Методы и функции JavaScript

abs	forward	setDate
acos	getDate	setHours
alert	getDay	setMinutes
anchor	getHours	setMonth
asin	getMinures	setSeconds
atan	getMonth	setTime
back	getSeconds	setTimeout
big	getTime	setYear
blink	getTimezoneOffset	sin
blur	getYear	small
bold	go	sqrt
ceil	indexOf	strike
charAt	isNaN	sub
clearTimeout	italics	submit
click	lastIndexOf	substring
close (объект document)	link	sup
close (объект window)	log	tan
confirm	max	toGMTString
cos	min	toLocaleString
escape	open (объект document)	toLowerCase
eval	open (объект window)	toUpperCase
exp	parse	unescape
fixed	parseFloat	UTC
floor	parseInt	write
focus	pow	writeln
fontcolor	prompt	
fontsize	random	

Метод abs

Возвращает абсолютное значение числа.

Синтаксис:

```
Math.abs(number)
```

number любое числовое выражение или свойство существующего объекта.

Метод:

Math

Метод acos

Возвращает арккосинус числа (в радианах).

Синтаксис:

```
Math.acos(number)
```

number числовое выражение между -1 и 1 или свойство существующего объекта.

Метод:

Math

Описание:

Метод acos возвращает числовое значение между 0 и Пи. Если значение *number* находится за пределами данного диапазона, возвращаемое значение всегда будет 0.

Смотрите также:

- методы asin, atan, cos, sin и tan.

Метод alert

Отображает диалоговое окно Alert с сообщением и кнопкой OK.

Синтаксис:

```
alert("message")
```

Метод:

window

Описание:

Метод alert используется для отображения сообщения, не требующего решения пользователя. Аргумент*message* определяет сообщение, которое содержит диалоговое окно. Хотя alert является методом объекта window вам не нужно определять *windowReference*, при его вызове. Например, *windowReference*.alert() необязательно.

Смотрите также:

- методы confirm, prompt.

Метод anchor

Создает HTML якорь, который используется как гипертекстовая ссылка.

Синтаксис:

```
text.anchor(nameAttribute)
```

text любая строка или свойство существующего объекта.
nameAttribute любая строка или свойство существующего объекта.

Метод:

string

Описание:

Метод anchor используется с методами write или writeln для программного создания и отображения якоря в документе. Якорь создается с помощью метода anchor, а write или writeln используется для отображения якоря в документе.
В синтаксисе строка *text* представляет собой текст, который увидит пользователь. Строка *nameAttribute*представляет собой атрибут NAME тага <A>.
Якоря, созданные с помощью метода anchor становятся элементами массива anchors. Информацию о массиве anchors смотрите в объекте anchor.

Смотрите также:

- метод link

Метод asin

Возвращает арксинус числа (в радианах).

Синтаксис:

```
Math.asin(number)
```

number числовое выражение между -1 и 1 или свойство существующего объекта.

Метод:
Math
Описание:
Метод asin возвращает числовое значение между -Пи/2 и Пи/2. Если значение *number* находится за пределами данного диапазона, возвращаемое значение всегда будет 0.
Смотрите также:
- методы acos, atan, cos, sin, tan.

Метод atan
Возвращает арктангенс числа (в радианах).
Синтаксис:

```
Math.atan(number)
```

number любое числовое выражение или свойство существующего объекта, представляющее собой тангенс угла.
Метод:
Math
Описание:
Метод atan возвращает числовое выражение между -Пи/2 и Пи/2.
Смотрите также:
- методы acos, asin, cos, sin, tan.

Метод back
Позволяет вернуться на предыдущий URL в списке посещенных URL'ей.
Синтаксис:

```
history.back()
```

Метод:
history
Описание:
Этот метод выполняет действие равносильное выбору пользователем кнопки Back в окне Navigator'a. Метод back также равносилен history.go(-1).
Смотрите также:
- методы forward, go.

Метод big
Вызывает строку, отображаемою большим шрифтом, как если установить ей таг <BIG>.
Синтаксис:

```
stringName.big()
```

stringName любая строка или свойство существующего объекта.
Метод:
string
Описание:
Для форматирования и отображения строки в документе метод big используется с методами write или writeln.
Смотрите также:
- методы fontsize, small.

Метод blink

Вызывает мигающую строку, как если установить ей таг <BLINK>.

Синтаксис:

```
stringName.blink()
```

stringName любая строка или свойство существующего объекта.

Метод:

string

Описание:

Для форматирования и отображения строки в документе метод blink используется с методами write или writeln.

Смотрите также:

- методы bold, italics, strike.

Метод blur

Изменен в Navigator 3.0.

Убирает фокус с указанного объекта.

Синтаксис:

```
1. password.blur()

2. select.blur()

3. textName.blur()

4. textareaName.blur()
```

password любое значение атрибута NAME объекта password или элемент массива *elements*.
select любое значение атрибута NAME объекта select или элемент массива *elements*.
textName любое значение атрибута NAME объекта text или элемент массива *elements*.
textareaName любое значение атрибута NAME объекта textarea или элемент массива *elements*.

Метод:

password, select, text, textarea.

Описание:

Метод blur используется для удаления фокуса с указанного элемента формы.

Смотрите также:

- методы focus, select.

Метод bold

Вызывает строку, отображаемую жирным шрифтом, как если установить ей таг <B>.

Синтаксис:

```
stringName.bold()
```

stringName любая строка или свойство существующего объекта.

Метод:

string

Описание:

Для форматирования и отображения строки в документе метод bold используется с методами write или writeln.

Смотрите также:

- методы blink, italics, strike.

Метод ceil

Возвращает ближайшее целое числа, округленного в большую сторону или равное числу.

Синтаксис:

```
Math.ceil(number)
```

number любое числовое выражение или свойство существующего объекта.

Метод:

Math

Смотрите также:

- метод floor.

Метод charAt

Возвращает символ указанный в *index*.

Синтаксис:

```
stringName.charAt(index)
```

stringName любая строка или свойство существующего объекта.
index любое целое число от 0 до *stringName*.length-1 или свойство существующего объекта.

Метод:

string

Описание:

Символы в строке индексируются слева направо. Индексом первого символа является 0, индексом последнего символа - *stringName*.length-1. Если вы указали *index* превышающий количество символов в строке, JavaScript возвратит пустую строку.

Смотрите также:

- методы indexOf, lastindexOf.

Метод clearTimeout

Окончание задержки, установленной методом setTimeout.

Синтаксис:

```
clearTimeout(timeoutID)
```

timeoutID задержка, установка которой была возвращена предыдущим вызовом метода setTimeout.

Метод:

frame, window

Описание:

Смотрите описание метода setTimeout

Смотрите также:

- метод setTimeout

Метод click

Имитирует щелчок мыши на выбранном элементе формы.

Синтаксис:

```
1.    buttonName.click()
2.
3.
4.    radioName[index].click()
5.
6.
7.    checkboxName.click()
```

```
8.
9.
```

buttonName любое значение атрибута NAME объектов button, reset или submit или элемент массива*elements*.
radioName значение атрибута NAME объекта radio или элемент массива *elements*.
index целое число, представляющее кнопку radio в объекте radio.
checkboxName любое значение атрибута NAME объекта checkbox или элемент массива *elements*.

Метод:

button, checkbox, radio, reset, submit.

Описание:

Результат действия метода click изменяется в зависимости от вызываемого элемента:

- для button, reset и submit выполняется одинаковое действие - нажатие кнопки.
- для radio - выбор кнопки radio.
- для checkbox - отметка галочкой checkbox и установка значения на on.

Метод close (объект document)

Закрывает поток вывода и завершает вывод данных в рабочую область Navigator'a для отображения.

Синтаксис:

```
document.close()
```

Метод:

document

Описание:

Метод close закрывает поток вывода, открытый методом document.open(). Если поток был открыт для рабочей области Navigator'a, метод close завершает вывод содержимого потока на экран. Таги стиля шрифта, такие как <BIG> и <CENTER>, автоматически закрывают поток вывода. Метод close также останавливает "meteor shower" в иконе Navigator'a и отображает "Document: Done" в строке состояния.

Смотрите также:

- методы open, write, writeln.

Метод close (объект window)

Изменен в Navigator 3.0.
Закрывает указанное окно.

Синтаксис:

```
windowReference.close()
```

windowReference ссылка на окно, как описано в объекте window.

Метод:

window

Описание:

Метод close закрывает указанное окно. Если вы объявляете close без указания *windowReference*, то JavaScript закрывает текущее окно.
В событиях вы должны указывать window.close() вместо обычно используемого close(). Объявление close() без определения имени объекта равносильно document.close().

Смотрите также:

- метод open

Метод confirm

Отображает диалоговое окно с указанным сообщением и кнопками OK и Cancel.

Синтаксис:

```
confirm("message")
```

message любая строка или свойство существующего объекта.

Метод:

window

Описание:

Метод confirm используется для принятия пользователем решения, требующего выбора OK или Cancel. Аргумент *message* определяет сообщение, которое требует решения пользователя. Метод confirm возвращает true, если пользователь выбрал OK, и false, если пользователь выбрал Cancel.

Хотя confirm является методом объекта window, вам не нужно указывать *windowReference* при его вызове. Например, *windowReference*.confirm() является необязательным.

Смотрите также:

- методы alert, prompt

Метод cos

Возвращает косинус числа.

Синтаксис:

```
Math.cos(number)
```

number числовое выражение, представляющее собой размер угла в радианах или свойство существующего объекта.

Метод:

Math

Описание:

Метод cos возвращает числовое значение между -1 и 1, которое представляет собой косинус угла.

Смотрите также:

- методы acos, asin, atan, sin, tan.

Функция escape

Возвращает ASCII значение аргумента, закодированного в ISO Latin-1.

Синтаксис:

```
escape("string")
```

string не буквенно-числовая строка в ISO Latin-1 кодировке или свойство существующего объекта.

Описание:

Функция escape не является методом, связанным с любым объектом, но является частью самого языка.

Значение, возвращаемое функцией escape, является строкой вида "%xx", где *xx* является ASCII кодировкой символа в аргументе. Если аргументом функции escape является буквенно-числовым символом, то функция escape возвращает тот же символ.

Смотрите также:

- функцию unescape.

Функция eval

Функция eval выполняет строку-аргумент и подставляет полученное значение вместо себя.

Синтаксис:

```
eval("string")
```

string любая строка, представляющая собой JavaScript выражение, команду или последовательность команд. Выражение может включать переменные и свойства существующего объекта.

Описание:

Функция eval является встроенной функцией JavaScript. Она не является методом, связанным с любым объектом, но является частью самого языка.
Аргументом функции eval является строка. Не используйте eval для вычислений арифметических выражений. JavaScript вычисляет арифметические выражения автоматически. Если аргумент представляет собой выражение, eval вычисляет выражение. Если аргумент представляет собой одно или более JavaScript команд, то eval выполняет команды.
Если вы построили арифметическое выражение как строку, вы можете использовать eval для ее вычисления.

Метод exp

Возвращает e^{number}, где *number* является аргументом, а *e* является экспонентой, основанием натурального логарифма.

Синтаксис:

```
Math.exp(number)
```

number любое числовое выражение или свойство существующего объекта.

Метод:

Math

Смотрите также:

- методы log, pow.

Метод fixed

Вызывает строку, отображаемую моноширинным шрифтом, как если установить ей таг <TT>.

Синтаксис:

```
stringName.fixed()
```

stringName любая строка или свойство существующего объекта.

Метод:

string

Описание:

Для форматирования и отображения строки в документе метод fixed используется с методами write и writeln.

Метод floor

Возвращает ближайшее целое числа, округленного в меньшую сторону или равное числу.

Синтаксис:

```
Math.floor(number)
```

number любое числовое выражение или свойство существующего объекта.

Метод:

Math

Смотрите также:

- метод ceil.

Метод focus

Изменен в Navigator 3.0.
Устанавливает фокус на определенный объект.

Синтаксис:

```
1. password.focus()
2. select.focus()
3. textName.focus()
4. textareaName.focus()
```

password любое значение атрибута NAME объекта password или элемент массива *elements*.
select любое значение атрибута NAME объекта select или элемент массива *elements*.
textName любое значение атрибута NAME объекта text или элемент массива *elements*.
textareaName любое значение атрибута NAME объекта textarea или элемент массива *elements*.

Метод:

password, select, text, textarea.

Описание:

Метод focus используется для установки фокуса на указанный элемент формы. Вы можете затем программно ввести значение в элемент или позволить пользователю ввести значение.

Смотрите также:

- методы blur, select.

Метод fontcolor

Вызывает строку, отображаемую установленным цветом, как если поместить ее в таг <FONT COLOR=*color*>.

Синтаксис:

```
stringName.fontcolor(color)
```

stringName любая строка или свойство существующего объекта.
color строка или свойство существующего объекта, определяющая цвет как шестиразрядное шестнадцатиричное число (RGB) или как одно из строковых названий в списке Color Value.

Метод:

string

Описание:

Для форматирования и отображения строки в документе метод fontcolor используется с методами write и writeln.
Если вы определяете *color* как шестиразрядное шестнадцатиричное число вы должны использовать формат rrggbb.
Метод fontcolor анулирует значение, установленное в свойстве fgColor.

Метод fontsize

Вызывает строку, отображаемую установленным размером шрифта, как если поместить ее в таг <FONT SIZE=*size*>.

Синтаксис:

```
stringName.fontsize(size)
```

stringName любая строка или свойство существующего объекта.
size целое число от 1 до 7 или строка, представляющая собой целое со знаком (+ или -) от 1 до 7, или свойство существующего объекта.

Описание:

Для форматирования и отображения строки в документе метод fontsize используется с методами write и writeln.
Когда вы определяете *size* как целое, вы устанавливаете размер *stringName* в один из семи специфицированных размеров. Когда вы определяете *size* как "-2", вы устанавливаете размер шрифта*stringName* относительно размера, установленного в таге .

Смотрите также:

- методы big, small.

Метод forward

Загружает следующий URL в списке посещенных URL'ей.

Синтаксис:

```
history.forward()
```

Метод:

history

Описание:

Этот метод выполняет действие равносильное выбору пользователем кнопки Forward в окне Navigator'a. Метод forward также равносилен history.go(1).

Смотрите также:

- методы back, go.

Метод getDate

Возвращает число месяца для указанной даты.

Синтаксис:

```
dateObjectName.getDate()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:

Date

Описание:

Значение, возвращаемое getDate, является целым числом от 1 до 31.

Смотрите также:

- метод setDate

Метод getDay

Возвращает день недели для указанной даты.

Синтаксис:

```
dateObjectName.getDay()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:

Date

Описание:

Значение, возвращаемое getDay, является целым числом, соответствующим дню недели: ноль для воскресенья, один для понедельника, два для вторника и так далее.

Метод getHours

Возвращает часы для указанной даты.

Синтаксис:

```
dateObjectName.getHours()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:

Date

Описание:

Значение, возвращаемое getHours, является целым числом от 0 до 23.

Смотрите также:

- метод setHours.

Метод getMinutes

Возвращает минуты для указанной даты.

Синтаксис:

```
dateObjectName.getMinutes()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:

Date

Описание:

Значение, возвращаемое getMinutes, является целым числом от 0 до 59.

Смотрите также:

- метод setMinutes.

Метод getMonth

Возвращает месяц для указанной даты.

Синтаксис:

```
dateObjectName.getMonth()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:

Date

Описание:

Значение, возвращаемое getMonth, является целым числом от 0 до 11. Ноль соответствует январю, один - февралю и так далее.

Смотрите также:

- метод setMonth.

Метод getSeconds

Возвращает секунды в текущем времени.

Синтаксис:

```
dateObjectName.getSeconds()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:

Date

Описание:

Значение, возвращаемое getSeconds, является целым числом от 0 до 59.

Смотрите также:

- метод setSeconds.

Метод getTime

Возвращает числовое значение, соответствующее времени для указанной даты.

Синтаксис:

```
dateObjectName.getTime()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:

Date

Описание:

Значение, возвращаемое методом getTime, является числом миллисекунд, начиная с 1 января 1970 00:00:00. Вы можете использовать этот метод для назначения даты и времени другому объекту date.

Смотрите также:

- метод setTime.

Метод getTimezoneOffset

Возвращает смещение временной зоны в минутах относительно гринвичского меридиана.

Синтаксис:

```
dateObjectName.getTimezoneOffset()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:

Date

Описание:

Смещение временной зоны является разницей между местным временем и GMT (гринвичским временем). Сезонное время (зимнее, летнее) не дает возможности говорить об этом смещении как о константе.

Метод getYear

Возвращает год для указанной даты.

Синтаксис:

```
dateObjectName.getYear()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:

Date

Описание:

Значение, возвращаемое getYear, равно году минус 1900. Например, если год равен 1976, то возвращаемое значение равно 76.

Смотрите также:

- метод setYear.

Метод go

Загружает URL из списка посещенных URL'ей.

Синтаксис:

```
history.go(delta | location)
```

delta целое число или свойство существующего объекта, представляющее собой относительную позицию в списке посещенных URL'ей.
location строка или свойство существующего объекта, представляющая собой URL или его часть из списка посещенных URL'ей.

Метод:

history

Описание:

Метод go позволяет перейти на адрес, содержащийся в списке посещенных URL'ей, который указан вами в качестве аргумента метода go. Вы можете посмотреть этот список, выбрав History в меню Window. Последние 10 позиций списка также отображаются в меню Go.
Аргумент *delta* может быть положительным и отрицательным числом. Если *delta* больше нуля, то метод go переходит на URL вперед в списке посещенных URL'ей; в противном случае переход осуществляется на URL назад. Если *delta* равна 0, то Navigator перезагружает текущую страницу.
Аргумент *location* является строкой. *location* выбирает для загрузки ближайший адрес в списке посещенных URL'ей, содержащий подстроку *location*, указанную вами в качестве аргумента. Каждая часть URL содержит определенную информацию. Смотрите объект location, где описаны компоненты URL.

Смотрите также:

- методы back, forward.

Метод indexOf

Возвращает индекс позиции впервые встреченного искомого значения в вызванном объекте string. Поиск начинается с *fromIndex*.

синтаксис:

```
stringName.indexOf(searchValue, [fromIndex])
```

stringName любая строка или свойство существующего объекта.
searchValue строка или свойство существующего объекта, представляющая собой искомое значение.
fromIndex место в вызванной строке, с которого начинается поиск. Это может быть любое целое число от 0 до *stringName*.length-1 или свойство существующего объекта.

Метод:

string

Описание:

Символы в строке индексируются слева направо. Индекс первого символа равен 0, индекс последнего -*stringName*.length-1.
Если вы не указываете значение *fromIndex*, JavaScript принимает по умолчанию 0. Если *searchValue* не найден, JavaScript возвращает -1.

Смотрите также:

- методы charAt, lastIndexOf.

Функция isNaN

Изменена в Navigator 3.0.
На UNIX платформах проверяет аргумент, является ли он "NaN" (не числом).

Синтаксис:

```
isNaN(testValue)
```

testValue значение, которое вы хотите проверить.

Описание:
Функция isNaN является встроенной функцией JavaScript. Она не является методом, связанным с любым объектом, но является частью самого языка. Функция isNaN применяется только на UNIX платформах.
На всех платформах, за исключением Windows, функции parseFloat и parseInt возвращают "NaN", когда они принимают нечисловое значение. Значение "NaN" не является числом в любом случае. Вы можете вызывать функцию NaN для того, чтобы определить является ли результат parseFloat или parseInt "NaN". Если над "NaN" совершаются арифметические операции, то их результатами также будет "NaN".
Функция isNaN возвращает true или false.
Смотрите также:
- функции parseFloat, parseInt.

Метод italics
Вызывает строку, отображаемую курсивом, как если установить ей таг <I>.
Синтаксис:
```
stringName.italics()
```
stringName любая строка или свойство существующего объекта.
Метод:
string
Описание:
Для форматирования и отображения строки в документе метод italics используется с методами write или writeln.
Смотрите также:
- методы blink, bold, strike.

Метод lastIndexOf
Возвращает индекс впервые встреченного искомого значения в вызванном объекте string. Поиск по строке осуществляется в обратном направлении, начиная с *fromIndex*.
Синтаксис:
```
stringName.lastindexOf(searchValue, [fromIndex])
```
stringName любая строка или свойство существующего объекта.
searchValue строка или свойство существующего объекта, представляющая собой искомое значение.
fromIndex место в вызванной строке, с которого начинается поиск. Это может быть любое целое число от 0 до *stringName*.length-1 или свойство существующего объекта.
Метод:
string
Описание:
Символы в строке индексируются слева направо. Индекс первого символа равен 0, индекс последнего -*stringName*.length-1.
Если вы не указываете значение *fromIndex*, JavaScript принимает по умолчанию *stringName*.length-1 (конец строки). Если *searchValue* не найден, JavaScript возвращает -1.
Смотрите также:
- методы charAt, IndexOf.

Метод link
Создает гипертекстовую ссылку HTML, по которой можно перейти на другой URL.

Синтаксис:

```
linkText.link(hrefAttribute)
```

Метод:

string

Описание:

Для создания и отображения гипертекстовой ссылки в документе метод link используется с методами write или writeln. Создайте ссылку методом link, затем вызовите write или writeln для отображения ссылки в документе.

В синтаксисе строка *linkText* представляет собой текст, который увидит пользователь. Строка *hrefAttribute*представляет собой атрибут HREF тага <A>, это будет целевой URL. Каждая часть URL содержит определенную информацию. Смотрите объект location, где описаны компоненты URL.

Ссылки, созданные методом link, становятся элементами массива links.

Смотрите также:

- метод anchor.

Метод log

Возвращает натуральный логарифм числа (по основанию *e*).

Синтаксис:

```
Math.log(number)
```

number любое положительное числовое выражение или свойство существующего объекта.

Метод:

Math

Описание:

Если значение *number* находится за пределами диапазона, возвращенное значение всегда будет -1.797693134862316e+308.

Смотрите также:

- методы exp, pow.

Метод max

Возвращает большее число из двух.

Синтаксис:

```
Math.max(number1, number2)
```

number1 и *number2* любые числовые аргументы или свойства существующих объектов.

Метод:

Math

Смотрите также:

- метод min.

Метод min

Возвращает меньшее число из двух.

Синтаксис:

```
Math.min(number1, number2)
```

number1 и *number2* любые числовые аргументы или свойства существующих объектов.

Метод:

Math

Смотрите также:

- метод max.

Метод open (объект document)

Открывает поток для получения вывода методами write и writeln.

Синтаксис:

```
document.open(["mimeType"])
```

mimeType устанавливает любой из следующих типов документа:

```
text/html

text/plain

image/gif

image/jpeg

image/x-bitmap

plug-In
```

plug-In любой составной plug-in MIME тип, поддерживаемый Netscape'ом.

Метод:

document

Описание:

Метод open открывает поток для получения вывода методами write и writeln. Если *mimeType* является текстом или картинкой, то поток открыт в рабочую область Navigator'a; иначе, поток открыт на plug-in. Если документ уже существует в целевом окне, то метод open очищает его.

Для закрытия потока используйте метод document.close(). Метод close вызывает текст или картинку, которые были отправлены в рабочую область Navigator'a для отображения. После использования document.close(), введите document.open() снова, когда вы захотите начать вывод другого потока.

mimeType является необязательным аргументом, определяющим тип документа. Если вы не указываете*mimeType*, то метод open принимает по умолчанию text/html.

Описание *mimeType*:

- *text/html* определяет текст, содержащий ASCII текст в HTML формате.
- *text/plain* определяет текст, содержащий ASCII текст с символами конца строки, для ограничения отображаемых строк.
- *image/gif* определяет документ с закодированными байтами, содержащий GIF заголовок и размеры в пикселях.
- *image/jpeg* определяет документ с закодированными байтами, содержащий JPEG заголовок и размеры в пикселях.
- *image/x-bitmap* определяет документ с закодированными байтами, содержащий bitmap заголовок и размеры в пикселях.
- *plug-in* загружает определенный plug-in и использует его как место назначения для методов write и writeln. Например, "x-world/vrtml" загружет VR Scout VRML plug-in из Chaco Communications, а "aplication/x-director" загружает Macromedia Shockware plug-in.

Смотрите также:

- методы close, write, writeln.

Метод open (объект window)

Открывает новое окно web-броузера.

Синтаксис:

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

windowVar имя нового окна. Эта переменная используется при ссылках на свойства, методы и контейнеры окна.
URL определяет URL, открываемый в новом окне. Смотрите объект location, где описаны компоненты URL.
windowName имя окна, используемое в атрибуте TARGET тага <FORM> или <A>.
windowName может содержать только буквенно-цифровые символы или символ подчеркивания (_).
windowFeatures список через запятую любых из следующих опций или значений:

```
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
```

Вы можете использовать любой набор этих опций. Опции разделяются запятой. Не делайте пробелов между опциями.
pixels положительное целое число, определяющее размеры окна в пикселях.

Метод:

window

Описание:

Метод open открывает новое окно web-броузера клиента, что равносильно выбору New WebBrowser из меню File Navigator'a. Аргумент *URL* определяет URL, содержащийся в новом окне. Если *URL* является пустой строкой, то создастся пустое окно.
В событиях вы должны указывать window.open() вместо обычно используемого open(). Объявление open() без определения имени объекта равносильно document.open().
windowFeatures является необязательным списком перечисленных через запятую опций для нового окна. Булевы опции *windowFeatures* принимают значение true, если они определены без значений, или как yes или 1. Например, open("", "messageWindow", "toolbar") и open("", "messageWindow", "toolbar=1") как в первом, так и во втором случае опция toolbar принимает значение true. Если *windowName* не определяет существующего окна и вы не определяете *windowFeatures*, то все булевы опции *windowFeatures*принимают по умолчанию значение true. Если вы определяете любую из опций *windowFeatures*, то все остальные опции принимают значение false, если вы их не определите дополнительно.
Описание *windowFeatures*:

- *toolbar* создает стандартные рабочие инструменты Navigator'a, с такими кнопками как "Back" и "Forward".

- *location* создает поле ввода Location.
- *directories* создает кнопки стандартных директорий Navigator'a, такие как "What's New" и "What's Coll".
- *status* создает строку состояния внизу окна.
- *menubar* создает меню вверху окна.
- *scrollbars* создает горизонтальную и вертикальную прокрутки, когда документ больше, чем размер окна.
- *resizable* позволяет пользователю изменять размер окна.
- *width* определяет ширину окна в пикселях.
- *height* определяет высоту окна в пикселях.

Смотрите также:

- метод close.

Метод parse

Возвращает количество миллисекунд в строковом представлении даты, начиная с 1 января 1970 00:00:00, по местному времени.

Синтаксис:

```
Date.parse(dateString)
```

Метод:

Date

Описание:

Метод parse выдает дату в строковом представлении (например, "Dec 25, 1995") и возвращает количество миллисекунд, начиная с 1 января 1970 00:00:00 (по местному времени). Эта функция используется для установки значений даты, основанных на строковом значении, например, в сочетании с методом setTime и объектом Date.
Полученная строка представляет собой время, parse возвращает значение времени. Она принимается в стандартном синтаксисе даты IETF: "Mon, 25 Dec 1995 13:30:00 GMT". Она понимает континентальную US временную зону, но в основном, используется временная зона смещения, например "Mon, 25 Dec 1995 13:30:00 GMT+0430" (4 часа, 30 минут западнее Гринвича). Если вы не указали временной зоны, принимается местная временная зона. GMT и UTC считаются эквивалентными.
Так как функция parse является статическим методом Date, вы всегда используете ее как Date.parse(), а не как метод созданного вами объекта date.

Смотрите также:

- метод UTC

Функция parseFloat

Анализирует строковый аргумент и возвращает число с плавающей точкой.

Синтаксис:

```
parseFloat(string)
```

string строка, представляющая собой значение, которое вы хотите проанализировать.

Описание:

Функция parseFloat является встроенным объектом JavaScript. Она не является методом, связанным с любым объектом, но является частью самого языка.
Функция parseFloat анализирует строку-аргумент и возвращает число с плавающей точкой. Если встреченный им символ отличается от знака (+ или -), цифры (0-9), десятичной точки или экспоненты, то он возвращает значение до этой точки, игнорируя этот символ и все

последующие символы.
Если первый символ не может быть конвертирован в число, parseFloat возвращает одно из следующих значений:

- "пусто" на Windows платформах.
- "NaN" на любых других платформах указывает на то, что значение не является числом.

Смотрите также:

- методы isNaN, parseInt.

Функция parseInt

Анализирует строковый аргумент и возвращает целое число, определенное как основание.

Синтаксис:

```
parseInt(string [,radix])
```

string строка, которая представляет собой значение, которое вы хотите проанализировать.
radix целое число, представляющее собой основание, возвращаемого значения.

Описание:

Функция parseFloat является встроенным объектом JavaScript. Она не является методом, связанным с любым объектом, но является частью самого языка.
Функция parseFloat анализирует его первый аргумент-строку и пытается возвратить целое число, определенное как основание. Например, основание 10 означает перевод в десятичное число, 8 - восьмеричное, 16 - шестнадцатиричное, и т.д.
Если parseInt в указанном основании встречает символ, не являющийся числом, то он пропускает его и все следующие символы и возвращает целочисленное значение разобранное до точки. ParseInt усекает числа до целочисленных значений.
Если основание не определено или определено как 0, JavaScript принимает следующее:

- если ввод *string* начинается с "0x", то основание равно 16 (шестнадцатиричное).
- если ввод *string* начинается с "0", то основание равно 8 (восьмиричное).
- если ввод *string* начинается с любого другого значения, то основание равно 10 (десятичное).
- если первый символ не может быть конвертирован в число, parseFloat возвращает одно из следующих значений:
    - "пусто" на Windows платформах.
    - "NaN" на любых других платформах указывает на то, что значение не является числом.

Для арифметических целей значение "NaN" не является числом в любом случае. Вы можете вызвать функцию isNaN для того, чтобы определить является ли результат parseInt "NaN". Если "NaN" применить в арифметических операциях, то их результатами также будут "NaN".

Смотрите также:

- функции isNaN, parseFloat.

Метод pow

Возвращает *base* в степени *exponent*, т.е. $base^{exponent}$.

Синтаксис:

```
Math.pow(base, exponent)
```

base числовое выражение или свойство существующего объекта.
exponent числовое выражение или свойство существующего объекта. Если результат может

оказаться недопустимым значением (например, pow(-1, 0.5), то возвращенное значение равно нулю.

Метод:

Math

Смотрите также:

- методы exp, log.

Метод prompt

Отображает диалоговое окно с сообщением и полем ввода.

Синтаксис:

```
prompt(message, [inputDefault])
```

message любая строка или свойство существующего объекта; строка отображается как сообщение.
inputDefault строка, целое число или свойство существующего объекта, представляющая собой значение вводимое в поле по умолчанию.

Метод:

window

Описание:

Метод prompt используется для отображения диалогового окна, требующего ввода текста пользователем. Если вы не определяете первоначальное значение для *inputDefault*, то диалоговое окно отображает значение <undefined>.
Хотя prompt является методом объекта window, вам не нужно определять *windowReference*, при его вызове. Например, *windowReference*.prompt() является не обязательным.

Смотрите также:

- методы alert, confirm.

Метод random

Изменен в Navigator 3.0.
Возвращает случайное число между нулем и единицей. Этот метод применяется только на UNIX платфомах.

Синтаксис:

```
Math.random()
```

Метод:

Math

Метод setDate

Устанавливает число месяца для указанной даты.

Синтаксис:

```
dateObjectName.setDate(dayValue)
```

dateObjectName любое имя объекта date или свойство существующего объекта.
dayValue целое число от 1 до 31 или свойство существующего объекта, представляющего собой число месяца.

Метод:

Date

Смотрите также:

- метод getDate.

Метод setHours

Устанавливает часы для указанной даты.

Синтаксис:

```
dateObjectName.setHours(hoursValue)
```

dateObjectName любое имя объекта date или свойство существующего объекта.
hoursValue целое число от 0 до 23 или свойство существующего объекта, представляющее собой часы.

Метод:

Date

Смотрите также:

- метод getHours.

Метод setMinutes

Устанавливает минуты для указанной даты.

Синтаксис:

```
dateObjectName.setMinutes(minutesValue)
```

dateObjectName любое имя объекта date или свойство существующего объекта.
minutesValue целое число от 0 до 59 или свойство существующего объекта, представляющее собой минуты.

Метод:

Date

Смотрите также:

- метод getMinutes.

Метод setMonth

Устанавливает месяц для указанной даты.

Синтаксис:

```
dateObjectName.setMonth(month Value)
```

dateObjectName любое имя объекта date или свойство существующего объекта.
monthValue целое число от 0 до 11 (представляющее собой месяцы с января по декабрь) или свойство существующего объекта.

Метод:

Date

Смотрите также:

- метод getMonth.

Метод setSeconds

Устанавливает секунды для указанной даты.

Синтаксис:

```
dateObjectName.setSeconds(secondsValue)
```

dateObjectName любое имя объекта date или свойство существующего объекта.
secondsValue целое число от 0 до 59 или свойство существующего объекта.

Метод:

Date

Смотрите также:

- метод getSeconds.

Метод setTime

Устанавливает значение объекта date.

Синтаксис:

```
dateObjectName.setTime(timevalue)
```

dateObjectName любое имя объекта date или свойство существующего объекта.
timevalue целое число или свойство существующего объекта, представляющее собой количество миллисекунд, начиная с 1 января 1970 00:00:00.

Метод:

Date

Описание:

Метод setTime используется для добавления даты и времени другому объекту.

Смотрите также:

- метод getTime.

Метод setTimeout

Выполняет выражение по истечении установленного количества миллисекунд.

Синтаксис:

```
timeoutID=setTimeout(expression, msec)
```

timeoutID идентификатор, который используется только для окончания выполнения, используя метод clearTimeout.
expression строковое выражение или свойство существующего объекта.
msec числовое значение, числовой ряд или свойство существующего объекта в миллисекундах.

Метод:

frame, window

Описание:

Метод setTimeout выполняет выражение после установленного количества времени. Он не выполняет выражение многократно. Например, если метод setTimeout установлен на 5 секунд, то выражение выполнится через 5 секунд, но не каждые 5 секунд.

Смотрите также:

- метод clearTimeout.

Метод setYear

Устанавливает год для указанной даты.

Синтаксис:

```
dateObjectName.setYear(yearValue)
```

dateObjectName любое имя объекта date или свойство существующего объекта.
timevalue целое число больше чем 1900 или свойство существующего объекта.

Метод:

Date

Смотрите также:

- метод getYear.

Метод sin

Возвращает синус числа.

Синтаксис:

```
Math.sin(number)
```

number числовое выражение или свойство существующего объекта, представляющее собой величину угла в радианах.

Метод:

Math

Описание:

Метод sin возвращает числовое значение между -1 и 1, представляющее собой синус угла.

Смотрите также:

- методы acos, asin, atan, cos, tan.

Метод small

Выводит строку, отображаемую маленьким шрифтом, как если установить ей таг <SMALL>.

Синтаксис:

```
stringName.small()
```

stringName любая строка или свойство существующего объекта.

Метод:

string

Описание:

Для форматирования и отображения строки в документе метод small используется с методами write или writeln.

Смотрите также:

- методы big, fontsize.

Метод sqrt

Возвращает квадратный корень числа.

Синтаксис:

```
Math.sqrt(number)
```

number любое неотрицательное числовое выражение или свойство существующего объекта.

Метод:

Math

Описание:

Если значение *number* находится за пределами данного диапазона, возвращенное значение всегда будет 0.

Метод strike

Выводит строку, отображаемую как перечеркнутый текст, как если установить ей таг <STRIKE>.

Синтаксис:

```
stringName.strike()
```

stringName любая строка или свойство существующего объекта.

Метод:

string

Описание:

Для форматирования и отображения строки в документе метод strike используется с методами write или writeln.

Смотрите также:

- методы blink, bold, italics.

Метод sub

Выводит строку, отображаемую как нижний индекс, как если установить ей таг <SUB>.

Синтаксис:

```
stringName.sub()
```

stringName любая строка или свойство существующего объекта.

Метод:

string

Описание:

Для форматирования и отображения строки в документе метод sub используется с методами write или writeln.

Смотрите также:

- методы sup.

Метод submit

Передает форму.

Синтаксис:

```
formName.submit()
```

formName любая строка или свойство существующего объекта.

Метод:

form

Описание:

Метод submit передает указанную форму. Он выполняет такое же действие как кнопка submit.

Метод submit используется для передачи данных http-серверу. Метод submit возвращает данные, используя методы "get" или "post", определенные в свойстве method.

Смотрите также:

- объект submit.
- свойство onSubmit.

Метод substring

Возвращает подстроку объекта string.

Синтаксис:

```
stringName.substring(indexA, indexB)
```

stringName любая строка или свойство существующего объекта.
indexA любое целое число от 0 до *stringName*.length-1 или свойство существующего объекта.
indexB любое целое число от 0 до *stringName*.length-1 или свойство существующего объекта.

Метод:

string

Описание:

Символы в строке индексируются слева направо. Индекс первого символа равен 0, индекс последнего -*stringName*.length-1.

Если *indexA* меньше чем *indexB*, то метод substring возвращает подстроку, начиная с символа *indexA* и заканчивая символом перед *indexB*. Если *indexA* больше чем *indexB*, то метод

substring возвращает подстроку, начиная с символа *indexB* и заканчивая символом перед *indexA*. Если *indexA* равен *indexB*, то метод substring возвращает пустую строку.

Метод sup

Выводит строку, отображаемую как нижний индекс, как если установить ей таг <SUP>.

Синтаксис:

```
stringName.sup()
```

*stringName*любая строка или свойство существующего объекта.

Метод:

string

Описание:

Для форматирования и отображения строки в документе метод sup используется с методами write или writeln.

Смотрите также:

- методы sub.

Метод tan

Возвращает тангенс числа.

Синтаксис:

```
Math.tan(number)
```

number числовое выражение, представляющее собой величину угла в радианах, или свойство существующего объекта.

Метод:

Math

Описание:

Метод tan возвращает числовое значение, представляющее собой тангенс угла.

Смотрите также:

- методы acos, asin, atan, cos, sin.

Метод toGMTString

Переводит дату в строку, используя среднее гринвичское время (GMT).

Синтаксис:

```
dateObjectName.toGMTString()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:

Date

Описание:

Точный формат значения возвращаемого toGMTString зависит от платформы.

Смотрите также:

- методы toLocaleString.

Метод toLocaleString

Переводит дату в строку, используя местный часовой пояс.

Синтаксис:

```
dateObjectName.toLocaleString()
```

dateObjectName любое имя объекта date или свойство существующего объекта.

Метод:
Date
Описание:
Если вы для перевода даты используете toLocaleString, помните, что различные locales собирают строку в различных путях. Используйте методы getHours, getMinutes, getSeconds для получения более переносимых результатов.
Смотрите также:
- методы toGMTString.

Метод toLowerCase
Возвращает значение вызванной строки, переведенной в нижний регистр.
Синтаксис:
```
stringName.toLowerCase()
```
stringName любая строка или свойство существующего объекта.
Метод:
string
Описание:
Метод toLowerCase возвращает значение *stringName*, переведенное в нижний регистр. toLowerCase не изменяет значения *stringName*.
Смотрите также:
- методы toUpperCase.

Метод toUpperCase
Возвращает значение вызванной строки, переведенной в верхний регистр.
Синтаксис:
```
stringName.toUpperCase()
```
stringName любая строка или свойство существующего объекта.
Метод:
string
Описание:
Метод toUpperCase возвращает значение *stringName*, переведенное в верхний регистр. toUpperCase не изменяет значения *stringName*.
Смотрите также:
- методы toLowerCase.

Функция unescape
Возвращает ASCII строку для указанного значения.
Синтаксис:
```
unescape("string")
```
string строка или свойство существующего объекта, содержащие символы в любой из следующих форм:
- *"%integer"*, где *integer* - число между 0 и 255 (десятичное)
- *"hex"*, где *hex* - число между 0x0 и 0xFF (шестнадцатиричное)

Описание:
Функция unescape не является методом, связанным с каким-либо объектом, но является частью самого языка. Строка, возвращаемая функцией unescape, является рядом символов в ISO Latin-1 кодировке.

Смотрите также:

- функцию escape.

Метод UTC

Возвращает количество миллисекунд в объект date, начиная с 1 января 1970 00:00:00, GMT.

Синтаксис:

```
Date.UTC(year, month, day, [, hrs] [, min] [, sec])
```

year год после 1990.
month месяц между 0-11.
day день месяца между 1-31.
hrs часы между 0-23.
min минуты между 0-59.
sec секунды между 0-59.

Метод:

Date

Описание:

UTC берет параметры даты, разделенные запятой, и возвращает количество миллисекунд, начиная с 1 января 1970 00:00:00, GMT.
Так как UTC является статическим методом Date, используйте его как Date.UTC(), а не как метод созданного вами объекта date.

Смотрите также:

- метод parse.

Метод write

Пишет одно или более HTML выражений в документ в указанном окне.

Синтаксис:

```
document.write(expression1 [,expression2], ... [,expressionN])
```

с *expression1* по *expressionN* любое JavaScript выражение или свойство существующего объекта.

Метод:

document

Описание:

Метод write отображает любое количество выражений в окне документа. Вы можете определить любое JavaScript выражение методом write, включая числовое, строковое или логическое.
Метод write является таким же как метод writeln, но метод write не добавляет символа перевода на новую строку в конец выходной информации.
Метод write используется внутри тага <SCRIPT> или внутри события. События выполняются после закрытия документа, поэтому метод write по умолчанию откроет новый документ с *mimeType* text/html, если вы не укажете метод document.open() в событии.

Смотрите также:

- методы close, open, writeln.

Метод writeln

Пишет одно или более HTML выражений в документ в указанном окне, добавляя символ перевода на новую строку в конец выходной информации.

Синтаксис:

```
document.writeln(expression1 [,expression2], ... [,expressionN])
```

с *expression1* по *expressionN* любое JavaScript выражение или свойство существующего объекта.

Метод:

document

Описание:

Метод writeln отображает любое количество выражений в окне документа. Вы можете определить любое JavaScript выражение методом write, включая числовое, строковое или логическое.

Метод writeln является таким же как метод write, но метод writeln добавляет символ перехода на новую строку в конец выходной информации. HTML игнорирует символ новой строки, за исключением определенных тагов, таких как <PRE>.

Метод writeln используется внутри любого тага <SCRIPT> или внутри события. События выполняются после закрытия документа, поэтому метод writeln по умолчанию откроет новый документ с *mimeType*text/html, если вы не укажете метод document.open() в событии.

Смотрите также:

- методы close, open, write.