCodeName	Кодовое имя браузера. Определяет группу браузеров или т.н. "движок".
appName	Обычное название браузера, например "Netscape Navigator"
appVersion	Версия браузера, например "6.0"
cookieEnabled	Признак того, что браузер поддерживает cookie. Логическое значение (да/нет).
userAgent	Строка, которую высылает браузер серверу при запросе страницы. Строка содержит данные о типе браузера, его версии и поддерживаемых расширениях..
Методы	
javaEnabled()	Метод определяет возможность запуска Java-скриптов, логическое значение.

Вот таким скриптом можно быстро определить параметры собственного браузера:

```
<script type="text/javascript" language="javascript">
alert(
  'Кодовое имя: ' +
      window.navigator.appCodeName + '\n' +
  'Название браузера: ' +
      window.navigator.appName + '\n' +
  'Версия: ' +
      window.navigator.appVersion + '\n' +
  'Поддержка cookie: ' +
      window.navigator.cookieEnabled + '\n' +
  'Строка агента: ' +
      window.navigator.userAgent

  );
</script>
```

Объект navigator в основном используется при написании кроссбраузерных скриптов, т.к. позволяет определить какие блоки кода в каком браузере исполнять. К сожалению, использование объекта navigator не даёт стопроцентрой гарантии при определении типа браузера. Связано это с тем, что любой браузер (кроме, наверное, IE) позволяет легко изменить строки User-Agent на что угодно, в том числе на подпись браузера другого типа.

Поэтому более точным будет определение типа браузера по наличию или отсутствию объектов, специфичных для конкретной платформы. О кроссбраузерности написано уже немало книг и статей, поэтомы мы не будет глубоко вдаваться в эту тему.

Объект Screen

Объект screen предназначен для определения характеристик экрана.

объект с таким именем объявлен глобально. Для обращения к этому объекту используется следующий синтаксис

```
var a=screen.prepertyName;
var xw=screen.width;
```

Свойства:

height	Разрешение экрана по вертикал в точках.
width	Разрешение экрана по горизонтали в точках.
availHeight	Содержит высоту экрана в точках, доступную для использования. Т.е. height за вычетом пространства занятого постоянными элементами пользовательского интерфейса, такими как панель задач и т.д.
availWidth	Содержит ширину экрана в точках, доступную для использования. Т.е. width за вычетом пространства занятого постоянными элементами пользовательского интерфейса, такими как панель задач и т.д.
colorDepth	Определяет глубину цвета, текущего режима экрана. Для Internet Explorer это глубина одновременно отображаемых цветов, а в Netscape содержит общее количество цветов, глубину палитры. Количество одновременно отображаемых цветов для Netscape содержится в свойстве pixelDepth.
pixelDepth	Определяет глубину цвета, текущего режима экрана. Количество одновременно отображаемых цветов для Netscape. В Internet Explorer отсутствует. Для безпалитровых режимов эти два свойства одинаковые.

Объект document

Объект document, это предопределенный экземпляр, который ссылается на документ HTML, являющийся владельцем скрипта. Или на документ, который загружен в текущее окно броузера.

Невозможно создать программно новые экземпляры этого объекта. Экземпляр этого объекта создается тегом <HTML> при написании странички.

Этот объект создан для того, чтобы можно было управлять содержанием отображаемой в данный момент информации.

MIME-типы

В настоящий момент почти все документы, расположенные в Internet, являются документами HTML. Но очень быстро появляются новые языки т новые форматы записи различной информации предназначенные для распространения по глобальной паутине. Некоторые из них создаются специально для WWW, а некоторые мигрируют или адаптируются с различных типов компьютеров. Какие то, изначально представляют собой документ, предназначеный заменить HTML, другие же лишь чтобы быть его частью. Чтобы броузер мог различать и правильно отображать все эти типы, введена специальная классификация информации, которую назвали MIME-типами.

Эти типы представляют собой короткие текстовые строки, некоторые из которых мы приведем ниже:

`text/html`	– определяет тип документа в формате HTML.
`text/plain`	– Обычный текст.
`image/gif`	– изображение в формате GIF.
`image/jpeg`	– изображение в формате JPEG.
`image/bmp`	– изображение в формате BMP.

`image/wmf` – изображение в формате WMP.
`video/avi` – видеоклип AVI.
`video/mpeg` – видеоклип MPEG.
`plugIn` – плагин для броузера (подпрограмма, подключаемая к броузеру и позволяющая ему отображать новые форматы данных).
`x-world/vrml` – документ на языке VRML (язык для написания миров виртуальной реальности).

Методы объекта

Методы объекта позволяют нам открывать новый документ, писать в него и закрывать. С двумя методами объекта document мы уже познакомились, это write() и writeln(). Все остальные методы и свойства документа вызываются так же как и ранее изученные.

```
document.<Имя метода>(<Параметры>);
document.<Имя свойства>=<Новое значение>;
<Переменная>=document.<Имя свойства>;
```

Метод	Назначение
open(mimeType, replace)	Открывает новый документ. mimeType – необязательный параметр, определяющий какой именно документ будет записан.
close()	Закрывает документ и выводит его на экран.
write(expr) writeln(expr)	Записывает информацию в документ.
getSelection()	Взвращает строку, содержащую выделенную в документе информацию.

Свойства объекта

Свойство	Назначение
alinkColor="aqua"	Позволяет узнать или изменить свойство документа alink, который управляет цветом активизированных гиперссылок документа. Цвет представлен в виде строки, содержащей определение цвета.
linkColor	Позволяет узнать или изменить свойство документа link, который управляет цветом еще не посещенных гиперссылок документа.
vlinkColor	Позволяет узнать или изменить свойство документа vlink, который управляет цветом посещенных гиперссылок документа.
fgColor	Представляет собой строку, определяющую цвет текста в документе по умолчанию.
bgColor	Представляет собой строку, определяющую цвет фона документа.
title	Возвращает и позволяет изменить заголовок документа.
URL	Возвращает полный URL адрес для текущего документа.
lastModified	Возвращает строку, содержащую дату последней модификации документа.
referrer	Содержит ссылку на документ, из которого был вызван переход на текущий. (Не работает в Internet Explorer).

Элементы документа

Документ HTML состоит из форматированного текста и некоторых дополнительных элементов оформления или управления документа, таких как гиперссылки, изображения, формы и т.д.

Для их чтения управления ими из программы в объекте документ имеются свойства, представляющие массивы элементов документа, сгруппированные по типам.

images

Это свойство документа, содержащее массив изображений документа. Добавление новых изображений в документ через это свойство не возможен, но возможно изменение их свойств, загрузка на их место других картинок и использование информации о них.

```
document.images[0]
document.images[1]
```

Изображения нумеруются в порядке их следования в исходном тексте документа, начиная с 0. Свойства изображений мы рассмотрим когда будем изучать объект

Следубщий пример заменяет в HTML документе изображение, на другое.

```
document.images[0].src = "car1.gif"
```

anchors

Это свойство документа, содержащее массив якорей документа. Добавление новых якорей в документ через это свойство не возможен, но возможно изменение их свойств и использование информации о них.

forms

Это свойство содержит массив форм документа. Каждая форма представляет собой объект типа Form, и содержит свойства формы и массив элементов управления фрмы.

links

Массив гипертекстовых ссылок документа. Каждая гиперссылка представляет собой объект типа Link.

Имена элементов документа

Поскольку ссылаться на элементы документа по их индексам довольно неудобно, то соответствующим элементам документа присваиваются имена. Эти имена определяются в исходном тексте документа HTML с помощью атрибута NAME тегов. Этот атрибут применим к тегам <A>, <IMG>, <FORM> элементам управления форм. Но к сожалению имена объектов HTML, поддерживает меньше броузеров, чем массивы элементов.

Anchor

Вы можете ссылаться на объекты anchor в вашей программе, используя массив anchors. Этот массив содержит запись для каждого тега <a>, содержащего атрибут NAME по порядку встречаемости в документе. Например, если документ содержит три поименованных якоря, то эти якоря представлены как document.anchor[0], document.anchor[1], document.anchor[2].

Использование массива anchors:

```
document.anchors[index]
document.anchors.length
```

index целое число, представляющее якорь в документе.

Для получения количества якорей в документе используется свойство length: document.anchors.length.

Хотя массив *anchors* представляет собой поименованные якоря, значение anchors[*index*] является всегда нулевым. Но если в документе якоря именуются по порядку натуральными числами, вы можете использовать массив *anchors* и его свойство length для употребления имени якоря перед использованием его в операторах, таких как установка location.hash.

Элементы массива anchors окрыты открыты только для чтения. Например, выражение document.anchors[0]="anchor1" не имеет эффекта.

Свойства:

Объект anchor имеет единственное свойство.

Name – Содержит имя якоря, которое можно использовать для переходов.

Массив anchors имеет следующие свойства:

Length – определяет число поименованных якорей в документе.

Методы:

нет

События:

нет

Link

Каждый объект link является объектом location и имеет те же свойства как и объект location. Если объект link также является объектом anchor, то объект записан в массивах *anchors* и *links*.

Когда пользователь выбирает объект link и переходит в документ, обозначенный ссылкой (определенный HREF=*locationorURL*), то этот документ содержит URL документа источника.

Массив links

Вы можете ссылаться на объекты link в вашей программе, используя массив *links*. Этот массив содержит запись для каждого объекта link (тега <A HREF="">) по порядку встречаемости в документе. Например, если документ содержит три объекта link, то эти ссылки представлены так document.links[0], document.links[1] и document.links[2].

Использование массива *links*:

document.links[*index*]

document.links.length

index целое число, представляющее ссылку в документе.

Для получения количества ссылок в документе используется свойство length: document.links.length.

Элементы в массиве *links* открыты только для чтения. Например, выражение document.links[0]="link1" не имеет эффекта.

Свойства: Объект link имеет следующие свойства:

- hash определяет имя якоря в URL
- host определяет hostname:port часть URL'a
- hostname определяет хост и доменное имя или IP адрес сетевого хоста

- href определяет запись URL
- pathname определяет url-path часть URL'a
- port определяет коммуникационный порт, который сервер использует для коммуникаций
- protocol определяет начало URL, включая двоеточие
- search определяет запрос
- target отражает атрибут TARGET

Массив *links* имеет следующие свойства:

- length отражает количество ссылок в документе

Методы: – нет

События:

onClick
onDblClick
onKeyDown
onKeyPress
onKeyUp
onMouseDown
onMouseOut
onMouseUp
onMouseOver

Объект Location

Объект location представляет собой полный URL. Каждое свойство объекта location представляет собой отдельную часть URL.
Следующий формат URL показывает связь между location свойствами:

```
protocol//hostname:port pathname search hash
```

protocol – представляет собой начало URL, включая первое двоеточие.
hostname – представляет хост и доменное имя или IP адрес сетевого хоста.
port – представляет коммуникационный порт, который сервер использует для коммуникаций.
pathname – представляет url-path часть URL'a.
search – представляет любой запрос в URL'e, начинающийся со знака вопроса.
hash – представляет имя якоря фрагмент в URL'e, начинающийся со знака #.
Смотрите описание свойств ниже, где более детально описаны различные части URL, или свойство href.

Объект location имеет еще два свойства, не показанных в формате:

href – представляет полный URL.
host – представляет набор *hostname:port*.

Объект location содержится в объекте window. Если вы ссылаетесь на объект location без определения окна, то объект location представляется как текущий location.
Если вы ссылаетесь на объект location и определяете имя окна, например, *windowReference*.location.*propertyName*, то объект location представляется как location определенного окна.

Не путайте объект location со свойством location объекта document. Вы не можете изменить значение свойства location (document.location), но вы можете изменить значение свойств объекта location (window.location.*propertyName*). document.location является

строковым значением, которое обычно равно window.location.href, который устанавливается когда вы загружаете документ, но перенаправление может изменить его.

Синтаксис для общеизвестных типов URL:

Объект Image.

В документе есть массив images[i], каждый элемент которого, является объектом Image, т.е. картинкой, отображаемой в документе.

Объект Image можно создать используя конструктор

```
Var Img1=new Image(width, height);
```

width – ширина изображения.
height – высота изображения.

Для объекта Image определены следующие события

```
onAbort
onError
onKeyDown
onKeyPress
onKeyUp
onLoad
```

border	Reflects the BORDER attribute.
Complete	Boolean value indicating whether the web browser has completed its attempt to load the image.
Height	Reflects the HEIGHT attribute.
Hspace	Reflects the HSPACE attribute.
Lowsrc	Reflects the LOWSRC attribute.
Name	Reflects the NAME attribute.
Src	Reflects the SRC attribute.
Vspace	Reflects the VSPACE attribute.
Width	Reflects the WIDTH attribute.

Пример использования изображений:

```
<html>
<head>
<SCRIPT LANGUAGE="javaScript">
  var btnArray=new Array(6);
  var PictLoaded=false;
  for(i=0;i<6;i++) btnArray[i]=new Array(2);

  function LoadPic()
   {for(i=0;i<6;i++)
     {btnArray[i][0] =Image(165,40);
      btnArray[i][1] =Image(165,40);
      btnArray[i][0].src="button1n.gif";
      btnArray[i][1].src="button1s.gif";
     }
    PictLoaded=true;
   }

  function SetSelect(ButtonIndex)
   {window.status="Select  "+window.location.href;
    if (PictLoaded)
```

```
       document.images[ButtonIndex].src=
              btnArray[ButtonIndex][1].src;
    return true;
    }

   function SetUnSelect(ButtonIndex)
    {window.status="UnSelect";
      if (PictLoaded)
         document.images[ButtonIndex].src=
              btnArray[ButtonIndex][0].src;
      return false;
    }

   function Goto(Address)
    {
      parent.frames[1].document.location.href=Address;
    }

</SCRIPT>

</head>

<body bgcolor="#CCCCCC" OnLoad="LoadPic()">
<p align="center">
<img NAME=btnAbout src="button1n.gif" width="165" height="40"
alt="About"
     onMouseOver="SetSelect(0)" onMouseOut="SetUnSelect(0)">
<img NAME=btnHome src="button1n.gif" width="165" height="40"
alt="Home"
     onMouseOver="SetSelect(1)" onMouseOut="SetUnSelect(1)">
<img NAME=btnSearch src="button1n.gif" width="165" height="40"
alt="Search"
     onMouseOver="SetSelect(2)" onMouseOut="SetUnSelect(2)">
<img NAME=btnRegister src="button1n.gif" width="165" height="40"
alt="Register"
     onMouseOver="SetSelect(3)" onMouseOut="SetUnSelect(3)">
<img NAME=btnFAQ src="button1n.gif" width="165" height="40"
alt="F.A.Q."
     onMouseOver="SetSelect(4)" onMouseOut="SetUnSelect(4)">
<img NAME=btnHelp src="button1n.gif" width="165" height="40"
alt="Help"
     onMouseOver="SetSelect(5)" onMouseOut="SetUnSelect(5)"
     onClick="Goto('D:/Doc/HTML_JavaScript/WORK/Colors3.html')">
</p>
</body>
</html>
```

Формы

План

1. Доступ к формам
2. Объект форма
3. Доступ к элементам формы
4. Поля ввода (обычное и многострочное)
5. Button
6. CheckBox
7. RadioButton

Доступ к формам

Одна или несколько форм, принадлежат объекту document. Каждая форма в документе является отдельным объектом. Для ссылки на формы используется один из двух методов:

- Свойство документа forms, являющееся массивом форм
- По имени формы.

Примеры:

```
document.forms[0].name
document.MainForm.metod
```

Объект form имеет следующие свойства:

- *action* отражает атрибут ACTION
- *elements* массив, отражающий все элементы в форме
- *encoding* отражает атрибут ENCTYPE
- *length* отражает количество элементов в форме
- *method* отражает атрибут METHOD
- *target* отражает атрибут TARGET

Методы:

- *submit* вызывает передачу данных формы.
- *Reset* сбрасывает состояние формы в начальное

События:

- onSubmit
- onReset

Для ссылки на элементы формы можно пользоваться следующими способами:

1. *document.formName.buttonName.propertyName*
2. *document.forms[index].buttonName.methodName(parameters)*
3. *document.formName*.elements[*index*].*propertyName*
4. *document.formName*.elements[*index*].*methodName(parameters)*

ссылка на document в большинстве случаев может быть опущена.

Следующие объекты являются элементами объекта form:

- *button*
- *checkbox*
- *hidden*
- *password*
- *radio*
- *reset*
- *select*
- *submit*
- *text*
- *textarea*

Поля ввода (обычное и многострочное)

Поле ввода текста в HTML форме. Текстовое поле позволяет пользователю вводить слова, фразы или числовой ряд. Для определения объекта text используется стандартный HTML синтаксис с добавлением событий onBlur, onChange, onFocus, onSelect:

Свойства:

- *name* отражает имя элемента, заданное в исходном коде HTML. Этот атрибут присутствует для всех элементов.
- *type* отражает тип элемента. Для текста всегда text. Этот атрибут присутствует для всех элементов.
- *defaultValue* отражает атрибут VALUE
- *value* отражает текущее значение поля объекта text

Методы:

- *blur* Убирает фокус с указанного объекта.
- *focus* Устанавливает фокус на определенный объект.
- *select* Выделяет текст в поле ввода.

События:

- *onBlur*
- *onChange*
- *onFocus*
- *onSelect*

Для TextArrea дополнительно определены следующие события:

- *onKeyDown*
- *onKeyPress*
- *onKeyUp*

Объект Button

Нажимаемая кнопка в HTML форме.

Объект button является обычной кнопкой, которую вы можете использовать для выполнения действия, определенного вами. Кнопка выполняет скрипт, определенный событием onClick.

Свойства:

- *name* отражает атрибут NAME
- *value* отражает атрибут VALUE (надпись на кнопке).

Методы:

- *click* Эмитирует нажатие кнопки.

События:

- *onClick* Вызывается при нажатии кнопки.

Объект CheckBox

Контрольный переключатель в HTML форме. checkbox является сенсорным переключателем, позволяющим пользователю устанавливать значение on или off.

Свойство checked используется для определения checkbox, помеченного галочкой в настоящий момент. Свойство defaultChecked используется для определения checkbox, помеченного галочкой при загрузке формы.

Свойства:

- *checked* позволяет вам в программе установить какой checkbox будет помечен галочкой.
- *defaultChecked* отражает атрибут CHECKED.
- *name* отражает атрибут NAME.
- *value* отражает атрибут VALUE.

Методы:

- *click* Эмитирует нажатие кнопки.

События:

- *onClick*

Объект RadioButton

Установка статических кнопок (кнопок radio) в HTML форме. Установка кнопок radio позволяет пользователю выбрать один пункт из списка.

Все кнопки radio в группе кнопок radio используют одинаковое свойство name. Для обращения к отдельным кнопкам radio в вашей программе, используйте имя объекта с индексом, начинающимся с нуля, для каждой кнопки, также как вы это делали для массива,

```
forms: document.forms[0].radioName[0]    //это первая,
document.forms[0].radioName[1]    //это вторая и так далее.
```

Свойства:

- *checked* позволяет вам программно выбирать кнопку radio
- *defaultChecked* отражает атрибут CHECKED
- *length* отражает количество кнопок radio в объекте radio
- *name* отражает атрибут NAME
- *value* отражает атрибут VALUE

Методы:

- *click*

События:

- *onClick*

Объекты reset и submit

Кнопка сброса (кнопка reset)в HTML форме. Кнопка reset сбрасывает все элементы в форме в их значения, установленные по умолчанию.

Кнопка передачи данных (кнопка submit) в HTML форме. Кнопка submit вызывает передачу формы.

Имеют все теже свойства, что и обычная кнопка.

Объект password

Текстовое поле в HTML форме, значение которого на экране отображается звездочками (*). Когда пользователь вводит текст в это поле, звездочки (*) скрывают введенное значение.

Объект hidden

Текстовый объект формы, который не отображается в HTML форме. Объект hidden используется для передачи пар имя/значение при загрузке формы.

-