

آموزش جاوا اسکریپت

جاوا اسکریپت یک زبان اسکریپت نویسی تحت وب است که برای افزایش پویایی صفحات HTML استفاده می شود. از این زبان می توان برای ارزیابی اعتبار فرمهای HTML قبل از ارسال به سرور، شناسایی کاربران، ایجاد انواع منو ها و بسیاری کارهای دیگر استفاده کرد.

نکته : باید توجه داشت که جاوا اسکریپت یک زبان طرف کاربر (Client Side) است توسط کامپیوتر شخصی کاربر تفسیر می شود. کدهای مربوط به جاوا اسکریپت توسط کاربر نهایی قابل مشاهده هستند پس توصیه می شود از آن برای مقاصد مهم نظیر اتصال به دیتابیس استفاده نشود.

می توان کدهای جاوا اسکریپت را به صورت تعبیه شده در صفحات HTML استفاده کرد و یا کدهای جاوا اسکریپت را در یک فایل جداگانه با پسوند .js ذخیره کرد ولی در هر صورت آن فایلها هم باید توسط یک صفحه HTML فراخوانی شده و نتیجه آنها در صفحات HTML نمایان خواهد شد.

بهتر است قبل از شروع به یادگیری جاوا اسکریپت با HTML آشنایی داشته باشید.

برای آزمایش کردن اسکریپتهایی که می نویسید باید از مرورگری استفاده کنید که از جاوا اسکریپت پشتیبانی می کند. قابل ذکر است که تقریباً همه مرورگرهای جدید از جاوا اسکریپت پشتیبانی می کنند از جمله این مرورگرها می توان به اینترنت اکسپلورر (IE) که در ویندوز به صورت پیش فرض نصب می شود، فایرفوکس (Firefox) و اپرا (Opera) اشاره کرد. به خاطر داشته باشید که نسخه های قبل از نسخه 4.0 اینترنت اکسپلورر و Netscape از جاوا اسکریپت پشتیبانی نمی کنند.

نحوه استفاده از کدهای جاوا اسکریپت در صفحه

به طور کلی به سه طریق می توانیم از جاوا اسکریپت در یک صفحه استفاده کنیم:

1. وارد کردن اسکریپت در بخش **<head>** صفحه
2. وارد کردن جاوا اسکریپت در بخش **<body>** صفحه
3. استفاده از فایل اسکریپت خارجی

به یاد داشته باشید که می توانید از همه روشهای بالا به طور همزمان برای وارد کردن جاوا اسکریپت در صفحه استفاده کنید.

در دو روش اول کدهای جاوا اسکریپت به طور مستقیم وارد صفحه HTML می شوند. برای وارد کردن جاوا اسکریپت در این دو روش از تگ **script** استفاده می شود. تمامی کدهای جاوا اسکریپتی که می خواهید به طور مستقیم وارد صفحه کنید باید در تگ **script** قرار داشته باشند وگرنه مرورگر به جای به کار بردن آنها را به صورت متن در صفحه نمایش می دهد. معمولاً کدهای جاوا اسکریپت با ساختار زیر وارد صفحه می شوند :

```
<script language="javascript" type="text/javascript">
```

دارید در این قسمت وارد می شوند دستورات جاوا اسکریپتی که قصد استفاده از آنها را

```
</script>
```

در تگ بالا برای مشخص کردن نوع موتور اسکریپتی که مرورگر باید برای تفسیر کردن کد استفاده کند از دو شناسه **language** و **type** استفاده می کنیم.

وارد کردن اسکریپت در بخش <head>

در هنگام باز شدن یک صفحه وب عناصر موجود در بخش <head> قبل از سایر اجزای صفحه لود می شوند. معمولاً هنگامی کدهای جاوا اسکریپت در بخش <head> متن استفاده می شود که بخواهیم مطمئن شویم به هنگام نیاز کدهای ما لود شده اند.

در اینجا یک مثال از استفاده از جاوا اسکریپت در <head> را مشاهده می کنید:

```
<html dir="rtl">

<head>
<script language="javascript" type="text/javascript">
function message() {
document.write('<div align="center">در کدهای جاوا اسکریپت موجود در &lt;head&gt; کدهای جاوا اسکریپت موجود در &lt;div align="center">
<br> صفحه قبل از سایر اجزای صفحه لود می شوند. معمولاً هنگامی کدهای جاوا اسکریپت در بخش <head> متن استفاده می شود که بخواهیم مطمئن شویم به هنگام نیاز کدهای ما لود شده اند.
</div>');
}
</script>
</head>

<body>
<p align="center"><a href="javascript:message();">کنید روی این متن کلیک</a></p>
</body>

</html>
```

[برای مشاهده صفحه ساخته شده دستورات بالا را اجرا کنید.](#)

وارد کردن جاوا اسکریپت در قسمت بدنه متن HTML

این امکان وجود دارد که کدهای جاوا اسکریپت را در قسمت <body> صفحه وارد کنیم. در این صورت لود شدن آنها تا هنگامی که آن بخش از صفحه لود شود به عقب می افتد و اگر صفحه به صورت کامل باز نشود ممکن است اسکریپت ما کار نکند. مثال زیر یک نمونه از وارد کردن جاوا اسکریپت در <body> صفحه HTML را نشان می دهد:

```
<html dir="rtl">

<head>
<title>صفحه جاوا اسکریپت در بدنه</title>
</head>
```

```
<body>
<script language="javascript" type="text/javascript">
document.write('<br><h1 align="center">شده اسکرپت نوشته شده</h1>');
</script>
</body>

</html>
```

[برای مشاهده صفحه ساخته شده دستورات بالا را اجرا کنید.](#)

فرا خوانی اسکرپت خارجی به صفحه

گاهی اوقات ممکن است بخواهید از یک سری از کدهای جاوا اسکرپت برای چندین صفحه استفاده کنید و نمی خواهید همه آنها را به صورت جداگانه در هر صفحه بنویسید. در این مواقع می توانید کدهای مورد نظر خود را در یک فایل متنی وارد کنید و آنرا با پسوند .js ذخیره کنید. در هر صفحه ای که بخواهید از این فایل استفاده کنید کافی است آنرا در آن صفحه فرا خوانی کنید. (به یاد داشته باشید که در این فایلهای خارجی نباید از تگهای <script> که مخصوص HTML هستند استفاده کنید)

برای فرا خوانی این فایلها به صفحه از همان تگ **script** که در موارد قبل استفاده می کردیم کمک می گیریم با این تفاوت که در اینجا به آن یک شناسه **src** که آدرس اسکرپت ما به عنوان مقدار آن قرار گرفته است اضافه می کنیم.

برای مثال کد جاوا اسکرپت زیر را در نظر بگیرید :

```
var text = '<br><h1 align="center" dir="rtl" >متن با استفاده از جاوا اسکرپت نوشته شده است</h1>';
document.write(text);
```

ما این کد را در فایلی با نام external.js در همان پوشه ای که صفحه ما قرار دارد ذخیره کرده ایم. حالا با وارد کردن کد زیر در صفحه می توانیم فایل بالا را فراخوانی کنیم:

```
<script language="javascript" type="text/javascript" src="external.js"></script>
```

[برای مشاهده صفحه ساخته شده دستورات بالا را اجرا کنید.](#)

معرفی عملگرهای مورد استفاده در JavaScript

در جاوا اسکریپت برای جمع، تفریق، مقایسه و بسیاری از عملیات دیگر از عملگرها استفاده می کنیم. عملگرها به چند دسته تقسیم می شوند : عملگرهای محاسباتی، مقایسه ای، جایگزینی، منطقی، رشته ای، و شرطی. در ادامه به ارائه توضیحاتی برای هر نوع از این عملگرها خواهیم پرداخت.

عملگرهای محاسباتی

از این نوع عملگرها برای انجام عملیات ریاضی نظیر جمع و ضرب استفاده می شود. می توانید این عملگرها را در اینجا مشاهده کنید (در مثال زیر متغیر x را برابر 8 و متغیر y را برابر 4 در نظر بگیرید) :

نام	عملگر	مثال	نتیجه
جمع	+	$x + y$	12
تفریق	-	$x - y$	4
ضرب	*	$x * y$	32
تقسیم	/	x / y	2
باقیمانده	%	$9 \% y$	1
افزایش	++	x++	9
کاهش	--	x--	7

از عملگرهای افزایش و کاهش به ترتیب برای افزودن 1 به مقدار قبلی و کم کردن 1 از مقدار قبلی استفاده می شود.

عملگرهای جایگزینی

این عملگرها ترکیبی از عملگرهای دیگر هستند و به همین دلیل آنها را عملگرهای ترکیبی هم می نامند. می توانید این عملگرها را در جدول زیر مشاهده کنید (در مثال زیر متغیر x را برابر 8 و متغیر y را برابر 4 در نظر بگیرید) :

نام	عملگر	مثال	عبارت معادل
انتساب	=	x = 8	x = 8
انتساب جمع	+=	x += y	x = x + y
انتساب تفریق	-=	x -= y	x = x - y
انتساب ضرب	*=	x *= y	x = x * y
انتساب تقسیم	/=	x /= y	x = x / y
انتساب باقیمانده	%=	x %= y	x = x % y

عملگرهای مقایسه ای

از این عملگرها برای مقایسه دو مقدار استفاده می شود که شامل عملگرهای زیر هستند.

در جدول زیر مقادیر x و y و z را مطابق زیر در نظر بگیرید و توجه داشته باشید که مقدار متغیرهای x و y عددی و مقدار متغیر Z از نوع رشته ای است :

x = 4;
y = 8;
z = "4";

نام	عملگر	مثال	نتیجه
تساوی	==	x == z	True
همانی	===	x === z	False
نامساوی	!=	x != y	True

False	$x > y$	>	بزرگتر از
True	$x < y$	<	کوچکتر از
True	$x \geq z$	\geq	بزرگتر یا مساوی
True	$x \leq y$	\leq	کوچکتر یا مساوی

تفاوت عملگر تساوی با عملگر همانی در این است که در عملگر تساوی نوع متغیرها در نظر گرفته نمی شود و اگر مقدار متغیرها با هم برابر باشد حتی اگر از یک نوع هم نباشند نتیجه True خواهد بود. ولی در عملگر همانی باید متغیرها مقداری مساوی داشته باشند و از یک نوع هم باشند تا نتیجه True باشد.

عملگرهای منطقی

این عملگرها برای انجام عملیات منطقی بر روی دو عبارت به کار می روند (در مثال زیر متغیر x را برابر 8 و متغیر y را برابر 4 در نظر بگیرید) :

نتیجه	مثال	عملگر	نام
True	$!(x==y)$!	نقیض
False	$!(y==3)$		
True	$(x=4 \ \&\& \ y=8)$	&&	و
False	$(x>4 \ \&\& \ y=8)$		
True	$(x>4 \ \ y=8)$		یا
False	$(x>4 \ \ y>8)$		

عملگر رشته ای

از این عملگر برای چسباندن دو رشته به هم استفاده می شود. مثلاً برای اتصال دو متغیر از نوع رشته از این عملگر استفاده می شود.

نام	عملگر	مثال
عملوند رشته ای	+	a = 'طراحی وب'; b = 'با جاوا اسکریپت'; c = a+b;

در مثال بالا مقدار متغیر C این است : 'طراحی وب با جاوا اسکریپت'
برای ایجا فاصله بین کلمه وب و کلمه با می توانیم یک فاصله خالی به یکی از متغیرهای a یا b اضافه کنیم یا با استفاده از عملوند رشته ای یک فضای خالی به متغیر C اضافه کنیم :

```
a = 'طراحی وب';  
b = 'با جاوا اسکریپت';  
c = a+b;
```

```
-----  
a = 'طراحی وب';  
b = 'اسکریپت با جاوا';  
c = a+' '+b;
```

در نتیجه کد بالا مقدار متغیر C به این صورت خواهد بود : 'طراحی وب با جاوا اسکریپت'

عملگر شرطی

در جاوا اسکریپت نوعی عملگر وجود دارد که بسته به شرایط مقداری را به یک متغیر تخصیص می دهد به این عملگر عملگر شرطی می گویند. ساختار این عملگر به صورت زیر است :

```
VAR_NAME=(شرط)?value1:value2
```

این عملگر در صورتی که شرط برقرار باشد مقدار اول را به متغیر می دهد و در صورت برقرار نبودن شرط مقدار دوم را به متغیر تخصیص می دهد. به مثال زیر توجه کنید :

```
age_group=(age>18)?'بزرگسال':'زیر سن بلوغ';
```

در این مثال اگر متغیر age بزرگتر از 18 باشد مقدار 'بزرگسال' به متغیر age_group تخصیص داده می شود و در صورتی که متغیر age کوچکتر از 18 باشد مقدار 'زیر سن بلوغ' به عنوان مقدار متغیر age_group قرار می گیرد.

نمایش پیام هشدار در صفحه با استفاده از جاوا اسکریپت

آیا مایلید یکی از آن کادر هایی که ناگهان ظاهر می شوند و معلوم نیست از کجا آمده اند را به صفحه خود اضافه کنید؟ بسیار خوب بیایید با کادر هشدار شروع کنیم که به بیننده چیزهایی را که شما می خواهید می گوید.

این فرمان هشدار است:

(متنی که می خواهید به بازدید کننده نشان داده شود) alert

حالا برای استفاده از آن ما باید آنرا در یک فرمان دیگر قرار دهیم. بیایید آنرا در یک فرمان آشنا مثل فرمان onmouseover استفاده کنیم. اسکریپت مقابل باعث ظاهر شدن یک کادر هشدار وقتی که ما نشانگر ماوس خود را روی لینک به وجود آمده قرار می دهیم می شود. مثلاً در لینکی که نمی خواهیم بیننده کلیک کند:

```
<a href="alert.php" onmouseover="alert('Insert your message here')">
```

ماوس خود را روی این لینک قرار دهید

```
</a>
```

آنرا امتحان کنید. فقط ماوس خود را روی آن قرار دهید:

[ماوس خود را روی این لینک قرار دهید](#)

در اینجا به ارائه توضیحاتی در مورد دستورات جاوا اسکریپت استفاده شده در کد بالا می پردازیم :

• onmouseover :

این رویداد مسئولیت اجرای کد جاوا اسکریپت موجود در خود را وقتی که ماوس روی آن قرار می گیرد بر عهده دارد.

: alert(' ')

این تابع متنی را که به عنوان مقدار آن قرار گیرد در یک کادر محاوره ای با یک دکمه OK نمایش می دهد.

بسیار خوب، یک روش دیگر هم وجود دارد که بیشتر باعث گیج شده بیننده می شود. آن هم کادر هشدار به محض باز شدن پنجره است. شما با این روش می توانید مطالبی را که می خواهید به بیننده خود بگویید قبل از باز شدن صفحه به او نشان دهید.

برای ساختن این نوع کادر هشدار فرمان هشدار را بین <head> و </head> قرار دهید. برای این کار باید از تگهای <script> که در اول فصل با آن آشنا شدیم استفاده کنیم تا کدهای جاوا اسکریپت بین بقیه کدها تشخیص داده شوند:

```
<head>
```

```
<script language="javascript">
```

```
<!-- hide from old browsers
```

```
alert('به این صفحه خوش آمدید');
```

```
//-->
```



```
</script>
</head>
```

فرمان بالا یک کادر محاوره ای را وقتی که بیننده صفحه شما را باز می کند نشان می دهد. وقتی که OK را کلیک کنید بارگذاری صفحه ادامه پیدا می کند. آن را امتحان کنید.

آیا می خواهید چند گام جلو تر برویم؟ چندین فرمان هشدار را در تگ SCRIPT قرار دهیم تا چندین کادر هشدار پشت سر هم به محض باز شدن صفحه نمایش داده شوند و البته برای هر پیام باید دکمه OK را کلیک کنیم تا به پیام بعدی برویم. خودتان این را امتحان کنید .

این هم کدی که مسئول این پیامها است:

```
<head>
<title>javascript example 2</title>
<script language="javascript">
<!--
alert('لطفاً سایت ما را به منوی favorite اضافه کنید خود');
alert('از منوی favorite گزینه add to favorite را انتخاب کنید');
alert('میانبر هم انجام دهید می توانید این کار را با کلیدهای');
alert('کلید فقط کلید کنترل را نگهدارید و d را فشار دهید');
//-->
</script>
</head>
```

شما می توانید برای ظاهر شدن کادر هشدار از دکمه ها هم استفاده کنید. فقط کد را در جایی که می خواهید دکمه نشان داده شود قرار دهید. در این حالت شما نیازی به تگهای <script> هم ندارید. این هم یک نمونه از این نوع پیام هشدار:

این هم کد این اسکریپت:

```
<input type="button" value="اینجا را کلیک کنید اگر این صفحه را مطالعه کرده اید"
onclick="alert('باز هم سری به ما بزنید از اینکه به سایت ما مراجعه کردید متشکریم!')">
```

ما در این دکمه از رویداد **onclick** استفاده کردیم. کار این رویداد هم مانند **onmouseover** است با این تفاوت که کد مربوطه را در هنگام کلیک کردن روی آن اجرا می کند.

نوشتن متن با استفاده از جاوا اسکریپت

در جاوا اسکریپت این امکان وجود دارد که متنی را نوشته و در صفحه نمایش بدهیم. متن مورد نظر را می توانیم قالب بندی کنیم و از تگهای HTML نیز در آن استفاده کنیم. برای این کار از فرمان **document.write()** استفاده می کنیم که یک فرمان جاوا اسکریپت مخصوص چاپ کردن خروجی در صفحه است. روش کار به صورت زیر است :

نوشتن متنهای ساده

```
<script language="JavaScript" type="text/javascript">
document.write('بنویسید متن مورد نظر خود را اینجا');
</script>
```

البته نقطه ویرگولی که در آخر فرمان نوشته شده تنها در صورتی الزامی است که بیش از یک فرمان در اسکریپت خود داشته باشیم.

برای نوشتن متن به این صورت کافی است تا فرمان بالا را در جایی از صفحه HTML که می خواهید متن به نمایش در آید بنویسید.

نوشتن متنهای قالب بندی شده

با استفاده از فرمان **document.write** امکان نوشتن متنهای قالب بندی شده با استفاده از تگهای HTML هم وجود دارد. برای اینکار فقط کافیست متن HTML مورد نظر خود را در تابع **document.write()** قرار دهید. برای مثال به کد زیر توجه کنید :

```
<script language="JavaScript" type="text/javascript">
document.write('<h1>!جاوا اسکریپت نوشته شده است این متن با استفاده از</h1>');
</script>
```

برای مشاهده نتیجه این مثال دستورات بالا را اجرا کنید.

تابع **document.write()** با مقدار چند سطری

اگر بخواهیم متنی که با استفاده از جاوا اسکریپت متنی را بنویسیم که در صفحه در چند سطر نمایش داده شود به راحتی می توانیم با وارد کردن **
** در متن مورد نظر یا با سایر تگهای HTML این کار را انجام دهیم. اما اگر بخواهیم مقداری را که به تابع **document.write()** می دهیم در چند سطر وارد کنیم باید مقدار مورد نظر را در هر سطر از هم جدا کرده و با استفاده از عملوند رشته ای **+** آنها را در چند سطر به هم پیوند بزنیم.

نکته : نباید بدون استفاده از عملگرهای مربوطه مقدار تابع بالا را در بیش از یک سطر بنویسید. در صورتی که متن مورد نظر خود را به این گونه در چند سطر بنویسید تابع کار نمی کند و احتمالاً با پیغام خطا مواجه می شوید.

اگر کد خود را به صورت زیر بنویسید با خطا مواجه خواهید شد :

```
<script language="JavaScript" type="text/javascript">
document.write('First part of text
Second part of text');
</script>
```

برای اینکه با خطا مواجه نشوید و کد شما کار خود را انجام دهد باید آنرا به صورت زیر بنویسید :

```
<script language="JavaScript" type="text/javascript">
document.write('First part of text'
+
'Second part of text');
</script>
```

معرفی دستور شرطی **if...else** و کاربردهای مختلف آن

از دستورات شرطی برای انجام دستوراتی به شرط برقراری یک رابطه دیگر استفاده می شود. در طراحی صفحات وب بسیار پیش می آید که می خواهیم در شرایط خاصی صفحه یک رفتار مشخص داشته باشد و در موارد دیگر رفتار صفحه تفاوت داشته باشد. در این مواقع می توانیم از دستور شرطی **if** استفاده کنیم. با این دستور مشخص می کنیم اگر شرط خاصی برقرار بود یک عمل خاص انجام شود. ساختار این دستور به شکل زیر است :

دستور شرطی **if**

```
if (condition)
{
در اینجا قرار می گیرد دستوری که در صورت درستی شرط اجرا می شود
}
```

به یک مثال برای دستور **if** توجه کنید :

```
var test = confirm('می شود اگر این پیام را تأیید کنید دستور شرطی اجرا');

if (test == true) {
alert('و این پیام به نمایش در آمد شما دکمه مورد نظر را کلیک کردید');
}
```

برای اینکه کد بالا را آزمایش کنید دستورات بالا را اجرا کنید و در پیامی که نمایش داده می شود دکمه OK را کلیک کنید. اگر دکمه Cancel را کلیک کنید شرط بالا برقرار نمی شود و بدون اجرا شدن دستور شرطی صفحه مثال باز می شود.

دستور شرطی **if...else**

برخی مواقع ممکن است بخواهیم دستور شرطی یک شرط را چک کند تا در صورت برقرار بودن آن شرط یک فرمان خاص را اجرا کند و در صورتی که شرط برقرار نبود یک فرمان دیگر را به اجرا در آورد. در این موارد می توانیم از دستور **if** به همراه **else** استفاده کنیم. ساختار این دستور به شکل زیر است :

```
if (condition) {
شوند دستوراتی که در صورت برقراری شرط اجرا می
}
else {
```

دستوراتی که در صورت عدم برقراری شرط اجرا می شوند
}

در کد بالا ابتدا شرط بررسی می شود و در صورت برقراری آن دستورات مربوطه اجرا می شوند ولی در صورتی که شرط برقرار نباشد دستورات مربوط به قسمت **else** اجرا می شود.

```
var yourmark = prompt('یک شماره از 0 تا 20 وارد کنید');
```

```
if (yourmark > 10) {  
  alert('شماره ای که وارد کرده ای از 10 بزرگتر');  
}  
else {  
  alert('نیست عبارتی که شما وارد کرده اید از 10 بزرگتر');  
}
```

برای مشاهده نتیجه مثال بالا دستورات بالا را اجرا کنید. و در کادری که نمایش داده می شود یک عدد از 0 تا 20 وارد کنید و دکمه OK را کلیک کنید.

استفاده از دستور **if..else if..else** (دستورات شرطی تو در تو)

ممکن است در نظر داشته باشیم تا در صورت برقراری یک شرط یک دستور خاص اجرا شود و در صورتی که شرط برقرار نبود شرط دیگری بررسی شود و دستور مربوط به آن اجرا شود و در صورتی که هیچ یک از شرطهای قبل برقرار نبود دستور دیگری اجرا شود. در این مواقع می توانیم از دستور **if..else if..else** قبلی استفاده کنیم. به صورت زیر :

```
if (condition 1) {  
  شوند دستوراتی که در صورت درستی شرط اول اجرا می  
}  
else if (condition 2) {  
  می شوند دستوراتی که در صورت درستی شرط دوم اجرا  
}  
else {  
  می شوند دستوراتی که زمانی که هیچکدام از شرطها صحیح نباشد اجرا  
}
```

در این روش محدودیتی از نظر تعداد شرطها وجود ندارد و می توانیم به تعداد نامحدود از دستور **else if** استفاده کنیم ولی در مواردی که تعداد این شرطها خیلی زیاد می شود بهتر است به جای این روش از دستور **switch** که بعداً توضیح داده خواهد شد استفاده کنیم.

در اینجا مثال قبل را با کمی تغییر برای این روش استفاده می کنیم :

```
var yourmark = prompt('یک شماره به دلخواه وارد کنید');
```

```

if (yourmark > 0 && yourmark < 100 ) {
alert('وارد شده بین 0 تا 100 است شماره');
}
else if (yourmark > 100) {
alert('شماره وارد شده از 100 بزرگتر است');
}
else {
alert('وارد شده یا 0 است یا به جای عدد از حروف استفاده کرده اید عبارت');
}

```

[برای مشاهده نتیجه مثال دستور if...else if...else دستورات بالا را اجرا کنید .](#)

دستور شرطی switch در جاوا اسکریپت

با استفاده از دستور if امکان این وجود دارد که برنامه را حداکثر به یکی از دو جهت ممکن هدایت کنیم. یعنی اگر شرط مورد نظر برقرار بود اسکریپت یک سری کد را اجرا می کند و در صورت عدم برقراری شرط یک سری کد دیگر اجرا می شود. اگر بخواهیم برای اسکریپت خود مسیرهای بیشتری در نظر بگیریم یک راه موجود استفاده از دستورات if تو در تو است. این روش ممکن است در موارد جزئی کارایی داشته باشد اما اگر تعداد شرطها زیاد شود ما را مجبور می کند از تعداد زیادی if و else در اسکریپت خود استفاده کنیم که این مسئله هم باعث طولانی شدن برنامه و هم باعث گیج کننده شدن آن می شود از طرفی امکان اشتباه را هم بالا می برد.

راه حل دیگر استفاده از دستور شرطی **switch** است. این دستور را می توان در هنگامی که می خواهیم یک متغیر را با چندین مقدار مقایسه کنیم و مقدار متناسب با آنرا پیدا کنیم کار برد دارد. ساختار این دستور به صورت زیر است :

```

switch(x)
{
مقدار اول case:
با مقدار اول اجرا می شود\این کد در صورت برابری متغیر
break
مقدار دوم case:
با مقدار دوم اجرا می شود\این کد در صورت برابری متغیر
break
default:
با هیچکدام از مقادیر ذکر شده برابر نباشد\این کد در صورتی اجرا می شود که مقدار متغیر
}

```

در این دستور مقدار متغیر معرفی شده در قسمت **switch** با مقادیر آورده شده در هر **case** مقایسه می شود. در صورت مساوی بودن مقادیر کد همان case اجرا می شود و با استفاده از دستور **break** آورده شده در هر قسمت برنامه پس از اجرای کد آن قسمت به خارج از دستور switch هدایت می شود. در غیر این صورت برنامه به سراغ case بعدی می رود و در صورتی که مقدار متغیر مورد نظر با هیچکدام از case ها برابر نبود کد قسمت **default** اجرا خواهد شد.

قابل ذکر است که استفاده از default اختیاری است و شما بنا به نیاز خود می توانید از آن استفاده کرده یا آنرا حذف کنید. در دستور switch محدودیتی در تعداد case ها وجود ندارد و می توانید به هر تعدادی که نیاز داری از این مورد استفاده کنید.

در این قسمت به ذکر یک نمونه برای دستور switch می پردازیم :

به کدی که در اینجا آورده شده است توجه کنید. پس از اجرای این کد کادری باز می شود که از شما میخواهد عددی را وارد کنید. با وارد کردن عدد در این کادر و تأیید آن پیامی متناسب با روز معادل آن عدد در هفته ظاهر می شود. البته ما این کار را فقط برای اعداد 1 تا 3 انجام دادیم، شما در صورت تمایل می توانید آنرا تکمیل کنید :

```
var x=prompt('لطفاً یک عدد بین 1 تا 7 وارد کنید');

switch (x) {
case '1':
alert('شماره وارد شده مربوط به روز « شنبه » است');
break
case '2':
alert('شماره وارد شده مربوط به روز « یکشنبه » است');
break
case '3':
alert('شماره وارد شده مربوط به روز « دوشنبه » است');
break
case '4':
alert('شماره وارد شده مربوط به روز « سه شنبه » است');
break
case '5':
alert('شماره وارد شده مربوط به روز « چهارشنبه » است');
break
case '6':
alert('شماره وارد شده مربوط به روز « پنجشنبه » است');
break
case '7':
alert('شماره وارد شده مربوط به روز « جمعه » است');
break
default:
alert('روز مربوط به شماره وارد شده در این برنامه وجود ندارد');
}
```

[برای مشاهده مثال بالا در عمل دستورات بالا را اجرا کنید](#)

رویداد onmouseover

وقت اون رسیده که نوشتن اولین اسکریپت خود را شروع کنید. این یکی راحت‌تر چون مجبور نیستیم درگیر اضافه کردن تگهای SCRIPT شویم. این اسکریپت کوچک به دلخواه شما وقتی نشانگر ماوس روی یک لینک قرار می‌گیرد متنی را در نوار وضعیت مرورگر نشان می‌دهد. به مثال نگاه کنید:

```
<a href="JavaScript_2.htm"
onmouseover="window.status='آموزش جاوا اسکریپت';return true">
!!!دهید ماوس خود را روی این لینک قرار
</a>
```

شما هم امتحان کنید و ببینید چگونه کار می‌کند. نشانگر ماوس خود را روی لینک زیر قرار دهید ولی کلیک نکنید (اگر کلیک کنید، چون در کد بالا آدرس لینک، آدرس همین صفحه است پس از کلیک کردن دوباره همین صفحه باز می‌شود). حالا پایین پنجره مرورگر خود به نوار وضعیت نگاه کنید باید این جمله را ببینید: «آموزش جاوا اسکریپت»

بسیار خوب، این چیزی است که با فرمان **onmouseover** اتفاق می‌افتد:

1. **onmouseover=" "** این شکل فرمان **onmouseover** است. مرورگر انتظار فرمان دیگری بین دو علامت دابل کوت (") را دارد.
2. **'آموزش جاوا اسکریپت'** **window.status='آموزش جاوا اسکریپت'** این فرمان به مرورگر می‌گوید که جمله ای را در نوار وضعیت بنویسد. شما می‌توانید جمله بین دو علامت (') را با آنچه که می‌خواهید در نوار وضعیت به نمایش در آید جایگزین کنید.
3. **return true** عبارت را تأیید کرده و مرورگر آن را نمایش می‌دهد.

دلیل استفاده از علامت سینگل کوت (') در فرمان **window.status** این است که **window.status** خود قسمتی از فرمان **onmouseover** است که از (") استفاده کرده است. اگر دوباره از علامت دابل کوت (") استفاده کرده بودیم ممکن بود مرورگر در مورد اینکه چه کاری باید انجام دهد گیج می‌شد. چون ممکن بود علامت دابل کوت دوم را به منزله پایان فرمان **onmouseover** تلقی کند و فرمان جدیدی را با **window.status** شروع کند: **onmouseover=""window.statuse**

مشکلی که ممکن است در اینجا به وجود آید پاک نشدن متنی است که می‌خواستید در نوار وضعیت نمایش داده شود. دو راه برای از بین بردن این مشکل وجود دارد. یکی استفاده از فرمان **onmouseout** است، و دیگری فرا خواندن تابعی که متن را پس از مدتی پاک می‌کند. راه دوم نیازمند استفاده از توابع و اسکریپت تگها است. در اینجا آسانترین راه را به شما نشان می‌دهم: استفاده از فرمان **onmouseout** :

```
<a href="JavaScript_2.htm"
onmouseover="window.status='متن خود را اینجا قرار دهید'; return true"
onmouseout="window.status=''; return true">
قرار دهید لینک خود را اینجا
</a>
```

این فرمان متن نوار وضعیت را وقتی که نشانگر ماوس روی آن نیست ناپدید می کند.

در حقیقت ما همان کار قبل را تکرار کردیم ولی به جای متن از فضای خالی در فرمان **window.status** استفاده کردیم.

این اسکریپت جالب و مفید است و می تواند به بازدید کنندگان صفحه شما کمک کند تا بهتر بدانند لینک آنها را به چه صفحه ای می برد. و می توانند توضیحات شما را در مورد لینک ببینند.

دکمه ها و موارد استفاده آنها

ما برای نوشتن اسکریپت با استفاده از دکمه ها، ابتدا باید بدانیم چگونه می توان یک دکمه را در یک صفحه قرار داد. برای این کار شما باید از تگهای `<FORM>` در اطراف تگ دکمه استفاده کنید. در مثال زیر نمونه یک HTML که یک دکمه را در یک صفحه قرار می دهد آورده شده است:

```
<FORM>
<INPUT type="button" value="اینجا کلیک کنید" name="button1">
</FORM>
```

این فرمان یک دکمه را در صفحه شما قرار می دهد، اما اگر روی آن کلیک کنید اتفاقی نمی افتد...

در زیر کار هر قسمت از فرمان بالا را می بینیم:

1. **<FORM>**
این قسمت یک فرم را ایجاد می کند بنابراین این ما می توانیم یک دکمه بسازیم.
2. **<INPUT>**
این تگ به ما اجازه می دهد نوعی ناحیه ورودی بسازیم.
3. **type="button"**
این فرمان ناحیه ورودی ما را به عنوان یک دکمه تعریف می کند.
4. **value="اینجا کلیک کنید"**
این متنی است که افراد روی دکمه می بینند. شما می توانید هر چیزی را که می خواهید بینندگان روی دکمه ببینند اینجا بنویسید.
5. **name="button1"**
شما می توانید برای مراجعات بعدی یا احتمالاً استفاده در یک اسکریپت به دکمه خود یک نام بدهید.

احتمالاً شما نمی خواهید یک دکمه بسازید که کاری انجام نمی دهد. در زیر یک فرمان جاوا اسکریپت آورده شده است که مرورگر را مجبور می کند وقتی دکمه را کلیک کردید عکس العمل نشان دهد:

```
onClick="javascript command"
```

فقط این فرمان را در تگ INPUT قرار دهید که دکمه را با آن ساختید، مانند زیر:

```
<INPUT type="button" value="اینجا کلیک کنید" name="button1" onClick="" >
```

برای مثال ما می توانیم مانند فرمان `onmouseover` در این دکمه نیز استفاده کنیم و متنی را در نوار وضعیت پنجره چاپ کنیم!


```
<FORM>
<INPUT type="button" value="اینجا کلیک کنید" name="Button2"
onClick="window.status='متن خود را اینجا بنویسید'; return true">
</FORM>
```

حالا دکمه را فشار دهید و به متن نوار وضعیت نگاه کنید:

شما می توانید با استفاده از دکمه ها به بینندگان خود اجازه دهید رنگ زمینه صفحه را برای خودشان تغییر دهند. فقط از فرمان زیر به جای فرمان window.status استفاده کنید:

```
document.bgColor='رنگ مورد نظر به زبان انگلیسی نوشته شود'
```

فرمان زیر را به عنوان تمرین فرمان onClick وارد کنید:

```
<FORM>
<INPUT type="button" value="Change Background Color" name="button3" onClick="document.bgColor='yellow'">
</FORM>
```

حالا شما دکمه ای مانند زیر دارید که با کلیک کردن بر روی آن زمینه صفحه به رنگ زرد در می آید!

شما می توانید به هر تعداد که می خواهید از این دکمه ها اضافه کنید، فقط مطمئن شوید که آنها گزینه بازگشت به حالت اولیه را نیز داشته باشند. اسکریپت زیر به شما چهار گزینه می دهد: سفید، زرد، قرمز و آبی. می توانید از دکمه «بازگشت به رنگ اولیه!» برای بازگرداندن تصویر زمینه این صفحه استفاده کنید. کدی که در این صفحه از آن استفاده شده است به دلیل وجود تصویر زمینه کمی با کد زیر متفاوت است ولی کارایی آن تفاوتی ندارد.

```
<FORM>
<INPUT type="button" value="به سفید تغییر رنگ زمینه"
name="button6" onClick="document.bgColor='white'"><br>
<INPUT type="button" value="تغییر رنگ زمینه به زرد"
name="button3" onClick="document.bgColor='yellow'"><br>
<INPUT type="button" value="تغییر رنگ زمینه به قرمز"
name="button4" onClick="document.bgColor='red'"><br>
<INPUT type="button" value="تغییر رنگ زمینه به آبی"
name="button5" onClick="document.bgColor='#66CCFF'">
</FORM>
```

برای دکمه چهارم به جای نام رنگ از معادل هگزادسیمال رنگها استفاده شده است. حالا شما سه دکمه دارید، آنها را امتحان کنید!

در اسکریپت آخری که در این قسمت بررسی می کنیم، اجازه می دهیم تا یک دکمه به عنوان یک لینک عمل کند. به این منظور فقط فرمان زیر را برای فرمان onClick استفاده کنید:

```
window.location='URL'
```

برای مثال این هم اسکریپتی که شما را به صفحه آموزش جاوا اسکریپت ما می برد:

```
<FORM>
<INPUT type="button" value="اسکریپت آموزش جاوا" name="button6"
onClick="window.location='./'/">
</FORM>
```

آموزش کار با تاریخچه مرورگر با استفاده از جاوا اسکریپت

مرورگرها به صورت پیش فرض آدرس و حتی خود صفحاتی را که کاربر مشاهده می کند ذخیره می کنند. این آدرسها به ترتیبی که کاربر آنها را مشاهده کرده است در **تاریخچه (history)** مرورگر ثبت می شوند و دکمه های Back و Forward پنجره مرورگر بر اساس این تاریخچه کار می کنند. با استفاده از جاوا اسکریپت می توانیم به این تاریخچه دسترسی داشته باشیم و با کمک آن مسیر کاربران را هدایت کنیم.

ما برای کار با تاریخچه مرورگر در اینجا از دکمه های HTML استفاده می کنیم که با کمک رویداد onclick در هنگام کلیک کردن روی این دکمه ها دستورات مورد نظر ما اجرا خواهند شد. در اینجا می توانید کد HTML مربوط به ساخت این دکمه ها را مشاهده کنید (البته مقدار مربوط به شناسه onclick مربوط به دستورات جاوا اسکریپت است که بعداً وارد خواهد شد) :

```
<input type="button" value="متن روی دکمه" onclick="">
```

دستورات مختلفی برای کار با تاریخچه مرورگر وجود دارد که همه آنها از شیء **history** استفاده می کنند. در ادامه می توانید تعدادی از این دستورات را که با بیشتر مرورگرها کار می کنند مشاهده کنید

history.back()

از این فرمان می توانید جهت هدایت کاربران به صفحه ای که قبل از صفحه کنونی در پنجره مرورگر مشاهده کرده اند استفاده کنید. این دستور دقیقاً مانند دکمه **Back** در پنجره مرورگر عمل می کند و در صورتی که صفحه کنونی اولین صفحه ای باشد که از زمان باز شدن پنجره نمایش داده شده باشد این دستور کاربردی ندارد. برای مشاهده یک نمونه از کاربرد این دستور می توانید لینک را کلیک کنید :

در اینجا می توانید کد مورد استفاده در مثال بالا را مشاهده کنید :

```
<input type="button" value="کلیک کنید برای بازگشت به صفحه قبل اینجا را" onclick="history.back()">
```

history.forward()

در صورتی که بازدید کننده قبلاً از دکمه Back در مرورگر خود استفاده کرده باشد یا با استفاده از دستور history.back() به صفحه قبلی بازگشته باشد می توانیم از دستور **history.forward()** برای هدایت بازدید کننده به صفحه ای که پس از صفحه کنونی باز کرده است استفاده کنیم. این دستور کاربردی مشابه دکمه **Forward** در پنجره مرورگر دارد و کاربرد آن عکس کاربرد history.back() است.

دکمه زیر با استفاده از دستور **history.forward()** شما را به صفحه ای که برای مثال مربوط به بازگشت به صفحه قبلی مشاهده کرده اید هدایت می کند :

می توانید کد استفاده شده در ساخت دکمه بالا را در اینجا مشاهده کنید :

```
<input type="button" value="نمایش صفحه بعدی" onclick="history.forward()">
```

history.go()

با استفاده از دستورهای **history.back()** و **history.forward()** فقط امکان این وجود دارد که کاربران را یک صفحه به عقب یا جلو هدایت کنیم. اگر بخواهیم بازدید کننده را به چندین صفحه عقب یا جلو هدایت کنیم باید از دستور **history.go()** استفاده کنیم.

1- استفاده از مقداری مثبت برای **history.go()**

اگر برای دستور **history.go()** از مقداری مثبت استفاده شود کاربر به همان تعداد صفحه در تاریخچه مرورگر به جلو هدایت می شود. به کد زیر توجه کنید :

```
<input type="button" value="متن روی دکمه" onclick="history.go(3)">
```

کد بالا دکمه ای را می سازد که کلیک کردن بر روی آن همان کاری را انجام می دهد که سه بار فشار دادن دکمه Forward در پنجره مرورگر انجام می دهد.

در صورتی که دستور به صورت **history.go(1)** نوشته شود مانند دستور **history.forward()** عمل می کند.

2- استفاده از مقدار صفر برای **history.go()**

در صورتی که این دستور به صورت **history.go(0)** نوشته شود باعث باز بارگذاری مجدد همان صفحه خواهد شد. برای مثال به کد زیر و دکمه ساخته شده با آن توجه کنید :

```
<input type="button" value="بارگذاری مجدد صفحه" onclick="history.go(0)">
```

این هم دکمه ساخته شده :

3- استفاده از مقادیر منفی برای **history.go()**

اگر بخواهیم کاربر را چند صفحه در تاریخچه مرورگر به عقب هدایت کنیم می توانیم برای دستور **history.go()** از مقادیر منفی استفاده کنیم. برای مثال به کد زیر توجه کنید :

```
<input type="button" value="متن روی دکمه" onclick="history.go(-3)">
```

این کد دکمه ای را می سازد که وقتی کاربر آنرا کلیک کند به سومین صفحه ای که قبل از صفحه کنونی در پنجره مرورگر بازدید کرده است هدایت می شود. کاربرد این دکمه در اینجا مشابه سه بار فشار دادن دکمه Back در پنجره مرورگر است.

history.length

با استفاده از دستور history.length می توانید تعداد صفحاتی را که در تاریخچه مرورگر وجود دارد پیدا کنید. یعنی با استفاده از این دستور می توانیم به تعداد صفحاتی که از زمان باز شدن پنجره مرورگر در آن نمایش داده شده است پی ببریم. برای مثال می توانید با کلیک کردن بر روی لینک زیر تعداد صفحات ذخیره شده در تاریخچه پنجره مرورگر خود را پیدا کنید :

در اینجا می توانید کد مورد استفاده در ساخت لینک بالا را مشاهده کنید :

```
<a href="javascript:alert(history.length)">  
صفحات موجود در تاریخچه مرورگر نمایش تعداد  
</a>
```

درخواست اطلاعات از کاربر

بیا فرض کنیم می خواهیم نام فردی را قبل از ورود به صفحه پرسیم و سپس نام آنها را در صفحه برای خودشان نمایش دهیم. این کار را با استفاده از جاوااسکریپت می توان انجام داد. فرمانی که برای این کار استفاده می شود به این صورت است:

```
prompt('سؤال شما');
```

این فرمان یک پنجره را قبل از باز شدن صفحه نمایش می دهد که سؤال شما در آن پرسیده می شود و بیننده می تواند جواب خود را در کادری که برای او در نظر گرفته شده بنویسد. دو سینگل کوت بعدی جواب پیش فرض شما برای سؤال است. اگر شما چیزی بین این دو سینگل کوت ننویسید، بیننده با یک لیست خالی، آماده برای نوشتن مواجه می شود. این پیغام قبل از بارگذاری صفحه نمایش داده می شود پس شما می توانید پاسخ بیننده را در صفحه نمایش دهید. اگر مثالی در این مورد می خواهید لینک زیر را کلیک کنید. در صفحه مثال نام شما پرسیده می شود و پس از وارد کردن آن، نام شما در صفحه جدید نمایش داده می شود

حالا به سراغ اسکریپتی که این کار را انجام می دهد می رویم. توجه داشته باشید که دستورات prompt و if/else در بخش HEAD صفحه وارد می شوند ولی نام کاربر در بخش BODY صفحه مشاهده می شود.

```
<html dir="rtl" >  
  
<head>  
<script language="javascript">  
<!--  
var yourname=prompt('لطفاً نام خود را وارد کنید');  
if ( (yourname=="") || (yourname==null) )
```

```

{
yourname="ناشناس";
}
//-->
</script>
</head>

<body>
<script language="javascript">
<!--
document.write("<center><h1>سلام " +yourname+ " ، به سایت ما خوش آمدید "</h1></center>");
//-->
</script>
</body>

</html>

```

اولین چیزی که اتفاق می افتد این است که متغیر yourname مقداری را که کاربر برای آن وارد می کند به خود می گیرد. پس مقدار متغیر yourname معادل استرینگ است که نام کاربر را می سازد.

دستور if/else نام « ناشناس » را در صورتی که کاربر نامی را وارد نکند به عنوان مقدار متغیر yourname قرار می دهد. این دستور چک می کند که اگر نامی وارد نشود (null) و یا اگر فضای خالی وارد شود، مقدار « ناشناس » را به متغیر yourname می دهد.

در بخش body متن دوباره از تگ script استفاده می کنیم تا کدهای جاوا اسکریپت از بقیه متن html متمایز شود. در این قسمت ما از یک فرمان جدید یعنی document.write(""); استفاده کرده ایم. این همان چیزی است که به متغیر yourname اجازه نوشته شدن در صفحه html را می دهد. ما در این پیغام دو استرینگ را به علاوه متغیر yourname در صفحه می نویسیم. متغیر yourname در این فرمان بین دو دابل کوت قرار نمی گیرد چون یک مقدار است نه یک استرینگ، (البته مقدار متغیر yourname یک استرینگ است). به دلیل اینکه متغیر yourname یک استرینگ نیست ما از دو علامت بعلاوه در ابتدا و انتهای آن استفاده می کنیم. این سری نوشته ها مرورگر را وادار می کند ابتدا یک استرینگ سپس مقدار متغیر و سپس استرینگ دوم را بنویسد. چون ما در حال نوشتن یک کد جاوا اسکریپت هستیم برای اینکه بتوانیم متنی را در صفحه بنویسیم نمی توانیم مانند نوشتن متن در html عمل کنیم، و باید متن مورد نظر را بین دابل کوت و در قالب استرینگ در صفحه بنویسیم.

شاید متوجه شده باشید تگهای پایانی h1 و center به گونه ای متفاوت با قبل نوشته شده اند (</h1>). این تفاوت بدین دلیل است که اسلش در جاوا اسکریپت به معنای تقسیم است و برای جلوگیری از اشتباه شدن آن با علامت تقسیم از یک یک اسلش قبل از آن استفاده می کنیم. در نتیجه استفاده از یک یک اسلش (\) و بلافاصله بعد از آن یک اسلش (/)



این کتاب از وبلاگ میهن کتاب دانلود شده است.
mihanketab.blogfa.com

برای دریافت اطلاعات بیشتر
و آگاهی از جدیدترین کتابهای اضافه شده،
به صفحه فیس بوک میهن کتاب پیوندید.
facebook.com/mihanketab

