

روش اول این است که با استفاده از یکی از زبان‌های برنامه‌نویسی سمت سرویس‌دهنده، صفحه مذکور پیاده‌سازی شود. در این حالت وقتی کاربر اطلاعات موردنیاز را وارد و روی دکمه «محاسبه» کلیک می‌کند، این اطلاعات برای سرویس‌دهنده ارسال شده و پس از انجام محاسبات لازم، نتیجه برگشت داده می‌شود. طبیعتاً این کار مستلزم یک بار رفت و برگشت اطلاعات و اجرا عملیات روی سرویس‌دهنده و واحدی زمان‌بر است.

روش دوم پیاده‌سازی، استفاده از زبان‌های سمت سرویس‌دهنده مانند جاوا اسکریپت است. در این حالت، فرمول موردنیاز برای محاسبه سود بانکی (در قالب یک قطعه کد) در هنگام درخواست نمایش اولیه صفحه برای رایانه کاربر فرستاده می‌شود. هنگامی که کاربر، اطلاعات موردنیاز را وارد و روی دکمه «محاسبه» کلیک می‌کند، کد اجرا شده و نتیجه روی صفحه ظاهر می‌شود. در این حالت، رفت و برگشت اطلاعات وجود ندارد، عملیات روی سرویس‌دهنده کاهش می‌یابد و عملیات با سرعت فوق‌العاده بیش‌تری انجام می‌شود.

فرض کنید همین کار برای اعتبارسنجی داده‌های وارد شده در یک فرم اینترنتی انجام گیرد. به جای این که فرم با اطلاعات ناقص چندین بار برای سرویس‌دهنده ارسال شود و پاسخ خطا از سمت سرویس‌دهنده برای رایانه کاربر ارسال گردد، می‌توان با استفاده از کدهای جاوا اسکریپت، عملیات اعتبارسنجی را روی مرورگر کاربر انجام داد تا ضمن جلوگیری از رفت‌وبرگشت اطلاعات و افزایش بی‌مورد ترافیک شبکه، فشار کاری روی سرویس‌دهنده هم کاهش یابد. این کار در مواقعی که قرار است برنامه‌ای بزرگ با ده‌ها هزار کاربر نوشته شود (مانند نرم‌افزار ثبت‌نام اینترنتی کنکور سراسری) اهمیت خود را به خوبی نشان می‌دهد.

علاوه بر این‌ها امروزه با جستجو در اینترنت می‌توانید صدها قطعه کد جاوا اسکریپت بیابید که توسط برنامه‌نویسان خلاق برای ایجاد جلوه‌های بصری درون مرورگر نوشته شده‌اند و ظاهر و کارایی صفحات را به نحو خیره‌کننده‌ای افزایش می‌دهند. نمایش ساعت درون صفحه، تغییر ظاهر اشاره‌گر ماوس، ظاهر نمودن پیغام روی صفحه و ... تنها بخشی از قابلیت‌های این زبان اسکریپت‌نویسی است.



هنگام معرفی جاوا اسکریپت همواره این نکته هم مورد تأکید قرار می‌گیرد که علی‌رغم تشابه اسمی، رابطه‌ای میان جاوا اسکریپت و زبان جاوا وجود ندارد و حتی نگارش این زبان اسکریپتی به زبان C نزدیک‌تر است.

در مجموع می‌توان موارد زیر را به عنوان کاربردهای اصلی جاوا اسکریپت ذکر کرد:

- خواندن عناصر صفحه و افزودن عناصر و اجزاء جدید به صفحه
- دستکاری یا جابه‌جا کردن عناصر و نوشته‌ها

- اجرای محاسبات ریاضی روی داده‌ها
- واکنش نسبت به وقوع رویدادهایی نظیر بارگذاری صفحه یا کلیک شدن دکمه
- استخراج اطلاعاتی مانند نسخه مرورگر، زمان و تاریخ، ابعاد صفحه و ... از مرورگر کاربر
- اعتبارسنجی داده‌های وارد شده درون فرم و نمایش پیغام‌های مناسب

جاوا اسکریپت یک زبان اسکریپت‌نویسی شیء‌گرا^۱ محسوب می‌شود که کاربرد آن در طراحی صفحات وب روزبه‌روز افزایش می‌یابد. امروزه عباراتی مانند AJAX (با تلفظ ای جکس) یا JQuery (با تلفظ جی کوئری) در میان طراحان صفحات وب زیاد به گوش می‌خورد که هر دو منشاء جاوا اسکریپتی دارند.

AJAX که از عبارت Asynchronous JavaScript and XML به معنی «ترکیب ناهمگام جاوا اسکریپت و XML» گرفته شده، به گونه‌ای قابلیت‌های جاوا اسکریپت و XML را ترکیب نموده تا لازم نباشد با وقوع هر رویدادی در صفحه وب، یک بار عملیات رفت و برگشت صفحه به سرور انجام شود.

برای نمونه احتمالاً در موتورهای جستجوی معروف مانند Google یا Yahoo مشاهده کرده‌اید که کاربر به محض وارد کردن عبارت موردنظر، با فهرستی از پیشنهاد‌های موتور جستجو مواجه می‌شود که پیش از این توسط سایر کاربران مورد جستجو قرار گرفته‌اند. در این حالت، صفحه بدون این‌که تازه‌سازی (Refresh) شود عبارت وارد شده را برای موتور جستجو ارسال نموده و نتایج را به صورت پویا نمایش می‌دهد.

در پشت این اتفاق بدیع، فن‌آوری AJAX کار ارسال و دریافت اطلاعات را انجام می‌دهد و برنامه‌نویسی جاوا اسکریپت، اهمیت و قدرت خود را به رخ می‌کشد.



JQuery را نیز می‌توان به صورت کتابخانه‌ای آماده از جاوا اسکریپت تصور کرد که برای ساده کردن کدنویسی جاوا اسکریپت ایجاد شده است. برای استفاده از این فن‌آوری شما با مراجعه به وب‌سایت www.jquery.com آخرین نسخه موجود را در قالب یک فایل متنی با پسوند .js دانلود و به صفحه وب خود اضافه می‌کنید. سپس با به‌کارگیری متدهای JQuery که با استفاده از جاوا اسکریپت نوشته شده می‌توانید به سادگی، جلوه‌های بصری زیبا و پیچیده‌ای را به وب‌سایت خود اضافه نمایید.

احتمالاً نمونه‌ای از کاربرد JQuery را در وبسایت‌های ارایه‌دهنده تصاویر مشاهده کرده‌اید که با کلیک روی تصویر کوچک، پیش‌نمایش آن درون صفحه نشان داده می‌شود و پس‌زمینه صفحه تیره یا مات می‌گردد. در واقع این کار با استفاده از کدهای جاوا اسکریپت انجام می‌شود که روش دشوار آن، نگارش کامل کد و روش ساده‌تر استفاده از کتابخانه آماده و متدهای JQuery است.



این جلوه تصویری که LightBox نام دارد، یکی از رایج‌ترین موارد استفاده از JQuery است که در بسیاری از وبسایت‌ها مورد استفاده قرار می‌گیرد.

۱۷-۴ افزودن اسکریپت به صفحه HTML

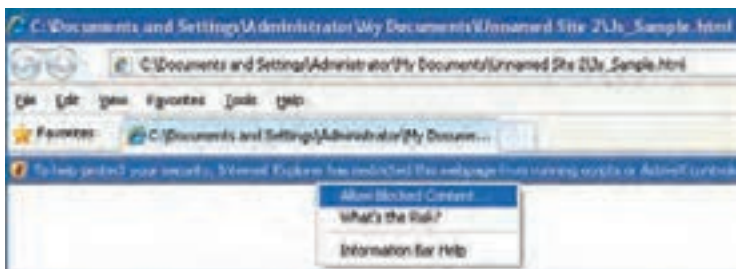
کدهای جاوا اسکریپت را می‌توان به دو صورت درون صفحات وب گنجانید. روش اول این است که اسکریپت‌ها را مستقیماً درون کدهای HTML بنویسید. در روش دوم که متداول‌تر و اصولی‌تر است، کدها درون یک فایل متنی با پسوند js نوشته می‌شوند و سپس درون صفحه موردنظر به این فایل ارجاع داده می‌شود. (دقیقاً مانند کاری که در هنگام استفاده از فایل‌های CSS فراگرفتید).

فعالاً در ابتدای کار، با روش اول یعنی گنجانیدن کدهای جاوا اسکریپت در میان کدهای HTML آشنا خواهید شد. در این روش باید به نحوی اسکریپت‌ها را از سایر کدها جدا کنید. برای انجام این کار، لازم است اسکریپت‌ها را درون برچسب `<script></script>` قرار دهید. از آن‌جا که علاوه بر جاوا اسکریپت، اسکریپت‌های دیگری نظیر VBScript هم وجود دارد که می‌توان درون کدهای HTML جای داد، باید با مقداره‌ی مشخصه type معین نمایید که از چه نوع اسکریپتی استفاده شده است. برای به‌کارگیری جاوا اسکریپت باید مشخصه type را با مقدار "text/javascript" مقداره‌ی کنید.

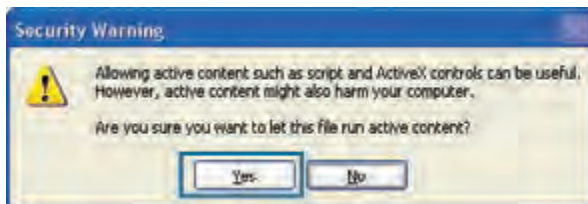
کد زیر را درون یک ویرایشگر متنی وارد و به صورت صفحه HTML ذخیره نمایید.

```
<html>
<head>
</head>
<body>
<p>
<script type="text/javascript">
document.write("My first JavaScript")
</script>
</p>
</body>
</html>
```

هنگامی که صفحه را درون مرورگر باز می‌کنید یک نوار زردرنگ در بالای پنجره ظاهر می‌شود و اعلام می‌کند که یک اسکریپت می‌خواهد روی رایانه شما اجرا گردد و مرورگر وب به دلیل مسایل امنیتی آن را مسدود نموده است. برای اجرای اسکریپت، روی این پیغام کلیک نموده و گزینه Allow Blocked Content را انتخاب نمایید.



بلافاصله پیغامی ظاهر می‌شود و برای اجرای این دستور از شما کسب تکلیف می‌کند. با کلیک روی دکمه Yes اسکریپت موجود اجرا و نتیجه روی صفحه نشان داده می‌شود.

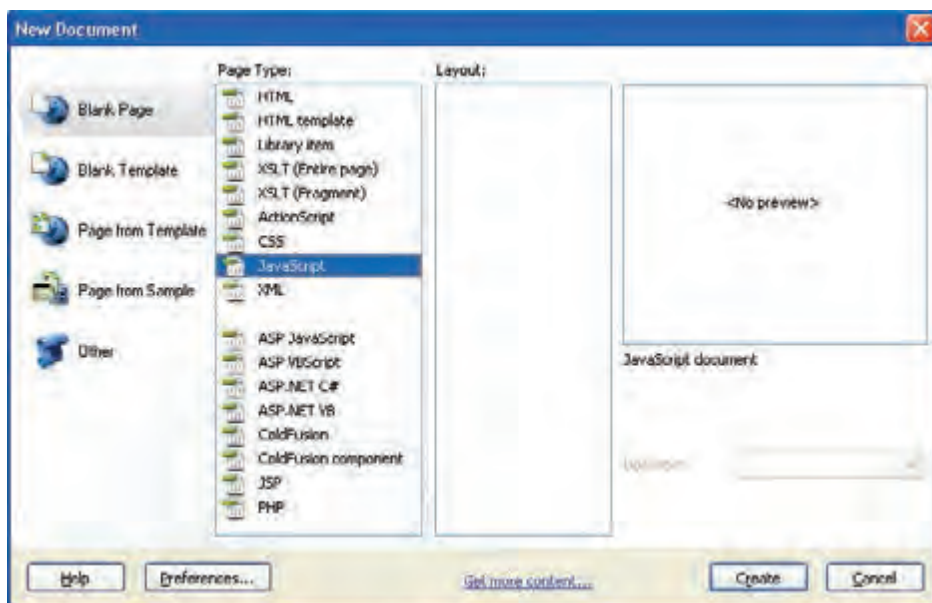


در این مثال، عبارت "My first JavaScript" روی صفحه ظاهر می‌گردد.

اما همان گونه که پیش از این توضیح داده شد، روش اصولی تر برای استفاده از کدهای جاوا اسکریپت در صفحه، قرار دادن آن‌ها درون یک فایل خارجی و ارجاع دادن به آن درون صفحه وب است. چراکه به این ترتیب همه صفحات می‌توانند از اسکریپت‌های نوشته شده در فایل استفاده کنند و نیاز به نگارش مجدد آن‌ها نیست. برای انجام این کار به روش زیر عمل کنید.

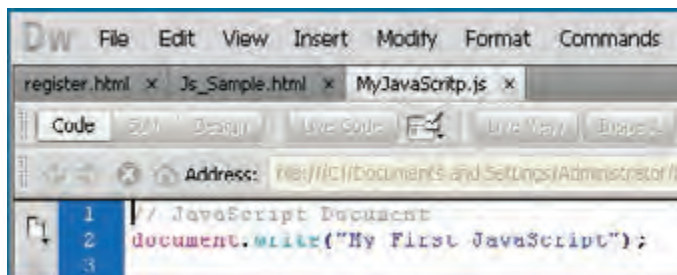
۱. از منوی File دستور New را اجرا کنید.

۲. در ستون Page Type گزینه JavaScript را انتخاب نمایید.



۳. دکمه Create را کلیک نمایید.

۴. کد موجود در تصویر صفحه بعد را درون صفحه بنویسید.



۵. با اجرای دستور Save، فایل را درون یکی از پوشه‌ها (برای مثال پوشه Scripts) با نام MyJavaScripts ذخیره نمایید. این فایل با پسوند .js ذخیره خواهد شد.

۶. درون بخش سرصفحه یا بدنه، کد زیر را اضافه نمایید.

```
<script type="text/javascript" src="Scripts/MyJavaScripts.js"> </script>
```

در این کد، مشخصه src با نشانی نسبی فایل حاوی کدهای جاوا اسکریپت مقداردهی شده است.

۷. تغییرات را ذخیره‌سازی و صفحه وب را درون مرورگر مشاهده نمایید.

۸. کد جاوا اسکریپت همانند حالت‌های قبلی به اجرا در می‌آید.

توجه داشته باشید چنان‌چه در درج این کدها اشتباه تایپی داشته باشید، مثلاً نام یا مسیر فایل خارجی را به درستی وارد نکنید، اسکریپت اجرا نخواهد شد و متأسفانه هیچ پیغام خطایی هم درون مرورگر ظاهر نمی‌شود.

اغلب طراحان وب، کدهای جاوا اسکریپت را درون یک فایل خارجی ذخیره نموده و در صفحه به آن ارجاع می‌دهند. چنان‌چه لازم باشد کد، هم‌زمان با بارگذاری صفحه به اجرا دربیاید، برچسب‌های `<script></script>` را درون بدنه صفحه قرار می‌دهند. اما غالباً اجرای اسکریپت‌ها مشروط به وقوع یک رویداد (مثلاً کلیک شدن یک دکمه) خواهد بود؛ به همین دلیل ارجاع به فایل خارجی عمدتاً درون سرصفحه (برچسب‌های `<head></head>`) گذاشته می‌شود.

هنگامی که در محیط نرم‌افزار Dreamweaver یک فایل جدید js ایجاد می‌کنید، در ابتدای صفحه عبارتی خاکستری‌رنگ به صورت زیر ظاهر می‌شود:

```
// JavaScript Document
```

این کد تأثیری در نتیجه اجرای اسکریپت ندارد چون با افزوده شدن نویسه‌های //، تبدیل به توضیح^۱ شده است. هنگامی که کدهای جاوا اسکریپت و هر زبان دیگر زیاد یا پیچیده می‌شود، برای درج توضیح پیرامون عمل‌کرد هر بخش از عمل‌گرهای توضیحی استفاده می‌شود که هنگام اجرای کد توسط مرورگر نادیده گرفته می‌شوند.

از عمل‌گر // برای ایجاد توضیحات یک‌سطری استفاده می‌شود و چنان‌چه توضیح بیش از یک سطر است، باید آن را درون عمل‌گر /* */ قرار دهید.

کد درون فایل MyJavaScripts.js را به صورت زیر تغییر دهید.

```
// JavaScript Document
```

```
عبارت درون پرانتز را روی صفحه می‌نویسد//
document.write("My First JavaScript");
```

1 . Commnet

یک پنجره ظاهر و پیغام /*
 *درون پرانتز را نمایش می دهد
 window.alert("به دنیای جاوا اسکریپت خوش آمدید");

پس از ذخیره‌سازی تغییرات، صفحه وب را درون مرورگر باز کنید.
 اسکریپت اجرا می‌شود اما اثری از توضیحات نیست.



اغلب مرورگرها به کاربران امکان می‌دهند تا اجرای اسکریپت‌ها را محدود کنند. به همین این احتمال وجود دارد که کدهای جاوا اسکریپت شما روی مرورگر برخی کاربران نمایش داده نشود. برای بررسی این وضعیت و اعلام هشدار به کاربر می‌توانید از برچسب `<noscript>` استفاده کنید. محتویات این برچسب تنها در صورتی ظاهر خواهد شد که مرورگر کاربر، از اجرای اسکریپت‌ها جلوگیری نماید.

کد صفحه وب را به صورت زیر تغییر دهید.

```
<html>
<head></head>
<body><p>
<script type="text/javascript" src="Scripts/MyJavaScripts.js"></script>
<noscript>اجرای کدهای جاوا اسکریپت روی رایانه شما غیرفعال است</noscript>
</p>
</body></html>
```

با نمایش صفحه درون مرورگر، پیغام زیر ظاهر خواهد شد و در صورت کلیک کاربر روی نوار زردرنگ و انتخاب گزینه Allow Blocked Content، کدهای جاوا اسکریپت به اجرا در خواهد آمد و پیغام ناپدید می‌شود.



گاهی اوقات ممکن است در کد صفحات XHTML با عبارتی به صورت زیر مواجه شوید؛

```
<script type="text/javascript">
<!--
document.write("My first JavaScript")
//-->
</script>
```

قرار دادن اسکریپت‌ها درون علامت‌های `<!-- -->` این مزیت را دارد که چنانچه مرورگر کاربر قدیمی باشد و احياناً از کدهای جاوا اسکریپت پشتیبانی نکند، دستور موجود درون این عملگر نادیده گرفته می‌شود و لذا مشکلی در نمایش صفحه به وجود نخواهد آمد.

۵-۱۷ آشنایی با مفهوم شیء‌گرایی

همان‌طور که پیش از این اشاره گردید، جاوا اسکریپت یک زبان شیء‌گرا محسوب می‌شود. بهتر است پیش از ورود به مبحث کدنویسی با جاوا اسکریپت، با مفهوم شیء‌گرایی آشنایی مختصر اما دقیقی داشته باشیم؛ چرا که در مثال ساده‌ای که درون همین بخش مرور کردیم، از مفاهیم شیء‌گرایی استفاده نمودیم و ادامه کتاب نیز از آن بهره خواهیم برد.

تا پیش از تولید زبان‌های شیء‌گرا، برنامه‌ها به صورت ساخت‌یافته ایجاد می‌شدند. یعنی مجموعه‌ای از توابع، عملیات موردنیاز را پیاده‌سازی می‌کردند و برنامه‌نویس با احضار توابع، ارسال پارامترهای موردنیاز و دریافت نتایج، کارهای موردنظر را انجام می‌داد.

شیء‌گرایی در واقع نوعی نگاه و روش جدید در برنامه‌نویسی محسوب می‌شود که به دنیای پیرامون ما شبیه‌تر است. یک مشتری را در یک نرم‌افزار بانک‌داری در نظر بگیرد. این مشتری تعدادی «خصوصیت» و تعدادی «رفتار» دارد. برای مثال می‌توان شماره ملی، نام، نام خانوادگی و موجودی حساب را جزو خصوصیات وی در نظر گرفت. همچنین رفتارهایی مانند واریز وجه، وام گرفتن و ... را به عنوان رفتارهای وی برشمرد. حال اگر ما مفهوم مشتری را به صورت مجموعه‌ای از خصوصیت‌ها و رفتارها در نظر بگیریم، می‌توانیم نمونه‌هایی از این مفهوم را تولید کرده و خصوصیات و رفتارهای ویژه را به آن نسبت دهیم.

در برنامه‌نویسی شیء‌گرا، به این مفهوم، Class، به خصوصیت‌های آن Property و به رفتارها، Method گفته می‌شود. وقتی هم یک نمونه واقعی از آن مفهوم ایجاد می‌شود می‌گوییم یک Object یا شیء را ایجاد کرده‌ایم و منشاء نام‌گذاری این روش برنامه‌نویسی هم همین مفهوم است.

وقتی از یک کلاس، شیئی ساخته شد می‌توانیم با استفاده از خصوصیت‌ها و رفتارهای آن، نیازهای برنامه‌نویسی خود را رفع کنیم و عملیات‌های پیچیده را به سادگی انجام دهیم. فرض کنید از مفهوم مشتری (Customer) یک نمونه واقعی به نام (Customer1) ساخته‌ایم. حال می‌توانیم خصوصیت‌های او را مقداردهی کنیم؛ مثلاً بنویسیم:

```
Customer1.Name=حمید
```

یا عملیات واریز وجه او را که نوعی رفتار محسوب می‌شود با یک متد پیاده‌سازی کنیم و مثلاً بنویسیم:

```
Customer1.Deposite(مبلغ)
```

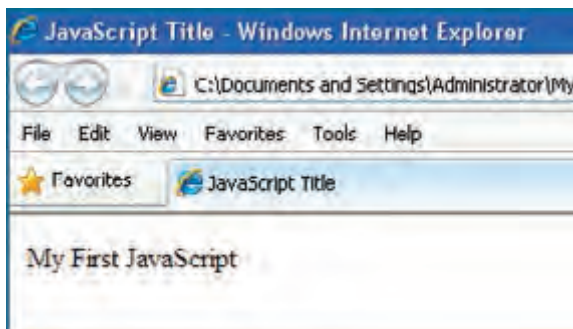
که در این رفتار باید مبلغ واریزی را برای متد ارسال کنیم تا برای مثال در پایگاه داده ثبت نماید. حال به کدی که در همین فصل نوشتیم برمی‌گردیم. برای این‌که عبارتی روی صفحه نوشته شود از دستور ("message") document.write استفاده کردیم. با توضیحاتی که ارائه شده می‌توانید حدس بزنید که document شیئی است که از روی مفهوم صفحه تولید شده یعنی یک نمونه واقعی از مفهوم صفحه است و اشاره به صفحه جاری در مرورگر دارد. متد write() هم عبارتی را می‌گیرد و روی صفحه درج می‌کند. با این توضیحات، به نظر شما کد زیر چه کاری انجام می‌دهد؟

```
document.title="JavaScript Title"
```

بله! خصوصیت عنوان صفحه (title) را با عبارت "JavaScript Title" مقداردهی می‌کند و لذا در نوار عنوان مرورگر، همین عبارت نشان داده می‌شود. اگر کد زیر را در فایل خارجی جاوا اسکریپت وارد و تغییرات را ذخیره نمایید، هنگام باز کردن صفحه در مرورگر، چیزی شبیه به تصویر زیر خواهید دید.

```
document.write("My First JavaScript");
```

```
document.title="JavaScript Title";
```



استفاده از روش شیء‌گرایی در تولید نرم‌افزار، پیچیدگی‌های فرایند برنامه‌نویسی را کاهش می‌دهد و ضمناً شما می‌توانید از مفاهیم تولید شده (کلاس‌ها) در برنامه‌های دیگر هم استفاده کنید. برای نمونه، خصوصیت‌ها و رفتارهای تعریف شده برای کلاس دانش‌آموز در نرم‌افزار مدیریت مدرسه می‌تواند با تغییرات اندکی در نرم‌افزار آموزشی دانشگاه هم مورد استفاده قرار گیرد و به این ترتیب، هزینه و زمان تولید نرم‌افزار به شکل قابل‌ملاحظه‌ای کاهش خواهد یافت.

در فصل آینده هنگام بررسی «مدل اشیاء سند» با اشیاء متداول در جاوا اسکریپت و متدهای آن‌ها بیشتر آشنا خواهید شد.



چکیده فصل

جاوا اسکریپت زبانی است که برای بهبود قابلیت‌های نمایشی و افزایش کارایی صفحات وب ایجاد شده است.

اسکریپت، قطعه کدی است که درون کد دیگری نوشته می‌شود و بدون آن که مانند کدهای زبان‌های برنامه‌نویسی، کامپایل شود، خطبه‌خط به اجرا در می‌آید.

از جاوا اسکریپت برای دستکاری عناصر صفحه، انجام محاسبات ریاضی، واکنش نسبت به رویدادها و فرمان‌های کاربر و نیز اعتبارسنجی داده‌های فرم استفاده می‌گردد.

کدهای جاوا اسکریپت را می‌توان مستقیماً میان کدهای HTML درج کرد و یا درون یک فایل خارجی قرار داد.

در برنامه‌نویسی شیء‌گرا، از یک مفهوم (Class) که دارای تعدادی خصوصیت (Property) و رفتار (Method) است، یک نمونه (شیء یا Object) ساخته می‌شود و در فرایند برنامه‌نویسی مورد استفاده قرار می‌گیرد.



پرسش‌ها و تمرین‌ها

۱. تفاوت‌ها و شباهت‌های اسکریپت‌نویسی و برنامه‌نویسی را شرح دهید.
۲. با جستجو در اینترنت، پرکاربردترین ابزارهای اسکریپت‌نویسی را نام ببرید.
۳. با استفاده از جاوا اسکریپت چه نوع قابلیت‌هایی را می‌توان به یک وب‌سایت اضافه کرد؟
۴. کلاس «پرنده» را در نظر بگیرید. چه نوع خصوصیت‌ها و رفتارهایی را می‌توان برای این کلاس متصور شد؟ چند شیء قابل نمونه‌سازی از این کلاس را نام ببرید.



فصل هجدهم



آشنایی با

مدل اشیاء سند

(DOM)

هدف‌های رفتاری



پس از مطالعه این فصل از فراگیر انتظار می‌رود:

۱. با مفهوم مدل اشیاء سند آشنا شود.
۲. نحوه به‌کارگیری اشیاء را در یک سند فراگیری.
۳. بتواند از متدهای هر شیء استفاده نماید.

کلیات

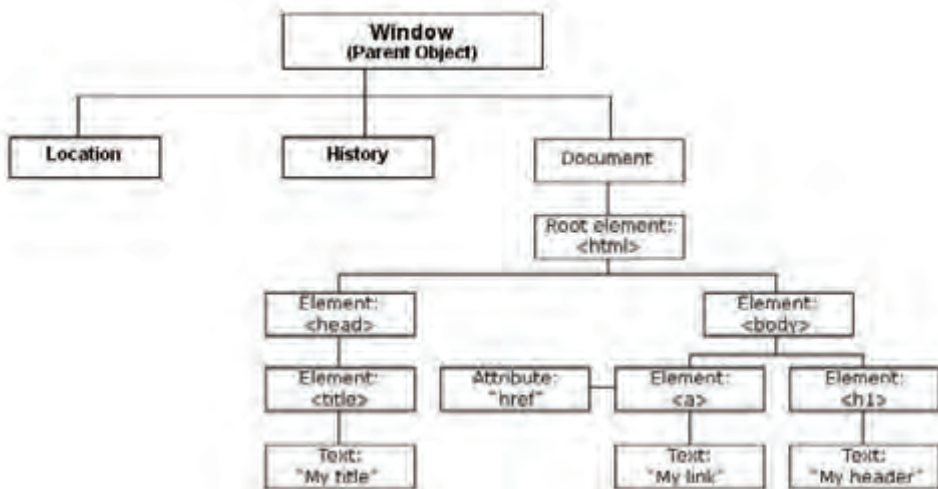
زبان جاوا اسکریپت اگر به اشیاء موجود در یک صفحه مانند پیوندها، تصاویر، نوشته‌ها و ... دسترسی نداشته باشد، نمی‌تواند کاری بیش از انجام محاسبات ریاضی یا کار با رشته‌های ساده را انجام دهد. برای ایجاد یک صفحه وب با قابلیت تعامل با کاربر یعنی صفحه‌ای که در آن به کارهای کاربر واکنش‌های مناسب نشان داده شود، اسکریپت باید بتواند به اجزاء تشکیل‌دهنده سند دسترسی داشته باشد.

کدهای جاوا اسکریپت از طریق خصوصیت‌ها (Properties)، متدها (Methods) و رویدادها (Events) با مرورگر و اشیاء درون سند تعامل دارند و این مجموعه رابطی را تشکیل می‌دهد که «مدل اشیاء سند» یا DOM^۱ نامیده می‌شود.

۱-۱۸ آشنایی با مفهوم DOM

DOM مدلی است که دسترسی به عناصر موجود در یک سند HTML را درون یک ساختار سلسله‌مراتبی فراهم می‌آورد و به این ترتیب اسکریپت می‌تواند محتوا، ساختار و سبک یک صفحه را به صورت پویا تغییر دهد. DOM مشخص می‌کند که کدامیک از خصوصیت‌های اشیاء موجود در صفحه می‌تواند توسط اسکریپت خوانده شود یا تغییر یابد. همچنین متدهایی را که می‌توانند برای ایجاد تغییر در صفحه فراخوانی شوند تعیین می‌نماید. به تصویر صفحه بعد دقت کنید.

1 . Document Object Model



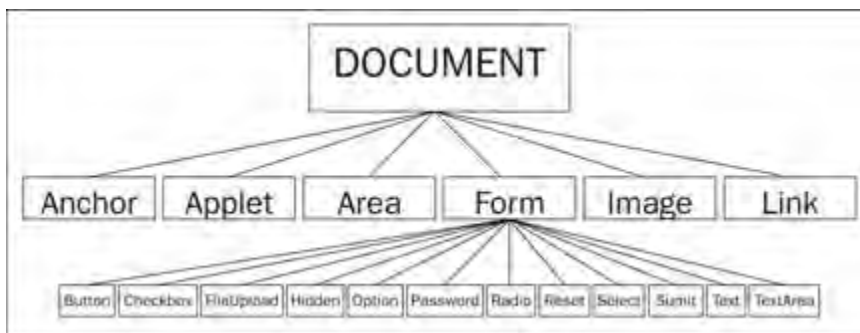
این تصویر، ساختار سلسله‌مراتبی DOM را نشان می‌دهد. این ساختار از window به عنوان شیء والد شروع می‌شود و هر شیء زیرمجموعه‌ای دارد تا به پایین‌ترین رده یعنی عناصر و خصوصیت‌های آن‌ها برسد. window به عنوان شیء والد دارای سه زیرمجموعه به صورت زیر است:

■ **document**: این شیء تمامی برچسب‌های موجود در صفحه وب یعنی همه نوشته‌ها، پیوندها، تصاویر و ... را در برمی‌گیرد.

■ **history**: شیء تاریخچه، شامل تمامی نشانی‌هایی است که توسط کاربر از زمان باز شدن مرورگر مشاهده شده است.

■ **location**: این شیء هم اطلاعات کلی پیرامون نشانی اینترنتی صفحه را در خود جای داده است.

حال اگر به ساختار سلسله‌مراتبی شیء document دقیق‌تر نگاه کنید خواهید دید که برای مثال دکمه‌ای که درون یک فرم قرار گرفته، درون مدل به عنوان زیرمجموعه‌ای از شیء document محسوب خواهد شد.



۲-۱۸ کار با اشیاء و متدها

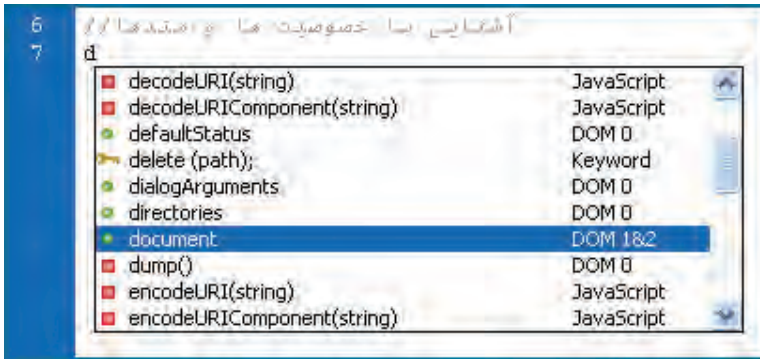
همان طور که پیش از این در بررسی مفهوم شیء گرایي اشاره شد، در زبان های شیء گرا مانند جاوا اسکریپت، هر شیء می تواند تعدادی خصوصیت و رفتار (متد) داشته باشد. با این توضیح می توان حدس زد که شیئی مانند document هم باید تعدادی خصوصیت و متد داشته باشد. برای مشاهده این خصوصیت ها و متدها به روش زیر عمل نمایید.

۱. در محیط Dreamweaver به سراغ فایل MyJavaScripts.js بروید.

۲. حرف d را تایپ و کلیدهای ترکیبی Ctrl+Space را فشار دهید.

۳. منویی ظاهر می شود که حاوی اشیاء، متدها، خصوصیت ها و رویدادهایی است که این حرف در نام آنها وجود دارد. این منو که برای کمک به کدنویسان برای یافتن عبارات مورد نظر طراحی شده و امروزه در اغلب محیط های کدنویسی دیده می شود Intellisense نام دارد.

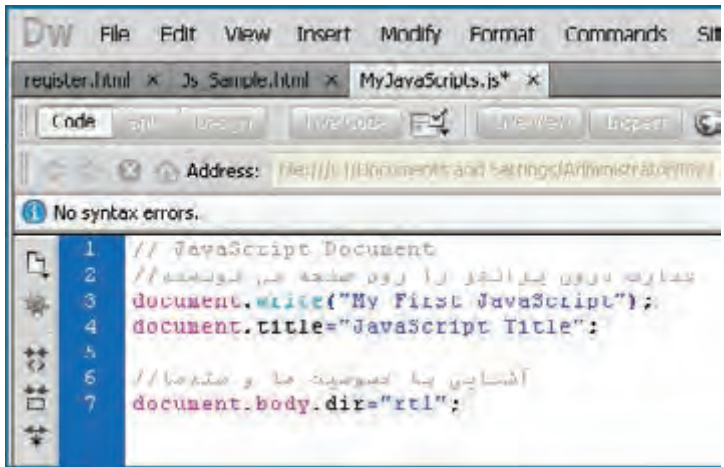
۴. با استفاده از دکمه های جهتی صفحه کلید، به سراغ گزینه document بروید و کلید Enter را فشار دهید تا درون صفحه درج شود.



۵. یک نقطه تایپ کنید. لیست اشیاء، متدها و خصوصیت های زیرمجموعه document ظاهر می گردد. گزینه body را انتخاب و کلید Enter را فشار دهید.

۶. مجدداً یک نقطه تایپ کنید تا خصوصیت ها و متدهای شیء body ظاهر شود.

۷. خصوصیت dir را انتخاب و آن را با "rtl" مقداردهی کنید.

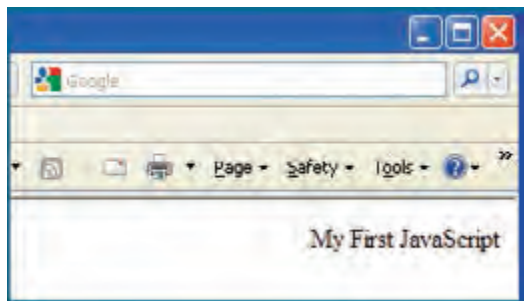


۸. علامت نقطه ویرگول (;) را در انتهای کد قرار داده و تغییرات را ذخیره کنید.

۹. صفحه‌ای را که به این فایل خارجی ارجاع دارد درون مرورگر باز نمایید.

۱۰. همان‌طور که می‌بینید عبارت درج شده توسط متد write() به سمت راست صفحه منتقل شده، چراکه

مشخصه dir از شیء body (که همه اشیاء درون صفحه را دربرمی‌گیرد) در اثر اجرای کد جاوا اسکریپت با "rtl" مقدار دهی شده است.



۱۱. کد زیر را به صفحه وب اضافه کنید تا سه پیوند در آن ایجاد شود.

```
<a id="google_link" href="http://www.google.com">Click to Visit Google</a><br>
<a id="yahoo_link" href="http://www.yahoo.com">Click to Visit Yahoo</a><br>
<a id="msn_link" href="http://www.MSN.com">Click to Visit MSN</a><br>
```

۱۲. کد زیر را به فایل جاوا اسکریپت اضافه نمایید.

```
document.linkColor="Green";
```

۱۳. پس از ذخیره‌سازی همه تغییرات، نتیجه را درون مرورگر ببینید.



در نتیجه مقداردهی مشخصه linkcolor از شیء document با مقدار "Green"، همه پیوندهای صفحه سبزنگ شده‌اند.

همان‌گونه که پیش از این اشاره شد، هر یک از اشیاء موجود در صفحه دارای خصوصیات و رفتار مخصوص به خود هستند. برخی از خصوصیات اشیاء را می‌توان در زمان اجرای اسکریپت تغییر داد که از آن‌ها به عنوان خصوصیت‌های «نوشتنی/خواندنی» یاد می‌کنیم.

برخی دیگر از خصوصیت‌ها هم «فقط خواندنی» هستند و امکان دستکاری آن‌ها وجود ندارد. در ادامه مطالب این بخش قصد داریم نگاه دقیق‌تری به اشیاء موجود در DOM و متدهای پرکاربرد هر یک داشته باشیم. ضمناً بررسی کنیم که کدام یک از خصوصیت‌های شیء، «فقط خواندنی» هستند و کدام یک را می‌توان تغییر داد.

۱۸-۳ شیء window

window حاوی متدهایی است که برای بازکردن یا بستن پنجره، تغییر ابعاد آن، ظاهر کردن پیغام و ... مورد استفاده قرار می‌گیرند. مهم‌ترین متدهای شیء window در جدول زیر قید شده‌اند.

کارکرد	متد
پنجره‌ای را ظاهر نموده و پیغام درون پرانتز را به کاربر نشان می‌دهد. این پنجره دارای یک دکمه OK است که با کلیک روی آن بسته می‌شود.	alert("msg")
پنجره جاری را می‌بندد.	close()
پنجره‌ای با دکمه OK و Cancel ظاهر می‌کند تا کاربر، عملیات جاری را تأیید یا لغو نماید.	confirm("msg")
پنجره را به مختصات قید شده درون پرانتز می‌برد. این مختصات نسبت به گوشه بالا و سمت چپ نمایشگر محاسبه می‌گردد.	moveTo(x,y)
پنجره را به میزان قید شده درون پرانتز جابه‌جا می‌کند.	moveBy(dx,dy)
پنجره جدیدی را باز می‌کند.	open()
محتویات صفحه جاری را برای چاپگر ارسال می‌کند.	print()
پنجره‌ای را برای ورود داده توسط کاربر ظاهر می‌کند. عبارت "msg" به عنوان پیغام نشان داده می‌شود و چنانچه پارامتر اختیاری Default Value را هم مقداردهی کنید، به عنوان مقدار پیش‌فرض در کادر متنی ورود داده قرار می‌گیرد.	prompt("msg", Default Value)



جاوا اسکریپت یک زبان حساس به بزرگی و کوچکی^۱ حروف است؛ بنابراین باید نام اشیاء، متدها و خصوصیت‌ها را دقیقاً با رعایت این نکته تایپ نمایید.



کدی بنویسید که پیغام «به وب‌سایت من خوش آمدید» را روی صفحه نمایش دهد.

```
window.alert("به وب سایت من خوش آمدید");
```



در همه مثال‌های این فصل، کد جاوا اسکریپت را درون فایل خارجی قرار داده‌ایم و یک ارجاع به فایل در بدنه صفحه به صورت زیر ایجاد کرده‌ایم:

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



کدی بنویسید تا به محض بارگذاری صفحه، صفحه اول گوگل را در پنجره جدیدی به ابعاد ۴۰۰×۴۰۰ باز نماید.

```
window.open('http://www.google.com',' ', 'width=400,height=400');
```

روش نگارش این متد به صورت `window.open(URL,name,specs,replace)` است. در این متد، می‌توانید به جای `specs` که مشخصات پنجره را تعیین می‌کند، ابعاد آن را قرار دهید. ضمناً هر چهار پارامتر این متد اختیاری هستند.

1 . Case Sensitive

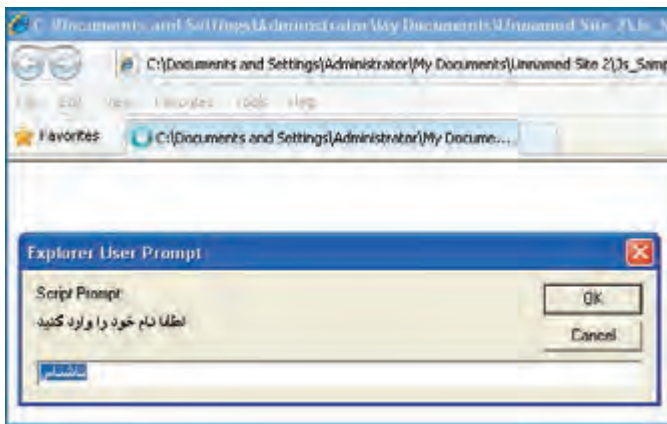


کدی بنویسید تا به محض باز شدن صفحه، نام کاربر از وی پرسیده شود. سپس نام وارد شده به همراه عبارت «خوش آمدید» روی صفحه درج گردد.

```
document.write("ناشناس", "لطفا نام خود را وارد کنید" + window.prompt(" " + "خوش آمدید");
```

بررسی کد:

- هنگام اجرای اسکریپت، مرورگر تلاش می‌کند عبارت موجود درون متد Write() را درون صفحه درج کند اما با متد prompt() مواجه می‌شود و ابتدا آن را اجرا می‌کند.
- در اثر اجرای این دستور، پنجره‌ای ظاهر شده و نام کاربر را درخواست می‌کند. چون پارامتر اختیاری دوم با عبارت «ناشناس» مقداردهی شده است، این عبارت درون کادر متنی قرار می‌گیرد.



- به محض وارد کردن نام و کلیک شدن دکمه OK، متد prompt مقدار وارد شده را به عنوان خروجی برمی‌گرداند.
- این خروجی با عبارت خوش آمدید و یک فاصله (" ") ترکیب شده و توسط متد write روی صفحه درج می‌شود.



عملگر + در جاوا اسکریپت برای اتصال رشته‌های متنی به یکدیگر کاربر دارد.

شیء window علاوه بر متدهای فوق، متدها و خصوصیت‌های دیگری هم دارد. برای نمونه با مقداردهی خصوصیت status می‌توانید عبارتی را درون نوار وضعیت^۱ مرورگر درج نمایید. البته توجه داشته باشید که مرورگرهای مختلف ممکن است در تفسیر کدهای جاوا اسکریپت به صورت یکسان عمل نکنند و لذا احتمال دارد کدی که در مرورگر IE به خوبی کار می‌کند در FireFox نتیجه مطلوبی نداشته باشد.



کدی بنویسید که پنجره را به میزان ۱۰۰ نقطه نسبت به بالای نمایشگر پایین بیاورد و زمان و تاریخ جاری را درون نوار وضعیت مرورگر درج کند.

```
window.moveBy(0,100);
window.status=Date();
```



بررسی کد: تابع Date() زمان و تاریخ جاری را برمی‌گرداند و خصوصیت status شیء پنجره را با آن مقداردهی می‌کند.



با هر بار تازه‌سازی (Refresh) صفحه چه اتفاقاتی می‌افتد؟

1 . Status Bar



اولاً زمان جدید درون نوار وضعیت درج خواهد شد و ثانیاً پنجره به میزان ۱۰۰ نقطه دیگر پایین می‌آید چون در واقع کد جاوا اسکریپت یکبار دیگر اجرا شده و پنجره را نسبت به وضعیت فعلی، تغییر مکان می‌دهد.

۴-۱۸ شیء document

هر یک از عناصر HTML که درون پنجره مرورگر بارگذاری می‌شوند تبدیل به اشیاء سند (Document Object) می‌گردند و می‌توان از طریق کدهای جاوا اسکریپت به آن‌ها دسترسی پیدا کرد. توجه داشته باشید که شیء document بخشی از شیء window محسوب می‌شود و می‌توان با استفاده از کد window.document به آن دسترسی پیدا کرد اما به منظور خلاصه‌نویسی صرفاً از نام شیء document استفاده می‌کنیم. مهم‌ترین خصوصیت‌های این شیء در جدول زیر بررسی شده‌اند.

نوع	کارکرد	خصوصیت
خواندنی/نوشتنی	محتوی عنوان صفحه است	title
خواندنی	محتوی نشانی کامل صفحه جاری است.	URL
خواندنی	نشانی صفحه‌ای را نشان می‌دهد که صفحه جاری را بارگذاری کرده است.	referrer

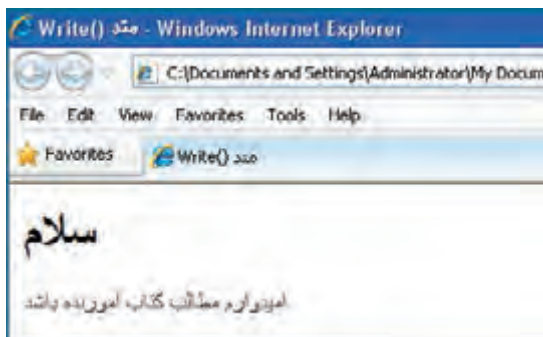
شیء document دارای متدی به نام write() است که به دلیل کاربرد زیاد، در بخش‌های قبل با کارکرد آن آشنا شدید. از این متد برای درج یک عبارت درون صفحه وب استفاده می‌شود. توجه داشته باشید که در عبارتی که درون پرانتز قید می‌شود، استفاده از کدهای HTML هم مجاز است.



کد جاوا اسکریپت زیر ضمن تغییر عنوان صفحه، یک سرفصل و یک پاراگراف ایجاد خواهد کرد.

```
document.title="متد Write()"
```

```
document.write("<p>امیدوارم مطالب کتاب آموزنده باشد</p><h1>سلام</h1>");
```



علاوه بر Write() متد دیگری به صورت writeIn() هم وجود دارد که علاوه بر نوشتن عبارت، یک برچسب
 به انتهای نوشته اضافه می‌کند تا عناصر بعدی در ابتدای سطر بعد قرار گیرند.

در هنگام تعریف DOM این نکته مهم را بیان کردیم که استفاده از «مدل اشیاء سند» امکان دسترسی به تک تک عناصر موجود روی صفحه و دست کاری آن‌ها را فراهم می‌آورد و بخشی از توانایی جاوا اسکریپت به همین نکته نهفته است. اکنون می‌خواهیم متدهایی از شیء document را بررسی کنیم که این دسترسی را امکان پذیر می‌سازند.

کارکرد	متد
دسترسی به نخستین عنصر با شناسه داده شده را فراهم می‌آورد.	getElementById()
دسترسی به همه عناصر موجود در صفحه با نام داده شده را امکان پذیر می‌سازد.	getElementsByName()
دسترسی به همه عناصر با برچسب مشخص شده را فراهم می‌آورد.	getElementsByTagName()



کد زیر را به صفحه وب اضافه کنید تا سه پاراگراف ایجاد شود.

```
<p id="p1">پاراگراف اول</p>
```

```
<p id="p2">پاراگراف دوم</p>
```

```
<p id="p3">پاراگراف سوم</p>
```

حال در فایل جاوا اسکریپت، کد زیر را بنویسید:

```
document.getElementById("p2").innerHTML="این متن با جاوا اسکریپت اضافه شده است";
document.getElementById("p1").style.backgroundColor="gray";
```


تغییرات را ذخیره نموده و نتیجه کار را در مرورگر ببینید.



بررسی کد:

■ هنگامی که متد `getElementById` فراخوانی می‌شود، این متد، شناسه عنصر موجود در صفحه را دریافت نموده و به سراغ آن می‌رود.

■ به این ترتیب شما به عنوان کدنویس می‌توانید خصوصیات «نوشتنی» آن عنصر را تغییر دهید و یا خصوصیات «فقط خواندنی» را استخراج نمایید.

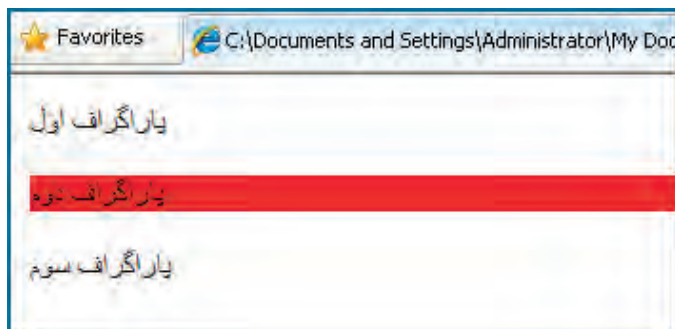
■ چنانچه چند عنصر با شناسه یکسان درون صفحه وجود داشته باشد، این متد به سراغ اولین عنصر یافت شده می‌رود و بقیه را نادیده می‌گیرد.



کد زیر را جایگزین عبارات موجود در فایل جاوا اسکریپت نموده و تغییرات را ذخیره نمایید.

```
document.getElementsByTagName("p").item(1).style.backgroundColor="red";
```

نمایی صفحه به صورت زیر تغییر می‌کند:



بررسی کد:

متد `getElementsByTagName()` نام یکی از برچسب‌های موجود در صفحه را دریافت نموده و مجموعه این برچسب‌ها را درون یک مجموعه^۱ قرار می‌دهد. در این مثال، برچسب `p` به عنوان سازنده پاراگراف برای متد فرستاده شده و در نتیجه سه پاراگراف در مجموعه نتایج قرار گرفته است.

حال برای دسترسی به عنصر موردنظر باید از متد `item()` استفاده کنیم.

این متد، نمایه یا `Index` عنصر موجود در مجموعه را دریافت نموده و آن عنصر را برمی‌گرداند.

توجه داشته باشید که در مجموعه‌ها و آرایه‌ها برای دسترسی به عنصر اول از نمایه صفر استفاده می‌شود؛ بنابراین در مثال فوق، `item(1)` بیان‌گر دومین عنصر موجود در مجموعه یعنی پاراگراف دوم است. به این ترتیب، عنصری از صفحه که قصد دسترسی به آن را داریم مشخص می‌شود و می‌توانیم خصوصیت‌های آن را بخوانیم یا تغییر دهیم. در این مثال، پس‌زمینه پاراگراف، قرمز شده است.

کدنویسان جاوا اسکریپت با استفاده از این متدهای پرکاربرد می‌توانند به تک‌تک عناصر موجود در صفحه دسترسی پیدا کنند، اطلاعات موردنظر را از آن‌ها استخراج نمایند و یا تغییرات دلخواه را روی ساختار صفحه وب و اطلاعات در حال نمایش روی آن ایجاد کنند.

۵-۱۸ شیء `history`

شیء `history` یا تاریخچه حاوی نشانی‌هایی است که توسط کاربر درون مرورگر وب مشاهده شده است. این شیء دارای یک خصوصیت پرکاربرد به نام `length` است که تعداد نشانی‌های موجود در لیست تاریخچه را در خود جای داده است. متدهای پرکاربرد این شیء هم که در اکثر مرورگرها پشتیبانی می‌شوند، در جدول زیر نشان داده شده‌اند.

کارکرد	متد
نشانی قبلی موجود در تاریخچه مرورگر را بارگذاری می‌کند و معادل کلیک کاربر روی دکمه <code>Back</code> مرورگر است.	<code>back()</code>
نشانی بعدی موجود در تاریخچه مرورگر را بارگذاری می‌کند و معادل کلیک کاربر روی دکمه <code>Forward</code> مرورگر است.	<code>forward()</code>
نشانی خاصی را از لیست تاریخچه مرورگر بارگذاری می‌کند.	<code>go()</code>



صفحه‌ای به نام test.html ایجاد و در آن پیوندی به صفحه Js_Sample.html قرار دهید. ضمناً در آن کدی برای نمایش تعداد عناصر موجود در لیست تاریخچه تعبیه نمایید. سپس صفحه Js_Sample.html را به گونه‌ای طراحی کنید که در آن تعداد عناصر موجود در لیست تاریخچه نمایش داده شود و ضمناً پیوندی برای «برگشت به صفحه قبل» در آن قرار گیرد.

کد صفحه test.html

```
<body>
<a href="Js_Sample.html">برای مشاهده صفحه حاوی کد جاوااسکریپت کلیک کنید</a>
<br/>
<script type="text/javascript">
document.writeln("تعداد نشانی های لیست تاریخچه " + history.length);
</script>
</body>
```

کد فایل MyJavaScript.js

```
document.writeln("تعداد نشانی های لیست تاریخچه " + history.length);
```

کد صفحه Js_Sample.html

```
<body>
<p id="p1">پاراگراف اول</p>
<p id="p2">پاراگراف دوم</p>
<p id="p3">پاراگراف سوم</p>
<script type="text/javascript" src="Scripts/MyJavaScripts.js">
</script>
<br/>
<a href="javascript:history.back();">بازگشت به صفحه قبل</a>
</body>
```

حال تغییرات را ذخیره نموده و صفحه test.html را باز کنید. همان‌طور که مشاهده می‌کنید، دکمه‌های Forward و Back مرورگر غیرفعال هستند و تعداد نشانی‌های تاریخچه صفر است.



روی پیوند کلیک کنید تا به صفحه بعد بروید. در این صفحه، تعداد نشانی‌ها به عدد ۱ تغییر می‌یابد چون صفحه قبل (test.html) در لیست تاریخچه ثبت شده است. ضمناً با کلیک روی پیوند «بازگشت به صفحه قبل»، به صفحه test.html منتقل می‌شوید و مثل این است که روی دکمه Back مرورگر کلیک کرده‌اید.



بررسی کد:

در این مثال، از سه روش گنجاندن کدهای جاوا اسکریپت درون صفحه وب استفاده کردیم که در روش اول را در بخش‌های قبل بررسی کرده بودیم: روش اول؛ گنجاندن اسکریپت‌ها در میان کدهای HTML و روش دوم؛ ارجاع به یک فایل جاوا اسکریپت خارجی در صفحه. حال با روش سوم که کدنویسی خطی یا Inline نامیده می‌شود آشنا خواهید شد. به کد زیر دقت کنید:

```
<a href="javascript:history.back();" >بازگشت به صفحه قبل</a>
```

در این کد، به جای درج نشانی یک صفحه در مشخصه href، با قرار دادن عبارت Javascript به مرورگر اعلام کرده‌ایم که قصد اجرای یک متد جاوا اسکریپت را داریم و این متد پس از علامت: درج شده است. کل عبارت هم درون علامت " قرار گرفته است. به این ترتیب، کاربر با کلیک روی پیوند، به جای رجوع به یک

صفحه، یک قطعه کد جاوا اسکریپت را اجرا خواهد کرد که در این مثال، برگشت به صفحه قبل است.



به روشی که در مثال قبل آموختید، پیوندی ایجاد کنید تا وقتی کاربر روی آن کلیک می‌کند، وبسایت گوگل را در پنجره‌ای به ابعاد ۴۰۰ در ۴۰۰ نقطه ببیند.

```
<body>
<a href="javascript:window.open('http://www.google.com','_blank',
'width=400,height=400')">برای جستجو در گوگل کلیک کنید</a>
</body>
```

در فصول بعدی کتاب با روش اصولی و استاندارد اجرای کدهای جاوا اسکریپت آشنا خواهید شد. در این روش، عملیات موردنظر درون یک تابع^۱ قرار داده می‌شود و اجرای مشروط به وقوع یک رویداد مانند بارگذاری صفحه، کلیک شدن یک دکمه و ... خواهد شد.

۱۸-۶ شیء location

این شیء حاوی اطلاعاتی در مورد نشانی صفحه جاری و نیز متدهایی برای کار با نشانی‌ها است. مهم‌ترین خصوصیت‌های این شیء عبارتند از:

نوع	کارکرد	خصوصیت
خواندنی	نشانی کامل صفحه جاری را برمی‌گرداند.	href
خواندنی	نام دامنه وبسایت جاری را برمی‌گرداند.	host

همچنین می‌توانید از متدهای زیر برای کار با نشانی‌ها بهره ببرید.

کارکرد	متد
صفحه‌ای که نشانی آن درون پرانتز درج شده را فراخوانی می‌کند.	assign("URL")
همانند دکمه Refresh مرورگر، صفحه جاری را مجدداً بارگذاری می‌نماید.	reload()
صفحه‌ای که نشانی آن درون پرانتز درج شده را جایگزین صفحه جاری می‌کند.	replace("URL")



کدی بنویسید که هنگام باز شدن صفحه، نشانی کامل آن را بنویسید. ضمناً حاوی پیوندهای زیر باشد:

الف) پیوند به وبسایت آموزش و پرورش

ب) پیوند برای تازه‌سازی (Refresh) صفحه

```
<script type="text/javascript">
document.writeln(location.href);
</script>
<br/>
```

```
<a href="javascript:location.assign('http://www.medu.ir')">وزارت آموزش و پرورش</a><br/>
```

```
<a href="javascript:location.reload();" >تازه سازی</a><br/>
```

```
</body>
```

۶-۱۸ شیء navigator

با استفاده از این شیء می‌توانید اطلاعاتی نظیر نوع و نسخه مرورگر کاربر را استخراج نمایید. پرکاربردترین خصوصیت‌های این شیء عبارتند از: `appName` و `appVersion` که به ترتیب، نام مرورگر و نسخه آن را برمی‌گردانند.

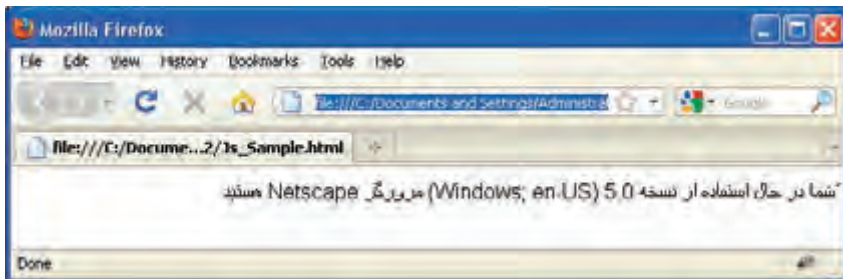
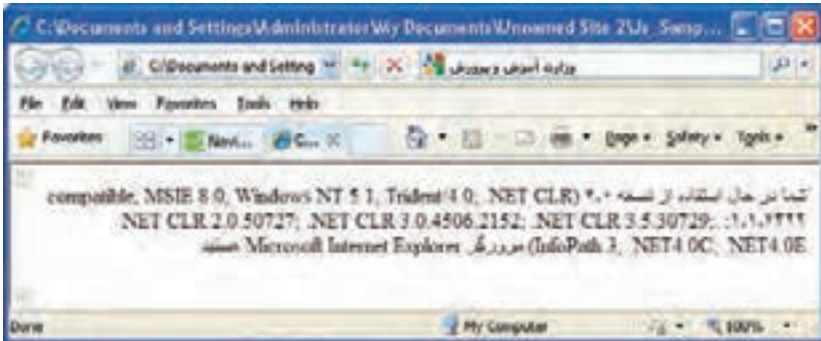


کدی بنویسید که با باز شدن صفحه، نوع و نسخه مرورگری که کاربر در حال استفاده از آن است را برگرداند.

```
<html ><head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body dir="rtl">
<script type="text/javascript">

document.writeln(" شما در حال استفاده از نسخه " + navigator.appVersion + " مرورگر " + navigator.appName + " هستید ");
</script>
</body></html>
```

نتیجه کار را در دو مرورگر IE و Firefox مشاهده می کنید.





چکیده فصل

- DOM مدلی است که دسترسی به عناصر موجود در صفحه را فراهم می‌آورد.
- هر یک از اشیاء در این مدل دارای تعدادی خصوصیت و تعدادی متد هستند.
- با استفاده از متدها و خصوصیت‌ها می‌توان ساختار و نحوه نمایش صفحه وب را در مرورگر کاربر تغییر داد.
- شیء window دارای متدهایی برای ایجاد، بستن و دستکاری پنجره‌ها است.
- با استفاده از متدهای شیء document می‌توان به تک‌تک عناصر صفحه دسترسی پیدا کرد.
- شیء history امکان دسترسی به تاریخچه مرورگر را فراهم می‌آورد.
- کار با نشانی‌ها از طریق شیء location امکان‌پذیر است.
- اطلاعات مربوط به مرورگر کاربر از طریق شیء navigator قابل دستیابی است.



پرسش‌ها و تمرین‌ها

۱. DOM چیست؟ موارد کاربرد آن را توضیح دهید.
۲. متدهای پرکاربرد شیء window را نام ببرید.
۳. در یک صفحه وب، پیوندی ایجاد کنید که با کلیک روی آن تصویری در ابعاد 400×300 نقطه درون یک پنجره جدید ظاهر شود. ضمناً این پنجره باید پیوندی برای بستن داشته باشد.
۴. کدی بنویسید تا به محض باز شدن صفحه، نام کاربر پرسیده شود و سپس در یک پنجره پیغام (Alert) به نمایش دربیاید.
۵. با جستجو در وب، سایر تنظیمات قابل اعمال در پارامتر specs از متد window.open را بیابید.
۶. آیا همه خصوصیت‌ها و متدهای یک شیء در همه مرورگرها به یک صورت عمل می‌کنند؟



فصل نوزدهم



کار با

متغیرها، داده‌ها و توابع

هدف‌های رفتاری



پس از مطالعه این فصل از فراگیر انتظار می‌رود:

۱. با روش تعریف و نام‌گذاری متغیرها آشنا شود.
۲. انواع داده‌ای را در جاوا اسکریپت بشناسد.
۳. با عمل‌گرهای موجود در این زبان آشنا شود.
۴. روش تعریف تابع و استفاده از آن را فراگیرد.

کلیات

تا این بخش از کتاب با روش درج کدهای جاوا اسکریپت درون برچسب‌های HTML و نیز روش استفاده از اشیاء DOM، متدها و خصوصیت‌های آن‌ها آشنا شدید که می‌تواند بخشی از نیازهای شما را برای نوشتن کدهای موردنظر برطرف سازد. اما یادگیری هر زبان برنامه‌نویسی یا اسکریپت‌نویسی در استفاده از متدهای از پیش تعریف شده محدود نمی‌شود؛ بلکه باید با روش‌های موجود برای تعریف توابع موردنظر، فراخوانی آن‌ها و انجام محاسبات موردنیاز آشنا شوید. در این فصل ابتدا مفاهیم مرتبط با متغیرها و داده‌ها را مرور خواهیم کرد تا مهارت لازم برای تعریف و استفاده از توابع در انتهای فصل به دست آید.

۱-۱۹ مفهوم داده و متغیر

متغیر^۱ مخزنی است که از آن برای نگهداری اطلاعات استفاده می‌شود. متغیر را می‌توان به صورت بخش کوچکی از حافظه رایانه تصور کرد که در آن می‌توانید عدد، رشته و مواردی از این دست را ذخیره نمایید. آن چه در متغیر ذخیره می‌شود، داده^۲ نام دارد.

۲-۱۹ تعریف متغیرها

هر متغیر یک نام دارد که در طول اجرای برنامه با همین نام شناخته می‌شود. در جاوا اسکریپت برای تعریف متغیرها از کلمه کلیدی var استفاده می‌شود و پس از آن نام متغیر درج می‌شود. به کد زیر دقت کنید:

```
var Name;
```

در این قطعه کد، ابتدا متغیری با نام Name تعریف شده است. توجه داشته باشید که جاوا اسکریپت یک زبان حساس به بزرگی و کوچکی حروف است، لذا متغیر name کاملاً از Name متمایز خواهد بود. بنابراین می‌توانیم در ادامه کد، متغیر دیگری به این صورت تعریف نماییم:

```
var name;
```

1 . Variable

2 . Data

برای تعریف متغیرها باید موارد زیر را رعایت کنید:

■ نام متغیر صرفاً می‌تواند با یک حرف یا نویسه زیرخط^۱ (_) شروع شود.

■ نام متغیرها با بزرگی و کوچکی حروف حساس است.

■ بهتر است نام متغیر با کارکرد آن متناسب باشد تا کد خواناتر شود.

■ استفاده از نام‌های مشابه برای متغیرها ممکن است به از دست رفتن مقادیر در طول اجرای کد و بروز خطا و اشتباه منجر شود.

■ برای تعریف متغیر نمی‌توان از کلمات کلیدی^۲ یا رزرو شده^۳ استفاده کرد. این کلمات، واژه‌هایی هستند که توسط خود برنامه مورد استفاده قرار می‌گیرند.

برخلاف بسیاری از زبان‌های برنامه‌نویسی که باید در هنگام تعریف متغیر، نوع آن هم مشخص شود؛ یعنی تعیین گردد که قرار است در این متغیر عدد، رشته، تاریخ، مقدار منطقی و ... ذخیره شود، خوشبختانه در جاوا اسکریپت نیازی به این کار نیست.

۳-۱۹ کلمات کلیدی و رزرو شده

جاوا اسکریپت همانند هر زبان دیگری دارای مجموعه‌ای از «کلمات کلیدی» و «لغات رزرو شده» است که نمی‌توانند به‌عنوان نام متغیر تا تابع مورد استفاده قرار گیرند. چنانچه در مجموعه کدهای خود از کلمات کلیدی به عنوان متغیر و یا نام تابع استفاده نمایید، با پیغام خطا مواجه خواهید شد. کلمات کلیدی جاوا اسکریپت را در جدول زیر مشاهده می‌کنید.

break	else	new	var
case	finally	return	void
catch	for	switch	while
continue	function	this	with
default	if	throw	
delete	in	try	
do	instanceof	typeof	

1 . Under Score

2 . Key Words

3 . Reserved Words

کلمات رزرو شده هم از جهت ممنوعیت استفاده به عنوان نام تابع یا متغیر همانند کلمات کلیدی هستند. کلمات رزرو شده جاوا اسکریپت در جدول زیر فهرست شده‌اند.

abstract	enum	int	short
boolean	export	interface	static
byte	extends	long	super
char	final	native	synchronized
class	float	package	throws
const	goto	private	transient
debugger	implements	protected	volatile
double	import	public	

۱۹-۴ انتساب مقادیر به متغیرها

تعریف متغیر به خودی خود هیچ مزیتی ندارد، مگر آن که بتوان مقداری را درون متغیر ذخیره کرد و در طول اجرای برنامه از آن استفاده نمود. برای مقداردهی به متغیرها از عملگر انتساب (با علامت تساوی)، استفاده می‌شود. این مقدار در طول برنامه قابل خواندن یا تغییر است. به کد زیر دقت کنید:

```
var userName;
userName = "Ali";
userName = "Amir";
```

در این قطعه کد، ابتدا با استفاده از کلمه کلیدی `var`، متغیری با نام `userName` تعریف شده و در خط دوم مقدار `Ali` به آن نسبت داده شده است. تا این جا، مقدار متغیر `userName` برابر است با `Ali` ولی وقتی خط سوم اجرا می‌شود، مقدار `Amir` جایگزین مقدار قبلی می‌شود و اکنون مقدار متغیر `userName` عبارت است از `Amir`.

حال می‌خواهیم متغیری با نام `_score` ایجاد و یک عدد در آن ذخیره کنیم. کد زیر را در نظر بگیرید:

```
var _score;
_score=20;
```

توجه داشته باشید که می‌توان عملیات تعریف و مقداردهی متغیر را در یک سطر انجام داد و نوشت:

```
var _score=20;
```

چرا در هنگام ذخیره‌سازی مقدار `Ali` از علامت نقل قول استفاده شد اما برای ذخیره‌سازی عدد چنین کاری صورت نگرفت؟ پاسخ روشن است. نوع مقادیری که قرار است در متغیرها ذخیره شود، در روش انتساب مؤثر است.

برای آشنایی با مقادیری که می‌توان به متغیرها نسبت داد و روش صحیح انجام این کار، جدول زیر را مشاهده کنید.

روش انتساب	مقادیر قابل ذخیره‌سازی	نوع متغیر	
var b=true;	مقادیر منطقی شامل true یا false	Boolean	منطقی
var n=200;	اعداد صحیح مانند ۰، ۱۵۰، -۱۵۰ و ...	Integer	عددی
var p=3.14;	اعداد اعشاری مانند ۱۲،۲، ۱۸،۲۳ و ...	Float	
var char='c';	یک نویسه مانند a، ^، B و ...	Character	رشته‌ای
var s="My Name";	یک رشته مانند abc، b12b و ...	String	

حال اگر دو متغیر را به صورت زیر تعریف و مقداردهی کنیم:

```
var n=20;
var s="20";
```

در متغیر n عدد 20 ذخیره می‌شود اما در متغیر s رشته 20 ذخیره خواهد شد. بدیهی است از متغیر n می‌توان برای انجام محاسبات ریاضی استفاده کرد اما متغیر s صرفاً کاربرد رشته‌ای دارد و صرفاً حاوی نویسه‌های 2 و 0 است.

عملیات تعریف و مقداردهی متغیرها را می‌توانید با استفاده از یک کلمه‌کلیدی var هم انجام دهید. برای نمونه در کد زیر، سه متغیر از نوع منطقی، عددی و رشته‌ای تعریف گردیده و مقداردهی شده است:

```
var a=true,b=10,c="text";
```

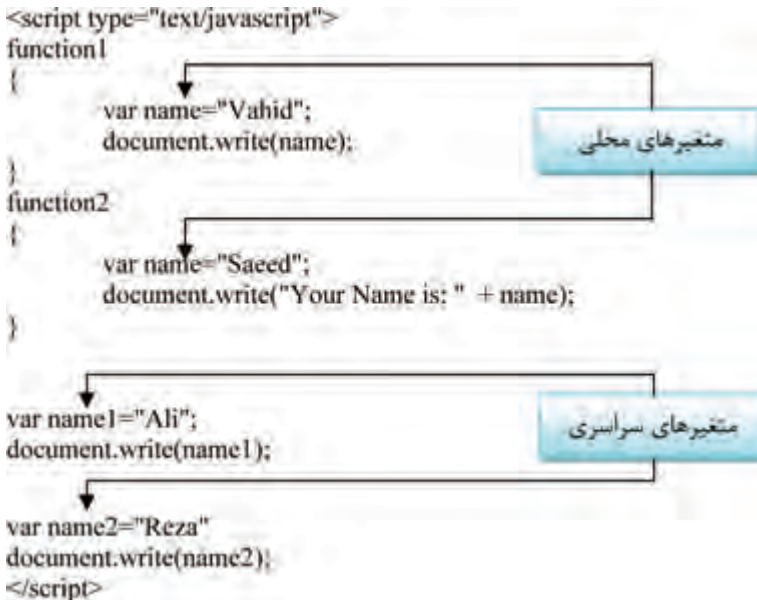
از آن‌جا که ممکن است در یک برنامه‌ها صدها متغیر وجود داشته باشد و کدنویس برای ردگیری نوع متغیرهای تعریف شده و کاربرد آن‌ها دچار سردرگمی گردد توصیه می‌شود در ابتدای نام متغیرها از حرفی که نشان دهنده نوع متغیر باشد استفاده شود. ضمناً نامی گویا که مشخص کننده کاربرد متغیر باشد انتخاب گردد. در جدول زیر مثال‌هایی از این روش تعریف را مشاهده می‌کنید.

مثال	پیشوند	نوع متغیر
var bChecked=false;	b	Boolean
var iCount=20;	i	Integer
var fScore= 15.25;	f	Float
var sText="My text";	s	String

۱۹-۵ متغیرهای محلی و سراسری

در همه زبان‌های برنامه‌نویسی هنگامی که نامی از متغیرها به میان می‌آید، در مورد طول عمر آن‌ها هم بحث می‌گردد؛ منظور از طول عمر یک متغیر، زمانی است که از تعریف متغیر تا از بین رفتن آن سپری می‌شود. فرض کنید متغیری را در ابتدای کد جاوا اسکریپت تعریف نموده‌اید، طول عمر این متغیر، از زمان تعریف تا زمان بسته شدن مرورگر خواهد بود و لذا در ادامه کد نمی‌توانید متغیر دیگری را به همین نام تعریف نمایید چون باعث تداخل خواهد شد. به چنین متغیرهایی، سراسری گفته می‌شود چون در سرتاسر برنامه وجود دارند و قابل استفاده هستند.

حال فرض کنید متغیری را درون یک تابع تعریف کنیم. طبیعتاً طول عمر این متغیر از زمان شروع فراخوانی تابع تا اتمام اجرای آن خواهد بود و لذا تعریف متغیرهای هم‌نام درون تابع‌های مجزا ایرادی ندارد. این نوع متغیرها، محلی نامیده می‌شود چون در یک محدوده خاص قابل دسترس می‌باشند.

۱۹-۶ انواع داده‌ای^۱ در جاوا اسکریپت

هر زبان برنامه‌نویسی یا اسکریپت‌نویسی برای اجرای عملیات موردنظر با مجموعه‌ای از مقادیر کار می‌کند. این مقادیر که دارای نوع و محدوده خاصی هستند باید در طول برنامه به‌گونه‌ای ذخیره شوند تا بتوان عملیات

موردنیاز برای محاسبه، مقداردهی یا خواندن مقادیر را بدون بروز اشکال انجام داد. لذا در هر زبان مجموعه‌ای از انواع داده‌ای تعریف می‌شود که کدنویس باید با ویژگی‌های هر یک از آنها آشنا باشد تا برای ذخیره‌سازی و بازیابی مقادیر با مشکل مواجه نشود.

انواع داده‌ای یا `DataType`های موجود در جاوا اسکریپت را می‌توان به دو دسته تقسیم‌بندی کرد:

■ انواع داده‌ای اولیه (Primitive)

■ انواع داده‌ای ارجاعی (Reference)

۱-۶-۱۹ انواع داده‌ای اولیه

همان‌گونه که پیش از این در بخش «انتساب مقادیر به متغیرها» دانستید، داده‌ها را می‌توان در سه نوع متغیر منطقی، عددی و رشته‌ای ذخیره نمود. این‌ها همان انواع داده‌ای اولیه هستند که در این بخش جزئیات مربوط به هریک را مرور خواهیم کرد.

◀ نوع داده‌ای منطقی یا **boolean**: در این نوع داده‌ای می‌توان مقادیر «درست» یا «نادرست» را ذخیره نموده و در طول برنامه عمدتاً برای بررسی حالت‌های شرطی مورد استفاده قرار می‌گیرد. مقدار درست با `true` و مقدار نادرست با `false` تعیین می‌گردد مانند:

```
var bChecked=true;
```

◀ نوع داده‌ای عددی یا **number**: این نوع داده‌ای می‌تواند حاوی اعداد صحیح یا اعشاری باشد مانند:

```
var iNum=-12;
var iCount=23;
var fNum=13.542;
var iBase8 = 070;
```

◀ نوع داده‌ای رشته‌ای یا **string**: برای ذخیره‌سازی رشته‌ها از این نوع داده‌ای استفاده می‌شود. برای مثال وقتی عبارت زیر را درون برنامه می‌نویسید:

```
var sText="Hello!";
```

درون حافظه ساختاری به صورت زیر تشکیل می‌شود و هریک از نویسه‌ها درون یکی از خانه‌ها قرار می‌گیرند. برای مثال حرف `o` در خانه‌ای با نمایه `۴` قرار خواهد گرفت.

H	e	l	l	o	!
0	1	2	3	4	5