

The JavaScript Programming Language

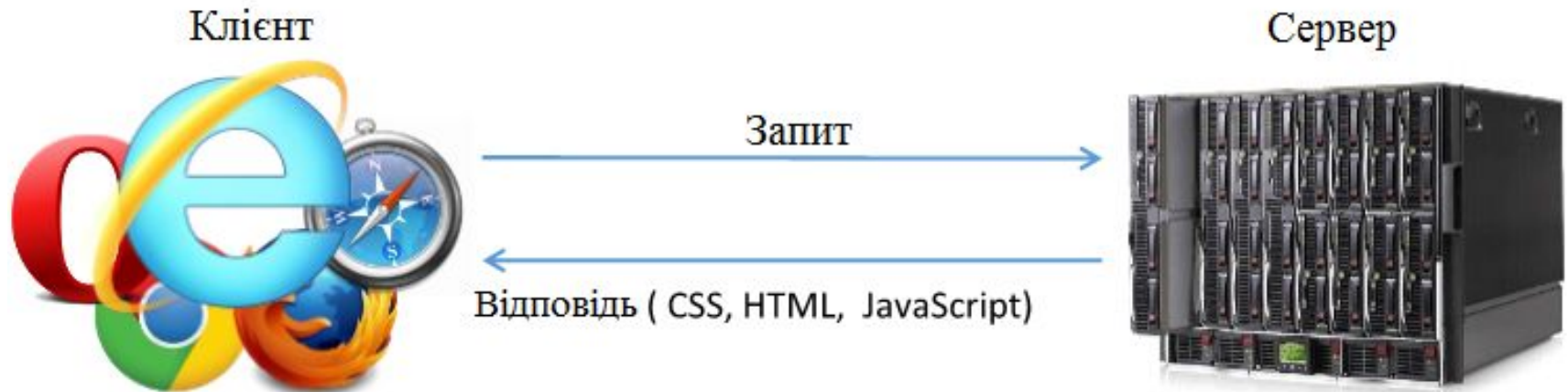
JavaScript – прототипно-орієнтована скриптова мова програмування.

Відповідно до стандарту ECMA-262 мова офіційно називається ECMAScript.

JavaScript – це не Java, мова Java розроблена компанією Sun Microsystems, JavaScript компанією Netscape та крім деякої схожості по синтаксису ці дві мови нічого між собою не пов'язує.

Мова JavaScript по синтаксису нагадує C, C++ та Java програмними конструкціями, такими як: цикли, умовні конструкції і оператори. Однак ця подібність обмежується синтаксичною схожістю.

Клієнтський JavaScript



Інтерпретатор JavaScript, яким оснащені браузери, дозволяє виконувати клієнтські сценарії завантажені разом з розміткою сторінки.

Роль клієнтського JavaScript

- Користувацький інтерфейс
 - Створення візуальних ефектів (анімація).
 - Виконання нескладних обчислень.
 - Перевірка даних введених користувачем.
 - Маніпуляція даними введеними користувачем у форми.
 - Пошук по даним, вбудованим в завантажену сторінку.
 - Збереження даних в cookies.
- Динамічна зміна вмісту сторінки (DHTML).
- Пряма взаємодія з сервером без перезавантаження всієї сторінки (AJAX).

Приклад клієнтського

JavaScript

JavaScript код виконується на сторони клієнта Web-браузером.

```
<html>
<head>
  <title>Клієнтський JavaScript</title>
  <script>
    for(var i = 1; i < 10; i++) {
      document.write(i + "<br/>");
    }
  </script>
</head>
<body>

</body>
</html>
```

1
2
3
4
5
6
7
8
9

Lexical structure

Лексична структура мови програмування – це набір елементарних правил, що визначають, як пишуться програми на мові.

Лексична структура JavaScript:

- ✓ Чутлива до регістру.
- ✓ Інструкції повинні закінчуватися крапкою з комою або починатися з нового рядка.
- ✓ Ігнорує пробіли і табуляції.

Comments

```
<!--Тег script-->
```

```
<script>
```

```
// Вы можете залишати як однорядковий коментар
```

```
alert("Hello world!");
```

```
/*
```

```
Так і багаторядковий
```

```
коментар
```

```
*/
```

```
</script>
```

```
<!--Завершення тега script-->
```

Зарезервовані слова

break

case

catch

continue

default

delete

do

else

false

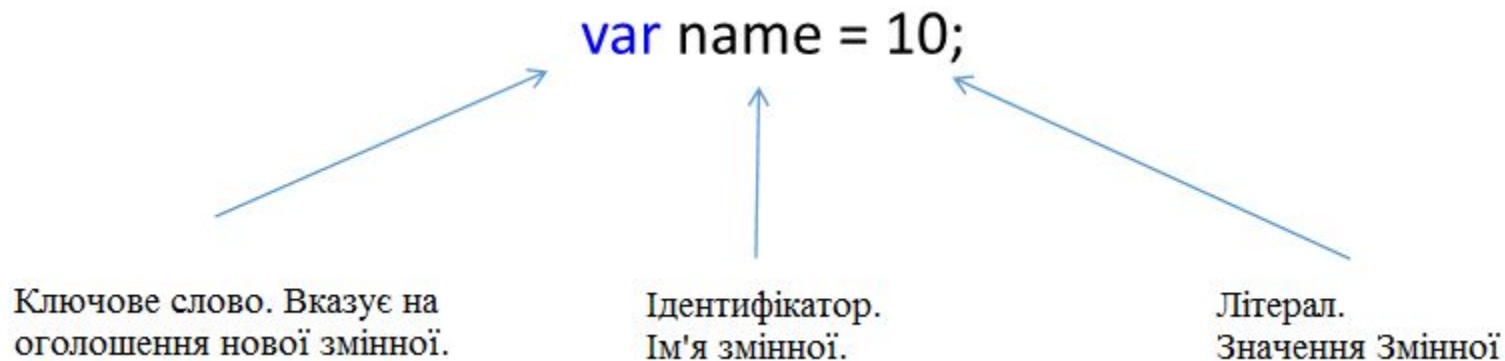
finally

for

function

Створення змінних

Змінна – це символ або слово, що означає деяке значення.



JavaScript – динамічно типізована, а не строго типізована мова програмування.

Правила іменування змінних

Ідентифікатор повинен бути одним словом:



myName



My Name

Ім'я не може починатися з цифри:



element21



21element

Першим символом імені може бути символ «\$» або «_»:



\$str



_name

При іменуванні часто використовуються правила **camelCase** або **PascalCase**

Data Types

Елементарні типи

- Числа (`Number`)
- Рядки (`String`)
- Логічні значення (`Boolean`)

Тривіальні типи

- `null` – відсутність значення
- `undefined` – невизначене значення

Складені типи

- Об'єкт (`Object`)
- Масив (`Array`)

Спеціальний тип

- Функція (`function`)

Літерали

Літерали – це найпростіші дані з якими може працювати програма.

Літерали цілого типу – цілі числа в поданні:

- ✓ десятичному, наприклад: 15, +5, -174.
- ✓ шістнадцятиричному, наприклад: 0x25, 0xff. Шістнадцятиричні числа включають цифри 0 - 9 і букви a, b, c, d, e, f. Записуються вони з символами 0x перед числом.
- ✓ вісімковому, наприклад: 011, 0543. Вісімкові числа включають тільки цифри 0 - 7.

Речові літерали – дробові числа. Ціла частина відділяється від дробової крапкою, наприклад: 99.15, -32.45. Можлива експоненціальна форма запису, наприклад: 2.73e-7. У звичному вигляді це 2.73X10-7, але в javascript знак множення і 10 замінюються символом -e-.

Логічні значення – є два: істина (true) і брехня (false).

Рядкові літерали – послідовність символів, укладена в одинарні або подвійні лапки. Наприклад: "ваше ім'я", "ваше ім'я".

Data Types

```
var a = 10; // тип Number
var b = 0.8; // тип Number
var c = "string"; // тип String
var d = 'string'; // тип String
var e = true; // тип Boolean
var f = false; // тип Boolean
var g = null; // тип null
var h = undefined; // тип undefined
var i = new Date(); // тип Object
var j = new Object(); // тип Object
var k = [1,2,3]; // Масив
var l = Array(); // Масив
var m = function() { }; // Функція
```

Змінні

Наприклад:

```
var test;  
var _test;  
var _my_test1;
```

Наприклад:

```
var a=15;  
var b=23.15;  
var c='Виконано';  
var s=true;
```

Кожна змінна має тип, який визначається значенням змінної. Так в нашому прикладі: змінні **a** і **b** мають тип **number**, змінна **c** має тип **string**, а змінна **s** – логічний тип.

Вирази

операція	назва
+	додавання
-	віднімання
*	множення
/	ділення
%	остача від ділення цілих чисел
++	збільшення значення операнда на одиницю
--	зменшення значення операнда на одиницю

Вирази

оператор	еквівалентний оператор присвоєння
$X+=Y$	$X=X+Y$
$X-=Y$	$X=X-Y$
$X*=Y$	$X=X*Y$
$X/=Y$	$X=X/Y$
$X\%=Y$	$X=X\%Y$

Присвоєння з операцією

Оператор	Приклад	Еквівалент
+=	a+=b	a=a+b
-=	a-= b	a=a-b
=	a= b	a=a*b
/=	a/=b	a=a/b
%=	a%= b	a=a%b
<<=	a<<=b	a=a<<b
>>=	a>>=b	a=a>>b
>>>=	a>>>=b	a=a>>>b
&=	a&=b	a=a&b
 =	a =b	a=a b
^=	a^=b	a=a^b

Порівняння двох значень

Результатом порівняння може бути тільки логічне значення: true або false:

операція	назва
<	менше
<=	менше або дорівнює
==	дорівнює
!=	не дорівнює
>=	більше або дорівнює
>	більше

Логічні операції:

&& - логічне І (AND),

|| - логічне АБО (OR),

! - логічне НЕ (NOT).

A	B	A&&B	A B	!A
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

Конкатенація рядків

```
var st1="Вітаю ";  
var st2="студент";  
var st3=st1+st2;
```

В результаті в змінній **st3** буде значення "Вітаю студент".

Оператор рівності та ідентичності

Рівність (==)

(1==1) = true

('text'=='text') = true

(null==undefined) = true

(1=='1') = true

(1==true) = true

(0==false) = true

Ідентичність (===)

(1===1) = true

('text'===text') = true

(null===undefined) = false

(1===1) = false

(1===true) = false

(0===false) = false

Таблиця пріоритетів операцій в порядку їх спадання

назва	позначення
інкремент	++
декремент	--
заперечення	!
унарний мінус	-
множення	*
ділення, остача від ділення	/,%
додавання	+
віднімання	-
порівняння	<, >, <=, >=
рівність	==
не рівність	!=
логічне І	&&
логічне АБО	
присвоєння	=, +=, -=, *=, /=, %=, !=

Javascript-код на сторінці

```
<html>  
  <head>  
    <title>Заголовок документа</title>  
    <link rel="stylesheet" href="style.css">  
    <script>  
    </script>  
  </head>  
  <body>  
  </body>  
</html>
```

JavaScript-код у зовнішньому файлі

```
<html>  
  <head>  
    <title>Заголовок документа</title>  
    <link rel="stylesheet" href="style.css">  
    <script src="script.js"></script>  
  </head>  
  <body>  
  </body>  
</html>
```

Наша перша програма

```
<html>
  <head>
    <title>Заголовок документа</title>
    <link rel="stylesheet" href="style.css">
    <script>                                document.write("Мій перший
Javascript-код.");
    </script>
  </head>
  <body>
  </body>
</html>
```


Обробка подій

```
<div onClick="addText();"></div>
```

Де *Click* – подія (клікання по div-у), *onClick* – обробник події, *addText()* – і'мя функції, яка спрацьовує при виникненні цієї події (кліканні по div-у).

Обробка подій

подія	коли виникає	обробник події
Blur	втрата об'єктом фокуса	onBlur
Change	користувач змінює значення елемента	onChange
Click	користувач клікає мишкою по об'єкту	onClick
DbClick	користувач робить подвійне клацання мишею по об'єкту	onDbClick
DragDrop	користувач перетягує мишею об'єкт	onDragDrop
Error	виникнення javascript-помилки	onError
Focus	вікно або елемент форми отримує фокус	onFocus
KeyDown	користувач натискає клавішу клавіатури	onKeyDown
KeyPress	користувач утримує клавішу клавіатури	onKeyPress
KeyUp	користувач відпускає клавішу клавіатури	onKeyUp

Обробка подій

подія	коли виникає	обробник події
Load	документ завантажується в браузер	onLoad
MouseDown	користувач натискає кнопку миші	onMouseDown
MouseOut	показчик миші виходить за межі елемента	onMouseOut
MouseOver	показчик миші поміщається над елементом	onMouseOver
MouseUp	користувач відпускає кнопку миші	onMouseUp
Move	користувач переміщує вікно	onMove
Reset	користувач натискає кнопку "reset" форми	onReset
Resize	користувач змінює розміри вікна або елемента	onResize
Select	користувач вибирає елемент форми	onSelect
Submit	користувач натискає кнопку "submit" форми	onSubmit

Функції для роботи з діалоговими вікнами:

alert() – Виводить модальне вікно з повідомленням.

Відвідувач не зможе продовжити роботу, поки не натисне на кнопку "ОК" в модальному вікні.

Приклад: `alert('Добрий день');`

prompt() – Виводить повідомлення у вікні з текстовим полем і двома кнопками: "ОК" і "СКАСУВАННЯ".

Повертає введені значення або null, якщо відвідувач натиснув на кнопку "СКАСУВАННЯ".

Як і в `alert()`, вікно – модальне, тобто відвідувач не може робити нічого іншого, поки не вибере одну з кнопок.

Приклад: `result = prompt("text", "value");`

confirm() – виводить вікно з питанням `question` з двома кнопками: "ОК" і "СКАСУВАННЯ"

Приклад: `var a = confirm("Ви – адміністратор?");`

Створення JavaScript- функцій

Функція – це іменована послідовність операторів (інструкцій). Будь-яка функція має наступний синтаксис:

```
function і'мя () {  
  оператор;  
  .....  
  оператор;  
}
```

Приклад JavaScript-функцій

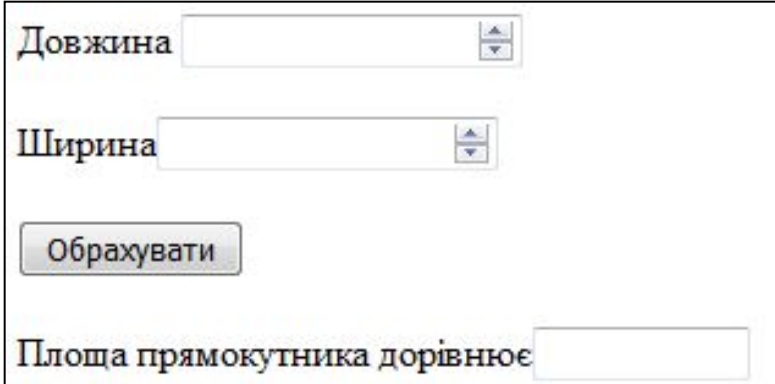
```
<html>  
  <head>  
    <title>Заголовок документа</title>  
    <link rel="stylesheet" href="style.css">  
    <script src="script.js"></script>  
  </head>  
  <body>  
    <div onClick="showMessage();">Push me</div>  
  </body>  
</html>
```

В script.js:

```
function showMessage(){  
  alert ("Ви клікнули по div-у");  
}
```

Розрахувати площу прямокутника

```
<html>
  <head>
    <title>Розрахунок площі прямокутника</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
  </head>
  <body>
    <form name="forma1">
      Довжина <input type="number" name="t1" size="10">
<br><br>
      Ширина<input type="number" name="t2" size="10"><br><br>
      <input type="button" name="button" value="
Обрахувати"><br><br>
      Площа прямокутника дорівнює<input type="text" name="res"
size="10">
    </form>
  </body>
</html>
```



The screenshot shows a web form with the following elements:

- A label "Довжина" followed by a text input field with a spinner control on the right.
- A label "Ширина" followed by a text input field with a spinner control on the right.
- A button labeled "Обрахувати".
- A label "Площа прямокутника дорівнює" followed by a text input field.

Розрахувати площу прямокутника

.....

```
<input type="button" name="button" value="Обрахувати"  
onClick="areaRectangle();"><br><br>
```

.....

Написання самої функції

```
function areaRectangle()  
{  
}
```

Тепер треба написати тіло функції. Для початку оголосимо три змінні: **a** – значення довжини прямокутника, **b** – значення ширини прямокутника, **s** – площа прямокутника:

```
function areaRectangle()  
var a;  
var b;  
var s;  
}
```


Розрахувати площу прямокутника

Значення (value) а має братися з поточної сторінки (document), з форми з ім'ям "forma1", з текстового поля з ім'ям "t1". Так це і записується `document.forma1.t1.value`, тобто перераховуються через крапку імена об'єктів від батьківського до потрібного. Останнім вказується необхідна властивість об'єкта (value).

```
function areaRectangle(){  
    var a=document.forma1.t1.value;  
    var b=document.forma1.t2.value;  
    var s=a*b;  
}
```

Розрахувати площу прямокутника

```
function areaRectangle(){  
    var a=document.forma1.t1.value;  
    var b=document.forma1.t2.value;  
    var s=a*b;  
    document.forma1.res.value=s;  
}
```

Приклад простої програми

```
<!DOCTYPE html>
<html>
<body>
<p id="demo">Click the button to change the text
in this paragraph.
</p>
<button onclick="myFunction()">Try it</button>
<script>
function myFunction() {
document.getElementById("demo").innerHTML = "Hello
World";
}
</script>
</body>
</html>
```