

Л8. Операції і структури JavaScript

Навчальні питання:

- 1. Використання вбудованих об'єктів і функцій**
- 2. Об'єктна модель браузера**

Література.

- 1. Флэнаган Д. JavaScript. Подробное руководство.— Пер. с англ.— СПб: СимволПлюс, 2008.— 992 с., ил.**
- 2. Вадим Дунаев Самоучитель JavaScript**

1. Використання вбудованих об'єктів і функцій

Об'єкт String

- Представлення рядків і операцій над ними.
- Створення зразка – `new String(str)`
- Параметр `str` задає рядковий вираз. Екземпляр створюється за умовчанням при такому описі змінною:
- **`var str="Сверкалось... Ськойкие сюды волчились у розвів."`**;
- Цей запис еквівалентний наступним:
- **`var str=new String("Сверкалось... Ськойкие сюды волчились у`**

Властивості об'єкту String

параметр	значення
length	<p>Довжина рядка.</p> <pre>var str= "Біг від н\я"; alert(str.length); // 9</pre>

Методи об'єкту String

метод	опис
<code>charAt(ind)</code>	<p>Повертає символ з вказаної позиції рядка (нумерація з нуля).</p> <pre>var str= new String("абвгд"); var ind=0;</pre>
<code>indexOf(subStr,startInd)</code> <code>indexOf(subStr)</code>	<p>Пошук підрядка <code>subStr</code>, починаючи з позиції <code>startInd</code> і до кінця рядка. Метод повертає позицію, з якою починається підрядок. Якщо <code>startInd</code> не заданий, пошук починається з нульової позиції. Якщо підрядок не знайдений, метод повертає -1.</p>
<code>lastIndexOf(subStr,startInd)</code> <code>lastIndexOf(subStr)</code>	<p>Пошук підрядка <code>subStr</code>, починаючи з позиції <code>startInd</code> і до початку рядка. Метод повертає позицію, з якою починається підрядок. Якщо <code>startInd</code> не заданий, пошук починається з останньої позиції рядка до її початку. Якщо підрядок не знайдений, метод повертає -1.</p>

<p>substring(ind1,ind2) substring(ind1)</p>	<p>Повертає підрядок від позиції ind1 до позиції ind2-1.</p> <p>Якщо ind1 менше нуля, він вважається рівним нулю.</p> <p>Якщо ind2 більше довжини рядка, він вважається рівним довжині рядка.</p> <p>Якщо ind1 рівний ind2, повертається порожній рядок.</p> <p>Якщо ind2 опущений, повертається кінець рядка, починаючи з позиції ind1.</p>
<p>toLowerCase()</p>	<p>Повертає рядок, перетворений до нижнього регістра (початковий рядок ніяк не міняється).</p>
<p>toUpperCase()</p>	<p>Повертає рядок, перетворений до верхнього регістра (початковий рядок ніяк не міняється).</p>

Об'єкт Math

- Математичні функції і константи.
- Об'єкт існує в єдиному екземплярі.
Програміст не може створити свій власний екземпляр цього об'єкту.

Властивості об'єкту Math

параметр	значення
E	Підстава натурального логарифма (константа Ейлера).
LN10	Натуральний логарифм числа 10.
LN2	Натуральний логарифм числа 2.
LOG10E	Логарифм числа E по підставі 10.
LOG2E	Логарифм числа E по підставі 2.
PI	Число пі.
SQRT1_2	Квадратний корінь з 1/2.
SQRT2	Квадратний корінь з 2.

Методи об'єкту Math

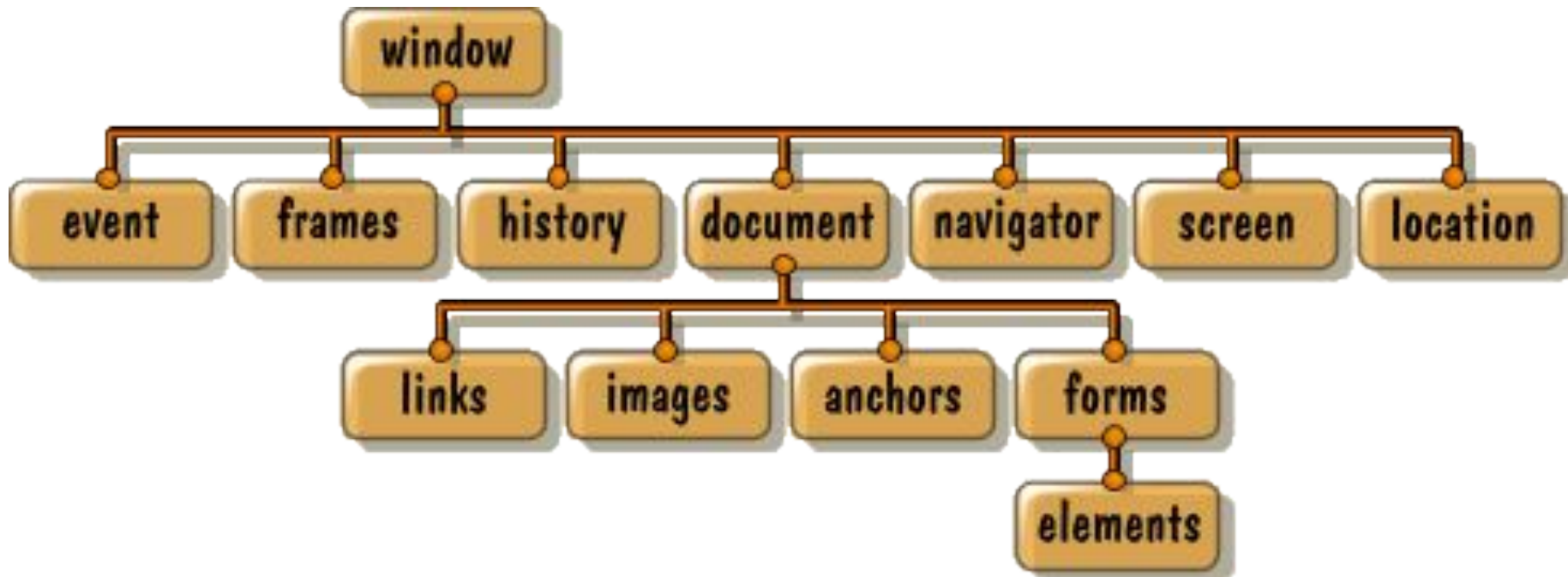
метод	опис
abs(arg)	Повертає абсолютне значення аргументу.
sin(arg) cos(arg) tan(arg)	Стандартні тригонометричні функції, аргумент указується в радіанах.
asin(arg) acos(arg) atan(arg)	Зворотні тригонометричні функції, повертають значення в радіанах.
exp(arg) log(arg)	Експонента і натуральний логарифм
ceil(arg)	Повертає найближче ціле число, більше або рівніше аргументу.
floor(arg)	Повертає найближче ціле число, менше або рівніше аргументу.

$\min(\text{arg1}, \text{arg2})$ $\max(\text{arg1}, \text{arg2})$	Мінімальне (максимальне) значення з двох числових аргументів.
$\text{pow}(\text{arg1}, \text{arg2})$	Повертає arg1 в ступені arg2 .
$\text{random}()$	Повертає випадкове число з діапазону $[0, 1]$.
$\text{round}(\text{arg})$	Округляє аргумент до найближчого цілого.
$\text{sqrt}(\text{arg})$	Корінь квадратний з аргументу.

2. Об'єктна модель браузера

Було сказано, що використовуючи JavaScript, можна перейти до технологій динамічного HTML. Для того, щоб змінювати вже намальований браузером екран або створювати нові вікна, потрібно проникнути в ієрархію об'єктів браузера, дістати доступ до його об'єктів. JavaScript надає таку можливість. У чому суть динамічного HTML-програмування? Браузер не тільки показує образи тегов на екрані комп'ютера. Браузер для кожного тега будує об'єкт, наділяючи його відповідними властивостями і методами. Всі ці об'єкти поміщаються в струнку ієрархію, яка називається об'єктною моделлю браузера. Доступ до об'єктної моделі можна виконати на JavaScript. Це дає наступні переваги: браузер постійно стежить за своєю об'єктною моделлю; якщо в ній щось міняється, ця зміна відразу відображається на екрані. Ось в цьому і є суть програмування динаміки на гіпертекстовій сторінці.

З точки програміста зору браузер є наступною ієрархією об'єктів:



Об'єкт window

Об'єкт window описує поточне вікно браузера і його вміст.

- Властивості об'єкту

параметр	значення
self	Властивість self є покажчиком на поточне вікно. Приклад використання цієї властивості показаний в розділі переміщення фокусу.
opener	Властивість opener є покажчиком на вікно батька. Використовуючи цей покажчик, можна в створеному вікні працювати зі всім тим, що належить "батьку", наприклад, використовувати "батьківські" скриптові функції і змінні.
status defaultStatus	Властивості status і defaultStatus містять відповідно тимчасовий і постійний вміст статусного рядка. Тимчасовий вміст міняється при наїзді мишачого курсора на посилання або при видачі браузером діагностичних повідомлень.

<code>dialogArguments</code>	Властивість <code>dialogArguments</code> повертає аргументи, що проходять через діалогове вікно, як масив. Дивитися приклад до методу <code>showModalDialog</code> .
<code>returnValue</code>	Властивість <code>returnValue</code> визначає значення, що повертається з вікна. Дивитися приклад до методу <code>showModalDialog</code> .

Методи об'єкту

метод	опис
open	Відкриває нове вікно браузера.
close	Закриває вікно браузера.
alert, prompt, confirm	Стандартні діалогові панелі.
showModalDialog	Відображає нове вікно як модальну діалогову панель.
blur	Відводить фокус з вікна.
focus	Перекладає фокус на вікно.
scroll	Показує документ у вікні із заданими абсолютними зсувами від його початку.

<code>scrollBy</code>	Переміщає документ у вікні на задані величини по відношенню до поточного положення.
<code>setInterval</code>	Указує функції виконуватися періодично через задану кількість мілісекунд.
<code>clearInterval</code>	Відмінняє дію методу <code>setInterval</code> .
<code>setTimeout</code>	Запускає функцію через задану кількість мілісекунд.
<code>clearTimeout</code>	Відмінняє дію методу <code>setTimeout</code> .

Об'єкт document

Об'єктом document є модель документа, побудованого браузером на екрані.

параметр	значення
title	Назва документа, визначена в теґе TITLE. У ІЕ цю властивість можна читати і міняти, в NN -- тільки читати.


```
<SCRIPT language="javascript">
<!--
  var oldTitle=document.title;
//-->
</SCRIPT>
<FORM>
  <INPUT type=button value=title
    onclick="alert(document.title) "><BR>
  <INPUT name=t type=text value="Новое назва">
  <INPUT type=button value="Изменить"
    onclick="document.title=this.form.t.value">
  <INPUT type=button value="Восстановить"
    onclick="document.title=oldTitle">
</FORM>
```

URL

Адреса сторінки. У ІЕ цю властивість можна читати і міняти, в NN -- тільки читати.

```
<FORM>
```

```
<INPUT type=button value="URL"
      onclick="alert(document.URL)
"><BR>
```

```
<INPUT name=u type="text"
value="00.htm">
```

```
<INPUT type=button value="ІЗМЕНИТЬ"
onclick="document.URL=this.form.u.value"
>
</FORM>
```

location

Адреса сторінки. Цю властивість можна читати і міняти як в ІЕ, так і в NN.

```
<FORM>
```

```
<INPUT type=button value="location"  
  onclick="alert(document.URL) "><BR>
```

```
<INPUT name=u type="text"  
value="00.htm">
```

```
<INPUT type=button value="Изменить"
```

```
onclick="document.location=this.form.u.val  
ue">
```

```
</FORM>
```

lastModified

Дата останньої зміни документа.

```
<DIV align=center>
```

```
<P>Этот документ мінявся останній раз:
```

```
<P>
```

```
<STRONG><FONT color=#FF4900>
```

```
<SCRIPT language="javascript">
```

```
<!--
```

```
    document.write(document.lastModified);
```

```
//-->
```

```
</SCRIPT>
```

```
</FONT>></STRONG>
```

```
</DIV>
```

<p>bgColor fgColor linkColor alinkColor vlinkColor</p>	<p>Кольори фону, тексту, не відвіданого посилання, активного посилання і відвіданого посилання (аналоги атрибутів bgcolor, text, link, alink, vlink тега BODY). Ці властивості доступні для читання і запису.</p>
<p>readyState</p>	<p>Повертає значення complete після повного завантаження документа. Властивість підтримується тільки в ІЕ.</p> <pre><FORM> <INPUT type=button value="readyState" onclick="alert(document.readyState) "> </FORM></pre>

Методи об'єкту

метод	опис
<code>open()</code>	Відкриває запис у вікно браузера. Колишній вміст вікна очищається.
<code>close()</code>	Закриває запис у вікно браузера.
<code>clear()</code>	Очищення вікна браузера.
<code>write(str)</code>	Записує текст і код HTML, що міститься в рядку <code>str</code> в документ.
<code>writeln(str)</code>	Записує текст і код HTML, що закінчується поверненням каретки (переходом на новий рядок). Перехід на новий рядок на екрані браузера буде помітний лише в тому випадку, якщо він виконується усередині тега <code>PRE</code> .

Напишемо функцію NR, яка виводитиме в документ незвичайну горизонтальну смужку:

```
<SCRIPT language="javascript">
```

```
<!--
```

```
    // Переклад числа num в 16-ричну систему  
    счисления
```

```
function ToHex(num)
```

```
{
```

```
    var ret="";
```

```
    var s = "0123456789ABCDEF";
```

```
    while(num)
```

```
{
```

```
    ret = s.charAt(num%16)+ret;
```

```
    num = Math.floor(num/16);
```

```
}
```

```
    return !ret ? "0":ret;
```

```
}
```

```
// Формування коду кольору у вигляді:#rrggbb
```

```
function RGB(r,g,b)
```

```
{  
  r=ToHex(r); if (r.length<2) r = "0"+r;  
  g=ToHex(g); if (g.length<2) g = "0"+g;  
  b=ToHex(b); if (b.length<2) b = "0"+b;  
  return "#"+r+g+b;  
}
```

```
// Виведення горизонтальної смуги з растяжкою сірого  
кольору
```

```
// win -- посилання на вікно, в яке виконується вивід
```

```
function HR(win)
```

```
{  
  var str("<TABLE border=0 cellspacing=0 cellpadding=0><TR>";  
  for(var i=80; i<240; i+=4)  
    str += "<TD width=12 bgcolor="+RGB(i,i,i)+">&nbsp;  </TD>";  
  str += "</TR></TABLE>"  
  win.document.write(str);  
}
```

```
HR(this);
```

```
//-->
```

```
</SCRIPT>
```



Колекції об'єкту document

Колекція -- це структура, схожа на масив. Елементи колекції є об'єктами і доступні як елементи масиву по індексу. Індксація традиційно починається з нуля. Як індекс можна використовувати ім'я об'єкту, якщо воно задане у відповідному теге за допомогою атрибуту name (або id). Наприклад, доступ до об'єкту, побудованого браузером для тега може бути отриманий за допомогою конструкції `document.images[0]`, якщо це перша картинка в HTML-файле або за допомогою конструкції `document.images["pic"]` незалежно від порядкового номера картини.

колекція	опис
all	<p>Колекція всіх тегов в тілі документа. За допомогою наступного скрипта і форми, можна міняти екранне зображення:</p> <pre data-bbox="440 535 1825 1320"><SCRIPT language=JavaScript><!-- var img = new Array (new Image(), new Image()); img[0].src="./pic/isp1a0.gif"; img[1].src="./pic/isp2a0.gif"; var ind=1; function ChangeImg() { document.all["pic0"].src=img[ind].src; ind=1-ind; }//--></SCRIPT><FORM> <INPUT type=button value=Изменить onclick="ChangeImg()"> </FORM></pre>

forms

Колекція всіх форм на сторінці.

Ця колекція сама містить колекцію elements -- список елементів даної форми (теги INPUT, SELECT, TEXTAREA).

Для доступу до форми з ім'ям forma використовується конструкція document.forms["forma"]. Для доступу до елемента форми з ім'ям element -- конструкція document.forms["forma"].elements["element"].

За допомогою наступного скрипта і форми, можна міняти вміст поля:

```
<SCRIPT language=JavaScript>
<!-- var titl = new Array( "Пригода Незнайка і його друзів"
"Незнайко в сонячному місті" "Незнайко на Місяці");
var ind=0; function ChangeTitl() {
document.forms["f"].elements["t"].value=  titl[ind];
ind = ind < 2 ? ++ind : 0; }//-->
</SCRIPT>
<FORM> <INPUT type=button value=Змінить
onclick="ChangeTitl()"> </FORM>
```

images

Колекція всіх картинок на сторінці.

**За допомогою наступного скрипта і форми,
можна міняти екранне зображення:**

```
<SCRIPT language=JavaScript>
```

```
<!-- var img = new Array (new Image(),  
new Image());
```

```
img[0].src="./pic/isp1a0.gif";
```

```
img[1].src="./pic/isp2a0.gif";
```

```
var ind=1;
```

```
function ChangeImg()
```

```
{ document.images["pic1"].src=img[ind].src;
```

```
ind=1-ind; }!-->
```

```
</SCRIPT>
```

```
<FORM> <INPUT type=button value=ІЗМЕНИТЬ  
onclick="ChangeImg()"> </FORM>
```

links

Колекція всіх посилань на сторінці.

Список всіх посилань на цій сторінці можна отримати за допомогою коду:

```
<SCRIPT language=JavaScript>
```

```
<!--function LinksList()
```

```
{ var str = "Список посилань цієї сторінки:\n ";
```

```
for(var i=0; i<document.links.length; i++)
```

```
str +=document.links[i].href+"\n";
```

```
  alert(str);}//-->
```

```
</SCRIPT>
```

```
<FORM> <INPUT type=button value="Ссылки"
```

```
onclick="LinksList()"> </FORM>
```

Об'єкт event

- Об'єкт event дозволяє скрипту отримати детальну інформацію про подію, що відбулася, і виконати необхідні дії.
- Об'єкт event доступний тільки під час самої події. Більш того, звертатися до нього можна тільки з обробників подій (функцій).

Властивості об'єкту

властивість	значення
srcElement	Дозволяє дізнатися джерело події. Ця властивість є об'єктом і само містить величезне число властивостей, що описують джерело, на якому відбулася подія. Наприклад: srcElement.tagName -- ім'я тега srcElement.type -- значення атрибуту type
type	Рядок, що містить тип події. <FORM> <INPUT type="button" value="type" onclick="alert(event.type) "> </FORM>
clientX clientY	Координати події відносно почала вікна, в якому браузер показує документ.
screenX screenY	Координати події відносно початку екрану монітора.

offsetX offsetY	Координати події відносно тега, на якому відбулася подія.
x y	Координати події відносно початку документа у вікні, в якому браузер показує документ. Початок документа може бути істотно "вище" за верхню межу вікна і "лівіше" за його ліву межу, якщо працюють лінійки протяжки.
button	Число, вказуюче натиснуту кнопку миші. <FORM> <INPUT type="button" value="button" onMouseDown="alert(event.button) "> </FORM>
keyCode	Код натиснутої клавіші.

<p>altKey ctrlKey shiftKey</p>	<p>Код натиснутої клавіші.</p> <pre><FORM> <INPUT type=button value="Спецклавиши" onKeyDown="this.form.Alt.value=event.altKey; this.form.Ctrl.value=event.ctrlKey; this.form.Shift.value=event.shiftKey;" onKeyUp ="this.form.Alt.value="; this.form.Ctrl.value="; this.form.Shift.value=";"> Alt<INPUT name=Alt type=text value="" size=7> Ctrl<INPUT name=Ctrl type=text value="" size=7> Shift<INPUT name=Shift type=text value="" size=7> </FORM></pre>
<p>returnValue IE4</p>	<p>Дозволяє заборонити або вирішити дію, наказану елементу.</p> <pre> >Перехід на іншій документ</pre>

Об'єкт history

- Об'єкт history містить інформацію про адреси сторінок, які відвідувалися в поточній сесії браузера. Ця інформація зберігається в списку History, довжина якого міститься у властивості length.

параметр	значення
length	Властивість length містить число елементів списку History, тобто число різних сторінок, які завантажував браузер в своє вікно під час поточного сеансу.

Методи об'єкту

метод	опис
<code>back()</code>	Завантажує попередню сторінку із списку History.
<code>forward()</code>	Завантажує наступну сторінку із списку History.
<code>go(n)</code>	<p>Завантажує n-ю сторінку із списку History по відношенню до поточної. Ціле число n може бути позитивним, негативним або нулем:</p> <p>$n < 0$ -- завантажуються попередні сторінки з History;</p> <p>$n = 0$ -- перевантажується поточна сторінка;</p> <p>$n > 0$ -- завантажуються наступні сторінки з History</p> <p>Наприклад, <code>history.go(-1)</code> еквівалентно <code>history.back()</code>, а <code>history.go(1)</code> еквівалентно <code>history.forward()</code>.</p>

Об'єкт location

Об'єкт location містить інформацію про адресу поточної сторінки і забезпечує методи для її перезавантаження або завантаження нової сторінки.

властивості об'єкту

Нижче приводиться повний список властивостей об'єкту location для цієї сторіночки:

hostname = www.bgsha.com

href =

http://www.bgsha.com/ru/academy/cit/textbooks/web/java_script/location.htm

host = www.bgsha.com

hash =

port =

pathname =

/ru/academy/cit/textbooks/web/java_script/location.htm

search =

protocol = http:

Властивості об'єкту

властивість	опис
href	Повна URL-адрес. Властивість доступна для читання і запису. При записі нової адреси, браузер завантажує у вікно нову сторінку.
hash	Рядок, наступний в URL за символом "#" (локальна мітка у файлі).
host	Містить частину URL адреси: "hostname:port".
hostname	Містить частину URL адреси: "hostname".
pathname	Містить частину URL адреси -- ім'я файлу (шлях до файлу після другого знаку "/" в повному URL).
port	Містить частину URL адреси: номер порту.
protocol	Містить початкову частину URL адреси - протокол (спосіб завантаження поточної сторінки).
search	Містить частину повної адреси URL - рядок запиту, наступний за символом "?".

Методи об'єкту

метод	опис
assign(URL)	<p>Завантажує сторінку із заданим URL. Дія методу повністю еквівалентна привласненню <code>location.href=URL</code>.</p> <p>Приклади</p> <pre><FORM> <INPUT type=button value=location.assign onclick="location.assign('ind.htm')"> </FORM></pre>
replace(URL)	<p>Заміщає поточну сторінку із заданим URL. Запис в списку History, відповідна поточної сторінки, заміщається на нову. Тепер повернутися до поточної сторінки по кнопці "Назад" браузера буде не можна.</p>
reload()	<p>Перенавантажує поточну сторінку. Цей метод можна використовувати з аргументом <code>true</code> і тоді браузер обов'язково перенавантажує сторінку з сервера. Якщо ж параметр відсутній або рівний <code>false</code>, браузер сам вирішує звідки перенавантажувати сторінку -- з кеша або сервера.</p>

Об'єкт navigator

- Об'єкт navigator містить інформацію про браузер, в якому виконується програма.

Властивості об'єкту

властивість	опис
appName	<p>Містить назву браузера. Для основних браузерів це: Microsoft Internet Explorer Netscape</p> <pre><FORM> <INPUT type="button" value="appName" onclick="alert(navigator.appName) "> </FORM></pre>
appVersion	<p>Дана властивість дозволяє визначити версію браузера.</p> <pre><FORM> <INPUT type="button" value="appVersion" onclick="alert(navigator.appVersion) "> </FORM></pre>

Визначення типу браузера

```
<SCRIPT language="javascript">
<!--
function defBrowser()
{
    var ver = parseFloat(navigator.appVersion);
    var nav = navigator.appName;
    if (nav == "Netscape")           ver = -ver;
    else if(nav != "Microsoft Internet Explorer") ver = 0;
    return ver;
}
//-->
</SCRIPT>
```

Події браузера

подія	опис
onLoad	Подія onLoad відбувається, коли браузер закінчує завантажувати документ у вікно. Цей атрибут можна використовувати в теґе BODY.
onUnload	Подія onUnload відбувається, коли користувач закриває поточний документ у вікні браузера (переходить по посиланню, по кнопках браузера Вперед/Назад, закриває браузер). Цей атрибут можна використовувати в теґе BODY.
onClick	Подія onClick відбувається при одноразовому клацанні миші по елементу форми або гіпертекстовому посиланню.
onDbClick	Подія onDbClick відбувається при подвійному клацанні миші по елементу форми або гіпертекстовому посиланню.

<code>onMouseOver</code> <code>onMouseOut</code>	Події <code>onMouseOver</code> і <code>onMouseOut</code> відбуваються коли мишачий курсор переміщається на елемент або відповідно йде за його межі.
<code>onMouseDown</code> <code>onMouseUp</code>	Події <code>onMouseDown</code> і <code>onMouseUp</code> відбуваються при натисненні кнопки миші на елементі або відповідно при відпуску кнопки.
<code>onKeyPress</code>	Подія <code>onKeyPress</code> відбувається при натисненні клавіші на клавіатурі і відпуску її.

onKeyDown і onKeyUp	Події onKeyDown і onKeyUp відбуваються при натисненні клавіші на клавіатурі і відповідно відпуску її.
onChange	Подія onChange відбувається при втраті елементом фокусу введення, якщо його значення було змінене з моменту отримання фокусу. Цей атрибут може використовуватися в тегах INPUT, SELECT і TEXTAREA.
onFocus onBlur	Події onFocus і onBlur відбуваються коли елемент отримує або відповідно втрачає фокус введення.
onReset	Подія onReset відбувається при очищенні форми. Воно використовується тільки як атрибут тега FORM.
onSubmit	Подія onSubmit відбувається при відправці форми. Воно використовується тільки як атрибут тега FORM.