BOM

LoftSchool
от мыслителя к создателю

Содержание

1

BOM

С помощью объектной модели браузера (BOM) возможно управление поведением браузера из JavaScript. BOM включает в себя несколько объектов.

2

Window

Объект window является корневым объектом JavaScript. Все объекты JavaScript, а также переменные и функции определяемые пользователем хранятся в объекте window. Объект **Window** в клиентском JavaScript является глобальным объектом. Это означает, что объект **Window** находится на вершине цепочки областей видимости и что его свойства и методы фактически являются глобальными переменными и функциями. Объект **Window** имеет свойство **window**, которое всегда ссылается на сам объект. Это свойство можно использовать для ссылки на сам объект, но обычно в этом нет необходимости, если требуется просто сослаться на свойство глобального объекта окна.

Свойства

Свойство	Описание
closed	Возвращает логическое значение (**true** или **false**) в зависимости от того, закрыто указанное окно или отрыто.

```
var newwin = window.open();

if (newwin.closed) {
   document.getElementById("mes").innerHTML
= "Окно newwin было закрыто.";
} else{
   document.getElementById("mes").innerHTML
= "Окно newwin открыто в данный момент.";
}
```

frames	Возвращает массив всех фреймов на странице (включая i**frame**). Элемент **iframe** является обычным узлом DOM, как и любой другой. Существенное отличие – в том, что с ним связан объект **window** внутреннего окна. Он доступен по ссылке **iframe.contentWindow**. Таким образом, **iframe.contentWindow.document** будет внутренним документом, **iframe.contentWindow.document.body** – его **<body>** и так далее. Элемент **<iframe>** является «двуличным». С одной стороны, это обычный узел DOM, с другой – внутри находится окно, которое может иметь совершенно другой URL, содержать независимый документ из другого источника. Внешний документ имеет полный доступ к **<iframe>** как к DOM-узлу. А тот к окну – если они с одного источника. Это приводит к забавным последствиям. Например, чтобы узнать об окончании загрузки **<iframe>**, мы можем повесить обработчик **iframe.onload.** По сути, это то же самое что **iframe.contentWindow.onload**, но его мы можем поставить лишь в случае, если окно с того же источника. Альтернативный способ доступа к окну фрейма – это получить его из коллекции **window.frames**.

frames

Есть два способа доступа:

window.frames[0]	доступ по номеру.
window.frames.iframeName	доступ по name ифрейма.

Обратим внимание: в коллекции хранится именно окно (**contentWindow**), а не DOM-элемент.
Внутри фрейма могут быть свои вложенные фреймы. Всё это вместе образует иерархию.

Ссылки для навигации по ней:

window.frames	коллекция «детей» (вложенных ифреймов)
window.parent	содержит ссылку на родительское окно, позволяет обратиться к нему из ифрейма.
window.top	содержит ссылку на самое верхнее окно (вершину иерархии).

Свойство **top** позволяет легко проверить, во фрейме ли находится текущий документ:

```
if (window == top) {
   alert( 'Этот скрипт выполняется в окне верхнего уровня' );
} else{
   alert( 'Этот скрипт исполняется во фрейме или дочернем окне!');
}
```

document	Возвращает объект **Document** данного окна.
history	Возвращает объект **History** данного окна.
length	Возвращает количество фреймов (включая **iframe**), которые находятся в данном окне.
location	Возвращает объект **Location** данного окна.
name	Устанавливает или возвращает имя данного окна.
navigator	Возвращает объект **Navigator** данного окна.
opener	Возвращает ссылку на окно, которое открыло данное.
parent	Возвращает родительское окно данного окна.
screen	Возвращает объект **Screen** данного окна.
self	Возвращает текущее окно.
top	Возвращает верхнее браузерное окно для данного окна.

Методы

Метод	Описание
alert()	Вызывает окно оповещения, которое содержит текст сообщения и клавишу OK.
blur()	Делает окно неактивным.
clearInterval()	Прекращает повторное выполнение кода заданного **setInterval()**.
clearTimeout()	Отменяет запланированное методом **setTimeout()** выполнение кода.
close()	Закрывает окно.

confirm()	Вызывает окно подтверждения содержащее текст сообщения и клавиши OK и Отмена.
	`//Выведем окно подтверждения и запишем результат в переменную x`
	`var x = confirm("Нажмите OK или Отмена.");`
	
	`//Если пользователь нажал OK`
	`if (x == true) {`
	`document.getElementById("mes").innerHTML = "Вы нажали кнопку OK."`
	`} else{ //Если пользователь нажал Отмена`
	`document.getElementById("mes").innerHTML = "Вы нажали кнопку Отмена."`
	`}`

focus()	Делает окно активным.

moveBy()	Смещает окно относительно его текущей позиции.
moveTo()	Перемещает окно на указанную позицию.
open()	Открывает новое окно браузера. **open(URL,имя,параметры)** Имя, устанавливает имя окна Является не обязательным параметром. Указывает способ открытия или имя окна. Возможные значения: • **_blank URL** загружается в новом окне • **_parent URL** загружается в родительском фрейме • **_self URL** заменяет текущую страницу Параметры отделенные запятой. Доступны следующие параметры: • ***height*** устанавливает высоту окна в пикселях. Минимальное значение 100. • ***location*** отвечает за отображение адресной строки (yes - отображать, no - неотображать). • ***menubar URL*** отвечает за отображение верхнего меню (**yes** - отображать, **no** - неотображать). • ***resizable*** отвечает за возможность изменения размеров окна (**yes** - можно изменять размер, **no** - нельзя изменять размер). • ***scrollbars*** отвечает за отображения полосы прокрутки (**yes** - отображать, **no** - неотображать). • ***width*** устанавливает ширину окна в пикселях. Минимальное значение 100.

print()	Распечатывает содержимое текущего окна.

prompt()	Вызывает окно запроса, побуждающее посетителя ввести в него определенные данные.
	`//Выведем окно запроса и запишем результат (введенные пользователем данные) в переменную x`
	`var x = prompt('Введите Ваше имя:', 'Имя');`
	
	`//Выведем введенные пользователем данные на страницу`
	`document.getElementById("mes").innerHTML = x;`

scrollBy()	Прокручивает содержимое окна на указанное количество пикселей.
scrollTo()	Прокручивает содержимое окна до указанных координат.
setInterval()	Вызывает функцию или выполняет код через определенные промежутки времени (указанные в миллисекундах).
setTimeout()	Вызывает функцию или выполняет код после указанного количества миллисекунд один раз.

3

Document

Свойства

С помощью данного объекта Вы сможете добавлять, изменять и удалять HTML элементы на странице из скриптов.

Свойство	Описание

cookie	Возвращает **cookie** связанные с данным документом.
	
	Куки - это небольшой объем данных, которые хранятся вэб браузером. Они позволяют Вам сохранять определенную информацию о пользователе и получать ее каждый раз, когда он посещает Вашу страницу. Каждый пользователь имеет свой собственный уникальный набор куков. Обычно куки используются веб сервером для выполнения таких функций как отслеживание посещений сайта, регистрации на сайте и сохранения сведений о заказах или покупках. Однако нам не нужно придумывать программу для вэб сервера чтобы использовать куки. Мы можем использовать их с помощью JavaScript.
	
	Создать куки можно следующим образом:
	`document.cookie = "name=значение; expires=дата; path=путь; domain=домен; secure";`
	
	получить весь сохраненный набор куков так:
	`var x = document.cookie;`

cookie	Для сохранения куки нужно присвоить **document.cookie** текстовую строку, которая содержит свойства куки, которые мы хотим создать: `document.cookie = "name=значение; expires=дата; path=путь;domain=домен; secure";` ***expires=дата*** Устанавливает дату истечения срока хранения куки. Дата должна быть представлена в формате, который возвращает метод **toGMTString()** объекта **Date**. Если значение expires не задано, куки будет удалено при закрытии браузера. ***path=путь*** Данная опция устанавливает путь на сайте, в рамках которого действует куки. Получить значение куки могут только документы из указанного пути. Обычно данное свойство оставляют пустым, что означает что только документ установивший куки может получит доступ к нему. ***domain=домен*** Данная опция устанавливает домен, в рамках которого действует куки. Получить значение куки могут только сайты из указанного домена. Обычно данное свойство оставляют пустым, что означает, что только домен установивший куки может получит доступ к нему. ***secure*** Данная опция указывает браузеру, что для пересылки куки на сервер следует использовать SSL. Очень редко используется. удаление куки из браузера происходит посредством установки срока хранения на одну секунду раньше текущего значения времени.

document.doctype	Позволяет узнать **doctype** документа.
domain	Возвращает доменное имя сервера, на котором размещается данный документ. `document.write(document.domain);`
forms	Возвращает массив содержащий все формы имеющиеся на странице.
images	Возвращает массив содержащий все картинки имеющиеся на странице.
links	Возвращает массив содержащий все ссылки имеющиеся на странице.
document.readyState	Позволяет узнать статус загрузки документа.
URL	Возвращает текущий URL.

4

Screen

Объект ***Screen*** содержит информацию об экране пользователя.

С помощью свойств данного объекта Вы можете узнать какое разрешение, а также какая глубина цвета установлена на экране пользователя.

Свойство **width** определяет ширину экрана пользователя, а свойство **height** высоту.

Свойства

Свойство	Описание
availHeight	Возвращает высоту экрана (исключая такие элементы интерфейса как полоса прокрутки, строка состояния, панель инструментов и т.д.).
availWidth	Возвращает ширину экрана (исключая такие элементы интерфейса как полоса прокрутки, строка состояния, панель инструментов и т.д.).
colorDepth	Возвращает битовую глубину цветовой палитры для отображения изображений.
height	Возвращает общую высоту экрана.
width	Возвращает общую ширину экрана.

5

Navigator

С помощью объекта ***navigator*** Вы можете определить какой браузер использует пользователь. Так же с помощью **navigator** Вы можете проверить может ли пользователь принимать **cookie** и включена ли у него поддержка Java.

Свойства

Свойство	Описание
appCodeName	Позволяет узнать кодовое имя браузера.
appName	Позволяет узнать имя браузера.
appVersion	Позволяет узнать версию браузера.
cookieEnabled	Проверяет включена или нет поддержка **cookie** у пользователя.
platform	Позволяет узнать платформу, под которую скомпилирован браузер пользователя (т.е. фактически узнать ОС, которую использует пользователь).
userAgent	Возвращает полную информацию о браузере пользователя (возвращает заголовок, который посылает браузер во время запроса страницы).

6

Location

С помощью объекта ***location*** Вы можете узнать любую информацию о URL текущего документа.

Свойства

Свойство	Описание
hash	Возвращает часть URL начиная со знака **#**
host	Возвращает часть URL содержащую домен сайта.
href	Возвращает URL целиком.
pathname	Возвращает часть URL содержащую путь к загруженному документу.
protocol	Возвращает часть URL содержащую название протокола.
search	Возвращает часть URL начиная со знака **?**

Методы

Метод	Описание
assign()	Загружает новый документ в том же окне браузера.
reload()	Загружает документ заново.

7

History

Объект ***History*** содержит список URL которые были посещены в данном окне браузера.

С помощью свойства **length** Вы можете узнать количество посещенных URL хранящихся в списке.

Методы

Метод	Описание
back()	Загружает предыдущий URL (если он есть)
forward()	Загружает следующий URL (если была нажата кнопка “назад”)
go()	Переходит на указанный URL по его **НОМЕРУ** в списке посещенных страниц. Число указывает смещение относительно текущей URL в списке. Данный параметр может принимать отрицательные значения.