



«پیش نمایش»

کتاب آموزش JavaScript

همراه با صدها مثال

- این کتاب مطابق با مطالب ارائه شده در سایت Beyamooz.com می باشد.
- JavaScript یکی از زبان های Script نویسی تحت وب است که با استفاده از آن می توان در صفحات وب توابع، اعتبارسنجی فرم ها، ارتباط برقرار کردن با سرور و... را اضافه نمود.
- در حال حاضر JavaScript یکی از محبوب ترین زبان های برنامه نویسی وب است. این زبان معمولاً با HTML و CSS برای ایجاد تعامل با کاربر در صفحات وب بکار می رود.

نسخه 1.2

فهرست مطالب

فصل 1: آموزش مقدماتی JavaScript 5

5-1-1	مقدمه JavaScript	5
11-2-1	محل قرار گیری کدهای JavaScript	11
17-3-1	شروع کار بار JavaScript	17
20-4-1	دستورات JavaScript	20
23-5-1	توضیحات (Comment) در JavaScript	23
26-6-1	متغیرهای JavaScript	26
7-1-7	انواع داده در JavaScript	در فایل اصلی
8-1-8	عملگرها در JavaScript	در فایل اصلی
9-1-9	دستورات شرطی در JavaScript	در فایل اصلی
10-1-10	دستور Switch در JavaScript	در فایل اصلی
11-1-11	توابع در JavaScript	در فایل اصلی
12-1-12	حلقه For در JavaScript	در فایل اصلی
13-1-13	حلقه while در JavaScript	در فایل اصلی
14-1-14	دستور Break در JavaScript	در فایل اصلی
15-1-15	حلقه For ... In در JavaScript	در فایل اصلی
16-1-16	مدیریت خطاها در JavaScript	در فایل اصلی
17-1-17	اعتبارسنجی (Validation) فرم ها در JavaScript	در فایل اصلی

فصل 2: اشیاء (Object) در JavaScript 33

33-1-2	مقدمه اشیاء (Object) در JavaScript	33
--------	------------------------------------	----

39.....	2-2- اعداد در JavaScript
42.....	2-3- رشته ها در JavaScript
در فایل اصلی	2-4- زمان در JavaScript
در فایل اصلی	2-5- آرایه ها در JavaScript
در فایل اصلی	2-6- شیء Boolean در JavaScript
در فایل اصلی	2-7- شیء Math در JavaScript
در فایل اصلی	2-8- شیء RegExp در JavaScript

49 فصل 3: آموزش مدل DOM

49.....	3-1- کاربرد DOM چیست؟
52.....	3-2- تغییر محتوای عناصر HTML
در فایل اصلی	3-3- تغییر ظاهر عناصر HTML
در فایل اصلی	3-4- رویدادها در JavaScript
در فایل اصلی	3-5- ایجاد و حذف عناصر در JavaScript

56 فصل 4: آموزش مدل BOM

56.....	4-1- مفهوم BOM چیست؟
56.....	4-2- شیء Window در JavaScript
در فایل اصلی	4-3- شیء Screen در JavaScript
در فایل اصلی	4-4- شیء Location در JavaScript
در فایل اصلی	4-5- شیء History در JavaScript
در فایل اصلی	4-6- شیء Navigator در JavaScript
در فایل اصلی	4-7- پنجره های نمایش پیغام در JavaScript
در فایل اصلی	4-8- زمانبندی رویدادها در JavaScript

9-4- متغییر Cookie در JavaScript.....	در فایل اصلی
10-4- خلاصه مطالب.....	در فایل اصلی

فصل 5: کتابخانه های JavaScript

1-5- کتابخانه های JavaScript.....	59
2-5- کتابخانه jQuery.....	در فایل اصلی
3-5- کتابخانه Prototype.....	در فایل اصلی

فصل 1: آموزش مقدماتی JavaScript

1-1- مقدمه JavaScript

JavaScript یک زبان اسکریپت نویسی تحت وب است.

JavaScript در میلیون ها صفحه وب برای اضافه کردن توابع، اعتبار سنجی فرم ها، ارتباط برقرار کردن با سرور و ... استفاده شده است.

یادگیری JavaScript آسان است و از آموزش آن لذت خواهید برد.

آموزش با استفاده از مثال های "خودتان امتحان کنید"

در هر فصل، مثال های "خودتان امتحان کنید" آورده شده است.

این آموزش شامل صدها مثال "خودتان امتحان کنید" است.

می توانید با ویرایشگری که در اختیارتان است، کد JavaScript را ویرایش کنید و با کلیک روی دکمه "ویرایش کردن" نتیجه آنرا مشاهده کنید.

مثال:

```
<script type="text/javascript">
function displayDate()
{
document.getElementById("demo").innerHTML=Date();
}
</script>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_events



خودتان امتحان کنید

برای مشاهده نتیجه کد بالا روی دکمه "خودتان امتحان کنید" کلیک کنید.

هم اکنون یادگیری JavaScript را آغاز کنید!

قبل از شروع آموزش چه چیزی را باید بدانید

پیشنهاد می شود اگر با بیاموزهای زیر آشنا نیستید، قبل از ادامه آنها را مطالعه نمایید:

1. آموزش HTML

2. آموزش CSS

JavaScript یک زبان اسکریپت نویسی است

JavaScript محبوب ترین زبان اسکریپت نویسی تحت وب است و با عمده مرورگرها مانند Firefox,

Internet Explorer, Opera, Chrome و Safari کار می کند.

JavaScript برای اضافه کردن تعامل بیشتر در صفحات HTML، بوجود آمده است.

JavaScript معمولاً در بین کدهای HTML قرار می گیرد (embedded).

JavaScript یک زبان تفسیر شده است. (یعنی اینکه اسکریپت ها بدون مقدمات کامپایل، اجرا می شوند)

هر کسی می تواند از JavaScript استفاده کند، بدون اینکه نیاز به خرید مجوز آن باشد.

در آینده، چه چیزی را آموزش خواهید دید

در زیر، آنچه در آینده به تفصیل فراخواهید گرفت آورده شده است:

آموزش JavaScript: آموزش ایجاد خروجی HTML

مثال:

```
document.write("<h1>This is a heading</h1>");  
document.write("<p>This is a paragraph</p>");
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_intro_document_write



خودتان امتحان کنید

💡 **توجه:** در کدهای JavaScript واقعی، از دستور `document.write()` استفاده نکنید. چون تمام صفحه دوباره بارگذاری می شود. با این وجود در اینجا دستور `document.write()` یک راه ساده برای ایجاد خروجی HTML در JavaScript را نشان می دهد.

آموزش JavaScript: آموزش واکنش در برابر رویدادها

کدهای JavaScript زمانی اجرا می شوند که صفحه بارگذاری شود (`onload`) اما این چیزی نیست که همیشه نیاز داریم.

بعضی مواقع می خواهیم زمانی که یک رویداد خاص اتفاق می افتد، کدمان اجرا شود، مانند زمانی که کاربر روی یک دکمه کلیک می کند.

مثال:

```
<button type="button" onclick="alert('Welcome!')">Click Me!</button>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_intro_alert



خودتان امتحان کنید

از تابع alert معمولاً اگر بخواهید، کاربر را از اطلاعات ارسال شده به Server مطمئن کنید، استفاده کنید.
(پیغام: عملیات با موفقیت انجام شد)

رویداد onclick، تنها یکی از رویدادهای HTML است و در آموزش های بعدی با آنها بیشتر آشنا خواهید شد.

آموزش JavaScript: آموزش تغییر محتوای عناصر HTML

یکی از بهترین قابلیت های JavaScript، دستکاری محتوای عناصر HTML است. (JavaScript می تواند محتویات عناصر HTML را بخواند و آنها را تغییر دهد)

مثال:

```
x=document.getElementById("demo") /*Find the element*/  
x.innerHTML="Hello JavaScript"; /*Change the content*/
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_intro_inner_html

خودتان امتحان کنید

شما دستور document.getElementById() را در آینده بسیار خواهید دید. در قسمت HTML DOM تعریف شده است.

DOM یا (Document Object Model) یک استاندارد رسمی در W3C است و برای دسترسی به عناصر HTML است. (در آینده بیشتر توضیح داده خواهد شد)

آموزش JavaScript: آموزش تغییر تصاویر HTML

این مثال، بصورت داینامیک، خصوصیت (src) عنصر را تغییر می دهد:

مثال:

```
element=document.getElementById('myimage')
if (element.src.match("bulbon"))
{
element.src="/pic_bulboff.gif";
}
else
{
element.src="/pic_bulbon.gif";
}
}
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_lightbulb

خودتان امتحان کنید

JavaScript بسیاری از خصوصیت های عناصر HTML را می تواند تغییر دهد.

آموزش JavaScript: آموزش تغییر ظاهر (Style) عناصر HTML

تغییر Style یک عنصر نسبت به تغییر خصوصیت کمی متفاوت است.

مثال:

```
x=document.getElementById("demo") /*Find the element */
x.style.color="#ff0000"; /*Change the style*/
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_intro_style

خودتان امتحان کنید

آموزش JavaScript: آموزش اعتبار سنجی فرم (Validate)

JavaScript می تواند برای معتبر ساختن اطلاعات فرم، قبل از اینکه به سرور ارسال شود استفاده شود. چنین

فرم هایی جلوی پردازش های اضافی توسط سرور را می گیرند.

مثال:

```
if isNaN(x) {alert("Not Numeric")};
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_intro_validate



خودتان امتحان کنید

آیا JavaScript و Java یکی اند؟

نه!

JavaScript و java از نظر مفهوم و طراحی، دو زبان کاملاً متفاوت هستند.

java (که توسط شرکت sun توسعه یافته است) قوی و بسیار پیچیده تر از JavaScript است و در دسته

بندی زبان هایی مانند C و C++ قرار می گیرد.

JavaScript = ECMAScript

JavaScript یک پیاده سازی از زبان استاندارد ECMAScript است. ECMA-262 استاندارد رسمی JavaScript است.

این زبان توسط "Brendan Eich" در Nets cape (با Navigator2.0) ابداع شده است و تقریباً در تمام مرورگرها از سال 1996 ظاهر شده است.

توسعه ی استاندارد ECMA_262 از سال 1996 و اولین ویرایش آن در 1997 توسط مجمع عمومی ECMA پذیرفته شد.

استاندارد ECMA به عنوان یک استاندارد ISO جهانی (ISO/IEC 16262) در سال 1998 تصویب شده است. توسعه استاندارد هنوز هم در حال پیشرفت است.

1-2- محل قرار گیری کدهای JavaScript

کدهای JavaScript در صفحه HTML باید بین تگهای `<script>` و `</script>` قرار داده شود. کدهای JavaScript را می توان در بدنه صفحه HTML (بین تگهای باز و بسته `<body>` و `</body>`) و یا در قسمت `<head>` صفحه قرار داد.

تگ `<script>`

از تگ `<script>`، برای وارد کردن کدهای JavaScript در صفحه HTML استفاده می شود. تگهای باز و بسته `<script>` و `</script>` می گوید، کدهای JavaScript کجا شروع و کجا پایان یافته است. خطوط بین `<script>` و `</script>` شامل کدهای JavaScript است:

```
<script>  
alert("My First JavaScript");  
</script>
```

مرورگر، کد JavaScript بین `<script>` و `</script>` را ترجمه و اجرا می کند.

💡 **توجه:** بدون تگ های باز و بسته `<script>` و `</script>`، مرورگر با دستورات بین آنها مانند یک متن

معمولی برخورد می کند و آنها را عیناً در صفحه HTML نمایش می دهد. (خودتان امتحان کنید)

💡 **توجه:** ممکن است برای تگ `<script>` خصوصیت `type` را تنظیم کنیم ("`text/javascript`"). نیاز به

انجام این کار نیست، چون در تمام مرورگرها و در HTML5 زبان اسکریپت پیشفرض، JavaScript است.

کدهای JavaScript در بدنه صفحه HTML

در مثال زیر، هنگام بارگذاری صفحه یا (onload)، خروجی HTML در قسمت <body> چاپ می شود:

مثال:

```
<!DOCTYPE html>
<html>
<body>
.
.
<script>
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph</p>");
</script>
.
.
</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_intro_document_write



خودتان امتحان کنید

همچنین در مثال زیر، تاریخ جاری سیستم، هنگام بارگذاری صفحه، داخل عنصر با شناسه "demo" نوشته می شود:

مثال:

```
<html>
<body>

<h1>My First Web Page</h1>
<p id="demo">My First Paragraph.</p>
<script type="text/javascript">
    document.getElementById("demo").innerHTML=Date();
</script>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_dom



خودتان امتحان کنید

💡 **توجه:** برای اطمینان بیشتر، کد JavaScript در پایین صفحه HTML قرار داده شده است (نباید قبل از

ایجاد عنصر <p> کدمان اجرا شود)

توابع و رویدادها در JavaScript

کدهای JavaScript در صفحه HTML، زمانی که صفحه بارگذاری شود (onload) اجرا خواهند شد. اما این چیزی نیست که ما همیشه نیاز داریم.

بعضی مواقع می خواهیم زمانی که یک رویداد خاص اتفاق می افتد، کدمان اجرا شود، مانند زمانی که کاربر روی یک دکمه کلیک می کند. در این مورد می توانیم کدمان را داخل یک تابع قرار دهیم.

رویدادها در حالت عادی با توابع ترکیب می شوند، مانند صدا زدن یک تابع هنگامی که رویداد رخ می دهد.

در آموزش های بعدی درباره توابع و رویدادهای JavaScript بیشتر خواهید آموخت.

اسکرپت ها در دو قسمت <body> و <head>

می توان به صورت نامحدود، هر تعداد خط اسکرپت را در صفحه HTML قرار داد، و همچنین می توانید اسکرپت

ها را همزمان در دو قسمت <body> و <head> بیاورید.

در حالت کلی، تمام توابع را در قسمت <head> صفحه و یا در پایین صفحه قرار می گیرد. این روش باعث می

شود که تداخلی با محتویات صفحه پیش نیاید.

کدهای JavaScript در قسمت <head>

در مثال زیر، زمانی که روی دکمه کلیک شود یا به عبارتی رویداد onclick عنصر مورد نظر رخ دهد، تابع

displayDate() صدا زده می شود:

مثال:

```
<html>
<head>
<script type="text/javascript">
function displayDate()
{
document.getElementById("demo").innerHTML=Date();
}
</script>

</head>
<body>

<h1>My First Web Page</h1>
<p id="demo"></p>
<button type="button" onclick="displayDate()">Display Date</button>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_events



خودتان امتحان کنید

همچنین در مثال ساده زیر، هنگامی که روی دکمه "Try it" کلیک شود، محتوای عنصر با شناسه "demo" تغییر می کند:

مثال:

```
<!DOCTYPE html>
<html>
<head>

<script>
function myFunction()
{
document.getElementById("demo").innerHTML="My First JavaScript
Function";
}

</script>
</head>
<body>

<h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_where_to_head

خودتان امتحان کنید

یک تابع JavaScript در قسمت <body>

در این مثال، تابع "myFunction" در انتهای <body> قرار داده شده است و وقتی کاربر روی دکمه "Try it" کلیک کند، myFunction صدا زده می شود:

مثال:

```
<!DOCTYPE html>
<html>
<body>

<h1>My Web Page</h1>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>

<script>
function myFunction()
{
document.getElementById("demo").innerHTML="My First JavaScript
Function";
}
</script>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_where_to_body



خودتان امتحان کنید

استفاده از یک فایل JavaScript خارجی

کدهای JavaScript را همچنین می توان در یک فایل خارجی قرار داد.

فایل های خارجی JavaScript اغلب شامل کدهایی است که در چندین صفحه HTML مختلف استفاده می شود.

این فایل ها را باید با فرمت ".js" ذخیره نمود.

توجه: اسکریپت خارجی نمی تواند شامل تگ های باز و بسته <script> و </script> باشد.

برای استفاده از یک فایل اسکریپت خارجی در صفحه HTML، باید خصوصیت "src" تگ <script> را با آدرس فایل، تنظیم کنید:

مثال:

```
<!DOCTYPE html>
<html>
<body>
<script src="/myScript.js"></script>
</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_externalexample

خودتان امتحان کنید

1-3- شروع کار بار JavaScript

JavaScript بطور معمول برای دستکاری عناصر HTML استفاده می شود.

تغییر محتوای عناصر HTML

در JavaScript برای دسترسی به یک عنصر HTML از طریق ID می توان از دستور

document.getElementById() استفاده کرد.

این متد بوسیله شناسه یا (ID) یک عنصر، به آن دسترسی پیدا می کند.

در مثال زیر، محتوای عنصر با شناسه "demo" به مقدار جدید، تغییر می کند:

مثال:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<p id="demo">My First Paragraph</p>
<script>
document.getElementById("demo").innerHTML="My First JavaScript";
</script>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_dom



خودتان امتحان کنید

همچنین در مثال زیر، تاریخ جاری سیستم، داخل عنصری با شناسه "demo" نوشته می شود عنصر (<p>):

مثال:

```
<html>
<body>

<h1>My First Web Page</h1>

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

<script type="text/javascript">
document.getElementById("demo").innerHTML=Date();
</script>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_show_curr_date



خودتان امتحان کنید

دستور document.write()

در مثال زیر، یک عنصر <p> با محتوی "My First JavaScript" در ادامه فایل HTML نوشته می شود:

مثال:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>
<script>
document.write("<p>My First JavaScript</p>");
</script>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_write



خودتان امتحان کنید

💡 **توجه:** در کدهای JavaScript واقعی، از دستور document.write() استفاده نکنید. چون تمام صفحه

دوباره نوشته می شود. با این وجود در اینجا دستور document.write() یک راه ساده برای ایجاد خروجی در

JavaScript را نشان می دهد. (به مثال زیر توجه نمایید)

مثال:

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Web Page</h1>

<p>My First Paragraph.</p>

<button onclick="myFunction()">Try it</button>

<script>
function myFunction()
{
document.write("Oops! The document disappeared!");
}
</script>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_write_over

خودتان امتحان کنید

JavaScript در Windows 8

Microsoft در Windows 8 برای ایجاد برنامه های تحت ویندوز، JavaScript را پشتیبانی می کند. در آینده بطور قطع، JavaScript هم در اینترنت و هم در ویندوز استفاده خواهد شد.

1-4- دستورات JavaScript

JavaScript، ترتیبی از دستورات است که توسط مرورگر اجرا می شود.

حساس به حروف کوچک و بزرگ

برخلاف HTML کدهای JavaScript به حروف کوچک و بزرگ حساس است (case sensitive)، بنابراین هنگامی که کد می نویسید به حروف بزرگ به دقت نگاه کنید.

دستورات JavaScript

یک دستور JavaScript در واقع یک فرمان به مرورگر است. هدف این دستورات این است که به مرورگر بگویید که چه کاری باید انجام دهد.

دستور JavaScript زیر به مرورگر می گوید که رشته "Hello world" را بر روی صفحه وب بنویسد:

```
document.write("Hello world");
```

به طور معمول در پایان هر دستور قابل اجرا، یک سمیکالن (;) اضافه می کنیم. براساس استانداردهای JavaScript، قرار دادن سمیکالن، اختیاری است و مرورگر فرض می کند که آخر خط، انتهای دستور است. به همین خاطر در اغلب مثال ها سمیکالن را در آخر خط نمی بینید.  **نکته:** استفاده از سمیکالن، این امکان را به شما می دهد که چندین دستور را در یک خط بنویسید.

کدهای JavaScript

کدهای JavaScript (یا فقط JavaScript) یک ترتیبی از دستورات JavaScript است.

هر دستور JavaScript به ترتیبی که نوشته شده است، توسط مرورگر اجرا می شود.

این مثال، دو عنصر با شناسه "demo" و "myDIV" را دستکاری می کند:

مثال:

```
document.getElementById("demo").innerHTML="Hello Dolly";  
document.getElementById("myDIV").innerHTML="How are you?";
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_statements



خودتان امتحان کنید

بلوکی از دستورات

دستورات JavaScript می توانند با هم در یک بلوک باشند.

بلوک ها با "{" شروع و با "}" خاتمه می یابند.

هدف از این بلوکها این است که یک ترتیبی از دستوراتی که با هم اجرا می شوند ایجاد شود.

یک مثال خوب برای گروه بندی دستورات در یک بلوک، توابع هستند.

این مثال، تابعی را برای دستکاری دو عنصر، صدا می زند:

مثال:

```
function myFunction()  
{  
  
document.getElementById("demo").innerHTML="Hello Dolly";  
document.getElementById("myDIV").innerHTML="How are you?";  
  
}
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_blocks



خودتان امتحان کنید

در فصل های بعدی با توابع بیشتر آشنا خواهید شد.

فضاهای خالی

JavaScript فاصله های اضافی را در نظر نمی گیرد. برای خواناتر شدن کد می توانید فضاهای خالی را اضافه نمایید. دو خط کد زیر یکسان هستند:

```
var person="Hege";  
var person = "Hege";
```

شکستن یک خط کد

با استفاده از "\" می توانید یک خط کد را بشکنید. مثال زیر بدرستی نمایش داده خواهد شد:

```
document.write("Hello \  
World!");
```

یک خط کد را مانند زیر نمی توانید بشکنید:

```
document.write \  
("Hello World!");
```

1-5- توضیحات (Comment) در JavaScript

توضیحات را می توان برای شرح چگونگی انجام یک بلوک از دستورات به کد JavaScript اضافه کرد، در این صورت کد خواناتری خواهیم داشت.

توضیحات یک خطی با "//" شروع می شود.

در مثال زیر، از توضیحات یک خطی، برای شرح کدها استفاده شده است:

مثال:

```
<script type="text/javascript">
// چاپ یک عنوان
document.write("<h1>This is a heading</h1>");
// چاپ دو پاراگراف
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_comments1



خودتان امتحان کنید

توضیحات چند خطی در JavaScript

توضیحات چند خطی با /* شروع شده و با */ تمام می شوند.

در مثال زیر، از توضیحات چند خطی استفاده شده است:

مثال:

```
<script type="text/javascript">
/*
کدهای زیر یک عنوان و
دو پاراگراف را چاپ خواهد کرد
*/
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_comments2



خودتان امتحان کنید

استفاده از توضیحات برای جلوگیری از اجرا

در مثال زیر، یک خط کد، به توضیحات تبدیل شده و از اجرای آن جلوگیری می کنیم (این راه می تواند برای کشف خطا مناسب باشد):

مثال:

```
<script type="text/javascript">
//document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
</script>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_comments3



خودتان امتحان کنید

در مثال زیر، بلوکی از کدها به توضیحات تبدیل شده است و مانند مثال قبلی از اجرای آنها جلوگیری خواهد شد (این راه می تواند برای کشف خطا مناسب باشد):

مثال:

```
<script type="text/javascript">
/*
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
document.write("<p>This is another paragraph.</p>");
*/
</script>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_comments4



خودتان امتحان کنید

قرار دادن توضیحات در انتهای خطوط

در مثال زیر، توضیحات در پایان یک خط ، قرار داده شده است:

مثال:

```
<script type="text/javascript">
document.write("Hello"); // Write "Hello"
document.write(" Dolly!"); // Write " Dolly!"
</script>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_comments5



خودتان امتحان کنید

1-6- متغیرهای JavaScript

متغیرها، ظرفی برای ذخیره اطلاعات اند:

مثال:

```
var x=5;  
var y=6;  
var z=x+y;
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_data1



خودتان امتحان کنید

آیا جبر را از زمان مدرسه به یاد دارید؟

$x=5, y=6, z=x+y$

آیا به یاد می آورید که یک حرف (مانند x) می تواند برای نگهداری یک مقدار (مثل 5) به کار رود و شما می توانید طبق اطلاعات بالا مقدار z را برابر 11 ارزیابی کنید.

این حروف، متغیر نامیده می شوند و آنها را می توان برای نگهداری مقادیر ($x=5$) و یا عبارات ($z=x+y$) استفاده کرد.

متغیرهای JavaScript

مانند جبر، متغیرهای JavaScript نیز برای نگهداری مقادیر و یا عبارات به کار می روند.

متغیرها می توانند اسمی کوتاه داشته باشند مانند X و یا کمی توصیفی باشند مانند Carname

قواعد نامگذاری متغیرهای JavaScript

- نام متغیرها به حروف کوچک و بزرگ حساس (case sensitive) هستند. (y و Y دو متغیر متفاوت اند)
- نام متغیر می تواند با حروف و یا آندرلاین و یا \$ شروع شود.

💡 **توجه:** چون JavaScript Case-sensitive است، نام متغیرهای آن نیز Case-sensitive است.

انواع داده در JavaScript

در متغیرهای JavaScript می توان، انواع دیگر داده، مانند: متن را نیز ذخیره نمود. ("علی احمدی"=person)
در JavaScript یک متن مانند "علی احمدی" یک رشته نامیده می شود.
در JavaScript انواع مختلف متغیر وجود دارد، اما در حال حاضر، فقط به متغیرهای عددی و رشته ای می پردازیم.

زمانی که یک مقدار رشته ای را به یک متغیر انتساب می دهید، باید آنرا در کوتیشن یا دابل کوتیشن (' یا ") قرار دهید.

زمانی که یک عدد را به یک متغیر انتساب می دهید، نیازی به کوتیشن نیست. اگر آنرا در کوتیشن قرار دهید با آن متغیر، مانند یک رشته برخورد خواهد شد.

مثال:

```
var pi=3.14;  
var person="John Doe";  
var answer='Yes I am!';
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_data2



خودتان امتحان کنید

مثال

مقدار یک متغیر در طول اجرای اسکریپت می تواند تغییر کند. شما می توانید به یک متغیر از طریق نام آن برای تغییر یا نمایش مقدارش دستیابی پیدا کنید.

این مثال به شما چگونگی دسترسی و مقداردهی یک متغیر را نشان خواهد داد.

اعلان (ایجاد) متغیرهای JavaScript

در JavaScript ایجاد متغیر بیشتر اوقات به اعلان متغیر معروف است.

شما می توانید متغیرهای JavaScript را با کلمه کلیدی `var` اعلان کنید:

```
var x;  
var carname;
```

بعد از تعریف به صورت بالا متغیرها خالی هستند(هنوز هیچ اطلاعاتی داخل آنها نیست)، اگر چه شما می توانید آنها را موقع تعریف مقداردهی نمایید.

```
var x=5;  
var carname="Volvo";
```

بعد از اجرای دستورات بالا، متغیر `x` مقدار 5 و `carname` مقدار "volvo" را در خود نگه می دارد.

 **توجه:** اگر یک متغیر JavaScript را بعد از مقدار دهی، دوباره اعلان کنید، مقدار اولیه اش را از دست نخواهد داد.

تعریف چندین متغیر، در یک دستور

می توانید چندین متغیر را در یک دستور اعلان نمایید. فقط دستور را با **var** شروع کنید و متغیرها را با **کاما** از هم جدا نمایید:

```
var lastname="Doe", age=30, job="carpenter";
```

اعلان ها، می توانند در چندین خط باشند:

```
var lastname="Doe",  
age=30,  
job="carpenter";
```

Value = undefined

در برنامه های کامپیوتری، اغلب متغیرها بدون مقدار اعلان می شوند این متغیرها مقدار **undefined** را خواهند داشت.

بعد از اجرای دستور زیر، متغیر **carname** مقدار **undefined** را خواهد داشت:

```
var carname;
```

متغیرهای محلی (LOCAL) در JavaScript

متغیرهایی که داخل یک تابع اعلان شده اند، تنها داخل همان تابع، قابل دسترسی اند. (متغیرهای با قلمرو محلی)

شما می توانید در توابع مختلف، متغیرهای محلی با نام های یکسان داشته باشید. (متغیرهای محلی تنها بوسیله تابعی که آنها را اعلان کرده است شناخته می شوند)

متغیرهای محلی بمحض اینکه عملیات تابع تکمیل شد، حذف می شوند.

در فصل های بعدی، درباره توابع بیشتر خواهید آموخت.

متغیر های عمومی (Global) در JavaScript

متغیرهایی که خارج از توابع اعلان شده اند، عمومی می شوند، و در تمام اسکریپت و توابع داخل یک صفحه به آنها دسترسی خواهد بود.

زمانی که یک صفحه وب را ببندید، متغیرهای عمومی حذف خواهند شد.

مقداردهی متغیرهای اعلان نشده

اگر به متغیری که هنوز اعلان نشده مقداری را اختصاص دهید، آن متغیر به طور خودکار اعلان خواهد شد.

مثال:

```
x=5;  
carname="Volvo";
```

اگر متغیرهای x و carname پیش از این وجود نداشته باشند، به طور خودکار، به عنوان متغیرهای عمومی اعلان خواهند شد(البته اگر دستورات بالا خارج از توابع باشد)

محاسبات در JavaScript

همانند جبر ، شما می توانید عملگرهای محاسباتی را با متغیرهای JavaScript به کار ببرید.

مثال:

```
y=5;  
x=y+2;
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_oper_add



خودتان امتحان کنید

شما با انواع عملگرهای JavaScript در فصل های بعدی بیشتر آشنا خواهید شد.

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

1. [کتاب آموزش JavaScript](#) (کتاب پیش رو)

2. [کتاب آموزش HTML](#)

3. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل 2: اشیاء (Object) در JavaScript

2-1- مقدمه اشیاء (Object) در JavaScript

در JavaScript هر چیزی می تواند یک شیء باشد: یک رشته، یک عدد، یک آرایه، یک تابع و ... علاوه بر این، در JavaScript قابلیت تعریف اشیاء نیز وجود دارد.

اشیاء (Object) در JavaScript

در JavaScript چندین شیء توکار (**built-in**) مانند: رشته، تاریخ، آرایه و ... وجود دارد. یک شیء، تنها یک نوع خاص داده است و می تواند شامل **property** و **method** هایی باشد.

دسترسی به خصوصیات (property) یک شیء

اشیاء، به متغیرهای خاصی دسترسی دارند که به آنها **property** می گویند. (مقادیر وابسته به یک شیء)

نحوه دسترسی به یک **property**:

```
objectName.propertyName
```

در مثال زیر، برای پیدا کردن طول رشته، از خصوصیت **length** استفاده شده است:

```
var message="Hello World!";  
var x=message.length;
```

مقدار **x**، بعد از اجرای کد بالا:

دسترسی به متدهای (Method) یک شیء

Method در واقع عملیاتی است که روی یک شیء اجرا می شود.

نحوه ی فراخوانی یک Method:

```
objectName.methodName()
```

در مثال زیر، با استفاده از متد toUpperCase() حروف رشته "message" حروف بزرگ تبدیل شده:

```
var message="Hello world!";
var x=message.toUpperCase();
```

مقدار x، بعد از اجرای کد بالا:

```
HELLO WORLD!
```

ایجاد اشیاء در JavaScript

در JavaScript قابلیت تعریف اشیاء، وجود دارد.

برای ایجاد یک شیء دو روش مختلف وجود دارد:

1. تعریف و ایجاد یک نمونه مستقیم از شیء

2. تعریف شیء با استفاده از یک تابع به عنوان سازنده (Constructor) و سپس ایجاد یک نمونه جدید از شیء

1- تعریف و ایجاد یک نمونه مستقیم از شیء

در مثال زیر، یک نمونه مستقیم از یک شیء ایجاد شده و سپس چهار property به آن اضافه شده است:

مثال:

```
person=new Object();  
person.firstname="John";  
person.lastname="Doe";  
person.age=50;  
person.eyecolor="blue";
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_create_object



خودتان امتحان کنید

کد بالا را می توان به صورت زیر نیز نوشت:

مثال:

```
person={firstname:"John",lastname:"Doe",age:50,eyecolor:"blue"};
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_create_object1



خودتان امتحان کنید

2- تعریف شیء با استفاده از یک تابع به عنوان سازنده (Constructor) و سپس ایجاد نمونه

در مثال زیر، از یک تابع به عنوان سازنده برای ساخت شیء، استفاده شده است:

مثال:

```
function person(firstname,lastname,age,eyecolor)
{
this.firstname=firstname;
this.lastname=lastname;
this.age=age;
this.eyecolor=eyecolor;
}
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_create_object2

خودتان امتحان کنید

کلمه کلیدی "this" در مثال بالا، به شیء ای که در دست است اشاره دارد.

ایجاد نمونه از یک شیء

زمانی که سازنده ی شیء وجود داشته باشد، برای ایجاد نمونه، مانند زیر عمل نمایید:

```
var myFather=new person("John","Doe",50,"blue");
var myMother=new person("Sally","Rally",48,"green");
```

اضافه کردن Property به اشیاء

این کار بسادگی دادن آن یک مقدار، به شیء موجود است.

به فرض اینکه شیء person قبلاً وجود داشته باشد، می توانید property های age ,lastname

firstname , و eyecolor را مانند زیر به آن اضافه کنید:

```
person.firstname="John";  
person.lastname="Doe";  
person.age=30;  
person.eyecolor="blue";  
  
x=person.firstname;
```

مقدار x، بعد از اجرای کد بالا:

John

اضافه کردن متدها (Method) به اشیاء

متدها، توابعی هستند که به اشیاء ضمیمه شده اند.

تعریف متدهای یک شیء، داخل تابع سازنده (constructor) انجام می شود:

```
function person(firstname,lastname,age,eyecolor)  
{  
  this.firstname=firstname;  
  this.lastname=lastname;  
  this.age=age;  
  this.eyecolor=eyecolor;  
  
  this.changeName=changeName;  
  function changeName(name)  
  {  
    this.lastname=name;  
  }  
}
```

تابع `changeName()` مقدار پارامتر ورودی `name` را به خصوصیت `lastname` اختصاص می دهد:

مثال:

دسترسی به متد `changeName()`

```
myMother=new person("Sally","Rally",48,"green");  
myMother.changeName("Doe");
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_create_object3



خودتان امتحان کنید

JavaScript می داند که شیء `person` ی که شما درباره آن صحبت می کنید کدام است، بنابراین در تابع سازنده، عبارت `this` با `mymother` جابه جا می شود.

کلاس ها در JavaScript

JavaScript یک زبان شیء گرا (object oriented) است، اما از کلاس ها استفاده نمی کند.

در JavaScript کلاس تعریف نمی شود و همچنین از یک کلاس شیء ای ساخته نمی شود (مانند بیشتر زبان های شیء گرا).

JavaScript براساس یک نمونه اولیه است (prototype based) و از کلاس های استفاده نمی کند.

دستور for...in در JavaScript

دستور `for...in` در JavaScript داخل `property` های یک شیء، حلقه می زند.

نحوه استفاده:

```
for (variable in object)
{
  code to be executed
}
```

💡 **توجه:** کد داخل حلقه `for...in`، برای تک تک `property` های شیء، اجرا خواهد شد.

مثال: حلقه زدن داخل `property` های شیء `person`:

مثال:

```
var person={fname:"John",lname:"Doe",age:25};

for (x in person)
{
  txt=txt + person[x];
}
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_object_for_in



خودتان امتحان کنید

2-2- اعداد در JavaScript

JavaScript تنها یک نوع متغیر عددی دارد.

اعداد در JavaScript

متغیرهای عددی می توانند صحیح یا اعشاری باشند.

مثال :

```
var pi=3.14; // Written with decimals
var x=34; // Written without decimals
```

اعدادی که بسیار بزرگ یا بسیار کوچک هستند را می توان بصورت "نماد علمی" نوشت:

مثال :

```
var y=123e5; // 12300000
var z=123e-5; // 0.00123
```

تمام اعداد در JavaScript اعداد 64 بیتی و درمبنای 10 هستند

JavaScript یک زبان متنوع نیست. در اینجا برخلاف بیشتر زبان های برنامه نویسی دیگر، انواع مختلف اعداد مانند: integers, short, long و ... وجود ندارد.

تمام اعداد در JavaScript بر مبنای 10 و 64 بیتی (8 بایتی) و با ممیز شناور هستند.

دقت اعداد

اعداد صحیح، در حداکثر 15 رقم و همیشه به صورت دقیق مطرح می شوند.

حداکثر تعداد اعشار، 17 رقم است، اما حساب ممیز شناور همیشه 100٪ دقیق نیست:

مثال:

```
var x=0.2+0.1;
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_inaccurate



خودتان امتحان کنید

Hexadecimal و Octal

در JavaScript اعداد ثابتی که با صفر شروع می شوند، به صورت octal (هشت هشتی) و اعدادی که با صفر و x شروع می شوند به صورت Hexadecimal تفسیر می شوند.

مثال:

```
var y=0377;  
var z=0xFF;
```

🔗 **توجه:** هرگز اعداد را با صفر مقدم استفاده نکنید، مگر اینکه یک تبدیل octal نیاز باشد.

خصوصیت ها (Property) و متدهای (Method) مربوط به اعداد

: Property

- MAX VALUE
- MIN VALUE
- NEGATIVE INFINITY
- POSITIVE INFINITY
- NaN
- prototype
- constructor

: Method

- toExponential()
- toFixed()
- toPrecision()
- toString()
- valueOf()

2-3- رشته ها در JavaScript

شیء string، برای ذخیره و دستکاری متن استفاده می شود.

رشته ها (String) در JavaScript

در متغیرها یا اشیاء رشته ای، می توان یک سری از کاراکترها مانند "beyamooz" را ذخیره نمود. زمانی که یک مقدار رشته ای را به یک متغیر انتساب می دهید، باید آنرا در کوتیشن یا دابل کوتیشن (' یا ") قرار دهید.

مثال:

```
var carname="Volvo XC60";
var carname='Volvo XC60';
```

می توانید از علامت کوتیشن (') داخل رشته استفاده کنید، البته تا زمانی که با علامتی که رشته را احاطه کرده، تطابق نداشته باشد:

مثال:

```
var answer="It's alright";  
var answer="He is called 'Johnny'";  
var answer='He is called "Johnny"';
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_datatypes_string



خودتان امتحان کنید

راه حل مشکل بالا (تطابق کوتیشن های داخل رشته با احاطه کننده رشته)، استفاده از علامت (\) قبل از کوتیشن های داخل رشته است: (برای توضیح بیشتر، به انتهای این مقاله مراجعه نمایید)

مثال:

```
var answer='It\'s alright';  
var answer="He is called \"Johnny\"";
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_datatypes_string



خودتان امتحان کنید

می توانید به هر کدام از کاراکترهای یک رشته، از طریق مکانش (index)، دسترسی داشته باشید:

مثال:

```
var character=carname[7];
```

🔗 **توجه:** ایندکس رشته از صفر شروع می شود، یعنی اولین کاراکتر [0] است، دومین کاراکتر [1] و ...

طول رشته (String length)

برای فهمیدن طول یک رشته، می توان از خصوصیت **length**، استفاده نمود:

مثال:

```
var txt="Hello World!";  
document.write(txt.length);  
  
var txt="ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
document.write(txt.length);
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_string_length



خودتان امتحان کنید

پیدا کردن یک رشته داخل رشته ای دیگر

در مثال زیر، متد **indexOf()**، اولین مکان رخداد رشته "welcome" را در شیء **str**، برمی گرداند:

مثال:

```
var str="Hello world, welcome to the universe."  
var n=str.indexOf("welcome");
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_string_indexof



خودتان امتحان کنید

💡 **توجه:** اگر رشته مشخص شده پیدا نشد، عدد **-1** برمی گردد.

متد **lastIndexOf()**، بجای اینکه جستجو را از ابتدای رشته شروع کند از آخر رشته شروع می کند.

تطبیق یک رشته در محتوای رشته ای دیگر

در مثال زیر ، اگر پارامتر ورودی متد **match()** ، در شی **str** یافت شد، مقدار یافت شده را برمی گرداند وگرنه مقدار **null** را بر می گرداند:

مثال:

```
var str="Hello world!";
document.write(str.match("world") + "<br>");
document.write(str.match("World") + "<br>");
document.write(str.match("world!"));
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_string_match



خودتان امتحان کنید

جایگزینی در محتوای یک رشته

در مثال زیر، متد **replace()**، مقدار پارامتر **"Microsoft"** را در شیء **str** جستجو می کند و تمام آنها را با مقدار **"Beyamooz.com"** جایگزین می کند:

مثال:

```
str="Please visit Microsoft!"
var n=str.replace("Microsoft","Beyamooz.com");
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_string_replace



خودتان امتحان کنید

حروف بزرگ (Upper Case) و حروف کوچک (Lower Case)

در مثال زیر، متد **toUpperCase()** کاراکترهای شیء **txt** را به حروف بزرگ و متد **toLowerCase()** به حروف کوچک تبدیل می کند:

مثال:

```
var txt="Hello World!"; // String
var txt1=txt.toUpperCase(); // txt1 is txt converted to upper
var txt2=txt.toLowerCase(); // txt2 is txt converted to lower
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_string_toupper



خودتان امتحان کنید

تبدیل رشته به یک آرایه

در مثال زیر، با استفاده از متد **split()** شی **txt** را براساس کارکتر **","** جدا کرده و به ترتیب در یک آرایه قرار می دهد:

مثال:

```
txt="a,b,c,d,e" // String
txt.split(","); // Split on commas
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_string_split



خودتان امتحان کنید

استفاده از کارکترهای خاص در یک رشته

علامت **(\)** برای وارد کردن کاراکترهای خاص مانند: **(\)**، **(')** و یا رفتن به خط جدید در یک رشته، استفاده می شود.

مثال:

```
var txt="We are the so-called \"Vikings\" from the north.";
document.write(txt);
```

در JavaScript یک رشته با علامت " یا ' احاطه می شود. بنابراین رشته بالا شکسته خواهد شد : We are

the so-called

راه حل این مشکل، استفاده از (\) قبل از دابل کوتیشن های داخل رشته است:

```
var txt="We are the so-called \"Vikings\" from the north.";
document.write(txt);
```

خروجی به صورت روبرو خواهد بود: We are the so-called "Vikings" from the north

در جدول زیر، لیستی از دیگر کاراکترهایی که می تواند با (\) داخل متن استفاده شود، آورده شده است:

خروجی	کد
single quote	\'
double quote	\"
backslash	\\
new line	\n
carriage return (رفتن سر خط)	\r
tab	\t
backspace	\b
form feed	\f

متدها (Method) و خصوصیات (Property) مربوط به رشته ها

: Properties

- length
- prototype
- constructor

: Methods

- charAt()
- charCodeAt()
- concat()
- fromCharCode()
- و ...

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

1. [کتاب آموزش JavaScript](#) (کتاب پیش رو)

2. [کتاب آموزش HTML](#)

3. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل 3: آموزش مدل DOM

3-1- کاربرد DOM چیست؟

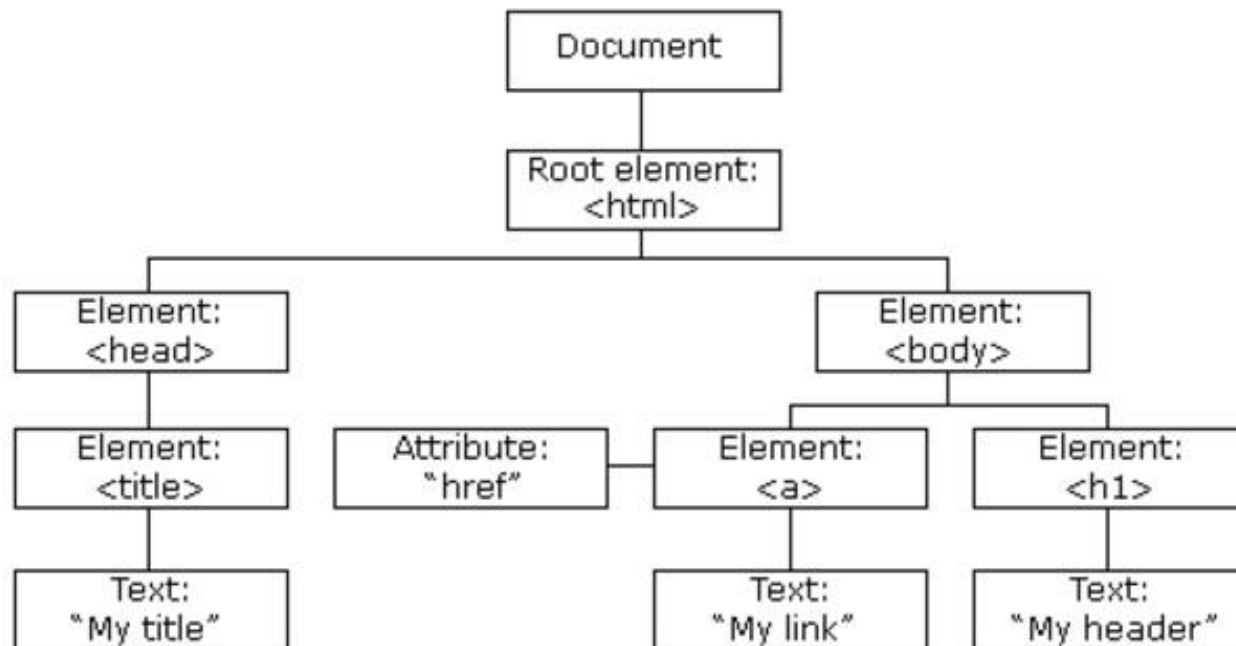
JavaScript با استفاده از مدل DOM، می تواند به تمام عناصر یک صفحه HTML دسترسی داشته باشد.

مدل DOM یا (Document Object Model)

زمانی که یک صفحه ی وب بارگذاری می شود، مرورگر از آن صفحه، یک مدل شیء گرا (Document Object Model) ایجاد می کند.

مدل DOM یک ساختار درختی از اشیاء موجود در صفحه HTML می سازد:

درخت DOM



JavaScript با استفاده از مدل شیء گرای DOM، یک صفحه HTML داینامیک یا پویا ایجاد می کند:

- JavaScript می تواند، تمام عناصر یک صفحه HTML را تغییر دهد.
- JavaScript می تواند، تمام خصوصیات عناصر یک صفحه HTML را تغییر دهد.
- JavaScript می تواند، تمام سبک های CSS یک صفحه HTML را تغییر دهد.
- JavaScript می تواند، به تمام رویدادهای یک صفحه HTML واکنش نشان دهد.

پیدا کردن عناصر HTML

خیلی اوقات می خواهید، با JavaScript عناصر HTML را دستکاری کنید.

برای انجام این کار، ابتدا باید عنصر مورد نظر را پیدا کنید. برای دسترسی به عنصر، راه های مختلفی وجود دارد:

- پیدا کردن عنصر از طریق id
- پیدا کردن عنصر، از طریق نام عنصر
- پیدا کردن عنصر، از طریق نام class

پیدا کردن عنصر، از طریق id

در مدل DOM، آسان ترین راه برای پیدا کردن یک عنصر، از طریق id یا شناسه آن است.

در مثال زیر، به عنصر با شناسه "intro" دسترسی پیدا کرده ایم:

مثال:

```
var x=document.getElementById("intro");
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=try_dom_getelementbyid

خودتان امتحان کنید

اگر عنصر مورد نظر پیدا شد، تابع getElementById() آنرا مانند یک شی برمی گرداند (داخل متغیر x).
اگر عنصر مورد نظر پیدا نشد، محتوای متغیر x خالی (null) خواهد بود.

پیدا کردن عنصر، از طریق نام عنصر

در مثال زیر، ابتدا به عنصر با شناسه "main" و سپس به تمام عناصر با نام <p> داخل "main" دسترسی پیدا کرده ایم:

مثال:

```
var x=document.getElementById("main");  
var y=x.getElementsByTagName("p");
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=try_dom_getelementsbytagname

خودتان امتحان کنید

🔗 **توجه:** دسترسی به عنصر، از طریق نام class در IE نسخه 5، 6، 7 و 8 کار نمی کند.

آموزش DOM

در مطالب بعدی این فصل، به موارد زیر خواهیم پرداخت:

- چگونه در مدل DOM محتوای (innerHTML) عناصر HTML را تغییر دهیم.
- چگونه در مدل DOM ظاهر عناصر HTML را با استفاده از CSS تغییر دهیم.
- چگونه در مدل DOM به رویدادهای یک صفحه، واکنش نشان دهیم.
- چگونه در مدل DOM عناصر HTML را در یک صفحه، حذف و یا اضافه کنیم.

3-2- تغییر محتوای عناصر HTML

مدل DOM به JavaScript اجازه می دهد تا محتوای عناصر HTML را تغییر دهد.

دستور `document.write()`

با JavaScript می توان، محتوای HTML داینامیک ایجاد کرد:

Date:

در JavaScript، با دستور `document.write()` می توان بطور مستقیم، خروجی HTML چاپ کرد:

مثال:

```
<!DOCTYPE html>
<html>
<body>

<script>
document.write(Date());
</script>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=trydhtml_date



خودتان امتحان کنید

🔗 **توجه:** هرگز از دستور `document.write()` بعد از اینکه صفحه بارگذاری شد استفاده نکنید. چون باعث

بارگذاری مجدد صفحه می شود.

تغییر محتوای عناصر HTML

آسان ترین راه برای تغییر محتوای یک عنصر HTML، استفاده از خصوصیت `innerHTML` است.

نحوه استفاده:

```
document.getElementById(id).innerHTML=new HTML
```

در مثال زیر، محتوای عنصر با شناسه "p1" تغییر داده شده است:

مثال:

```
<html>
<body>

<p id="p1">Hello World!</p>

<script>
document.getElementById("p1").innerHTML="New text!";
</script>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_change_innerhtml



خودتان امتحان کنید

در مثال زیر، محتوای عنصر با شناسه "header" تغییر داده شده است:

مثال:

```
<!DOCTYPE html>
<html>
<body>

<h1 id="header">Old Header</h1>

<script>
var element=document.getElementById("header");
element.innerHTML="New Header";
</script>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=trydhtml_dom_innertext



خودتان امتحان کنید

توضیح مثال:

- صفحه HTML بالا، شامل یک عنصر <h1> با شناسه "header" است.
- با استفاده از مدل DOM عنصر با شناسه "header" را پیدا کرده ایم.
- با استفاده از خصوصیت innerHTML در JavaScript محتوای عنصر را تغییر داده ایم.

تغییر خصوصیت عناصر HTML

نحوه استفاده:

```
document.getElementById(id).attribute=new value
```

مثال زیر، خصوصیت src عنصر را تغییر داده است:

مثال:

```
<!DOCTYPE html>
<html>
<body>



<script>
document.getElementById("image").src="/landscape.jpg";
</script>

</body>
</html>
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=trydhtml_dom_image



خودتان امتحان کنید

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

1. [کتاب آموزش JavaScript](#) (کتاب پیش رو)

2. [کتاب آموزش HTML](#)

3. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل 4: آموزش مدل BOM

4-1- مفهوم BOM چیست؟

مدل BOM یا (Browser Object Model) به JavaScript این اجازه را می دهد تا با مرورگر صحبت کند.

BOM یا (Browser Object Model)

استاندارد رسمی برای مدل BOM وجود ندارد.

از آنجایی که مرورگرهای امروزی برای تعامل بیشتر با JavaScript تقریباً به خصوصیت ها و متدهای یکسانی مجهز هستند، اغلب خصوصیت ها و متدهای BOM را نیز دارا هستند.

4-2- شیء Window در JavaScript

شیء window در تمام مرورگرها پشتیبانی می شود و به پنجره مرورگر اشاره دارد.

تمام اشیاء، توابع و متغیرهای عمومی در JavaScript، عضوی از شیء window هستند.

متغیرهای عمومی، خصوصیت های (property) شیء window هستند.

توابع عمومی، متدهای (method) شیء window هستند.

حتی شیء document در مدل DOM نیز یکی از خصوصیت های شیء window است.

```
window.document.getElementById("header");
```

در کد بالا، قسمت window را می توان حذف کرد:

```
document.getElementById("header");
```

اندازه پنجره مرورگر

برای معین کردن اندازه پنجره مرورگر، می توان سه خصوصیت مختلف را بکار برد (اندازه مرورگر شامل toolbars و scrollbars نمی شود):

مرورگرهای Internet Explorer, Chrome, Firefox, Opera, Safari :

- window.innerHeight
- window.innerWidth

مرورگر 5, 6, 7, 8: Internet Explorer

- document.documentElement.clientHeight
- document.documentElement.clientWidth

یا

- document.body.clientHeight
- document.body.clientWidth

یک راه حل عملی در JavaScript (پوشش دهنده تمام مرورگرها):

مثال:

```
var w=window.innerWidth
|| document.documentElement.clientWidth
|| document.body.clientWidth;

var h=window.innerHeight
|| document.documentElement.clientHeight
|| document.body.clientHeight;
```

http://beyamooz.com/try_it_yourself/tryit.php?filename=tryjs_win_inner



خودتان امتحان کنید

مثال بالا، عرض و ارتفاع مرورگر را نمایش می دهد. (بدون toolbars/scrollbars)

متدهای دیگر شیء window

- window.open(): یک پنجره جدید باز می کند.
- window.close(): پنجره جاری را می بندد.

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

1. [کتاب آموزش JavaScript](#) (کتاب پیش رو)

2. [کتاب آموزش HTML](#)

3. [کتاب آموزش CSS](#)

موفق و پاینده باشید

فصل 5: کتابخانه های JavaScript

5-1- کتابخانه های JavaScript

بعضی کتابخانه های معروف JavaScript:

- **jQuery**
- **MooTools**
- **Prototype**

کتابخانه های JavaScript

برنامه نویسی پیشرفته در JavaScript (بخصوص پیچیدگی کارکردن با مرورگرهای مختلف) می تواند اغلب، مشکل و زمان گیر باشد.

برای برخورد با این مشکل، کتابخانه های زیادی در JavaScript گسترش یافته است.

کتابخانه های JavaScript را اغلب JavaScript framework نیز می نامند.

در این آموزش، نگاهی به بعضی از کتابخانه های محبوب JavaScript خواهیم انداخت:

- **jQuery**
- **MooTools**
- **Prototype**

تمام این کتابخانه ها، توابعی برای انجام کارهای معمول در JavaScript مانند: دستکاری عناصر، کار با Ajax و متحرک سازی را دارند.

jQuery

امروزه jQuery محبوب ترین کتابخانه JavaScript است.

در jQuery برای دسترسی به عناصر HTML و دستکاری آنها از گزینشگرهای CSS استفاده شده است. (CSS selector)

بعضی از شرکت های بزرگی که از jQuery استفاده کرده اند:

- Google
- Microsoft
- IBM
- Netflix

اگر می خواهید درباره jQuery مطالب بیشتری را بخوانید، لطفاً به لینک [jQuery HOME](#) مراجعه نمایید.

Prototype

Prototype یکی از کتابخانه های JavaScript است که یک API ساده برای انجام کارهای معمول در JavaScript فراهم نموده است.

API مخفف کلمات (Application Program Interface) به معنی "واسط برنامه های کاربردی" است و کتابخانه ای شامل خصوصیت ها و متدهایی برای دستکاری عناصر HTML است.

Prototype کارآیی JavaScript را با فراهم نمودن کلاس ها و ارث بری، افزایش می دهد.

MooTools

کتابخانه MooTools ، نیز frameworkی است که یک API ساده را برای برنامه نویسی آسان تر در JavaScript فراهم نموده است.

MooTools، همچنین شامل توابعی برای افکت دهی و متحرک سازی است.

CDN

CDN مخفف کلمات Content Delivery Networks است. (تحويل محتوی، روی شبکه)

شما همیشه می خواهید که صفحات وبتان، تا جای ممکن سریع و کم حجم باشند و مرورگر نیز تا جای ممکن آنها را cache کند.

از کتابخانه های JavaScript، در وب سایت های مختلف، استفاده می شود، بنابراین احساس می شود که باید یک مکان مشترک برای استفاده عموم از این کتابخانه ها روی اینترنت ایجاد شود.

CDN این مشکل را حل کرده است. CDN، یک شبکه از چندین کتابخانه عمومی JavaScript است.

Google، یک CDN رایگان از کتابخانه های JavaScript فراهم نموده است:

- jQuery
- Prototype
- MooTools
- Dojo
- Yahoo! YUI

برای استفاده از کتابخانه های JavaScript در وب سایتتان، فقط کافی که آدرس URL آن کتابخانه را با استفاده از تگ <script> به صفحه اضافه نمایید:

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js">  
</script>
```

پایان پیش نمایش این فصل، ادامه در فایل اصلی ...!

توجه: پیش نمایش بالا، تنها با هدف جلب اطمینان شما از محتوای اصلی تهیه شده است. بنابراین اگر مورد تأیید شما قرار گرفت لطفاً از طریق سایت Beyamooz.com اقدام به خرید فایل کامل آن نمایید.

لینک خرید کتاب های آموزشی

1. [کتاب آموزش JavaScript](#) (کتاب پیش رو)

2. [کتاب آموزش HTML](#)

3. [کتاب آموزش CSS](#)

موفق و پاینده باشید

همراهان خوبی باشید و با

ری پست کردن ( Repost) این محتوی در [لینکدین پیاموز](#)

ما را در تولید دوره های آموزشی مفید

حمایت کنید.

از طریق صفحه لینکدین پیاموز، ما را **فالو** کنید.

+ Follow

برای باخبر شدن از آخرین دوره های آموزشی، کافیه در

لینکدین پیاموز روی **آیکن زنگ** کلیک کنید.



Beyamooz

