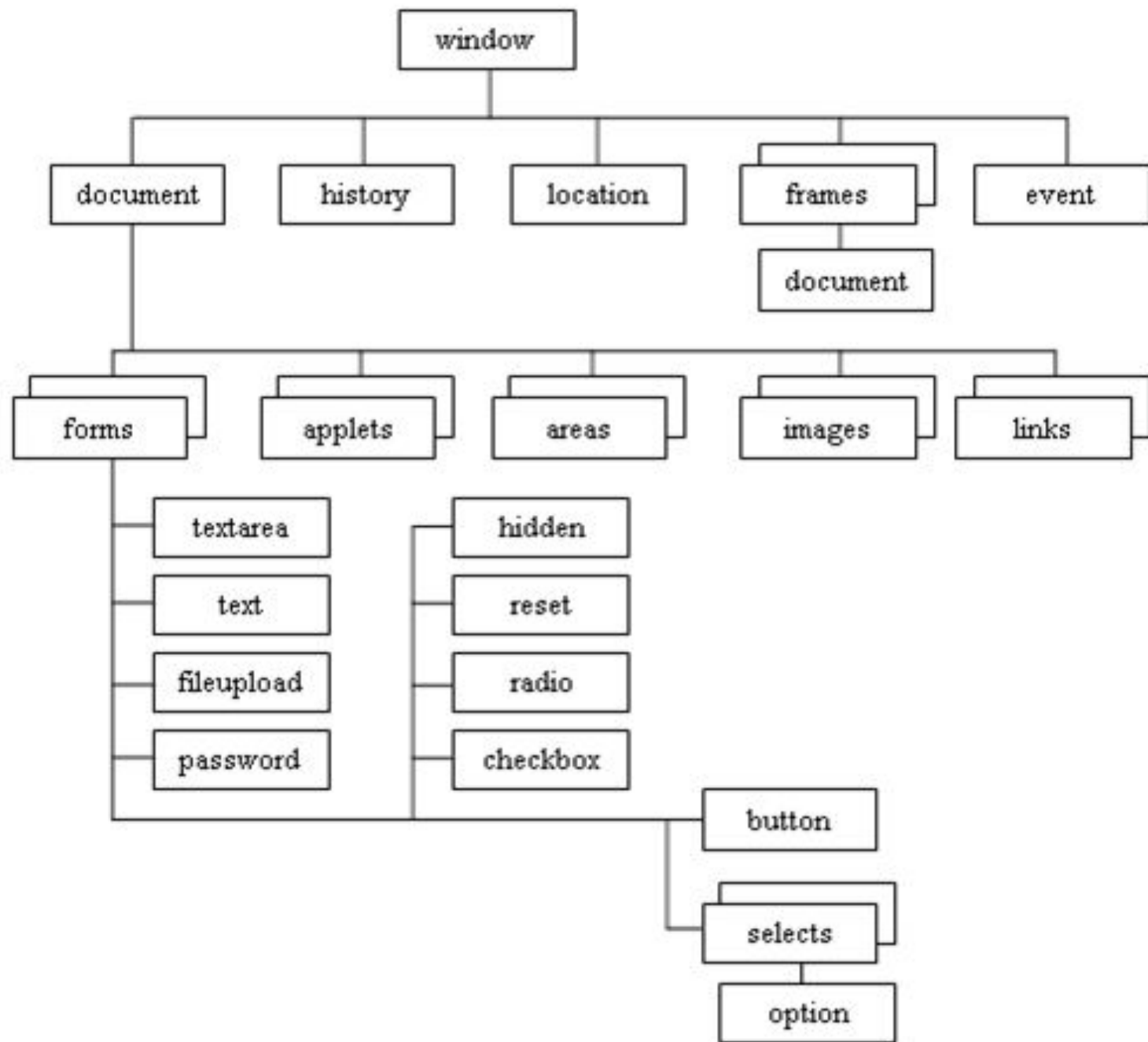


Объектная модель браузера и документа

Объектная модель браузера – это набор связанных между собой объектов, обеспечивающих доступ к содержанию страницы и ряду функций браузера.

Объектная модель браузера – множество объектов, которые создает браузер при загрузке страницы.

В рамках объектной модели браузера существует еще **объектная модель документа** (Document Object Model – DOM), она отвечает за сам html документ.



Объекты браузера

Имя объекта	Описание
window	Объект верхнего уровня в иерархии объектов языка JavaScript.
document	Содержит свойства, которые относятся к текущему HTML-документу, например имя каждой формы, цвета, используемые для отображения документа, и др.
location	Содержит свойства, описывающие местонахождение текущего документа, например адрес URL.
navigator	Содержит информацию о свойствах браузера.
history	Содержит информацию обо всех ресурсах, к которым пользователь обращался во время текущего сеанса работы с браузером.

Объект Window

Объект Window (окно)

Является объектом высшего уровня. Создается автоматически при запуске браузера.

При обращении к свойствам и методам окна либо свойствам и методам объектов, расположенных в иерархической структуре ниже объекта window, сам объект Window может не указываться.

При обращении к свойствам и методам окна не требуется явной ссылки на него.

Способы обращения:

1. явная ссылка: `window.alert(“ ”);`
2. неявная ссылка: `alert(“ “);`

Свойства

- ✓ **defaultStatus** – текстовое сообщение, которое по умолчанию выводится в строке состояния окна браузера;
- ✓ **name** – заголовок окна, который задается с помощью аргумента `windowName` метода `open()`;
- ✓ **parent** – синоним, используемый для обращения к родительскому окну.

Свойства

- ✓ **self** - синоним, используемый для обращения к текущему окну;
- ✓ **status** - текст временного сообщения в строке состояния окна браузера;
- ✓ **top** - синоним, используемый для обращения к главному окну браузера.

Методы

1. open () - создание окна (открывает новое окно браузера);

Является основным методом объекта window.

Формат:

open ([URL_ документа[, имя_окна[, атрибуты_окна]])

- **URL_документа** - описывает документ, который должен быть отображен в указанном окне.
- **имя окна** - если данный параметр задает имя текущего окна, то документ с указанным URL заменяет Web-страницу, которая отображается в окне браузера. Если заданное имя не совпадает с именем текущего окна, то браузер открывает новое окно и выводит в нем указанный документ.

- **атрибуты_окна** — если при вызове метода должно быть открыто новое окно, то его внешний вид может быть определен с помощью приведенных ниже атрибутов.

- ✓ **menubar** — строка меню.
- ✓ **toolbar** — панель инструментов.
- ✓ **location** — панель со строкой Location.
- ✓ **directories** — панель с кнопками каталогов.
- ✓ **status** — строка состояния.
- ✓ **width** — ширина окна в пикселях.
- ✓ **height** — высота окна в пикселях.
- ✓ **resizable** — разрешение изменять размеры окна.

Все атрибуты должны быть оформлены в виде одной строки и разделяться запятыми. Пробелы между атрибутами недопустимы. Каждый атрибут задается в формате имя=значение (значение атрибута может быть пропущено).

- ✓ **close()** — закрывает текущее окно браузера;
- ✓ **alert()** — показывает окно предупреждения с сообщением и кнопкой ОК;
- ✓ **prompt()** — показывает окно приглашения с сообщением, текстовым полем и кнопками ОК и Cancel (Отмена);
- ✓ **confirm()** — показывает окно подтверждения с сообщением и кнопками ОК и Cancel (Отмена);

- ✓ **navigate()** — загружает другую страницу с указанным адресом;
- ✓ **blur()** — убирает фокус с текущей страницы; соответствующее событие — `onblur`;
- ✓ **focus ()** — устанавливает страницу в фокус; соответствующее событие — `onfocus`;
- ✓ **setInterval()** — указывает процедуре выполняться периодически через заданное количество миллисекунд;

- ✓ **setTimeout()** – запускает программу через заданное количество миллисекунд после загрузки страницы;

Пример: `setTimeout('alert("прошла секунда")', 1000)`

- ✓ **close ()** – закрытие окна;
- ✓ **print()** – вывод на печать содержимого окна.

Пример

```
<script type=text/javascript>  
function pr() {  
    print();  
}  
</script>
```

```
<form>
```

```
<input type='button' value='Напечатать содержимое  
данной страницы' onclick='pr()' />
```

```
</form>
```

Пример

```
win_Example = window.open  
("http://www.rambler.ru", "linkWin", "toolbar=no,  
scrollbars=yes")  
windows.close ()  
self.close ()  
close ()  
win_Example.Close ()
```

Пример

```
<html>
<body>
<script language="JavaScript">
function opWind() {
var myUrl = "http://narod.ru";
myWin=window.open(myUrl, "wind1",
"width=200,height=100,resizable=no,scrollbars=no,menubar=no"); }
</script>
<input type="button" value="Открыть окно"
onClick="opWind()">
<input type="button" value="Закреть окно"
onClick="myWin.close()">
</body> </html>
```

События

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

Объект Navigator

Объект navigator

Объект navigator представляет ряд свойств браузера.

Свойства:

- ✓ **appName** – содержит имя браузера (например, "Microsoft Internet Explorer")
- ✓ **appVersion** – содержит информацию о версии браузера.



Объект Location

Объект Location

Связан с текущим URL адресом. Его свойства позволяют получать информацию о host-машине, с которой в данный момент связан браузер.

Для объекта location методы, не определены, также объект связан с какими-либо обработчиками событий.

Свойства:

- ✓ **href** — полный URL-адрес в виде строки;
- ✓ **host** — имя компьютера в сети Интернет;
- ✓ **pathname** — часть адреса URL, описывающая каталог, в котором находится документ;
- ✓ **port** — номер порта, который использует сервер;
- ✓ **protocol** — начальная часть, определяющая протокол, за которой следует двоеточие, например «http:»;
- ✓ **hostname** - содержит имя хоста;
- ✓ **target** - соответствует атрибуту target в теге <href>

Объект History

Объект history

Объект history содержит список адресов URL, посещенных в данном сеансе. Объект history связан с текущим документом. Методы этого объекта позволяют загружать в браузер различные ресурсы и обеспечивают навигацию по посещенным ресурсам.

Свойства

- ✓ **current** – текущая страница
- ✓ **next** – следующая страница
- ✓ **previous** – предыдущая страница
- ✓ **length** – количество элементов в списке посещенных страниц.

Методы:

- ✓ **back()** – загружает предыдущую страницу из списка;
- ✓ **forward()** – загружает следующую страницу из списка;
- ✓ **go()** – загружает страницу с относительным номером n из списка (от 0 до $history.length-1$) или с указанным URL-адресом.

history.go(-3)

Объект Document

Содержит информацию о загруженной странице. Для каждой страницы создается один объект **document**. Некоторые его свойства соответствуют параметрам тэга `body`.

Объект **document** является центральным в иерархической объектной модели и представляет всю информацию о HTML-документе с помощью коллекций и свойств. Он также предоставляет множество методов и событий для работы с документами.

alinkColor	Цвет активной ссылки на странице
bgColor	Цвет фона по-умолчанию
body	Ссылка на элемент body
cookie	Cookie данной страницы. Изменение этого свойства приводит к сохранению новых cookie в браузере пользователя.
fgColor	Цвет текста по-умолчанию.
lastModified	Дата последнего изменения страницы
linkColor	Цвет ещё не посещённых ссылок на странице
location	Текущий адрес страницы
parentWindow	Ссылка на родительское окно
referer	Адрес страницы, с которой зашел пользователь
title	Заголовок страницы
vlinkColor	Цвет посещённых ссылок
url	Адрес текущего документа

События

<code>onclick()</code>	Клик в области документа
<code>ondblclick()</code>	Двойной клик в области документа
<code>onerror()</code>	Ошибка при загрузке документа
<code>onhelp()</code>	Пользователь вызвал справку (нажал F1)
<code>onkeydown()</code>	События, происходящие при нажатии кнопок или перемещении указателя мыши.
<code>onkeypress()</code>	
<code>onkeyup()</code>	
<code>onmousedown()</code>	
<code>onmousemove()</code>	
<code>onmouseout()</code>	
<code>onmouseup()</code>	

Методы

<code>clear()</code>	Очистка выделенной области документа
<code>close()</code>	Закрытие документа, ранее открытого для изменения.
<code>open()</code>	Открытие документа для изменения
<code>write()</code>	Запись текста