



آموزش JAVA SCRIPT

ارائه شده به : دکتر حسن حقیقی

توسط : سیده زهرا حسینی و سیده الهه جلمبادانی

تیر ۹۰

فهرست

تاریخچه

چند برنامه ساده

تفاوت جاوا اسکریپت با جاوا

قابلیت‌های جاوا اسکریپت

تگ اسکریپت

شیء جاوا اسکریپت چیست؟

انواع اشیاء

در کجا باید اسکریپت نوشت

چگونگی دریافت پیغام خطا

انواع خطاها

نکات مهم برای اسکریپت نویسی

۳

۴

۶

۷

۷

۱:

۱.۱

۱.۴

۱.۷

۱.۸

۲:

تاریخچه

جاوا اسکریپت را در ابتدا شخصی به نام Eich Brendan در شرکت Netscape با نام Mocha طراحی نمود. این نام بعداً به LiveScript و نهایتاً به جاوا اسکریپت تغییر یافت این تغییر نام تقریباً با افزوده شدن پشتیبانی از جاوا در مرورگر وب Navigator Netscape همزمانی دارد. اولین نسخه جاوا اسکریپت در نسخه ۳B۲.۰ این مرورگر در دسامبر ۱۹۹۵ معرفی و عرضه شد. این نام گذاری منجر به سردرگمی‌های زیادی شده و این ابهام را ایجاد می‌کند که جاوا اسکریپت با جاوا مرتبط است در حالی که این طور نیست. عده زیادی این کار را یک ترفند تجاری برای به دست آوردن بخشی از بازار جاوا که در آن موقع زبان جدید مطرح برای برنامه نویسی تحت وب بود می‌دانند.

ساختار این زبان به سی پلاس پلاس (C++) و جاوا شباهت دارد؛ که این امر برای سهولت یادگیری در نظر گرفته شده است. از همین رو دستورهای متداول مانند if, for, try..catch و ... در این زبان هم یافت می‌گردند.

به دلیل موفقیت عمده جاوا اسکریپت در نقش زبان نویسه ای سمت کارخواه (client side scripting language) برای صفحات وب، مایکروسافت یک نسخه سازگار از این زبان را ایجاد کرد و به علت مشکلات حقوقی آن را Jscript نامید. این زبان در نسخه ۳.۰ از مرورگر Internet Explorer و در آگوست ۱۹۹۶ داده شد. تفاوت های این دو زبان به حدی جزئی است که اغلب Jscript و جاوا اسکریپت به جای هم به کار می‌روند. هرچند که مایکروسافت در اینجا چند ده دلیل برای تفاوت Jscript با استاندارد ECMA مطرح می‌کند.

Netscape جاوا اسکریپت را به سازمان International Ecma برای استاندارد سازی ارسال کرده است و نتیجه نسخه استاندارد شده‌ای به نام Script ECMA است.

جاوا اسکریپت به یکی از پر طرفدار در وب تبدیل شده است. هر چند ابتدا بسیاری از برنامه نویسان حرفه‌ای زبان را کم ارزش تلقی می‌کردند چون مخاطبین آن نویسندگان صفحات وب و آماتورهای این چینی بودند ظهور AJAX بار دیگر جاوا اسکریپت را در معرض توجه قرار داد و برنامه نویسان حرفه‌ای بیشتری را به خود جذب نمود. نتیجه ازدیاد فریمورک و کتابخانه‌های جامعی در این زمینه، بهبود شیوه‌های رایج برنامه نویسی در جاوا اسکریپت و افزایش کاربرد جاوا اسکریپت خارج از وب است.

چند برنامه ساده به جاوا اسکریپت

برنامه‌های ورودی و خروجی در مرورگر وب:

```
<script type="text/javascript">
  alert ("Hello world");
  var x = prompt("Enter Name:", "Default name");
  alert ( "Hello, " + x )
</script>
```

برای استفاده ساده از عناصر HTML و DOM:

```
<script type="text/javascript">
  document.write("Hello world");
  function handle() {
    document.write("You entered: " + document.getElementById("myinput").value );
  }
</script>
<form>
<input type="text" id="myinput"/>
<input type="button" onclick="handle()" value="Click here" />
</form>
```

برنامه sort:

```
<script type="text/javascript">
  var myarray = [ ۱, ۵, ۸, ۲, ۳, ۰, ۲, ۹, ۱۱ ];
  var i = ۰;
  var j = ۰;
  for (i = ۰; i < myarray.length ; ++i )
  {
    for (j = ۰; j < myarray.length; ++j)
    {
      if (myarray[i] < myarray[j])
      {
```

```

        var tmp = myarray[i];
        myarray[i] = myarray[j];
        myarray[j] = tmp;
    }
}

for (i = 0; i < myarray.length; ++i)
{
    document.write (myarray[i] + " ");
}
</script>

```

برنامه swap:

```

<script type="text/javascript">
    var a = ۱۰, b = ۴۰;
    [a,b] = [b,a]
    alert(" " + a + " " + b);
</script>

```

برنامه factorial:

```

<script type="text/javascript">
    function factorial(x) {
        if (x > ۰)
            return x * factorial(x-۱);
        return ۱;
    }
    alert(factorial(۵));
</script>

```

برای یادگیری جاوااسکریپت، نیاز به دانستن زبان HTML و درک کاملی از طراحی و ساخت صفحات وب باید داشت چون جاوااسکریپت یک زبان مجزا از اچ تی ام ال نیست و باید همراه با کدهای HTML در یک صفحه

بکار رود تا کارآیی داشته باشد.

زبان اچ تی ام ال به تنهایی جوابگوی نیاز بینندگان وب سایتها نمی باشد چون آنها علاقه خاصی به جذابیتها و ارتباط متقابل میان خود و صفحات وب دارند که اچ تی ام ال نمی تواند جذابیت قابل توجهی به یک صفحه بدهد و یا با یک کاربر ارتباط برقرار کند. پس برای برطرف کردن این مشکلات، صفحات وب نیازمند یک زبان قوی تر می باشد که به همین منظور زبان جاوااسکریپت خلق شد.

دو مرورگر معروف وب، Internet Explorer و Netscape Navigator از این زبان پشتیبانی می کنند. شرکت مایکروسافت هم برای اینکه از نت اسکریپت در این زمینه عقب نماند، نسخه مربوط به خود را بنام Jscript تولید کرد.

تفاوت جاوااسکریپت با جاوا

۱. جاوااسکریپت متعلق به شرکت نت اسکریپت می باشد اما زبان جاوا متعلق به شرکت Sun Microsystems است.
۲. جاوااسکریپت در اصل یک زبان اسکریپت نویسی است نه یک زبان برنامه نویسی کامپیوتر، شاید بتوان اسکریپتها را یک نوع زبان برنامه نویسی خیلی سبک و ساده نامید اما جاوا یک زبان برنامه نویسی کاملاً پیچیده مانند زبان C است.
۳. جاوااسکریپت توسط یک برنامه مفسر یعنی interpreter در مرورگر تعریف و تفسیر شده تا قابل اجرا باشد و حتماً هم این کار در مرورگر انجام می شود اما برنامه های جاوا باید توسط یک کامپایلر Compiler ، کامپایل شده که کامپایل شدن یعنی جمع آوری اطلاعات مورد نظر و ترجمه آن اطلاعات به زبان قابل فهم برای کامپیوتر و تبدیل آن به یک برنامه کاربردی واحد و مستقل برای کاربر را گویند. به همین دلیل برنامه های جاوا را که اپلت applet می نامند، مستقل اجرا می شوند. البته اپلتهای جاوا را می توان در صفحات وب هم گنجانند.
۴. جاوااسکریپت فقط در صفحات وب تأثیرگذار است اما زبان جاوا برای خلق برنامه های پیچیده در کامپیوتر ساخته شده است.

قابلیتهای جاوااسکریپت

در حقیقت جاوااسکریپت یک ابزار برای طراحان وب و نویسندگان کدهای HTML می باشد. جاوااسکریپت دارای ترکیباتی ساده نسبت به زبانهای برنامه نویسی است که براحتی می توان آنها را داخل کدهای اچ تی ام ال یک صفحه قرار داد و از آنها استفاده کرد. یکی از قابلیت‌های ویژه جاوااسکریپت رایگان بودن آن است که اگر شما زبان HTML را بدانید، می توانید با کمی دانستن جاوااسکریپت کدهای آماده که بیش از ۲۰۰۰ نوع از آنها در وب وجود دارد را در صفحات خود بکار گیرید.

با استفاده از این زبان می توانید متنهای پویا و متحرک در صفحات و در نقاط مختلف مرورگرها مانند نوار وضعیت آنها خلق کنید. یکی دیگر از قابلیت‌های این زبان عکس العمل نشان دادن به وقایع و اتفاقات می باشد بطور مثال واکنش نسبت به کلیک یکی از دکمه های موس یا فشردن یک کلید بر روی کیبورد و یا انجام عملی هنگام بسته شدن صفحه مرورگر.

جاوااسکریپت حتی می تواند کدهای اچ تی ام ال را بخواند و یا تولید کند که توسط این خاصیت می توانید در هر زمان خاص یک متنی را تولید یا تغییر دهید و یا در زمانی مشخص عکسی را جایگزین عکسی دیگر کنید.

یکی از بهترین ویژگیهای جاوااسکریپت، اعتبار بخشیدن به اطلاعات می باشد مثلاً شما یک فرمی برای ارتباط با کاربر خود ساخته اید و می خواهید قبل از اینکه آن اطلاعات به سرور فرستاده شود، بازبینی شده تا اگر مشکلی وجود داشته باشد به کاربر اطلاع داده شود تا در صدد رفع آن برآید. چون جاوااسکریپت اصطلاحاً side Client است، یعنی در سیستم کاربر اجرا شده و مستقل از وب سرور است می تواند ترافیک روی سرور را کم کرده که این موضوع یک امتیاز مثبت برای میزبانان وب محسوب می شود

آغاز اسکریپت نویسی

تگ اسکریپت

جاوااسکریپت هم مانند زبان HTML از یکسری دستور و گرامر خاص خود پیروی می کند. البته این زبان به سادگی اچ تی ام ال نیست و بسیار هم حساس است، مثلاً در اچ تی ام ال فاصله بین تگها اهمیتی ندارد و مرورگر آنرا نادیده می گیرد اما در جاوااسکریپت فاصله باید متناسب باشد که اگر رعایت نشود با پیغام خطا

روبرو خواهید شد. در قسمت بعد با چگونگی پیغامهای خطا در جاوااسکریپت آشنا می شوید. همانگونه که اطلاع دارید کدهای جاوااسکریپت را باید در میان تگهای HTML قرار دهید. به همین دلیل یک تگ برای آن در نظر گرفته شده است:

```
<script> ... </script>
```

که کلیه دستورات جاوااسکریپت باید مابین آنها قرار گیرد. این تگهای اچ تی ام ال به مرورگر می فهمانند که در کجا این زبان آغاز شده تا مفسر جاوااسکریپت مرورگر فعال شود و در کجا پایان می پذیرد که برنامه مفسر هم کار خود را خاتمه دهد.

language

اما تگ script مانند بقیه تگها دارای خصوصیتی می باشد، یکی از آنها language است که مقدار آنرا باید برابر با javascript قرار دهید،

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

برای مقدار javascript می توانید نسخه آن را هم بنویسید. این زبان هم مانند اچ تی ام ال دارای نسخه های مختلفی است که هر کدام آنها با قابلیت های جدیدی در وب توسعه یافتند. نسخه های این زبان عبارتند از ۱.۰ ، ۱.۱ ، ۱.۲ ، ۱.۳ ، ۱.۴ و آخرین نسخه حال حاضر هم ۱.۵ می باشد که هم اکنون این نسخه با مرورگر مایکروسافت یعنی اینترنت اکسپلورر سازگاری ندارد. فعلاً مرورگر اینترنت اکسپلورر از نسخه ۱.۳ این زبان پشتیبانی می کند.

type

تگ اسکریپت خصوصیت type هم دارد که برای این زبان باید آنرا برابر با text/javascript قرار دهید. نکته قابل توجه اینست که اگر این خصوصیت را به این تگ اضافه کنید، مرورگر اینترنت اکسپلورر نسخه ۱.۵ این زبان را نادیده می گیرد و کد را اجرا می کند. شما می توانید با اجرای کد زیر هم اولین اسکریپت خود را تجربه کنید و هم خصوصیت type را در IE امتحان کنید.

خب مانند زبان اچ تی ام ال می توانید از یک ویرایشگر متن مانند برنامه Notepad استفاده کرده و کد زیر را در آن وارد کنید:

```
<html>
<head>
<title> My JavaScript </title>
</head>
<body>
<script language="javascript\۱.۵" type="text/javascript">
document.write(" Hello my friends ")
</script>
</body>
</html>
```

اجرای این کد

اکنون این فایل را با پسوند htm ذخیره کنید و سپس آنرا در مرورگر اینترنت اکسپلورر اجرا کنید که جمله Hello my friends را باید در صفحه ببینید. حالا خصوصیت type را حذف کنید و دوباره فایل را ذخیره و مشاهده کنید، هیچ متنی در صفحه دیده نمیشود که با تغییر نسخه ۱.۵ به ۱.۳ آن متن دوباره نمایان خواهد شد. نوشتن نسخه جاوااسکریپت اجباری نیست یعنی می توانید آنرا ننویسید.

حالا ببینیم که چه اتفاقی هنگام اجرای این کد می افتد. تگ اسکریپت و خصوصیات آن روشن است اما به عبارت زیر دقت کنید:

```
document.write(" Hello my friends ")
```

ببینیم در اینجا این عبارت چگونه برای مرورگر تفسیر می شود. document یعنی همان سند و فایل html که در آن این عملیات باید انجام شود و مرورگر به محض برخورد با این عبارت آماده می شود برای انجام کاری که بعد از یک نقطه نوشته می شود. در اینجا باید عمل write یعنی نوشتن را انجام دهد اما چی باید بنویسد، طبق قرارداد باید هر چیزی که در بین پرانتز هست را بنویسد و چون محتوای داخل پرانتز مابین دو علامت نقل قول یعنی " " قرار گرفته پس باید بصورت یک string در بیایند. string یعنی یک رشته کاراکتر که می تواند هم حرف باشد هم عدد و هم علامت و خاصیت string در این است که هر چیزی مابین دو علامت " " قرار گرفت عیناً باید در صفحه نوشته شود.

در جاوااسکریپت به document یک object یعنی شیء می گویند و هنگامی که یک نقطه بعد از آن گذاشته می شود آن شیء منتظر method خود می شود که در اینجا همان write است. به عبارت ساده این خط می گوید که آن شیء را بردار و در آن توسط متد نوشتن چیزی بنویس.

src

تگ script یک خصوصیت دیگر هم دارد که src است و با آن آشنا هستید، بله همانطور که حدس زدید برای آدرس دهی است. کدهای جاوااسکریپت را هم مانند استایل شیت می توانید در یک فایل جداگانه نوشته و با پسوند js ذخیره کنید سپس با خصوصیت src آدرس آن فایل را مشخص کنید. توجه داشته باشید که در آن فایل نباید دیگر تگ script را بنویسید. معمولاً در این حالت تگ اسکریپت در قسمت head گنجانده می شود تا با شروع صفحه فایل جاوااسکریپت فراخوانی شود.

```
<html>
<head>
<script language="javascript" src="jsfiles/name.js" type="text/javascript">
</script>
</head>
```

در بخشهای قبل با بعضی از object ها یا همان شیء های جاوااسکریپت مانند، شیء Date و آرایه ها آشنا شدید. در این صفحه قصد داریم در مورد شیء های دیگر در جاوااسکریپت صحبت کنیم.

شیء جاوااسکریپت چیست؟

وظیفه شیء ها را می توان شبیه به متغیرها دانست اما شیء ها بسیار کامل تر از متغیرها هستند چون آنها می توانند اطلاعات بیشتری را در خود ذخیره کنند و همچنین تفاوت دیگری هم که با متغیرها دارند اینست که متغیرها باید تعریف شوند تا جاوااسکریپت آنها را بشناسد اما بعضی از شیء ها در جاوااسکریپت ساخته و تعریف شده اند. این شیء های از قبل تعریف شده را درون ساخت یا Built-in می گویند مانند شیء Date. البته همانطور که میدانید باید همین اشیاء هم در ابتدای کد یکبار دیگر با دستور new تعریف شوند:

```
mydate = new Date ( )  
name = new Array ( )
```

پس از این دستور برای شروع یک object استفاده کنید.

کلیه شیء ها دارای خصوصیات و متدهای شناخته شده برای جاوااسکریپت هستند که آنها را با گذاشتن یک نقطه بعد از خود شیء بکار می بریم و می توانیم مقادیر مختلفی را برای آنها در نظر بگیریم.

انواع اشیاء

به غیر از شیء های درون ساخت Date و Array که در صفحات قبل با آنها آشنا شدید، شیء های درون ساخت دیگر جاوااسکریپت عبارتند از شیء String و Math و شیء هایی که مربوط به HTML DOM می شوند. در بخش های بعدی به ترتیب به توضیح هر کدام از این اشیاء خواهیم پرداخت.

دستور with

در این دستور شما می توانید یک شیء را به همراه خصوصیات و متدهایش بصورت پیش فرض در اسکریپت خود تعریف کنید. این دستور به نوعی کد شما را راحت تر و خلاصه تر می کند و مانند توابع فقط اسم تعیین شده را در کدها بکار می برید که جاوااسکریپت کلیه خصوصیات و متدهای آنرا برایش در نظر خواهد گرفت. قالب دستور with به شکل زیر است:

```
with ( name ) {  
    .... javascript code ....  
}
```

روش ساخت یک شیء

در بالا صحبت از یک سری اشیاء شد که جاوااسکریپت از قبل با آنها آشناست اما گاهی اوقات نیاز هست که یک شیء را خودتان تعریف کنید تا بهتر بتوانید اسکریپت خود را مدیریت کنید. البته توسط دستور with هم می توانید اینکار را انجام دهید، به این صورت که اسم شیء مورد نظر را بجای name در الگوی بالا بنویسید و خصوصیت و متد آنرا در خطوط بعدی بین { } تعریف کنید.

اما اگر بخواهید یک شیء را در اسکریپت خود بسازید نیاز به دو چیز دارید، ابتدا یک تابع سازنده که به آن Constructor می گویند و در آن نوع شیء تعیین می شود، سپس به نمونه شیء که از آن تابع استفاده می کند و به آن Instance می گویند که توسط دستور new تعریف می شود. این مطلب ممکن است کمی گیج کننده باشد اما به مثال زیر توجه کنید تا کاملاً به ساخت شیء توسط خودتان تسلط پیدا کنید.

می خواهیم یک شیء برای اتومبیل تعریف کنیم که خصوصیاتش مانند اسم، رنگ و مدل را داشته باشد. برای اینکار ابتدا یک تابع باید تعریف کنیم بنام car و خصوصیات آنرا به عنوان پارامترهایش در پرانتز بنویسیم و سپس این پارامترها را بر اساس خصوصیاتش مقداردهی می کنیم:

```
function Car ( name, model, color ) {  
  this.name = name  
  this.model = model  
  this.color = color  
}
```

به عبارت this دقت کنید، این یک دستور جاوااسکریپت است و به شیء که در حال تعریف آن هستیم، اشاره دارد و در حقیقت به مفسر جاوااسکریپت می فهماند که این خصوصیت شیء تعریف شده با این مقدار است و شما باید از این دستور استفاده کنید تا خصوصیت یک شیء را برایش تعریف کنید و مقدار آنها را هم مانند بالا مشخص کنید.

حالا باید یک شیء جدید برای آن بسازید که این شیء جدید را در متغیری بنام mycar ذخیره می کنیم:

```
mycar = new Car ( "Corvette", "Z۰۶", "Red" )
```

این خط کد به جاوااسکریپت می فهماند که یک شیء بنام mycar دارای خصوصیات اسم که Corvette و مدل Z۰۶ و رنگ Red می باشد. شما می توانید هر شیء دیگر که بخواهید با دستور new بسازید:

```
thecar = new Car ( "BMW", "Z۴", "Silver" )
```

الان می خواهیم از شیء اول اسم ماشین و از شیء دوم رنگ را در صفحه چاپ کنیم:

```
document.write ( mycar.name + "<br />" )  
document.write ( thecar.color )
```

اجرای کد

بعد از اینکه خصوصیتی برای یک شیء تعریف کردید حالا نوبت به متدهای شیء ساخته شده، می رسد. در اینجا متدی که تعریف می کنیم برای کلیه ماشینها می توانیم بکار ببریم. الگوی کلی یک متد به شرح زیر است:

```
object.methodname( ) = function_name
```

object شیء است که ساختیم، methodname نام متدی است که شما در نظر می گیرید و function name اسم تابعی که باید بسازید. سپس هر کجا که لازم بود از متد استفاده کنید ابتدا شیء و سپس با گذاشتن یک نقطه متد را به همراه پرانتز می نویسید که اگر این متد پارامتری هم داشت باید آنرا داخل پرانتز معین کنید.

متدی که می خواهیم برای شیء Car بسازیم را باید ابتدا بصورت یک تابع تعریف کنیم:

```
function displaycar ( ) {  
var result = " I like " + this.name + " " + this.model + " with " + this.color + " color."  
document.write ( result )  
}
```

حالا با استفاده از دستور this این تابع را به عنوان متد شیء Car تعریف می کنیم:

```
function Car ( name, model, color ) {  
this.name = name  
this.model = model  
this.color = color  
this.displaycar = displaycar  
}
```

بعد از تعریف توابع، شیء را توسط دستور new می سازیم و متد را اجرا می کنیم:

```
mycar = new Car ( "Corvette", "Z۰۶", "Red" )  
mycar.displaycar ( )
```

اجرای کد

دستور prototype

در طول اسکریپت شما، ممکن است به جایی برسید که نیاز داشته باشید یک خصوصیت و یا یک متد دیگر برای شیء ساخته شده در خارج از آن توابع تعریف شود که با استفاده از دستور prototype می توانید آنها را تعریف کنید. الگوی این دستور به این صورت می باشد:

object.prototype.method or property = value

کلیه دستورات بالا را توسط توابع و آرایه می توانید انجام دهید اما تعریف کردن شیء در اسکریپت های طولانی بسیار راحت تر و سریع تر از نوشتن بقیه دستورات می باشد.

در صفحات بعدی، با شیء های درون ساخت string ، math و همچنین شیء های مربوط به HTML DOM آشنا خواهید شد

در کجا باید اسکریپت نوشت

در حال حاضر شما میدانید که برای اجرا شدن کدهای جاوااسکریپت باید آنها را در میان تگهای اچ تی ام ال بکار برید. شاید این سؤال برایتان پیش بیاید که در کجا باید آنها را بنویسید، آیا جای خاصی دارند یا در هر کجا می توانید اینکار را انجام دهید. کدهای جاوااسکریپت می توانند در هر کجای سورس کد باشند اما باید ابتدا تشخیص دهید که بکاربردن هر قسمت از اسکریپت در کجا مناسب تر می باشد. بطور کل کدهای جاوااسکریپت به سه دسته تقسیم میشوند، یک سری از آنها در قسمت سر یعنی head جای می گیرند و یک سری در قسمت بدنه کدهای اچ تی ام ال و دسته سوم هم در یک فایل جداگانه و خارج از کدهای html با پسوند js قرار می گیرند که در ادامه به توضیح هر دسته خواهیم پرداخت.

نوشتن اسکریپت در قسمت head

کلیه کدهای جاوااسکریپت به محض باز شدن یک صفحه وب در مرورگر اجرا می شوند اما گاهی اوقات ما نیاز داریم تا یک اسکریپت زودتر از کدهای دیگر اجرا شده و یا آماده باشد تا در صورت لزوم فراخوانی شود. که اینگونه کدها را باید در قسمت سر و ما بین تگ head بنویسیم. بطور مثال شما نیاز به این دارید که به محض

ورود کاربر به صفحه در همان ابتدا پیغامی ظاهر شود و تا کاربر آنرا تأیید نکرده است صفحه تکمیل نشود. اسکریپت‌های قسمت سر همیشه آماده اجرا و یا آماده پاسخ به اتفاقی در صفحه وب هستند، که در این مورد شما مطمئن هستید اسکریپت مورد نظر قبل از هر چیزی در صفحه، اجرا خواهد شد. برای نوشتن کدهای جاوااسکریپت بصورت زیر عمل می کنید:

```
<html>
<head>
<title> ... </title>
<script language="javascript" type="text/javascript">
..... javascript codes .....
</script>
</head>
```

نوشتن اسکریپت در قسمت بدنه

هنگامیکه نیاز داشته باشید تا کدهای جاوااسکریپت در بدنه فایل html اجرا شود آنها را در قسمت body صفحه وارد می کنید،

```
<html>
<head>
</head>
<body>
<script language="javascript" type="text/javascript">
..... javascript codes .....
</script>
</body>
</html>
```

شما حتی می توانید توسط کدهای جاوااسکریپت قسمتی از یک صفحه را بسازید یعنی با ترکیب جاوااسکریپت و تگهای اچ تی ام ال در body یک عنصر در صفحه وارد کنید. کدهای زیر را بنویسید و اجرا کنید:

```
<html>
<head>
</head>
<body>
<script language="javascript" type="text/javascript">
```

```
document.write("<h\ align='center'> Hello my friends! </h\>");  
</script>  
</body>  
</html>
```

اجرای کد

لازم بذکر است که شما می توانید برای نظم دادن و جلوگیری از شلوغی قسمت بدنه یک اسکریپت بصورت function در قسمت head تعریف کنید و در قسمت بدنه آنرا فراخوانی کنید که در قسمت های بعد در این مورد بیشتر توضیح خواهم داد.

فایل خارجی جاوااسکریپت

شما می توانید برای حفظ نظم تگهای اچ تی ام ال در یک صفحه وب و یا برای جلوگیری از تکرار یک اسکریپت در صفحات یک وب سایت، کدهای جاوااسکریپت را در یک فایل جداگانه نوشته و با پسوند js آنها را ذخیره کنید سپس در هر قسمت و هر صفحه ای که لازم بود، وارد کنید. خط زیر را در برنامه Notepad نوشته و آنرا به اسم test.js ذخیره کنید:

```
document.write(" JavaScript external file ")
```

حالا یک فایل اچ تی ام ال بسازید و تگهای زیر را در آن وارد کنید سپس با یک اسم دلخواه آنرا در همان پوشه که فایل جاوااسکریپت را ذخیره کرده اید قرار دهید:

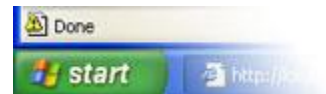
```
<html>  
<head>  
</head>  
<body>  
<script language="javascript" type="text/javascript" src="test.js">  
</script>  
</body>  
</html>
```

اگر مراحل را بدرستی انجام داده باشید، جمله زیر را در مرورگرتان مشاهده خواهید کرد.

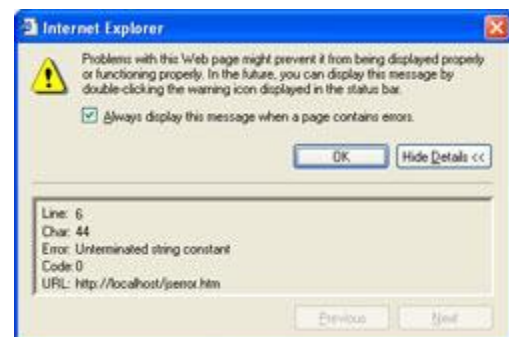
دقت کنید که اگر فایل جاوااسکریپت را در پوشه ای جدا از فایل html قرار می دهید حتماً در خصوصیت src آدرس دقیق آنرا بنویسید

چگونگی دریافت پیغام خطا

یکی از مواردی که در نوشتن کدهای جاوااسکریپت ممکن است زیاد با آن برخورد کنید، دریافت پیغام های خطا هنگام اجرای کدها در مرورگر می باشد، پس بهتر است در همین ابتدای کار با آنها آشنا شوید تا هنگام مشاهده خطا، تجربه کافی برای رفع آن داشته باشید. خطاهای جاوااسکریپت را به دو صورت متوجه می شوید یکی ظاهر شدن مثلی زرد با یک علامت تعجب در سمت چپ نوار وضعیت مرورگر مانند شکل زیر:



در بعضی از مرورگرها این نوار مشاهده نمی شود که می توانید برای مشاهده آن به منوی View مرورگر رفته و روی گزینه Status Bar کلیک کنید. حالت دیگر دیدن پیغامهای خطا، باز شدن یک پنجره در وسط مرورگر است که در این حالت جزئیات خطای ایجاد شده را هم می توانید مشاهده کنید. مانند شکل زیر:



توجه داشته باشید، در نسخه های جدید مرورگر اینترنت اکسپلورر این پنجره مشاهده نمیشود و ابتدا آنرا باید فعال کنید. برای فعال کردن آن به منوی Tools مرورگر رفته و گزینه Internet Options را انتخاب کنید، در پنجره باز شده به قسمت Advanced رفته و گزینه notification about script error Display a تیک بزنید، سپس دکمه Ok را زده تا کار شما تأیید شود. بهتر است که قبل از شروع کد نویسی جاوااسکریپت این عمل را انجام دهید تا هنگام بروز خطا در صدد رفع آن برآیید که به این کار اشکال زدایی یا Debugging می گویند.

اگر هم این گزینه را فعال نکردید، هنگام بروز خطا می توانید بر روی آن مثلث زرد رنگ در نوار وضعیت رفته و دوبار کلیک کنید که همان پنجره اعلان خطا باز خواهد شد.

انواع خطاها

اساساً دو نوع خطا برای جاوااسکریپت وجود دارد، خطاهای نحوی یا Syntax errors و خطاهای زمان اجرا یا Run time errors که در خطاهای نحوی یک مشکلی در ترکیبات کدها بوجود آمده مثلاً یک غلط املایی در یک دستور یا جا انداختن یک حرف یا علامت و در خطاهای زمان اجرا احتمالاً یک دستور بطور اشتباه جای یک دستور دیگر نوشته شده است که در هنگام اجرای کد، مفسر مرورگر نمی تواند این دستور را تجزیه تحلیل کند و خطا را اعلام می کند.

ایجاد یک خطا

خب حالا اجازه دهید تا این پیغامها را تجربه کنیم و ببینید که به چه راحتی توسط آنها می توانید اشکال زدایی کرده و کد خود را به سرعت اصلاح کنید. ممکن است که دهها بار با پیغام خطا روبرو شوید ولی ناامید نشوید چون حرفه ای ها هم بارها و بارها با آن مواجه می شوند. حالا کد زیر را با نام error.htm ذخیره کنید:

```
<html>
<head>
<title> Error message </title>
</head>
<body>
<script language="javascript" type="text/javascript">
document.write("I'm ready for first error message.");
</script>
</body>
</html>
```

اجرای کد

یکبار آنرا اجرا کنید و نتیجه را مشاهده کنید. حالا پرانتز دوم را حذف می کنیم، بدین صورت:

```
document.write("I'm ready for first error message.");
```

سپس فایل را دوباره ذخیره کرده و اجرا می کنیم. همانطور که انتظار داشتیم پنجره پیغام خطا باز شد، اگر جزئیات پیغام را نداری روی دکمه Details یا Show Details کلیک کنید تا ببینیم که چه اطلاعاتی را می توانیم بدست آوریم. در کادر جزئیات اولین گزینه Line می باشد که اگر مطابق کد بالا عمل کرده باشید باید در جلوی آن عدد ۷ نوشته شده باشد. این خط به ما می فهماند که ایراد در خط هفتم سورس کد است. دقت کنید، تنها خطهای مربوط به جاوااسکریپت را نباید بخوانید بلکه کلیه خطوط از بالا به پایین خوانده می شود، حتی خطهای خالی هم باید شمرده شوند.

دومین گزینه، کلمه Char است که بیانگر عدد شمارش شده کاراکتری که باعث بروز خطا شده است و در اینجا آن عدد ۵۲ می باشد. فراموش نکنید که هر فاصله بین کلمات هم یک کاراکتر به حساب میاید. شاید اینجا سؤالی برایتان پیش بیاید که چرا عدد ۵۲؟ در صورتی که در اینجا پرانتز کاراکتر ۵۱ است. باید یک نکته را خدمتتان عرض کنم که علامت ; در جاوااسکریپت یک کاراکتر اختیاری است که نشان دهنده پایان یک خط از کد می باشد و برنامه نویسان بر حسب عادتی که دارند از این علامت استفاده می کنند ، در اینجا مفسر جاوااسکریپت آنرا به عنوان یک کاراکتر قبل از کاراکتر اصلی خطا شناسایی کرده است.

گزینه بعدی Code می باشد که ظاهراً بطور معمول عدد صفر جلوی آن می باشد و هنوز نتوانستم کارآیی آنرا تشخیص بدهم و گزینه آخر هم URL است که آدرس فایل اچ تی ام ال است که در آن خطا تولید شده است.

در مجموع شما برای اشکال زدایی به سه گزینه اول نیاز دارید که با توجه به آنها بلافاصله می توانید محل بروز اشکال را تشخیص دهید. یک نکته را به خاطر بسپارید که در کدهای طولانی ممکن است با تعداد خطاهای زیادی برخورد کنید که در بیشتر مواقع برطرف کردن اولین خطا باعث رفع کلیه اشکالات می شود.

به عنوان جمله آخر این قسمت، همیشه به پنجره های اعلان خطا با دید مثبت نگاه کنید چون آنها به ما کمک می کنند تا اشکال را براحتی پیدا کنیم

نکات مهم برای اسکریپت نویسی

قسمت مقدماتی آموزش جاوا اسکریپت به پایان رسید و همانطور که مشاهده کردید دستورات این زبان، توضیح داده شد تا با استفاده از آنها بتوانید اسکریپتهایی برای صفحات وب خود بنویسید و یا اگر با اسکریپتی برخورد کردید، کاملاً آنرا درک کنید و مطابق نیاز خود آن را اصلاح کنید. فراموش نکنید که این دستورات و کدها را مرتب تمرین کرده و اسکریپتهای جدید و ساده برای خود بسازید تا این زبان در ذهن شما قرار گیرد. بهترین راه به خاطر سپردن دستورات جاوا اسکریپت خلق کدهایی برای خودتان و دیگران است.

هرگاه خواستید که اسکریپتی خلق کنید، ابتدا آنرا در ذهن خود بررسی کنید که این اسکریپت چه کاری باید انجام دهد و چه نتیجه ای را می خواهید از آن بگیرید، سپس افکارتان را روی کاغذ پیاده کنید با جزئیات کامل که چه اطلاعاتی را باید دریافت کنید و توسط چه دستوراتی آنرا پیش ببرید و در آخر چه نتیجه ای باید بدهد. هنگام خلق یک اسکریپت نکاتی را باید در نظر داشته باشید:

سعی کنید بعد از هر خط علامت ; را بگذارید تا پایان آن خط مشخص شود.

```
document.write ( " text string " );  
document.write ( myFunction ) ;
```

جاوا اسکریپت case sensitive است یعنی به حروف کوچک و بزرگ حساس است و بین آنها فرق می گذارد. در جاوا اسکریپت myFunction با myfunction تفاوت دارد.

دقت کنید کلیه علامتها مانند پرانتز، آکلاد و کروشه در هر جا که می نویسید باید دوباره آنها را ببندید.

```
(( [ ..... ] ))
```

جاوااسکریپت فاصله های اضافی را در نظر نمی گیرد، پس شما می توانید برای خوانا بودن کد خود فاصله هایی را اضافه کنید. دو خط زیر تفاوتی با هم ندارند:

```
var name="value";  
var name = " value " ;
```

در نظر داشته باشید که یک خط کد جاوااسکریپت حتماً باید در یک خط نوشته شود و نمی توانید ادامه آنرا در خط بعدی بنویسید، مگر اینکه به انتهای صفحه برسید و خود به خود به ابتدای خط بعد هدایت شوید. اما در میان رشته های متنی با گذاشتن علامت \ می توانید به خط بعد رفته و ادامه دهید.

```
document.write ( " Hello \  
my friends! " );
```

شما می توانید جلوی کدهای خود یک توضیح و یا یک نظریه را وارد کنید تا برای مراجعتهای بعدی یک سر نخ داشته باشید و سریع متوجه بشوید که آن قسمت از اسکریپت، چه کاری را انجام می دهد و یا اینکه شما یک اسکریپت برای یک فرد مبتدی می سازید و می خواهید بعضی از قسمتهای کد را توضیح دهید که برای این منظور ها می توانید در هر خط از اسکریپت که می خواهید علامت // گذاشته و سپس آن توضیح را بنویسید.

```
document.write( " text " ) ; // your comment
```

اگر هم نظریه یا توضیح شما بیشتر از چند کلمه است می توانید به صورت زیر عمل کنید:

```
/* This is a comment  
block. It contains  
several lines */
```

لازم بذکر است که این توضیحات و نظریات در صفحه نمایش داده نمی شوند.

در حال حاضر کلیه مرورگرها از جاوااسکریپت پشتیبانی می کنند ولی بعضی ممکن است از مرورگرهای قدیمی استفاده کنند و یا مرورگرهایی که قابلیت جاوااسکریپت را نداشته باشند، بکار برند. در این حالت شما یک تگ بعد از مجموعه اسکریپت خود به تگهای اچ تی ام ال صفحه اضافه می کنید به اسم `noscript` و متنی را داخل آن می نویسید تا به آن کاربر نشان داده شود و کاربر از کدهای جاوااسکریپت آن صفحه با خبر شود.

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

```
<noscript> Your browser does not support javascript.</noscript>
```

پایان