
Уеб технологии

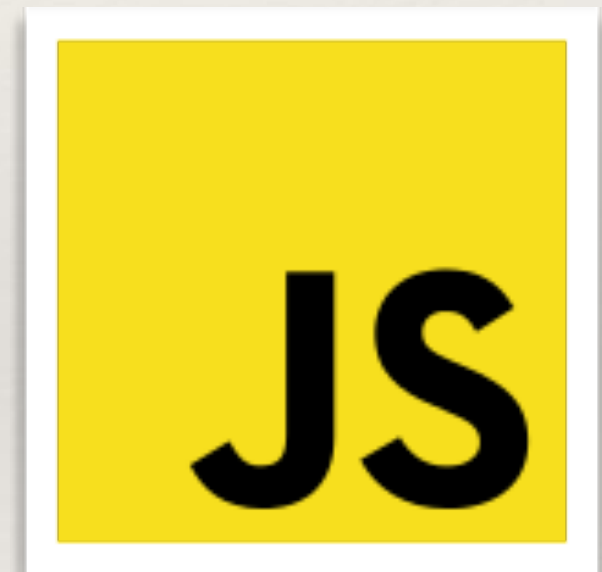
4 к., спец. БИС,
3 к., спец. “Информатика”
2016-2017 уч. година



Програмен език - JavaScript

JavaScript

- ❖ скриптов език, за създаване на интерактивни сайтове
- ❖ прилага се към HTML-а кода на страницата с цел добавяне на функционалност, работа с данни
- ❖ поддържа обектно-ориентиран и функционален стил на програмиране
- ❖ разработен от Netscape през 1995-та



JavaScript - възможности

- ❖ ефекти с изображения и HTML елементи, като скриване, показване, пренареждане, влачене, слайд шоу и др.
- ❖ управление на прозорци и рамки
- ❖ разпознаване на възможностите на браузъра
- ❖ използване на камерата и микрофона
- ❖ др.

JavaScript

- ❖ използва обектния модел на документа, в рамките на който всеки html контейнер може да се разглежда като съвкупност от свойства, методи, събития



JavaScript

❖ **обект**

обект.свойство

Пример:

Human - обект

Human.Name = 'Ivan' - свойство на обекта

❖ **метод**

obekt.metod(параметри);

Human.GetYear();

обект.метод = име на функция;

❖ **събитие**

OnBirth

❖ **клас**

Ivan - new Human;

JavaScript и HTML (1)

- ❖ JavaScript кодът може да се вмъква в HTML документа таг `<script>` `</script>`

```
<script>  
document.write("<h1>Здравей!</h1>");  
</script>
```

```
/* Сценарият ще се изпълни при отваряне на  
страницата, вмъкнат е JavaScript код, който изписва на  
екрана "Здравей!" */
```

JavaScript и HTML (2)

```
<form>  
<input type="button" value="Back"  
onClick="history.back()">  
</form>
```

/* При натискане на бутон се генерира
събитието onClick */

JavaScript и HTML (3)

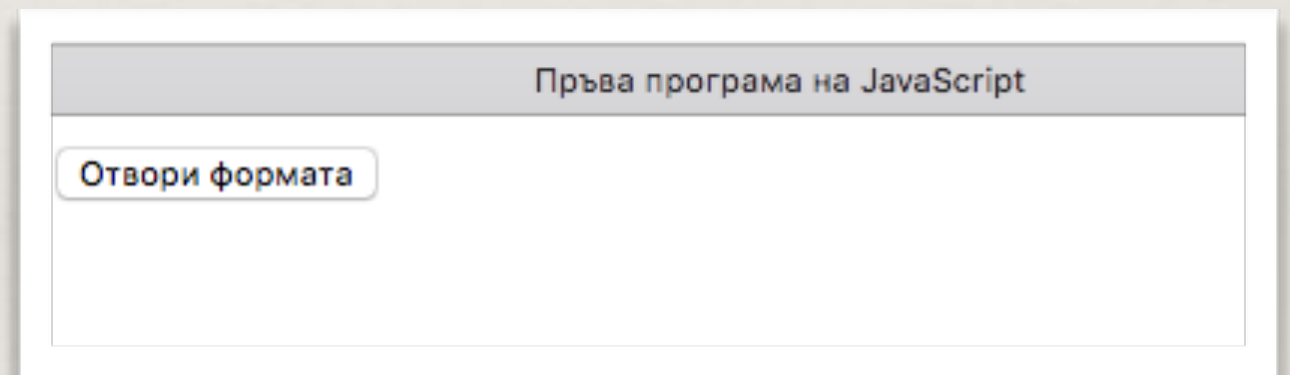
```
<form>  
<input type="button"  
value="Press me"  
onClick="openWindow( )">  
</form>
```

/* Необходимо е да се добави функцията
openWindow() в тага <script>*/

Вмъкнат JavaScript код

```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Прва програма на JavaScript</title>
<script language="javascript">
    function openWindow() {
        msgWindow=open( "cform.html" )
    }
</script>
</head>

<body>
<form>
<input type="button" value="Отвори формата" onClick="openWindow( )" >
</form>
</body>
</html>
```



Свързан JavaScript код

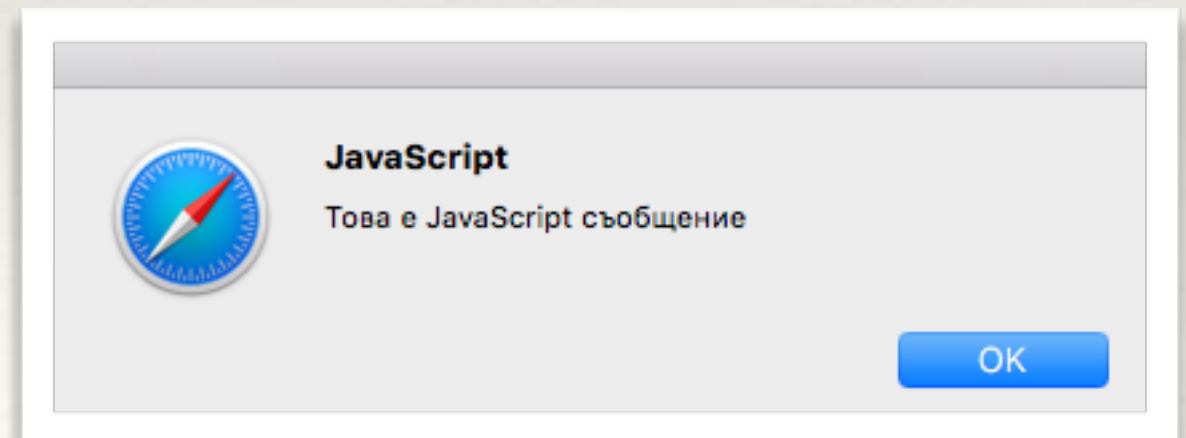
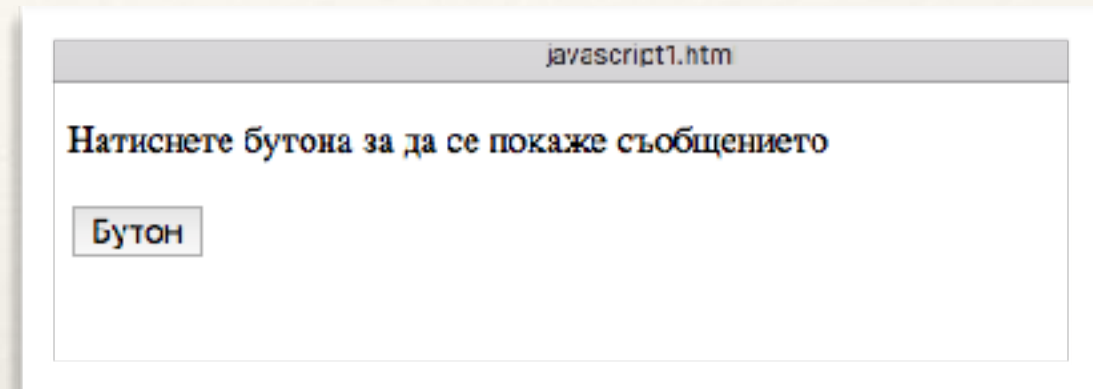
```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Първа програма на JavaScript</title>
<script src="javascript1.js" type="text/livescript">
</script>
</head>
<body>
<form>
<input type="button" value="Отвори формата" onClick="openWindow()">
</form>
</body>
</html>
```

```
// JavaScript Document
function openWindow() {
msgWindow=open( "cform.html" )
}
```

Метод alert()

- ❖ Метод alert() - прозорец със съобщение и бутон ОК

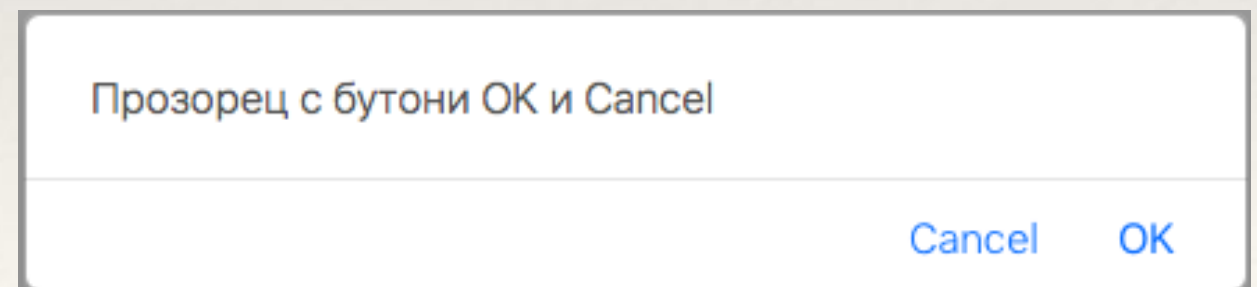
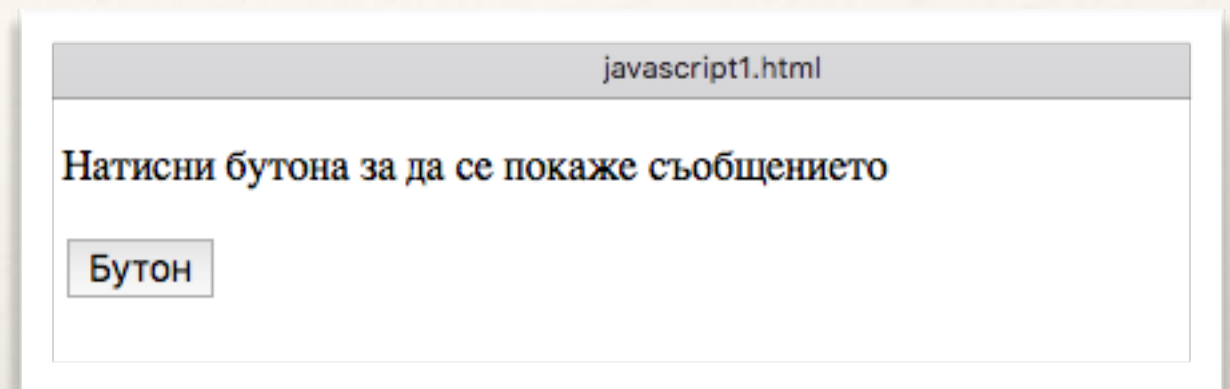
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<p>Натиснете бутона за да се покаже съобщението</p>
<button onclick="myFunction()">Бутон</button>
<script>
function myFunction() {
    alert("Това е JavaScript съобщение");
}
</script>
</body>
</html>
```



Метод confirm()

❖ Метод `confirm()` - прозорец с бутони ОК и Cancel

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<p>Натисни бутона за да се покаже съобщението</p>
<button onclick="myFunction()">Бутон</button>
<script>
function myFunction() {
    confirm("Прозорец с бутони ОК и Cancel");
}
</script>
</body>
</html>
```



Метод prompt()

- ❖ Метод `prompt()` - диалогов прозорец за показване на данни

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
</head>
```

```
<body>
```

```
<p>Натиснете бутона за да се отвори prompt box прозорец</p>
```

```
<button onclick="myFunction()">Бутон</button>
```

```
<p id="demo"></p>
```

```
<script>
```

```
function myFunction() {
```

```
    var person = prompt("Моля въведете вашето име", " ");
```

```
    if (person != null) {
```

```
        document.getElementById("demo").innerHTML =
```

```
        "Здравейте " + person + "! Как сте днес?";
```

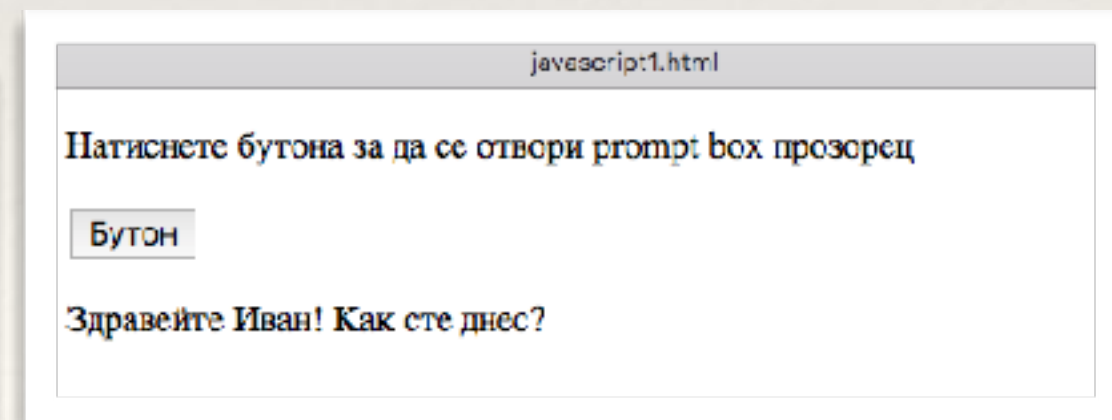
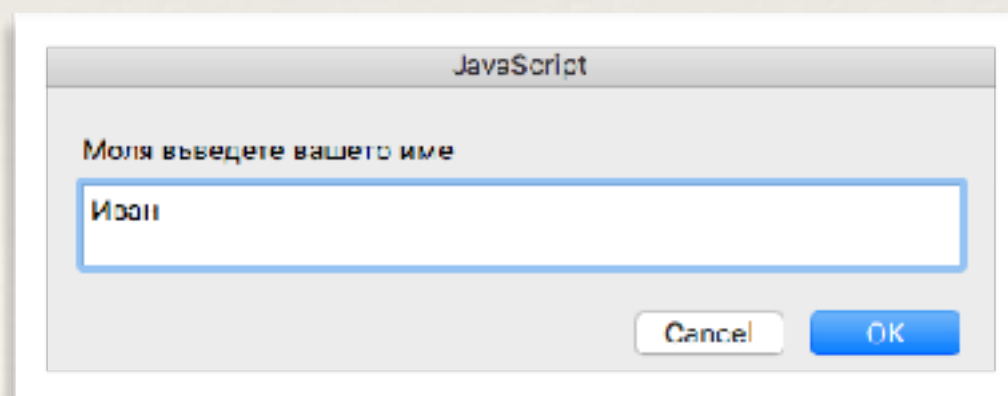
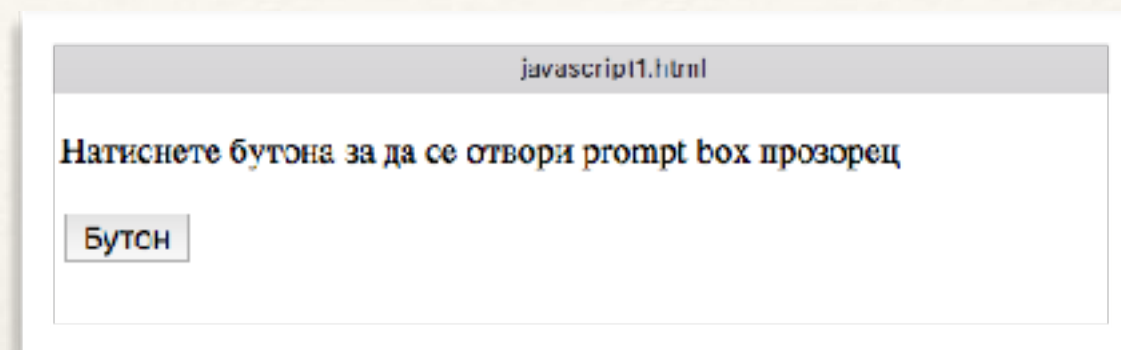
```
    }
```

```
}
```

```
</script>
```

```
</body>
```

```
</html>
```



Променливи в JavaScript

- ❖ променливите в JavaScript не са типизирани
- ❖ дефинират се чрез ключовата дума **var**, след която се изписва само името на променливата, ако тя няма стойност, ако има стойност, след името ѝ се приравнява приеманата по подразбиране стойност, на един команден ред могат да се изписват, разделени със запетая, множество променливи
- ❖ името на променливата трябва да започва с буква (A-Z) или със символ за подчертаване, след това в името може да има цифри, букви или знака \$, в JavaScript е от значение регистъра на символите, името на променливата не може да бъде запазена, ключова дума.
- ❖ Примери:
`var x, y1, userName, _user;`
неправилни имена - `1x`, `public`
`var my = 1; //инициализиране на променлива`
`var my = "hello";`

Променливи и типове данни

❖ В JavaScript променливите могат да съдържат следните типове данни:

❖ **number** - числа (цели и с плаваща запетая)

```
Num1 = 5;
```

```
Num2 = 5.5;
```

❖ **string** - низове

```
Str1 = "Hello";
```

```
Str2 = "World";
```

```
Str3 = null;
```

❖ **boolean** - логически тип - true, false

```
Bool1 = true;
```

❖ **function** - функции, може да се свърже функция с променлива като се напише името на функцията без скобите

❖ **object** - масиви, обекти, променливи със стойност null

```
var x = {firstName:"ИВАН", lastName:"Петров"};
```

//обект

```
var cars = ["Saab", "Volvo", "BMW"];
```

//масив

Стрингове

```
<body>
<p id="demo"></p>
<script>
var carName1 = "BMW 316i";
var carName2 = 'BMW 318i';
var answer1 = "It's alright";
var answer2 = "Темата е 'Облачни изчисления' ";
var answer3 = 'Темата е "Облачни изчисления" ';
document.getElementById("demo").innerHTML =
carName1 + "<br>" +
carName2 + "<br>" +
answer1 + "<br>" +
answer2 + "<br>" +
answer3;
</script>
</body>
```

BMW 316i
BMW 318i
It's alright
Темата е 'Облачни изчисления'
Темата е "Облачни изчисления"

Указва на браузъра да
изпише стойността за
елемента със съответното
id

Масиви

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<p id="demo"></p>
<script>
var cars = ["Opel", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars[2];
//Указва на браузъра да изпише стойността за елемента със
съответното id
</script>
</body>
</html>
```



BMW

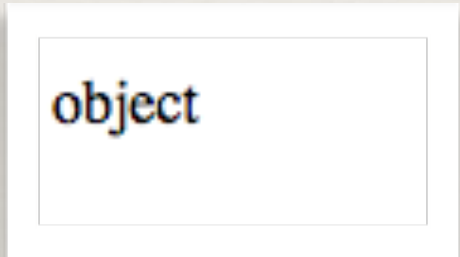
Обекти

```
<body>
<p id="demo"></p>
<script>
var person = {
    firstName : "Иван",
    lastName  : "Петров",
    age       : 50,
    eyeColor  : "blue"
};
document.getElementById("demo").innerHTML =
person.firstName + " е на " + person.age + " години.";
</script>
</body>
```

Иван е на 50 години.

Null

```
<body>
<p id="demo"></p>
<script>
var person = {firstName:"ИВАН",
lastName:"Петров", age:50,
eyeColor:"blue"};
var person = null;
document.getElementById("demo").innerHTML
= typeof person;
</script>
</body>
```



object

Оператор typeof

- ❖ Операторът връща типа на данните на променливите

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Оператор typeof</title>
</head>
<body>
<script>
var Num1, Num2, Str1, Str2, Str3, Bool1;
Num1=5;
Num2=5.5;
Str1="Hello";
Str2="World";
Str3=null;
Bool1=true;

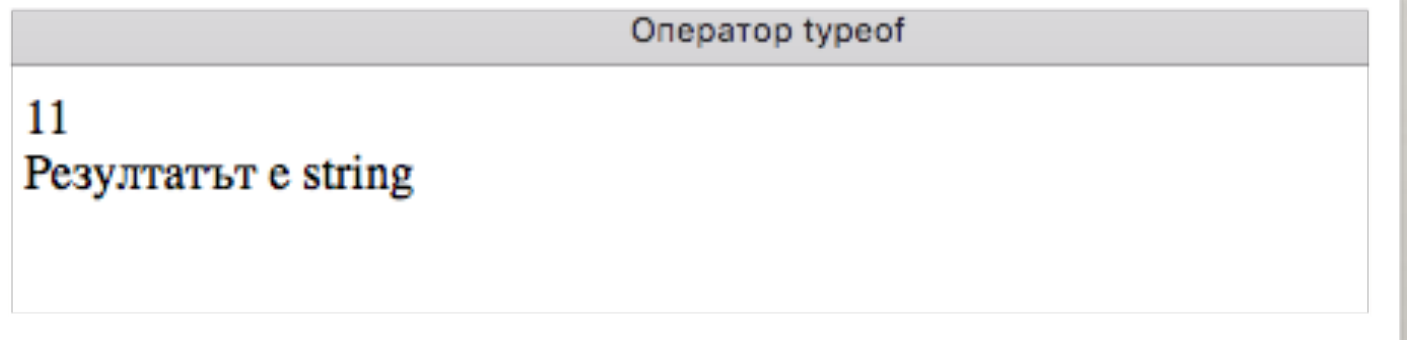
document.write("<br>Променливата Num1 е от тип "+typeof(Num1));
document.write("<br>Променливата Num2 е от тип "+typeof(Num2));
document.write("<br>Променливата Str1 е от тип "+typeof(Str1));
document.write("<br>Променливата Str2 е от тип "+typeof(Str2));
document.write("<br>Променливата Str3 е от тип "+typeof(Str3));
document.write("<br>Променливата Bool1 е от тип "+typeof(Bool1));
</script>
</body>
</html>
```

Оператор typeof

Променливата Num1 е от тип number
Променливата Num2 е от тип number
Променливата Str1 е от тип string
Променливата Str2 е от тип string
Променливата Str3 е от тип object
Променливата Bool1 е от тип boolean

Преобразуване на типовете

```
<script>
var A=1;
var B="1";
var C=A+B;
document.write(C);
document.write("<br>Резултатът е " + typeof(C));
</script>
</body>
</html>
```



- ❖ преобразуване на типовете
 - parseInt()
 - parseFloat()

Преобразуване на типовете

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
var x = 16 + 4 + "Слънце";
document.getElementById("demo").innerHTML = "Резултатът е: " + x;
</script>
</body>
</html>
</body>
</html>
```

Резултатът е: 20Слънце

Преобразуване на типовете

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
var x = "Слънце" + 16 + 4;
document.getElementById("demo").innerHTML = "Резултатът е: " + x;
</script>
</body>
</html>
</body>
</html>
```

Резултатът е: Слънце164

Локални и глобални променливи

- ❖ **локална променлива** - тази, която е обявена във функцията, достъпна само за дадена функция
- ❖ **глобална променлива** - обявява се в тялото сценария и е достъпна за всички функции от сценария, прието е да се обявяват в началото на сценария

Аритметични оператори

Оператор	Използване
+	събиране на две числа или обединяване на два стринга, ако действието се извършва с променливи от различен тип, като например число се добавя към стринг или обратното, то резултатът ще е стринг
-	изваждане
/	деление
*	умножение
%	връща остатък от делението
++	инкрементиране, може да се запише по два начина : $x++$ и $++x$, разликата е, че в първия случай, първо, се връща стойността и след това се прилага операцията, а при втория запис, първо, се изпълнява операцията и след това се връща, новата стойност на операнда
--	декрементиране, може да се запише по два начина : $x--$ и $--x$, аналогично на операция инкрементиране

Пример (1)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<h1>JavaScript оператори</h1>
<p>x = 5, y = 2</p>
<p>z = x + y</p>
<p id="demo"></p>
<script>
var x = 5;
var y = 2;
var z = x + y;
document.getElementById("demo").innerHTML = "z = " + z;
</script>
</body>
</html>
```



Указва на браузъра да изпише стойността за
елемента със съответното id

JavaScript оператори

x = 5, y = 2

z = x + y

z = 7

Пример (2)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<h1>JavaScript оператори</h1>
<p>x = 7</p>
<p>y = x++</p>
<p id="demo"></p>
<script>
var x = 7;
var y = ++x;
document.getElementById("demo").innerHTML = "y = " + y;
</script>
</body>
</html>
```

JavaScript оператори

x = 7

y = ++x

y = 8

Оператори за присвояване

Оператор	Използване
$=$	приравняване на стойности / изрази
$+=$	$x+=y$ е еквивалентно на $x=x+y$
$-=$	$x-=y$ е еквивалентно на $x=x-y$
$*=$	$x*=y$ е еквивалентно на $x=x*y$
$/=$	$x/=y$ е еквивалентно на $x=x/y$
$\%=$	$x\%=y$ е еквивалентно на $x=x\%y$

Оператори за сравнение

Оператор	Използване
<code>==</code>	равно
<code>===</code>	строго равно (сравнение за равенство и тип)
<code>!=</code>	различно
<code>!==</code>	не е равно и типът е различен
<code>></code>	по-голямо от
<code><</code>	по-малко от
<code>>=</code>	по-голямо или равно
<code><=</code>	по-малко или равно

Пример (2)

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<p>Присвояваме на x 5 и показваме стойността при
сравнението (x == 8):</p>
<button onclick="myFunction()">Бутон</button>
<p id="demo"></p>
<script>
function myFunction() {
    var x = 5;
    document.getElementById("demo").innerHTML = (x == 8);
}
</script>
</body>
</html>
```

Присвояваме на x 5 и показваме стойността при сравнението (x == 8):

Бутон

Присвояваме на x 5 и показваме стойността при сравнението (x == 8):

Бутон

false

Логически оператори

Оператор	Използване
!	отрицание (NOT)
&&	логическо и (AND)
	логическо или (OR)

Основни конструкции на езика

- ❖ условен оператор **if**
- ❖ оператор **switch**
- ❖ **ЦИКЛИ**

Условен оператор if

```
if (condition) {  
    код, който да се изпълни ако условието е true  
}
```

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="utf-8">  
</head>  
<body>  
<p>Display "Приятен ден!" ако часа е преди 18:00:</p>  
<p id="demo">Приятна вечер!</p>  
<script>  
if (new Date().getHours() < 18) {  
    document.getElementById("demo").innerHTML = "Приятен ден!";  
}  
</script>  
</body>  
</html>
```

Display "Приятен ден!" ако часа е преди 18:00:

Приятен ден!

Условен оператор if

```
if (condition) {  
    код, който да се изпълни ако условието е true  
} else {  
    код, който да се изпълни ако условието е false  
}
```

```
<body>  
<p>Натисни бутона за поздрав</p>  
<button onclick="myFunction()">Бутон</button>  
<p id="demo"></p>  
<script>  
function myFunction() {  
    var hour = new Date().getHours();  
    var greeting;  
    if (hour < 18) {  
        greeting = "Приятен ден";  
    } else {  
        greeting = "Приятна вечер";  
    }  
    document.getElementById("demo").innerHTML = greeting;  
}  
</script>  
</body>
```

Натисни бутона за поздрав

Бутон

Натисни бутона за поздрав

Бутон

Приятен ден

Условен оператор if

```
if (условие1) {  
    код, който да се изпълни ако условието е true  
} else if (условие2) {  
    код, който да се изпълни ако условие1 е false и условие2 е true  
} else {  
    код, който да се изпълни ако условие1 е false и условие2 е false  
}
```

```
<body>  
<p>Натисни бутона за поздрав</p>  
<button onclick="myFunction()">БУТОН</button>  
<p id="demo"></p>  
<script>  
function myFunction() {  
    var greeting;  
    var time = new Date().getHours();  
    if (time < 10) {  
        greeting = "Добро утро";  
    } else if (time < 19) {  
        greeting = "Добър ден";  
    } else {  
        greeting = "Добър вечер";  
    }  
    document.getElementById("demo").innerHTML = greeting;  
}  
</script>  
</body>
```

Натисни бутона за поздрав

Бутон

Натисни бутона за поздрав

Бутон

Добър ден

Оператор switch

```
switch(СТОЙНОСТЬ) {  
    case n:  
        програмен блок  
        break;  
    case n:  
        програмен блок  
        break;  
    default:  
        default програмен блок  
}
```


Оператор switch

Днес е неделя.

```
<body>
<p id="demo"></p>
<script>
var day;
switch (new
Date().getDay()) {
    case 0:
        day = "неделя";
        break;
    case 1:
        day = "понеделник";
        break;
    case 2:
        day = "вторник";
        break;
    case 3:
        day = "сряда";
        break;
```

```
        case 4:
            day = "четвъртък";
            break;
        case 5:
            day = "петък";
            break;
        case 6:
            day = "събота";
            break;
    }
    document.getElementById("demo
").
innerHTML = "Днес е " + day +
".";
</script>
</body>
```

Оператор switch

```
<body>
<p id="demo"></p>
<script>
var text;
switch (new Date().getDay()) {
    case 1:
    case 2:
    case 3:
    default:
        text = "Очакваме почивните дни.";
        break;
    case 4:
    case 5:
        text = "Скоро ще има почивни дни.";
        break;
    case 0:
    case 6:
        text = "Почивни дни са.";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
```

Почивни дни са.

Цикъл - for

- ❖ използва се когато даден програмен блок се изпълнява точно определен брой пъти

```
for (инициализация; условие; итерация) {  
    програмен код  
}
```

```
for (i = 0; i < cars.length; i++) {  
    text += cars[i] + "<br>";  
}
```

Цикъл - for (пример)

```
<body>
<p id="demo"></p>
<script>
var cars = [ "BMW", "Volvo", "Opel", "Ford" ];
var text = "";
var i;
for (i = 0; i < cars.length; i++) {
    text += cars[i] + "<br>";
}
document.getElementById( "demo" ).innerHTML = text;
</script>
</body>
```

BMW
Volvo
Opel
Ford

Цикъл - For/In

- ❖ използва се когато цикълът е за свойствата на обекта

```
var person = {fname:"ИВАН",  
lname:"Петров", age:25};  
var text = "";  
var x;  
for (x in person) {  
    text += person[x];  
}
```


Цикъл - For/In (пример)

```
<body>
<p id="demo"></p>
<script>
var txt = "";
var person = {fname:"ИВАН", lname:"Петров", age:
25};
var x;
for (x in person) {
    txt += person[x] + " ";
}
document.getElementById("demo").innerHTML = txt;
</script>
</body>
```

Иван Петров 25

Цикъл - while

- ❖ ИЗПОЛЗВА СЕ ЦИКЪЛЪТ ДОКАТО Е ИЗПЪЛНЕНО УСЛОВИЕТО

```
while (условие) {  
    програмен код, който се изпълнява  
}
```

```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```

Цикъл - while (пример)

```
<body>
<p>Натисни бутона</p>
<button onclick="myFunction()">Бутон</button>
<p id="demo"></p>
<script>
function myFunction() {
    var text = "";
    var i = 0;
    while (i < 10) {
        text += "<br>Числото е " + i;
        i++;
    }
    document.getElementById("demo").innerHTML = text;
}
</script>
</body>
```

Натисни бутона

Бутон

Натисни бутона

Бутон

Числото е 0
Числото е 1
Числото е 2
Числото е 3
Числото е 4
Числото е 5
Числото е 6
Числото е 7
Числото е 8
Числото е 9

Цикъл - do ...while

- ❖ ще се изпълни програмния код в тялото на цикъла преди да се провери условието, след това цикълът се повтаря докато е изпълнено условието

```
do {  
    code block to be executed  
}
```

```
while (condition);
```

```
do {  
    text += "The number is " + i;  
    i++;  
}  
while (i < 10);
```

Цикъл - do...while (пример)

```
<body>
<p>Натисни бутона</p>
<button onclick="myFunction()">БУТОН</button>
<p id="demo"></p>
<script>
function myFunction() {
    var text = ""
    var i = 0;
    do {
        text += "<br>Числото е " + i;
        i++;
    }
    while (i < 10)
    document.getElementById("demo").innerHTML = text;
}
</script>
</body>
```

Натисни бутона

Бутон

Натисни бутона

Бутон

Числото е 0
Числото е 1
Числото е 2
Числото е 3
Числото е 4
Числото е 5
Числото е 6
Числото е 7
Числото е 8
Числото е 9

Оператор Break

- ❖ break - прекъсва изпълнението на цикъла

```
<body>
```

```
<p>Цикъл с break</p>
```

```
<p id="demo"></p>
```

```
<script>
```

```
var text = "";
```

```
var i;
```

```
for (i = 0; i < 10; i++) {  
    if (i === 3) { break; }
```

```
    text += "Числото е " + i + "<br>";
```

```
}
```

```
document.getElementById("demo").innerHTML = text;
```

```
</script>
```

```
</body>
```

Цикъл с break

Числото е 0

Числото е 1

Числото е 2

Оператор Continue

- ❖ `continue` - прекъсва изпълнението на текущата итерация и преминава към следващата

```
<body>
<p>Цикъл с continue, пропуска изпълнението когато i=3</p>
<p id="demo"></p>
<script>
var text = "";
var i;
for (i = 0; i < 10; i++) {
    if (i === 3) { continue; }
    text += "Числото е " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
```

Цикъл с `continue`, пропуска изпълнението когато `i=3`

Числото е 0
Числото е 1
Числото е 2
Числото е 4
Числото е 5
Числото е 6
Числото е 7
Числото е 8
Числото е 9

Влагане на цикли

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<script>
var j = 1;
while (j<15) {
  for (k = 0; k < j; k++) document.write("*");
  document.write("<br>");
  j++;
}
</script>
</body>
</html>
```

```
*  
**  
***  
****  
*****  
******  
*******  
********  
*********  
**********  
***********  
************  
*****
```