

javascript



JavaScript

جاوا اسکریپت

مقدمه



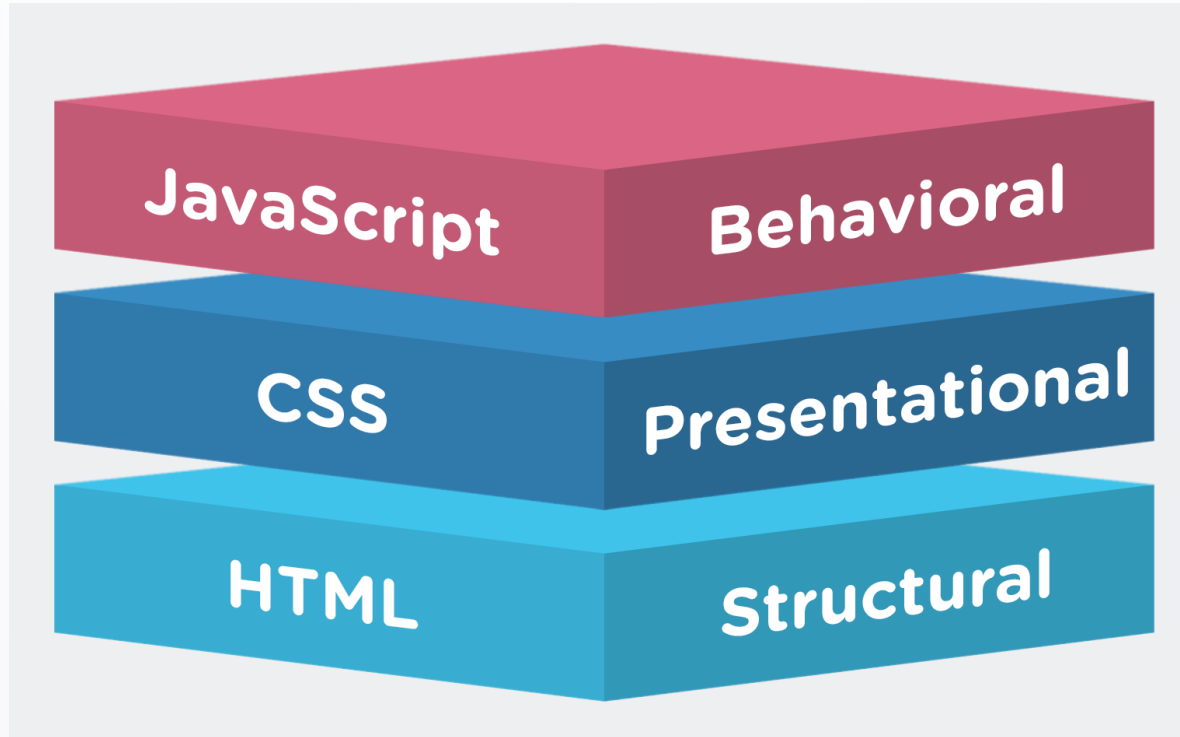
جاوا اسکریپت توسط فردی بنام برندان ایچ
در سال 1995 در شرکت نت اسکپی طراحی شد

کاربرد گسترده این زبان در سایتها و صفحات اینترنتی می باشد و به کمک این زبان
می توان به اشیاء داخل صفحات HTML دسترسی پیدا کرد و آنها را تغییر داد.

امروزه زبان جاوا اسکریپت بسیار پیشرو است و بسیار گسترده مورد استفاده قرار
می گیرد

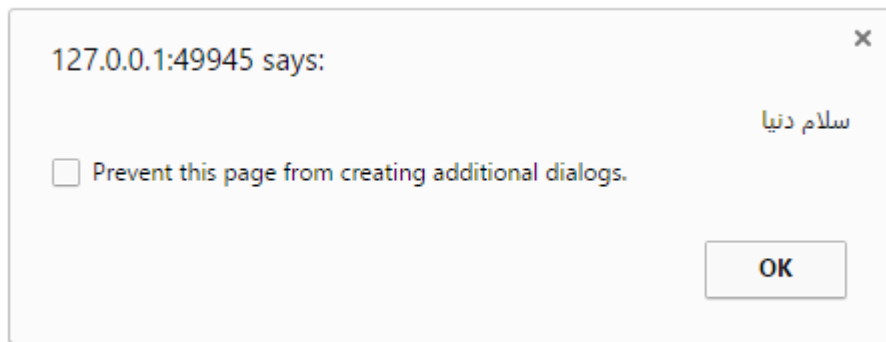


جاوا اسکریپت لایه رفتاری یک صفحه وب را تشکیل می‌دهد



چاپ سلام دنیا

```
alert("سلام دنیا");
```



- معمولاً در ابتدای یادگیری هر زبان برنامه نویسی به دانشجو یاد داده می شود که متن «سلام دنیا» را چاپ کند
- در اینجا ما با استفاده از تابع از پیش تعریف شده alert متن دلخواه خود را در صفحه وب بصورت یک کادر popup نمایش می دهیم که البته آزار دهنده است !



ایجاد کامنت برای مستند سازی

```
// این یک کامنت تک خطی است
```

- ایجاد یک کامنت تک خطی:

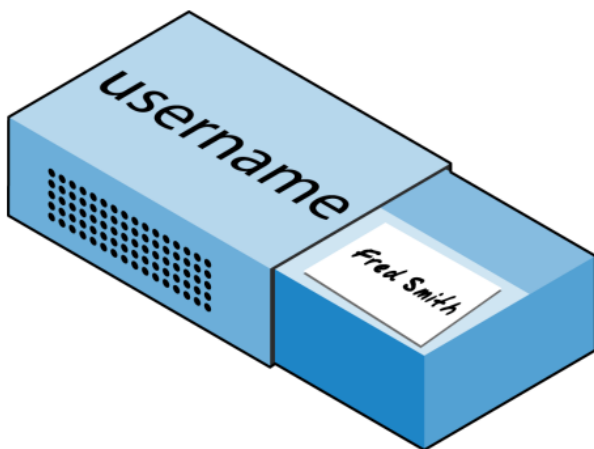
```
/*  
برای استفاده از  
چند خط کامنت برای  
مستند سازی از این شیوه استفاده می شود  
*/
```

- ایجاد یک کامنت
چند خطی:

- خاتمه دستور با semicolon (;) مشخص می شود .



متغیرها



```
var userName = "fred smith";
```



JavaScript

• دستور انتساب با علامت = مشخص می شود

- در نام گذاری متغیرها از حروف الفبا، اعداد، خط زیرین (-) و \$ باید استفاده کرد و نباید از فاصله در نام گذاری استفاده نمود ضمناً حرف اول آن باید از حروف الفبا و یا خط زیرین باشد.
- در جاوااسکریپت نیازی به تعریف نوع متغیر نیست و بعد از اولین مقدار دهی نوع آن مشخص می شود

متغیرها

```
var firstName = " ali"; // متغیر از نوع رشته  
var age = 21; // متغیر از نوع عدد صحیح  
var average=16.75; // متغیر از نوع اعشار  
var isMarried = false; // boolean متغیر از نوع
```

```
var firstName,age,average,isMarried;  
firstName="ali";  
age=21;  
average=16.75;  
isMarried=false;
```

شیوه راحتتر
تعریف متغیر
ها در یک خط



متغیرها

```
var firstName="ali",age=21,  
    average=16.75,isMarried=false;
```

تعریف و مقدار دهی چند
متغیر در یک خط

باید دقت نمود که اگر از عبارت `var` قبل از نام متغیر استفاده نکنیم باز هم آن متغیر تعریف می شود ولی امکان اشتباه وجود دارد پس برای پرهیز از اشتباه همیشه قبل از تعریف متغیر از کلمه کلیدی `var` استفاده میکنیم .

عملگرها

- عملگرهای ریاضی

عملگر	نام	مثال	نتیجه مثال
+	جمع	10+3	13
-	منها	10-3	7
*	ضرب	10*3	30
/	تقسیم	10/3	3.33333333333333
%	باقیمانده	10%3	1
**	توان	10**3	1000

- عملگرالحاق



JavaScript

```
var name = "mahdi", family = "asadi";  
alert(name + " " + family);
```



127.0.0.1:50225 says:

mahdi asadi

x

OK

عملگرهای ترکیبی

معادل	مثال	عملگر
<code>x = x + 5;</code>	<code>x += 5;</code>	<code>+=</code>
<code>x = x - 5;</code>	<code>x -= 5;</code>	<code>-=</code>
<code>x = x * 5;</code>	<code>x *= 5;</code>	<code>*=</code>
<code>x = x / 5;</code>	<code>x /= 5;</code>	<code>/=</code>
<code>x = x % 5;</code>	<code>x %= 5;</code>	<code>%=</code>
<code>x = x ** 5;</code>	<code>x **= 5;</code>	<code>**=</code>

عملگرهای مقایسه‌ای

عملگر	نام	جواب true خواهد بود اگر...	مثال : x = 4;	نتیجه
==	تساوی	طرفین عملگر با هم از نظر مقدار مساوی باشند	x==5	false
!=	نامساوی	طرفین عملگر با هم از نظر مقدار مساوی نباشند	x != 5	true
===	معادل	طرفین عملگر هم از نظر نوع و هم از نظر مقدار باید با یکدیگر مساوی باشند	x=== "4"	false
>	بزرگ‌تر	طرف سمت چپ عملگر از طرف سمت راست عمگ بزرگ‌تر باشد	x > 4	false
<	کوچک‌تر	طرف سمت چپ عملگر از سمت راست عملگر کوچک‌تر باشد	x < 4	false
>=	بزرگ‌تر یا مساوی	طرف سمت چپ عملگر از طرف سمت راست عمگ بزرگ‌تر یا مساوی باشد	x >= 4	true
<=	کوچک‌تر یا مساوی	طرف سمت چپ عملگر از سمت راست عملگر کوچک‌تر باشد	x <= 4	true



عملگرهای منطقی

عملگر	جواب true خواهد بود اگر ...	مثال	نتیجه
	اگر حداقل یکی از طرفین عملگر true باشد	True false	true
&&	اگر هر دو طرف عملگر true باشد	True && false	false
!	فقط بر روی یک عملوند اعمال می‌شود در صورتی که آن عملوند false باشد	! false	true

تابع prompt

- تابع از پیش تعریف شده prompt پیامی را در یک کادر popup به دید کاربر می‌رساند و از کاربر جواب می‌گیرد در مثال زیر برنامه از کاربر می‌خواهد نام خود را وارد کند و سپس از طریق alert به کاربر سلام می‌گوید

```
var name = prompt("لطفا نام خود را وارد کنید");  
alert("سلام " + name);
```



127.0.0.1:50225 says:

لطفا نام خود را وارد کنید

مهدی

OK

Cancel



127.0.0.1:50225 says:

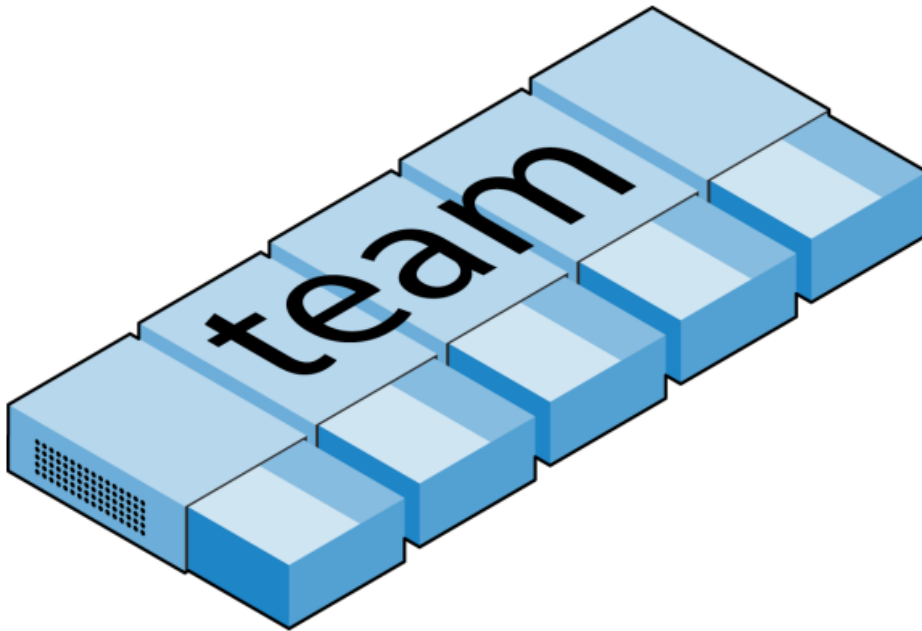
سلام مهدی

OK



JavaScript

آرایه



- آرایه متغیری است که چندین مقدار را در خود جای می‌دهد
- متغیر تیم آرایه ای است که از پنج بازیکن تشکیل شده:
رونالدو ، مسی، دایی،
عابدزاده، مجیدی



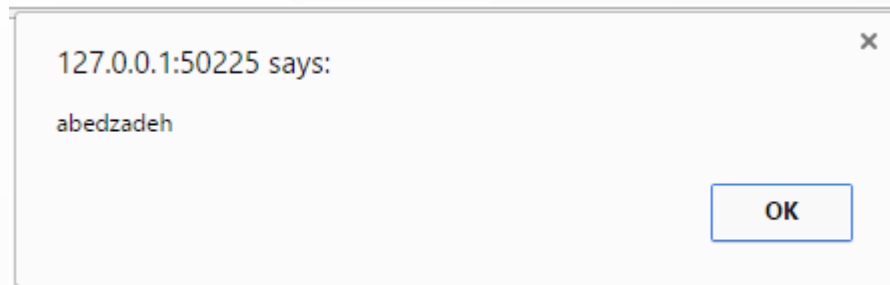
JavaScript

```
var team=["ronaldo","messi","daii",  
          "abedzadeh","majidi"];
```

دسترسی به عناصر آرایه با اندیس

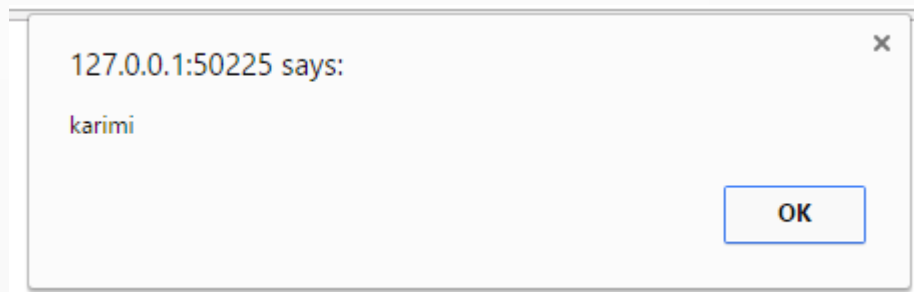
- دسترسی به عناصر آرایه با اندیس ها: در آرایه تیم ما پنج عنصر داریم که اولین آن رونالدو در مکان شماره صفر و آخرین آن مجیدی در مکان شماره 4 قرار دارد

```
var team = ["ronaldo", "messi", "dae",  
            "abedzadeh", "majidi"];  
alert(team[3]);
```



- می توان مقادیر عناصر آرایه را تغییر داد

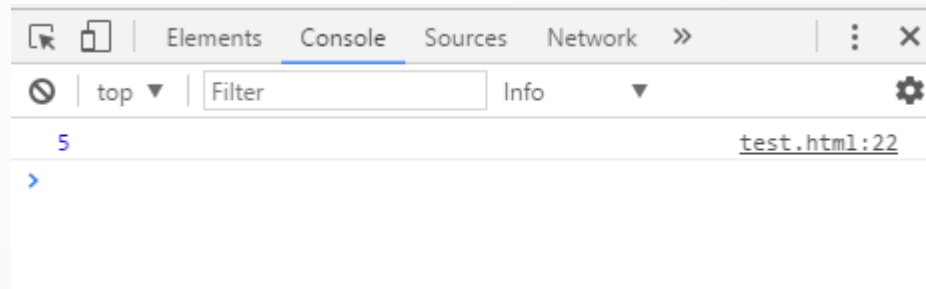
```
team[4] = "karimi";  
alert(team[4]);
```



برشمردن تعداد عناصر آرایه

- برای آنکه تعداد عناصر آرایه را بدست آوریم از یک property بنام length استفاده می کنیم لازم به ذکر است آرایه ها خود یک شی از پیش تعریف شده هستند ، در اشیاء برای دسترسی به property یک شی از نقطه (.) استفاده می کنیم (در ادامه با ویژگیهای اشیاء بیشتر آشنا خواهیم شد .)
- در مرورگر کروم قسمتی وجود دارد بنام کنسول که با فشار دادن کلیدهای (ctrl + shift+j) بطور همزمان این قسمت پدیدار می شود و می توانیم مستقلا دستورات جاوا اسکریپت را انجا اجرا کنیم. از داخل کد های صفحه جاوااسکریپتمان هم می توانیم خروجی به کنسول بفرستیم

```
var team = ["ronaldo", "messi", "daei",  
            "abedzadeh", "majidi"];  
console.log(team.length);
```

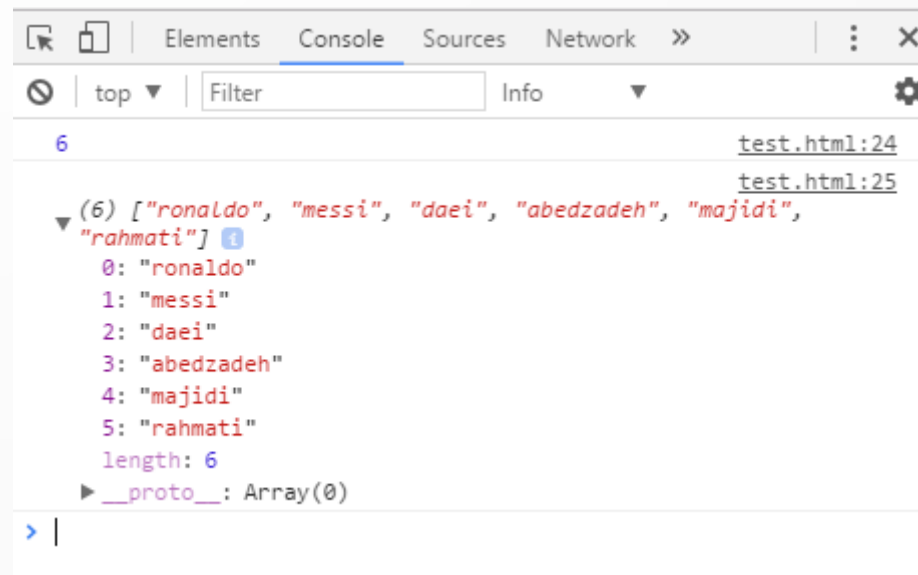


Log یک متد از شی از پیش تعرف شده console می باشد با نقطه می توان به متدهای یک شی دسترسی پیدا نمود این متد ارگومان ورودی که می گیرد را در کنسول چاپ می کند

افزودن یک عنصر به آرایه

- چون از طریق `length` می توانیم به تعداد عناصر یک آرایه دسترسی پیدا کنیم و از آنجا که اندیس آخرین عنصر یک آرایه یکی کمتر از تعداد عناصر آرایه می باشد لذا اگر بخواهیم عنصر جدیدی به آرایه مان اضافه نماییم باید عنصر جدید اندیس به مقدار `length` داشته باشد

```
var team = ["ronaldo", "messi", "daei",  
            "abedzadeh", "majidi"];  
var len = team.length;  
team[len] = "rahmati";  
console.log(team.length);  
console.log(team);
```

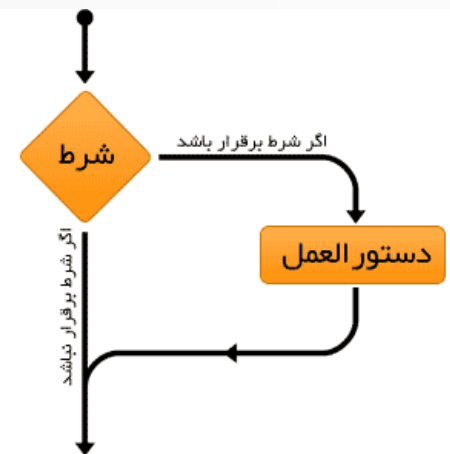


دستورات شرطی

- دستورات شرطی و حلقه ها جزو عناصر اصلی و جدایی ناپذیر هر زبان برنامه نویسی ساختار یافته هستند و تقریباً در تمامی برنامه ها کاربرد دارند. دستورات شرطی با برقراری شرط خاصی، مجموعه ای از دستورات عملی را اجرا میکنند.

- **دستور شرطی if**

- نحوه ی استفاده از دستور if به این شکل است که ابتدا شرط را بیان میکنیم و سپس با استفاده از آکولاد باز " { " و آکولاد بسته " } " دستورات عملی را بیان میکنیم. به ساختار زیر توجه کنید:



```
<?php
if (شرط) {
    دستور العمل
}
?>
```



مثال برای دستور شرطی if

- در مثال مقابل ابتدا نمره یک دانشجو را می گیریم در صورتی که بالاتر از 10 باشد به او تبریک می گوئیم

```
var mark = prompt("لطفا نمره خود را وارد کنید");  
if (mark >= 10)  
    console.log("تبریک شما قبول شدی");
```

در صورتی که کاربر نمره ای کمتر از 10 وارد کند در کنسول چیزی چاپ نمیشود.



127.0.0.1:50225 says:

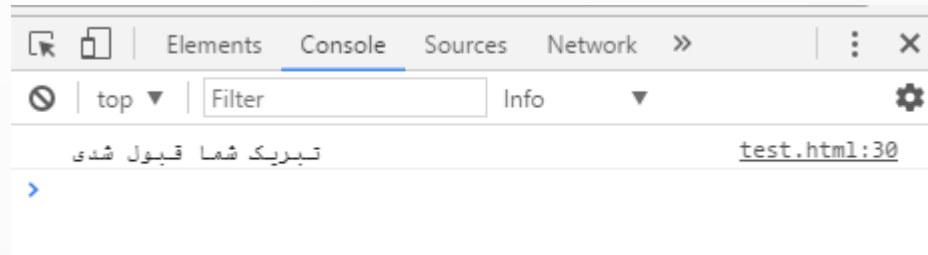


لطفا نمره خود را وارد کنید

13

OK

Cancel



JavaScript

دستور if..else

- فرق این دستور با دستور if در این است که شما حالت دومی را نیز برای شرط در نظر دارید، یعنی اگر شرط اول برقرار بود، دستوالعمل مناسب اجرا میشود ولی اگر شرط برقرار نبود دستوالعمل جایگزین اجرا میشود (بر خلاف دستور if که اگر شرط اول برقرار نبود هیچ دستوالعملی اجرا نمیشد).

```
var mark = prompt("لطفا نمره خود را وارد کنید");  
if (mark >= 10)  
    console.log("تبریک شما قبول شدی");  
else  
    console.log("متاسفم شما رد شدی");
```

127.0.0.1:50225 says:

لطفا نمره خود را وارد کنید

16

OK

Cancel

127.0.0.1:50225 says:

لطفا نمره خود را وارد کنید

6

OK

Cancel

تبریک شما قبول شدی



متاسفم شما رد شدی



JavaScript

Else if تو در تو

- اگر برای چک کردن مقدار یک عبارت باید چندین شرط چک شود در حالی که فقط یکی از آن شرطها جواب مورد نظر ما را داشته باشد می توانیم از else if های تو در تو استفاده کنیم

```
var abcd;  
var mark = prompt("لطفا نمره خود را وارد کنید");  
if ((mark < 5) && (mark >= 0))  
    abcd = "D";  
else if ((mark >= 5) && (mark < 10))  
    abcd = "C";  
else if ((mark >= 10) && (mark < 15))  
    abcd = "B";  
else if ((mark >= 15) && (mark <= 20))  
    abcd = "A";  
else  
    abcd = "wrong number";  
console.log(abcd);
```

- مثال از یک دانشجو نمره اش را در سیستم 0 تا 20 می گیریم و تبدیل به سیستم abcd می کنیم



دستور switch

- دستور switch در واقع بر حسب شرایط مختلف عملیات متفاوتی را انجام می دهد یا به عبارتی دیگر از این ساختار هنگامی استفاده می شود که لازم باشد بر حسب مقادیر مختلف یک متغیر یا عبارت خاص، دستورات متفاوتی اجرا گردد.

```
switch(expression) {  
  case n:  
    code block  
    break;  
  case n:  
    code block  
    break;  
  default:  
    default code block  
}
```

- رویه ی اجرای این دستور به شرح زیر می باشد.

عبارت switch ابتدا یک بار ارزیابی می گردد.

سپس مقدار عبارت با مقادیر هر یک از case ها به طور جداگانه مقایسه می گردد.

چنانچه مقدار عبارت با یکی از case ها مطابقت و همخوانی داشت، قطعه کد مربوطه اجرا می گردد.



مثال switch

```
var abcd;  
var mark = prompt("لطفا نمره خود را وارد کنید");  
console.log(mark);  
if ((mark < 5) && (mark >= 0))  
    abcd = "D";  
else if ((mark >= 5) && (mark < 10))  
    abcd = "C";  
else if ((mark >= 10) && (mark < 15))  
    abcd = "B";  
else if ((mark >= 15) && (mark <= 20))  
    abcd = "A";  
else  
    abcd = "wrong number";  
console.log(abcd);  
switch (abcd) {  
    case "A":  
        console.log("نمره ممتاز");  
        break;  
    case "B":  
        console.log("نمره خوب");  
        break;  
    case "C":  
        console.log("نمره ضعیف");  
        break;  
    case "D":  
        console.log("نمره بد");  
        break;  
    default:  
        console.log("نمره را اشتباه وارد کرده اید باید یک عدد بین 0 تا 20 وارد کنید");  
}
```



JavaScript

- مثال قبلی را با یک switch تکمیل کردیم

127.0.0.1:50225 says:

لطفا نمره خود را وارد کنید

12

OK

Cancel

	Elements	Console	Sources	Network	»		
⌵	top ▾	Filter	Info ▾				⚙
12						test.html:30	
B						test.html:41	
نمره خوب						test.html:47	
>							

دستورات حلقه - while

- در بسیاری از مواقع ما نیاز به تکرار اجرای یک یا چند دستور داریم برای این منظور از حلقه ها در برنامه نویسی استفاده می نماییم .

- **دستور while :**

```
while(شرط)
{
    دستورات برای تکرار
}
```

- حلقه while با کلمه کلیدی while شروع شده و در مقابل آن در داخل پرانتز شرطی که برای اجرای حلقه در نظر داریم را می نویسیم.
- حلقه while تا زمانی اجرا می شود که شرط داخل پرانتز مقدار true برگرداند.
- دستورات مرتبط با حلقه که باید تکرار شوند را داخل {} می نویسیم
- دستوراتی که داخل {} نوشته می شوند، می توانند یک یا چندین دستور باشند.
- دستورات مربوط به حلقه while در صورت برقرار نبودن شرط در ابتدای حلقه اجرا نخواهند شد.



دستورات حلقه - while

- یک مثال ساده برای چاپ اعداد 1 تا 10 با استفاده از حلقه while در کنسول

```
var counter=1;
while(counter<=10)
{
    console.log(counter);
    counter++;
}
```



1
2
3
4
5
6
7
8
9
10
>

- چاپ عناصر یک آرایه در با استفاده از حلقه while

```
var person=["saeid","irani",18],index=0;
while(index<person.length)
    console.log(person[index++]);
```



saeid
irani
18



دستورات حلقه do while-

- در حلقه while، شرط اجرای حلقه قبل از رسیدن به دستورات بررسی می شود، اما در ساختار do..while شرط در انتهای حلقه بررسی می شود، به همین خاطر و بر خلاف دستور while، دستورات حلقه حداقل یکبار اجرا می شوند. در زیر ساختار کلی حلقه do..while را مشاهده می کنید:

```
do
{
    // دستورات برای تکرار
} while(شرط)
```

```
var i=1;
do{
    console.log(i++);
}while(i<=10)
```



1
2
3
4
5
6
7
8
9
10
>

- در کد مقابل متن تشکر از خدا چاپ شده و به شرط حلقه که میرسد چون شرط برقرار نیست از حلقه خارج می شود



```
var godExist=true;
do{
    console.log("Thanks to god.");
}while(!godExist)
```

حلقه for

```
for(init; condition; increment)
{
    // دستورات برای تکرار
}
```

- ساختار کلی دستور for به صورت مقابل است:
همانطور که مشاهده می کنید این حلقه با کلمه کلیدی for آغاز شده و داخل پرانتز و مقابل کلمه کلیدی for سه بخش داریم که با کاراکتر ; از هم جدا شده اند.
- بخش init: متغیری که داخل حلقه استفاده می شود را تعریف و مقدار دهی اولیه می کنیم
- بخش condition: شرطی را مشخص می کند که حلقه بر اساس آن اجرا خواهد شد
- بخش increment: مشخص می کنید با هر بار تکرار، متغیری که در بخش init تعریف شده است چه افزایشی پیدا می کند .
- چاپ عناصر یک آرایه (کشورهای خاورمیانه) با استفاده از حلقه for:

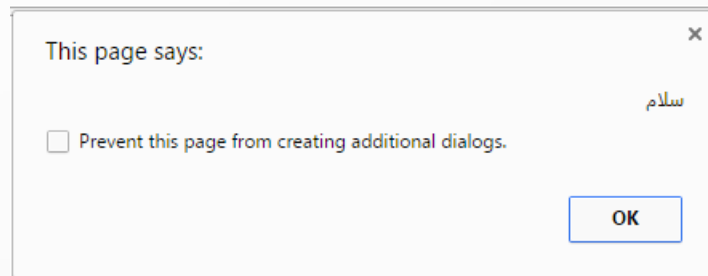


```
var middleEast=['iran','iraq','egypt','turkey'];
for(var i=0;i<middleEast.length;i++)
    console.log(middleEast[i]);
```

تابع

- در برنامه نویسی میتوانیم از توابع استفاده کنیم تا هم به ساختار کد نظم ببخشند و هم اینکه از تکرار کد جلوگیری کنیم. هنگامی که یک تابع تعریف میشود ما میتوانیم بدون توجه به ساختار داخلی آن بارها از آن استفاده کنیم. در واقع میتوان به آن به عنوان یک ماشین نگاه کرد که می توان با دادن ورودی به آن از آن خروجی مورد نظر را انتظار داشت .
- یک مثال ساده تابعی که وقتی فراخوانی می شود به بازدید کننده صفحه سلام می کند

```
function sayHello()  
{  
    alert("سلام");  
}  
sayHello();// نحوه فراخوانی تابع
```

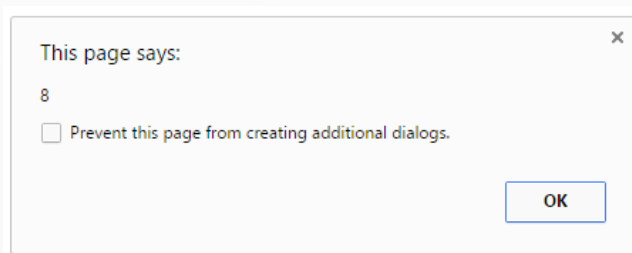


تابع با مقدار بازگشتی

- در بیشتر مواقع به توابعی نیاز داریم که یکسری مقدار ورودی می گیرند و یک مقدار را بر می گردانند

```
function add (num1,num2)
{
    var sum=num1+num2;
    return sum;
}
```

نحوه فراخوانی تابعی که دو مقدار ورودی می گیرد // `var z=add(5,3);`
`alert(z);`



تابعی که جمع دو
عددی را که
بصورت ورودی
گرفته حساب
می کند و با
استفاده از کلمه
کلیدی return
مقدار را بر می
گرداند

مقدمه‌ای بر اشیا در جاوا اسکریپت

در تصویر 3 شی
مشخص شده یک 1
هتل و 2 خودرو



JavaScript

مقدمه‌ای بر اشیا در جاوا اسکریپت

شی هتل property

های زیر را در اختیار دارد:

- نام هتل
- درجه
- تعداد اتاق
- اتاقهای رزرو شده
- آیا اتاق ژیمناستیک دارد
- یا خیر
- استخر دارد یا خیر



شی خودرو خصوصیات

یا property های زیر

را در اختیار دارد:

- شرکت سازنده
- سرعت فعلی
- رنگ
- نوع سوخت



JavaScript

مقدمه‌ای بر اشیا در جاوا اسکریپت

- رویدادهای شی هتل :
- رزرو اتاق
 - کنسل کردن رزرو اتاق



- رویدادهای شی ماشین :
- ترمز گرفتن
 - گاز دادن

مقدمه‌ای بر اشیا در جاوا اسکریپت

متدهای شی هتل:

makeBooking():

مقدار property **اتاقهای رزرو شده** را افزایش می‌دهد

cancelBooking():

مقدار property **اتاقهای رزرو شده** را کاهش می‌دهد

checkAvailability():

تعداد اتاق های هتل را منهای **تعداد اتاقهای رزرو شده** می‌کند و تعداد اتاقهای خالی را بر میگرداند



JavaScript



متدهای شی ماشین:

ChangeSpeed():

مقدار **سرعت فعلی ماشین** را کاهش یا افزایش می‌دهد

مقدمه‌ای بر اشیا در جاوا اسکریپت

در اینجا ابتدا رویداد **رزرو اتاق** رخ می‌دهد که این رویداد متد **makeBooking** را فراخوانی کرده که در نتیجه این متد مقدار **property** **اتاقهای رزرو شده** را افزایش می‌دهد

OBJECT TYPE: HOTEL			
EVENT	happens when:	method called:	PROPERTIES
1 book	reservation is made	makeBooking()	name Quay
cancel	reservation is cancelled	cancelBooking()	rating 4
			rooms 42
2 makeBooking()	increases value of <i>bookings</i> property		3 bookings 22
cancelBooking()	decreases value of <i>bookings</i> property		gym false
checkAvailability()	subtracts value of <i>bookings</i> property from value of <i>rooms</i> property and returns number of rooms available		pool true



مقدمه‌ای بر اشیا در جاوا اسکریپت

در اینجا ابتدا رویداد **گاز دادن** (فشار پا بر روی پدال گاز) روی میدهد که این رویداد متد **changeSpeed** را فراخوانی کرده و این متد مقدار **property سرعت فعلی** ماشین را افزایش می دهد



مقدمه‌ای بر اشیا در جاوا اسکریپت

هر مرورگر برای هر tab یا پنجره جدید از یک شی بنام window استفاده می‌کند که همانطور که در شکل پیداست یک property بنام location دارد که URL صفحه وب را نگهداری میکند



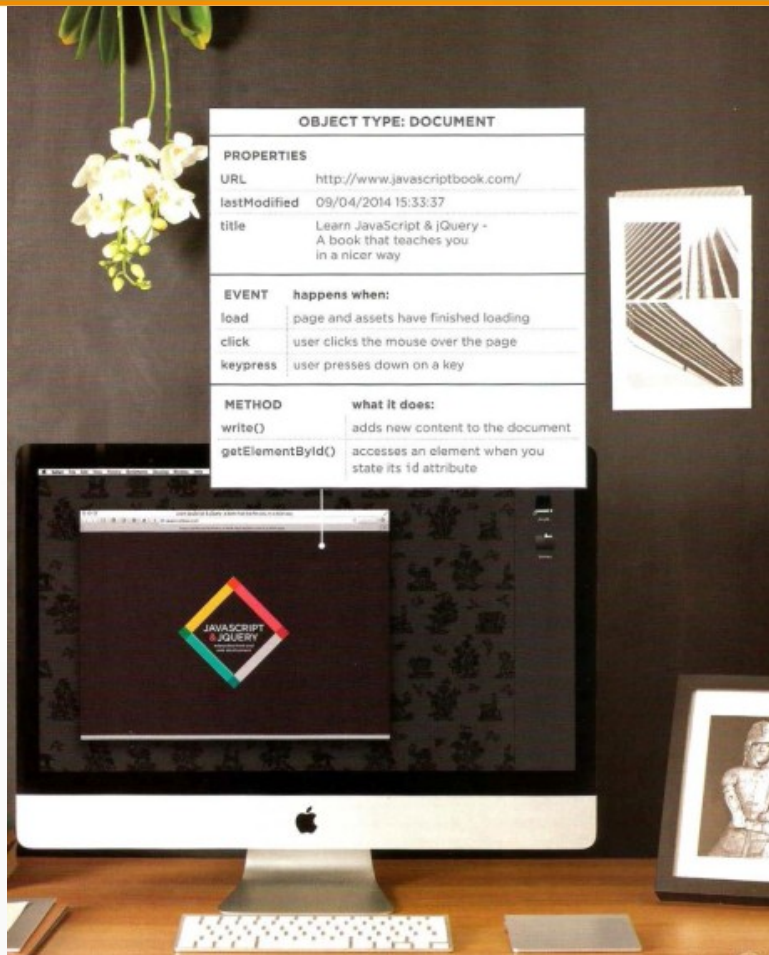
هر شی window برای نشان دادن محتویات صفحه وب خود از شی document استفاده می‌کند که دارای property های زیر است :

- URL
- title
- lastModified

مقدمه‌ای بر اشیا در جاوا اسکریپت

برای شی document می‌توان رویدادهای زیر را بر شمرد:

- load:
زمانی که صفحه وب بصورت کامل بالا بیاید یا اصطلاحاً لود شود
- click:
زمانی که با موس بر روی صفحه وب کلیک شود
- keypress:
زمانی که بر روی کلیدهای صفحه‌کلید ضربه زده می‌شود



برای شی document می‌توان متدهای زیر را برشمرد:

- write():
محتوای جدید به document اضافه می‌کند
- getElementById():
دسترسی به عناصر صفحه وب با استفاده از خاصیت id آنها

چگونگی نمایش صفحه وب در مرورگر

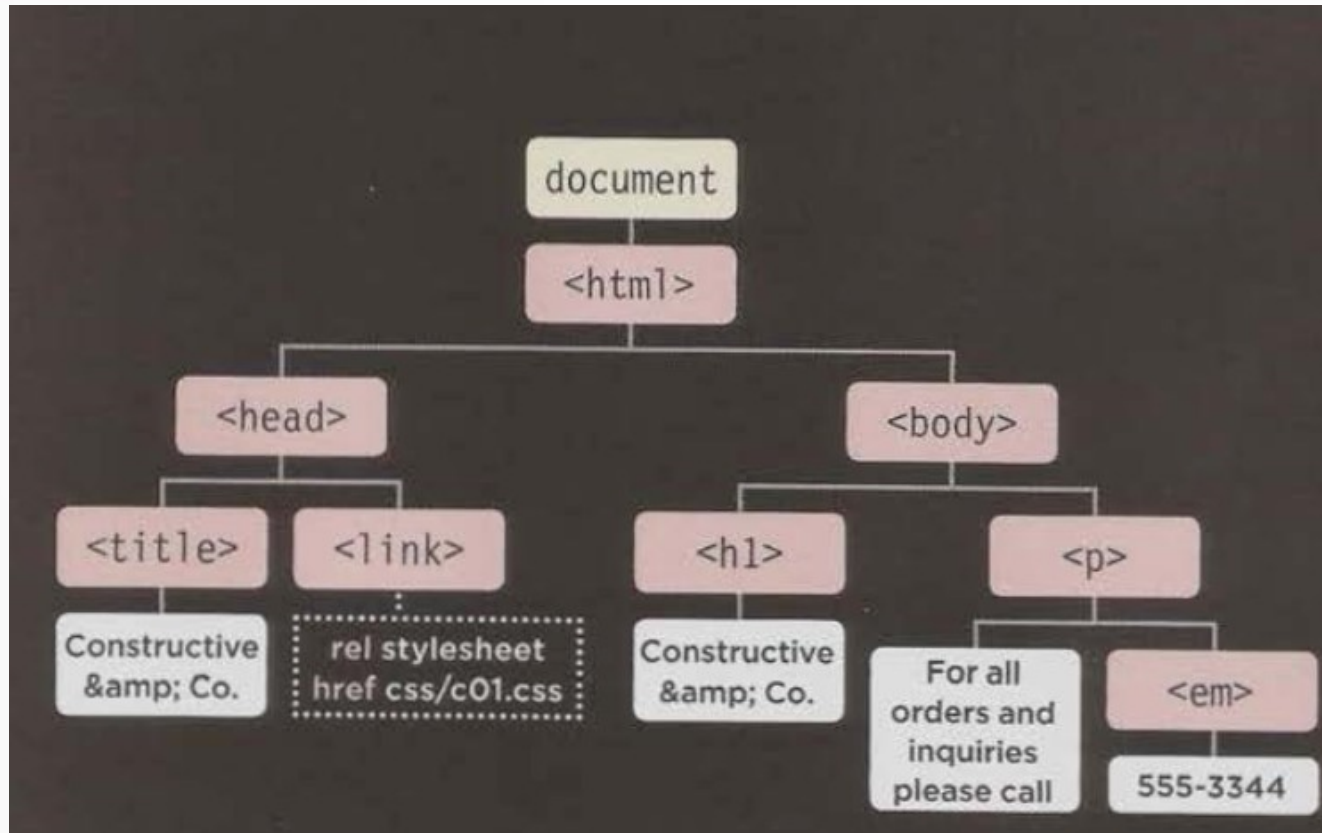


مرورگر صفحه
را html
دریافت می‌کند

```
<!DOCTYPE html>
<html>
  <head>
    <title>Constructive & Co.</title>
    <link rel="stylesheet" href="css/c01.css" />
  </head>
  <body>
    <h1>Constructive & Co.</h1>
    <script src="js/add-content.js"></script>
    <p>
      For all orders and inquiries please call
      <em>555-3344</em></p>
  </body>
</html>
```



چگونگی نمایش صفحه وب در مرورگر



مدل سلسله مراتبی و
درختی اش را ایجاد و
در حافظه ذخیره
می‌کند



چگونگی نمایش صفحه وب در مرورگر



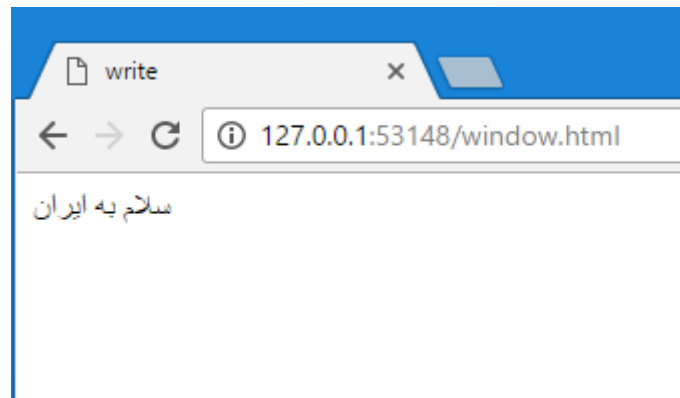
صفحه وب
نمایش داده
می شود



JavaScript

متد write

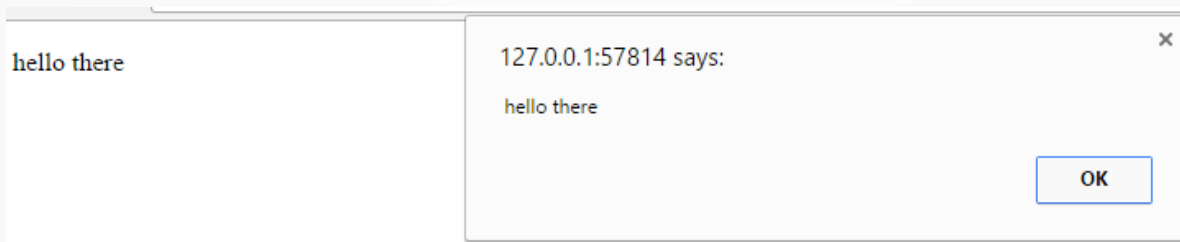
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>write</title>
</head>
<body>
<script>
document.write("سلام به ایران");
</script>
</body>
</html>
```



innerHTML و getElementById

- همانطور که قبلا ذکر شد با این متد شی document می توانیم به یک عنصر (تگ) صفحه وب با استفاده از ID آن عنصر دسترسی پیدا کنیم
- با استفاده از property innerHTML (خصوصیت) innerHTML می توانیم به محتوای عنصر دسترسی پیدا کنیم

```
<body>  
  <p id="p1"> hello there</p>  
  <script>  
    var x = document.getElementById("p1");  
    alert(x.innerHTML);  
  </script>  
</body>
```



تغییر محتوای عنصر

```
<body>
  <p id="p1"> hello there</p>
  <script>
    var x = document.getElementById("p1");
    x.innerHTML = "سلام";
  </script>
</body>
```

- میتوانیم محتوای عنصر را تغییر دهیم



سلام

- در ضمن می شود از تگهای html هم برای تغییر محتوای تگ انتخاب شده استفاده کرد

```
<body>
  <p id="p1"> hello there</p>
  <script>
    document.getElementById("p1").innerHTML = "<h1> سلام</h1>";
  </script>
</body>
```



سلام



متد `getElementsByTagName`

- این متد تمام تگهای مشخص شده را در یک آرایه جمع آوری میکند (بر اساس ترتیب حضور در صفحه)
در نتیجه با اندیس می توانیم به هر کدام از عناصر آرایه دسترسی داشته باشیم

```
<body>
  <ul>
    <li>iran</li>
    <li>iraq</li>
    <li>ireland</li>
  </ul>
  <script>
    document.getElementsByTagName('li')[2].innerHTML = "spain";
  </script>
</body>
```



- iran
- iraq
- spain



متد getElementByClassName

- توسط متد getElementByClassName میتوان عناصر را با توجه به نام کلاس انتخاب کرد و چون عناصر مختلفی در صفحه امکان گرفتن نام یک کلاس را دارند پس در یک آرایه گردآوری شده (بر اساس ترتیب حضور در صفحه) لذا با اندیس می توانیم به هر کدام از عناصر آرایه دسترسی داشته باشیم

```
<body>
  <p class="iran"> iran</p>
  <p> iraq </p>
  <h1 class="iran"> i love iran </h1>
  <script>
    document.getElementsByClassName("iran")[1].innerHTML = "iran is beautiful";
  </script>
</body>
```



iran

iraq

iran is beautiful



متد querySelectorAll

- توسط متد querySelectorAll میتوان مانند دستورات CSS به انتخاب عناصر پرداخت

```
<body>
  <p class="iran"> iran</p>
  <p id="iraq"> iraq </p>
  <h1 class="iran"> i love iran </h1>
  <script>
    document.querySelectorAll(".iran")[1].innerHTML = "ایران";
    document.querySelectorAll("#iraq")[0].innerHTML="عراق";
  </script>
</body>
```



iran
عراق
ایران



متد querySelector

- فرق این متد با متد querySelectorAll در این است که اولین عنصری را که پیدا کند انتخاب میکند و برمیگرداند پس نتیجه انتخاب بصورت آرایه نیست

```
<body>
  <ul>
    <li>iran</li>
    <li>iraq</li>
    <li>ireland</li>
  </ul>
  <script>
    document.querySelector("li").innerHTML = "ایران";
  </script>
</body>
```



```
• ایران
• iraq
• ireland
```



Property : style

- میتوان خصوصیت (property) style - عنصر انتخاب شده را تعیین نمود :

```
<body>
  <ul>
    <li>iran</li>
    <li>iraq</li>
    <li>ireland</li>
  </ul>
  <script>
    document.getElementsByTagName("li")[1].style.color="red";
  </script>
</body>
```



- iran
- iraq
- ireland

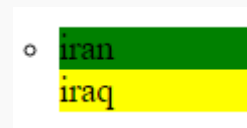
در مثال بالا عنصر li با شماره اندیس 1 انتخاب شده و خصوصیت رنگ فونت آن فرمز تعیین شده است .



Style property

- برای استایل دادن به عنصر انتخاب شده توسط جاوااسکریپت باید از همان عبارات CSS استفاده نمود فقط زمانی که این عبارات از ترکیب چند کلمه ساخته شده اند خط فاصله (-) بینشان حذف شده و بصورت camelcase نوشته می شوند بطور مثال background-color در CSS بصورت زیر در جاوااسکریپت نوشته می شود :
backgroundColor

```
<style>
  li{
    list-style-type: circle;
    background-color: green;
  }
</style>
</head>
<body>
  <ul>
    <li>iran</li>
    <li>iraq</li>
  </ul>
  <script>
    document.getElementsByTagName("li")[1].style.listStyleType="none";
    document.getElementsByTagName("li")[1].style.backgroundColor="yellow";
  </script>
</body>
```



متد `getAttribute`

- متد `getAttribute` مقدار attribute عنصر انتخاب شده را بر می گرداند:

```
<body dir="ltr">  
  hello world!  
  <script>  
    var dir = document.getElementsByTagName("body")[0].getAttribute("dir");  
    document.write("<h3> Direction: "+ dir+"</h3>");  
  </script>  
</body>
```



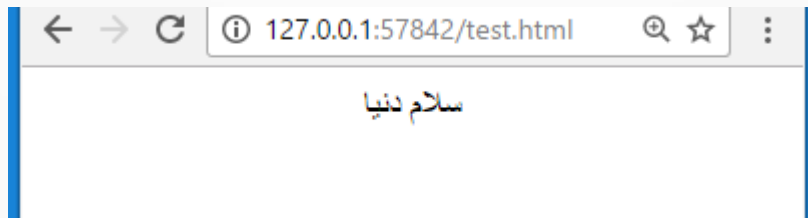
hello world!
Direction: ltr



متد setAttribute

- متد setAttribute برای attribute عنصر انتخاب شده مقدار تعیین می کند:

```
<body id="mybody" dir="ltr">  
سلام دنیا  
<script>  
    document.getElementById("mybody").setAttribute("dir","rtl");  
    document.getElementById("mybody").setAttribute("align","center");  
  
</script>  
</body>
```



Property :className

- با استفاده از خصوصیت className می توان برای عنصر انتخاب شده کلاس تعیین کرد :

```
<style>
    .green-blue{
        color :greenyellow;
        background:blue;
    }
</style>
</head>
<body>
    <ul>
        <li>iran</li>
        <li>iraq</li>
    </ul>
    <script>
        document.querySelector("li").className = "green-blue";
    </script>
</body>
</html>
```



- iran
- iraq



رویداد یا event

- رویداد یا event به رخدادهایی گفته می شود که در صورت رخ دادن یک تابع فراخوانی می شود
- رویداد mouseover : زمانی که موس بر روی عنصر مورد نظر برود این رویداد رخ می دهد
- رویداد mouseout : زمانی که موس از روی عنصر مورد نظر خارج شود این رویداد رخ می دهد
- رویداد click : زمانی که کاربر بر روی عنصر مورد نظر کلیک کند این رویداد رخ می دهد
- شیوه اول تعریف رویداد : رویداد داخل تگ html :

```
<body >
  <ul>
    <li id="iran" onmouseover="changeCountry()">iran</li>
    <li>iraq</li>
  </ul>
  <script>
    function changeCountry()
    {
      document.getElementById("iran").innerHTML= "ایران";
    }
  </script>
</body>
```

قبل از اینکه موس بر روی متن برود

- iran
- iraq

بعد از اینکه موس بر روی متن قرار گرفت

- ایران
- iraq



رویداد event

- شیوه دوم : استفاده از addEventListener

```
<style>
    .beautiful{
        color:chocolate;
        background-color: aquamarine;
    }
</style>
</head>
<body >
    <ul>
        <li id="iran" onmouseover="changecountry()" >iran</li>
        <li>iraq</li>
    </ul>
    <script>
        function changecolor()
        {
            document.getElementById("iran").className= "beautiful";
        }
        function changecountry()
        {
            document.getElementById("iran").innerHTML= "ایران";
            document.getElementById("iran").addEventListener("mouseout",changecolor);
        }
    </script>
</body>
```

- iran
- iraq

mouseover:

- ایران
- iraq

mouseout:

- ایران
- iraq



JavaScript

setInterval

- متد () setInterval به صورت متناوب و در فواصل زمانی معین، یک تابع را اجرا می کند
- این متد از متدهای شی window می باشد ولی می توان آنرا بدون پیشوند window نیز بکاربرد.

```
<script>
window.setInterval(sayhello,3000);
function sayhello(){
  alert("hello");
}
</script>
```

- این اسکریپت هر 3 ثانیه یکبار تابع alert را فراخوانی میکند



setInterval

- برنامه چاپ اعداد با استفاده از متد setInterval

در این برنامه تا زمانی که مرورگر را نبسته ایم اعداد به ترتیب بر روی صفحه وب ظاهر می شوند

```
<body>
  <script>
    setInterval(counter,200);
    var i= 0;
    function counter(){
      document.write (i++ +" ");
    }
  </script>
</body>
```



متد setInterval

- در هنگام کار با () setInterval ، متوجه شدید که این تابع به صورت متناوب و دوره ای ، یک کد یا تابع را پس از گذشت مدت زمان تعیین شده برای آن ، اجرا می کند . اگر کاربر یا برنامه این تابع را متوقف نسازد ، تا زمانی صفحه یا برنامه تحت وب در حال اجراست ، این تابع نیز اجرا شده و به کار خود ادامه خواهد داد .
- اما زمانی می رسد که شاید شما بخواهید پروسه این تابع را متوقف سازید .
- برای اینکه بتوانید سیر عملیات یک تابع () setInterval را متوقف سازید ، باید یک متغیر سراسری را در صفحه تعریف کرده و تابع () setInterval خود را در آن بریزید .
- مثال :

```
MyStop = setInterval ( " MyFunction " , 1000 )
```

clearInterval

- پس از اینکه تابع () setInterval موردنظر خود را در یک متغیر تعریف کرده اید ، هر زمان که بخواهید آن تابع را متوقف سازید ، باید متغیر مرتبط به آن را به عنوان یک پارامتر ، به تابع () clearInterval ، ارسال نمایید . در این صورت تابع از تکرار و تناوب باز خواهد ماند .
clearInterval (MyStop) ;

```
<body>
  <script>
    var numbers = setInterval(counter,200);
    var i= 0;
    function counter(){
      document.write (i++ +" ");
      if (i == 11)
        clearInterval(numbers);
    }
  </script>
</body>
```

در اینجا فقط اعداد 0 تا 10 چاپ می شوند .

