آشنایی با جاوا اسکریپت

- انواع زبان های برنامه نویسی تحت وب
  - سمت کلاینت
  - سمت سرور

تعریف متغیر

- قبل از اینکه از یک متغیر در یک برنامه ی جاوا اسکریپت استفاده کنید، شما باید آن را اعلان و یا تعریف کنید. متغیر ها با استفاده از کلمه ی کلیدی var به شکلی که در زیر آمده است، تعریف میشوند:
- var money; var name;

- نکته : از کلمه ی کلیدی var تنها یک بار برای تعریف و مقدار دهی یک متغیر با یک نام مشخص استفاده کنید. شما نباید یک متغیر را دو بار تعریف کنید.

نامگذاری متغیرها

نامگذاری متغیرها می بایست شرایط زیر را داشته باشد:

اولین کاراکتر متغیر می تواند یک حرف , یک underline (_) و یا یک علامت $ باشد.

بقیه کاراکترها می‌توانند از $، _ و یا هر حرف و عددی تشکیل شوند.

- شما نباید از هیچکدام از کلمات کلیدی و رزرو شده ی زبان جاوا اسکریپت به عنوان نام برای متغیر استفاده کنید. برای مثال تعریف متغیر هایی با نام های break و Boolean صحیح نیستند.

نامگذاری متغیرها

کلمات رزرو شده

Abstract	enum	int	short
Boolean	export	interface	static
Byte	extends	long	super
Char	final	native	synchronized
Class	float	package	throws
Const	goto	private	transient
Debugger	implements	protected	volatile
Double	import	public	

کلمات کلیدی

Break	else	new	var
Case	finally	return	void
Catch	for	switch	while
Continue	function	this	with
Default	if	throw	
Delete	in	try	
Do	instanceof	typeof	

مقداردهی متغیرها در جاوااسکریپت

```
Var  test ='ali';
```

```
var  test 1='ali' , test2='salam' ;
```

```
var  test_1='ali' , age=25;
```

- در جاوااسکریپت متغیرها می توانند مقدار اولیه نگیرند.

```
var  test ;
```

```
var  test ="hi" ;
alert(test);   // hi
test=55;
alert(test);   // 55
```

انواع داده ها در جاوااسکریپت

در جاوا اسکریپت پنج نوع داده اصلی به شرح زیر وجود دارد:

1) undefinedمتغیری که اعلان شود و مقدار دهی نشود.
2) nullنوع دیگر داده مقدار null است که به معنی خالی است. یک متغیر زمانی از نوع null است که اشاره به شی ای داشته باشد که وجود ندارد.
3) booleanنوع منطقی
4) numberنوع عددی(به اعداد صحیح integerو به اعداد اعشار floatمی گویند)
5) stringاین نوع می تواند برای ذخیره صفر یا چندین کاراکتر به کار رود.

انواع داده ها در جاوااسکریپت

```
var  oTemp ;
alert (typeof  oTemp) ;  // outputs "undefined"
alert (typeof oTemp2) ;  // outputs  "undefined"
```

```
var bFound = true;
var bLost = false;
```

```
var iNum = 55;
```

```
var fNum = 5.0;
```

```
var sColor1 = "blue";
var sColor2 = 'blue';
```

انواع داده ها در جاوا اسکریپت

- یک متغیر در جاوا اسکریپت می‌تواند شامل هر داده ای باشد. یک متغیر می تواند در یک لحظه حاوی یک رشته باشد و بعدا یک مقدار عددی بگیرد.

```
message = "hello";
message = 123456;
```

نمایش داده ها در جاوا اسکریپت

در جاوا اسکریپت میتوان داده ها را به چندین روش در مرورگر نشان داد :

- ✓ نوشتن در داخل یک عنصر HTML که با استفاده از خاصیت innerHTML انجام میشود.
- ✓ نوشتن در داخل خروجی HTML با استفاده از document.write ()
- ✓ نوشتن در یک پنجره ی پیغام با استفاده از window.alert ()
- ✓ نوشتن در داخل کنسول مرورگر با استفاده از console.log ()

تعریف کدهای جاوا اسکریپت

- HTML برای استفاده از جاوااسکریپت در صفحات، تگی به نام script را فراهم کرده است. عموما از این تگ در داخل تگ head صفحه استفاده می شود و می تواند یک ، دو یا سه خصوصیت را بگیرد.
- کد های جاوا اسکریپت را می توان در هر جایی از یک سند HTML قرار داد.
  - کد جاوا اسکریپت در قسمت<head>...</head>
  - کد جاوا اسکریپت در قسمت<body>...</body>
  - کد جاوا اسکریپت در قسمت <body>...</body> و <head>...</head>
  - نوشتن کد های جاوا اسکریپت در یک فایل جداگانه و سپس قرار دادن ارجاعی از آن در قسمت <head>...</head>

Body example

```
<html>
<head>
</head>

<body>

<script type="text/javascript">
document.write("This message written by JavaScript");
</script>

</body>
</html>
```

Internal example

```
<html>
<head>
<script type="text/javascript">
function message()
{
  alert("This alert was called with the onload event");
}
</script>
</head>

<body onload="message()">
</body>
</html
```

External example

```
<html>
<head>
  <script type="text/javascript" src="xyz.js"></script>
</head>
<body>
</body>
</html>
```

عملگر ها JavaScript

- عملگر های محاسباتی(Arithmetic)
- عملگر های مقایسه ای(Comparison)
- عملگر های منطقی یا رابطه ای(Logical)
- عملگر های انتساب(Assignment)
- عملگر های شرطی یا سه تایی(Conditional)

عملگر ها در JavaScript

- عملگر های محاسباتی

Operator	Description	Example	Result
+	Addition	x=2 y=2 x+y	4
-	Subtraction	x=5 y=2 x-y	3
*	Multiplication	x=5 y=4 x*y	20
/	Division	15/5 5/2	3 2,5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

example

```
<body>
<script type="text/javascript">
var a = 33;          var b = 10;          var c = "Test";
var linebreak = "<br />";
document.write("a + b = ");
result = a + b;
document.write(result);
document.write(linebreak);
document.write("a - b = ");
result = a - b;
document.write(result);
document.write(linebreak);
```

example

```
document.write("a / b = ");
result = a / b;
document.write(result);
document.write(linebreak);
document.write("a % b = ");
result = a % b;
document.write(result);
document.write(linebreak);
document.write("a + b + c = ");
result = a + b + c;
document.write(result);
```

example

```
document.write(linebreak);
a = ++a;
document.write("++a = ");
result = ++a;
document.write(result);
document.write(linebreak);
b = --b;
document.write("--b = ");
result = --b;
document.write(result);
document.write(linebreak);
</script>        </body></html>
```

عملگر ها در JavaScript

- عملگرهای انتساب

Operator	Example	Is The Same As
=	x=y	x=y
+=	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

عملگر ها در JavaScript

- عملگرهای مقایسه ای

Operator	Description	Example
==	is equal to	5==8 returns false
===	is equal to (checks for both value and type)	x=5 y="5" x==y returns true x===y returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

JavaScript Operators

- عملگرهای منطقی

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
||	or	x=6 y=3 (x==5 || y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

عملگر typeof

از عملگر typeof برای تشخیص نوع یک مقدار استفاده می شود. این عملگر یک پارامتر که می تواند یک متغیر و یا یک مقدار باشد را دریافت کرده و نوع آن را بر می گرداند.

این عملگر یکی از پنج مقدار زیر را بر می گرداند:

undefined:اگر متغیر از نوع Undefined است.

Boolean: اگر متغیر از نوع boolean باشد

توضیحات در جاوا اسکریپت

✓توضیحات تک خطی

// comment

✓توضیحات چند خطی

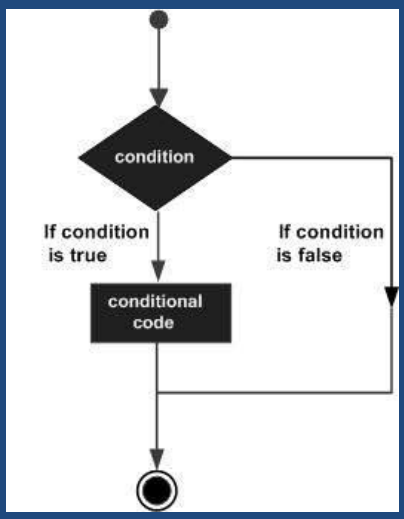
/* comment (avoid using this form…) */

ساختار شرطی if در جاوا اسکریپت

```
if (expression)
{
Statement(s) to be
   executed if expression
   is true
}
```

ساختار شرطی if در جاوا اسکریپت

```
if (expression)
{  Statement(s) to be
   executed if expression
   is true}
Else {  Statement(s) to
   be executed if
   expression is false}
```



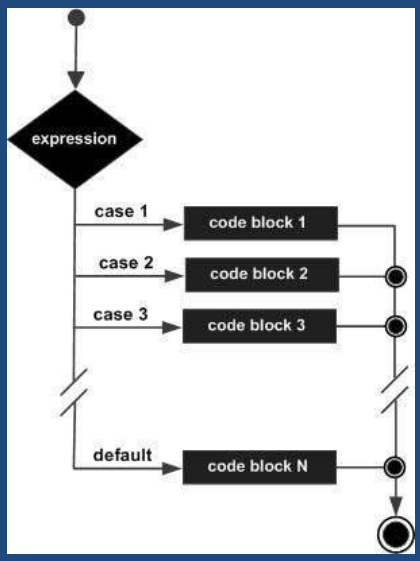
ساختار شرطی if در جاوا اسکریپت

```
if (condition) {
       statements;
} else if (condition) {
       statements;
} else {
       statements;
}
```

ساختار شرطی **switch** در جاوا اسکریپت

switch (expression)
{
case condition 1: statement(s) break;
case condition 2: statement(s) break; ...
case condition n: statement(s) break;
default: statement(s)
}

ساختار شرطی **switch** در جاوا اسکریپت

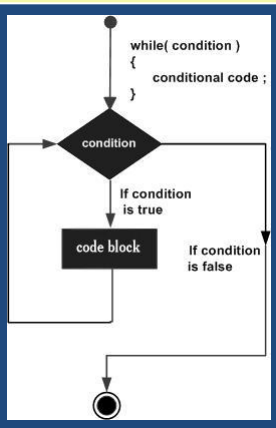


ساختارهای تکرار در جاوا اسکریپت

- ساختار تکرار while
- ساختار تکرار do while
- ساختار تکرار for

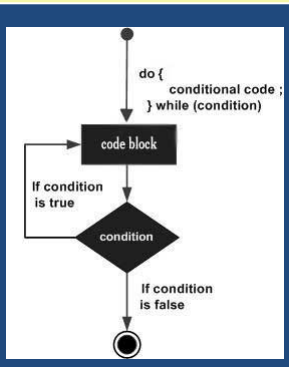
ساختار تکرار while

```
while (condition) {
       statements;
}
```
JS



ساختار تکرار do while

```
do {
    statements;
} while (condition);
```
JS



ساختار تکرار for

```
for (begin; condition; step) {
 statements;
}
```

تعریف آرایه ها

- آرایه یک متغیر مخصوص بوده که می تواند بیش از یک مقدار را همزمان در خود نگه دارد.

```
var array_name = [item1, item2, ...];
```

```
var aValues = new Array();
```

مقداردهی آرایه ها

```
var aColors = [“red”, “green”, “blue”];
```

```
var aColors = new Array(“red”, “green”, “blue”);
```

مقداردهی آرایه ها

```
var stringArray = ["one", "two", "three"];

var numericArray = [1, 2, 3, 4];

var decimalArray = [1.1, 1.2, 1.3];

var booleanArray = [true, false, false, true];

var mixedArray = [1, "two", "three", 4];
```

```
var stringArray = new Array();
stringArray[0] = "one";
stringArray[1] = "two";
stringArray[2] = "three";
stringArray[3] = "four";

var numericArray = new Array(3);
numericArray[0] = 1;
numericArray[1] = 2;
numericArray[2] = 3;

var mixedArray = new Array(1, "two", 3, "four");
```

آرایه

- یک آرایه نیز می‌توانند Objectباشند. به همین دلیل شما می توانید در یک آرایه مشخص، متغیرهایی با نوع های مختلف داشته باشید. حتی می توان در خانه های یک آرایه از یک شی استفاده نمود.

```
var student1 = {Name : "ali" , FName : “panahi" , age:22 , BDate:"1984/10/07"}
var student2 = {fName : “sara" , FName : "ahmadi" , age:20 , BDate:"1998/07/10"}
var student3 = {Name : “reza" , FName : “shojaee" , age:21 , BDate:"1995/03/10"}

var students = [student1 , student2 , student3]

alert(student1["Name"])
alert(students[1].firstName)
```

آرایه

بدست آوردن تعداد مقادیر درون یک آرایه در جاوا اسکریپت

```
var x = cars.length;
```

دسترسی به آخرین عنصر آرایه

```
var last = cars [cars.length-1];
```

مرتب سازی آرایه ها

- از دو تابع برای مرتب سازی عناصر آرایه استفاده میشود.
- تابعی به نام reverse() برای مرتب سازی عکس آرایه استفاده می شود.
- تابعی به نام sort() عناصر آرایه را به صورت صعودی بر حسب مقادیرشان مرتب می کند.

توابع (Function)

- تعریف تابع

```
function functionName()
{
  code to be executed
}
```

متغیری که در داخل یک function تعریف می‌شود، تنها داخل همان تابع قابل استفاده است

مثال:

```
function welcome( )
{
document.write ( "Just For Fun" )
}
```

توابع

- یک تابع می تواند، چندین متغیر را به عنوان پارامتر ورودی دریافت کند. پارامترهای یک تابع را باید در هنگام تعریف تابع، در پرانتز مقابل نام آن تعیین کرد، که پارامترها را با کاما از هم جدا می کنیم .
- در هنگام فراخوانی یک تابع که دارای پارامتر است، باید در پرانتز مقابل نام آن، مقادیر متناظر با پارامترهایش را اعلام کرد. که به این مقادیر آرگومان گفته می شود. این آرگومان باید از لحاظ تعداد و نوع کاملا یکسان با پارامترهای تعریف شده در تابع باشند .
- ساختار تعریف تابع به صورت زیر میباشد:

```
funciton Name ( Parametr 1 , Parametr 2 , ... )
{
Code
}
```

توابع

- یک تابع می تواند پس از انجام دستورات در نظر گرفته شده برای آن، مقداری را به عنوان خروجی به نقطه ای که از آن فراخوانی شده است، باز گرداند.
- برای تعیین مقدار بازگشتی یک تابع از دستور return استفاده کرده، که مقدار خروجی را درپرانتز مقابل آن دستور به شکل زیر تعریف می کنیم.

return (مقدار بازگشتی) ;

- نکته : مقدار بازگشتی ، می تواند یک رشته یا عدد ، یک متغیر و یا یک عبارت محاسباتی باشد.