فهرست مطالب:

فصل اول :

VBScript چیست؟

زبان VBScript زبانی Script می باشد که مستقیماً از Visual Basic نشأت گرفته است . زبانهای Script روایت کوچک شده زبانهای دیگر میباشند که برای اضافه شدن به صفحات وب طراحی شده اند . کد برنامه به یکی از زبانهای Script مستقیماً درون متن HTML درج شده و در هنگام اجرا به جای کامپایل یا تفسیر شدن ترجمه میشود .

برای آنکه متن به زبان Script را به یک صفحه HTML اضافه کنیم لازمست متن مورد نظر را میان تگهای **<SCRIPT>** و **</SCRIPT>** قرار دهیم . در حال حاضر مجموعاً دو نوع زبان Script مورد استفاده قرار می گیرند: VBScript (نشأت گرفته شده از Visual Basic) و Java Script (نشأت گرفته شده از Java) در این کتاب به مورد اول پرداخته و دومی را به عهده خواننده میگذاریم تا در کتابهای مربوطه مطالب مورد علاقه را پیدا نماید.

برای درج کد به زبان VBScript درون صفحه HTML باید به تگ <Script> پارامتر:

```
Language= "VBScript"
```

را اضافه نماییم که مشخص کننده زبان Script مورد استفاده می باشد.

گفتیم که زبانهای Script ترجمه می شوند و نه تفسیر این بدان معناست که هیچ گونه تغییر و تفسیرقبلی بر روی متن زبان Script صورت نمی پذیرد . هر خط کد درست قبل از اجرا شدن به زبان ماشین ترجمه می شود. و پس از اجرا روی هیچ سیستم ذخیره سازی (نظیر سخت دیسک و فلاپی و ...) ذخیره نگشته و اگر اجرای مجددی لازم باشد آن خط باید مجدداُ ترجمه شده و اجرا گردد. با این حال Visual Basic (که VBScript زاده آن میباشد.) یک زبان مفسری است یعنی متن Visaul Basic یکبار قبل از اجرا و به طور یکپارچه به زبان ماشین ترجمه میشود و نتیجه آن روی دیسک و به صورت یک فایل با پسوند EXE. ذخیره میگردد و زمانی که میخواهیم آن را اجرا کنیم کد تفسیر شده را اجرا میکنیم و نه کد اصلی را (که به آن کد مبدأ یا Source Code میگوییم.) . هر کدام از این دو سیستم معایب و مزایای خاص خود را دارا میباشند. که در ذیل بعضی از آنها را فهرست نموده ایم.

مترجم:

مزایا
• کد نوشته شده به راحتی اشکالزدای میشود. زیرا پس از تغییر احتیاجی به تفسیر مجدد نیست. • احتیاجی به مفسر جدا ندارد چرا که مترجم (که به همراه مرورگرمیباشد)خود وظیفه ترجمه را بر عهده میگیرد. • نگهداری و پشتیبانی آن راحتتر و سریعتر میباشد. چه از طرف نویسنده اصلی و چه از طرف برنامه نویسان دیگر

معایب
• اجرای آن کند میباشد چرا که هر دفعه کار ترجمه باید . بر روی خط به خط کد برنامه صورت پذیرد. • کد برنامه قابل رویت ودر دسترس میباشد که این امر باعث سادگی سو استفاده از سوی دیگران میشود. • استفاده کننده به کد دسترسی داشته و با تغییر ان میتواند اجرا را مختل و یا تغییر دهد.

مفسر:

مزایا
• کد تفسیر شده بسیار سریع اجرا میشود چرا که احتیاجی به ترجمه مجدد ان نیست. • کد تفسیر شده قابل باز شدن توسط دیگران نیست و احتیاجی به همراهی کد اصلی برنامه نیست. • کد اجرایی میتواند بسیار خلاصه و حتی به صورت یک فایل اجرایی یکتا باشد.

معایب
• به یک مفسر – موصل (Compiler-Linker) جهت انجام عمل تفسیر احتیاج میباشد. • کد اجرایی معمولاً حجم زیادی از دیسک را اشغال میکند زیرا که معمولاً مقداری از کدهای کتابخانه أی سیستم هم همراه آن وجود دارد. • ایجاد تغییرات مستلزم تفسیر مجدد میباشد.

کد VBScript میتواند برای اجرا بر روی رایانه مشتری (Client) و یا سرویس دهنده (Server) طراحی شود. تفاوت کار در این است که کدی که در سرور اجرا میشود توسط Client قابل رویت نیست و Client تنها پاسخ درخواستهایش را میبیند و نه کد اصلی اجرایی را. کدی که باید در سرور اجرا شود در صفحه وب مربوطه میان تگهای <%y%> قرار می گیرد مضاف بر اینکه نام صفحه مربوطه نیز تغییر یافته و به آن پسوند **asp.** (مخفف Active server page) اضافه گردد. البته صحبت در زمینه Asp از حوصله این کتاب خارج میباشد. با این حال در پایان مباحث اشاره ای به آن خواهیم داشت. در این کتاب هدف ما آموزش استفاده از VBScript برای اجرای بر روی رایانه مشتری می باشد.

متاسفانه VBScript تنها بر روی سیستمهایی قابل اجراست که از مرورگر **Internet explorer 4** یا بالاتر بهره گیری نمایند . که این باعث احتیاج به دستیابی به این مرورگر و الزاماً استفاده از سیستم عامل **Windows** میشود و کاربران سایر مرورگرها (نظیر) و سیستم عاملها (نظیر Linux,OS/2) امکان استفاده از آن را ندارند. (البته کاربران سیستمهای Apple Macintosch هم امکان استفاده از آن را دارند.)لذا اگر می خواهید سایت شما به صورت عمومی به نمایش در بیاید شاید أین موضوع کمی نگران کننده باشد ولی باید توجه داشت که بدون احتساب سرورها و سیستمهای بزرگ قریب به ۹۰ درصد کاربران اینترنت از ویندوز و تقریباً اغلب آنها از مرورگر پیش فرض آن یعنی Internet explorer استفاده مینمایند و بسیار بعید است که کسی از ویندوز و مرورگر آن و اینترنت استفاده نماید ولی کماکان بخواهد از روایتهای قدیم آن استفاده کند. که در این صورت هم به مصداق مثل "هرکه طاووس خواهد جور هندوستان کشد." ندیدن صفحه شما مشکل خود او بوده و باید وی دست از لجبازی برداشته همگام با تکنولوژی حرکت کند. سایر کاربران غیر Windows هم اغلب کاربران حرفه أی و دانشگاهی میباشند که خودشان در صورت احساس نیاز به سیستم عامل Windows و مرورگر آن دسترسی دارند.

فصل دوم :

مقدمات شی گرایی

تمام زبانهای برنامه نویسی سطح بالایی که در حال حاضر مورد استفاده قرار می گیرند سعی بر آن دارند تا زبانهایی شی گرا باشند. برنامه نویسی شی گرا (**OOP**:Object oriented programming) فلسفه ایست که بر اساس آن سعی بر آن داریم که به هر المان بکاررفته در برنامه به عنوا ن یک شی مجزا نگریسته شود. به عنوان مثال یک برنامه در حال اجرا یک شی تلقی میشود و در عین حال هر سطر نوشته هر گونه عکس، فیلم و ... بکار رفته در آن برنامه برای خود شیی مجزاست. همچنین محدوده کاری، پنجره های که برنامه در آن در حال اجراست یا متنی که درون آن پنجره در حال نمایش است نیز هر کدام اشیای مجزایی هستند.

اشیا ساختار خاصی دارند و به شیوه ای تعریف شده مرتب می گردند که به آن میپردازیم:

-خصوصیات(Properties)

هر شیء خصوصیات خاصی برای خود دارا میباشدکه توسط آن خصوصیات تعریف میشود به عنوان مثال متن درون صفحه شیی است که از خصوصیات آن میتوان به رنگ(color) اندازه (Size) و خطی (Font) که با آن نوشته شده اشاره کرد. هر شیء خصوصیاتی لاینفک داشته که مجموعه این خصوصیات همراه با متدهای مربوط کلاس

شیء را تشکیل میدهد. اشیاء مختلف طبق قاعده وارث و موروثی طبقه بندی می شوند که بر مبنای آن اشیایی با مرتبه بالاتر و اشیایی با مرتبه پایینتر بوجود می آید بدان معنا که ممکن است بعضی از اشیاء عضوی ازیک شیء دیگر باشند.(این شیء هم میتواند خود جزو خصوصیات شیء دیگری باشد.) به عنوان مثال یک سطر نوشته جزو خصوصیات متنی است که به آن تعلق دارد در عین این که خود نیز یک شیء می باشد. شیء والد شیئی است که جزو خصوصیات شیئی باشد که از آن متولد شده. هر شیء میتواند فرزندان بیشماری داشته ولی تنها باید یک پدر داشته باشد. هر شیء به طور پیش فرض تمام خصوصیات پدرش رابه ارث میبرد و در اصل از همان کلاسی پیروی میکند که از آن متولد شده ولی تمام یابخشی از این خصوصیات میتواند در هر لحظه توسط برنامه نویس و یا به تبع اعمال استفاده کننده از نرم افزار تغییر یابد.

- متدها (Methods)

علاوه بر خصوصیات اشیاء توابعی به همراه دارند که با اجرای آنها نتیجه مورد نظر بدست می آید.این توابع به **متد** معروف میباشند و علت نامگذاری آنها به این صورت اجتناب ازبه اشتباه گرفته شدن آنها با سایر توابع برنامه میباشد. به عنوان مثال شیء مربوط به متن فعال متدی جهت گرفتن ورودی متن (Text) دارد و شیء مربوط به تاریخ تابعی جهت گرفتن تاریخ و ساعت سیستم خواهد داشت و به همین منوال.

نمایه های دیگر شیء گرایی:

دو موضوع مهم دیگر در رابطه با شیء گرایی Instance و Implementation میباشد. Instance اشاره گری به شیء یا یکی از خصوصیات آن است و زمانی که از طریق یک Instance به خصوصیات شیء اشاره میکنیم در حقیقت در حال تاثیر گذاری بر روی خودش هستیم و نه یک کپی از آن واین موضوع زمانی اهمیت پیدا میکند که روی کپی از یک شیء تغییر ایجاد کنیم بدون آنکه خود شیء تاثیر پیداکند. امااگربر روی یک Instance از شیی تغییری ایجاد کنیم خودش هم تاثیر خواهد پذیرفت. Implementation در اصل نوشتن تابعی است که میخواهیم به کلاس اضافه شده تا تمام اشیاء کلاس به این

تابع به عنوان یکی از اعضا خود دسترسی داشته باشند. بحث در مورد این مقوله در حال حاضر از حوصله بحث خارج است در فصول ۱۲ و ۱۳ توضیحات بیشتری در این زمینه خواهیم داد.

شی گرایی در VBScript:

VBScript هم همانند همزادش Visual Basic از شیء گرایی پشتیبانی می نماید هر چند که این پشتیبانی نسبت به سایر زبانهای امروزی کمتر میباشد. در حقیقت ساختار این زبان تا حدود زیادی ساده می باشد . که این موضوع به علت روایتهای قدیمی Basic است که کلاً با روالها(Procedures) کار می کردند. زبانهای دیگر نظیر Java و C++ ساختاری بالنصبه با اصلوب تر و خشن تر دارند. به عنوان مثال Visual basic زبانی Case sensitive نیست بدان معنا که تفاوتی میان حروف کوچک و بزرگ قایل نمی باشد. و این تنها یکی از خصوصیات کوچک این زبان می باشد و در فصل بعدی با تعداد دیگری از آنها آشنا خواهیم شد. در فصول ۱۲ و ۱۳ لیست کامل اشیا، خصوصیات ومتدهای هر کدام به انضمام توضیحات تکمیلی در مورد شیء گرایی را آورده ایم.

فصل سوم:

آغاز VBScript

همانطور که قبلاً هم گفتیم زبانهای Script مستقیماً در درون متن HTML وارد میشود و اعمالی را که به طور عادی زبان HTML از عهده آنها بر نمی آید انجام میدهد. یک متن VBScriptبه طور عمومی به صورت زیر وارد یک صفحه وب میشود.

```
< Script LANGUAGE="VBScript">
VBScript متن
</Script>
```

مجدداً تاکید مینماییم که از این نحوه نوشتاری جهت استفاده در رایانه Client است ونه برای Server و به عنوان ASP .

کد VBScriptمیتواند هم در بخش Head و هم در بخش Body متن HTML وارد شود ولی عموماً آن بخشی از برنامه که باید قبل از نمایش صفحه HTML در حافظه قرار گیرد در بخش Head و بخشی که لازم است همراه صفحه اجرا شود در درون Body نوشته می شود. در ادامه مثالهای متعددی از استفاده در هر دو بخش را خواهیم دید که هرگونه شک و شبهه موجود در مورد مکان صحیح برای نوشتن متن برنامه را برطرف خواهد نمود. البته واضح است که در صورت نیاز می توان در هر دو بخش و به صورت همزمان کد VBScript داشت.

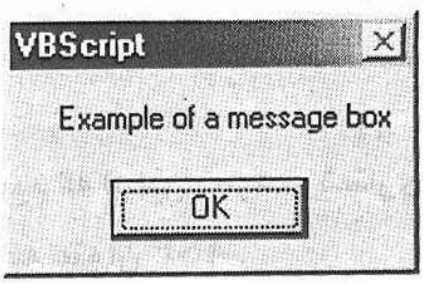
اولین مثال:

در مثالهایی که از این پس ذکر میکنیم خواهیم دید که چگونه باید به این زبان برنامه نوشت به طور عادی در هر خط تنها یک دستورالعمل با پارامترهای لازمش می آید.

یک مثال ساده از یک صفحه وب شامل کد VBScript چنین است :

```
<HTML>
     <HEAD>
        <TITLE>Message Box</TITLE>
     </HEAD>
     <BODY>
        <SCRIPT LANGUAGE ="VBScript">
           MSGBOX("example of a message box")
        </SCRIPT>
     </BODY>
</HTML>
```

کد فوق که شامل دستورالعمل **MSGBOX("Example of a message box")** می باشد تصویر مشابه شکل ۱-۳ پدید می آورد.



شکل ۱-۳

در مثال قبل مسائل بسیاری را می توان دید. اولاً ساختار کلی دستورات: همانطور که گفتیم در یک سطر دستور **MSGBOX** را به همراه پارامترهای لازم که در این مثال متن موردنمایش (به صورت متن مابین (" ")) بود را نوشتیم و در این سطر هیچ دستور دیگری را جای ندادیم. در یک سطر نباید بیش از یک دستورالعمل نوشت مگر در موارد خاصی که بعداً به آن خواهیم پرداخت. ثانیاً می بینیم که دستور **MSGBOX** یک جعبه پیغام (Message box) روی صفحه ظاهر مینماید و دکمه قبول (OK) را روی آن می گذارد. در

زمان اجرا تا وقتی که کاربر کلید OK را فشار ندهد اجرای برنامه متوقف است. و به این شیوه مطمئن میشویم که کاربر پیغام ما را دیده است. (البته تضمینی برای خوانده شدن آن وجودندارد زیرابسیاری از کاربران بدون توجه به متن مورد نمایش تنها کلید قبول را فشار میدهند .) متن مورد نمایش باید مابین پرانتز قرار گیرد و اگر از نوع رشته متنی (String) است باید در میان دو گیومه نوشته شود. بالطبع می توان به جای رشته متنی نام یک متغیر را بدون گیومه ها در درون پرانتز نوشت که در این صورت جعبه پیغام محتوای متغیر را نمایش خواهد داد. در مورد متغیرها و انواع داده ای جلوتر مفصل صحبت خواهیم کرد.

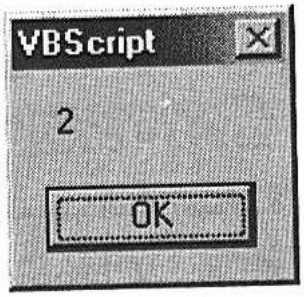
نکته آخر در مورد این مثال آنکه گفتیم در VBscript تفاوتی میان حروف کوچک و حروف بزرگ قایل نیستیم و این دستور به هر صورتی که نوشته شود مثلا" MsGbOX و msgbox در هر صورت درست اجرا میشود. و این که می بینید ما برای کلمات کلیدی از حروف بزرگ استفاده نموده ایم و برای سایر کلمات از حروف کوچک (به جز ابتدای جملات و اسامی خاص) صرفا" از یک قانون ساده و مناسب پیروی میکند: **بالا بردن خوانایی برنامه.** نویسنده نهایی میتواند از این قانون پیروی نموده ویا اینکه از هر ویرایشگری (Editor) با قوانین مخصوص به خود استفاده نماید. البته استفاده از یک ویرایشگر Chromacode بیشتر توصیه میشود. (ویرایشگر Chromacode ویرایشگری است که کلمات کلیدی در آن به یک رنگ متغیرها به رنگ دیگر و خلاصه هر چیز تعریف شده أی به رنگ خاص خود نمایش داده میشود. مانند ویرایشگر محیط Borland C++) که البته این ویرایشگر باید مثلما" برای کلمات زبان VBScript تنظیم شده باشد و الا درست عمل نخواهد کرد یکی از این ویرایشگرها نرم افزار Front page Express می باشد .

نظر به اینکه VBScript زبانی Case sensitive نیست (و از این نظر یکی از معدود زبانهای فعلی با این خاصیت به حساب می آید) یکبار دیگر بر روی این خاصیت تاکید میکنیم و صحبت در مورد آن را با مثال زیر ادامه میدهیم.

```
<HTML>
    <HEAD>
     <TITLE>
            A test to see that there is no
          diffrence between capital letters and
          small letters
```

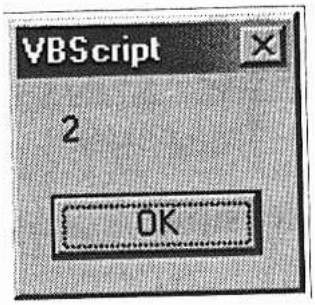
```
    </TITLE>
  </HEAD>
  <BODY>
      <SCRIPT LANGUAGE ="VBScript">
          variable = 1
          VARIABLE = 2
          MSGBOX(variable)
          MSGBOX(VARIABLE)
      </SCRIPT>
  </BODY>
</HTML>
```

این برنامه ابتدا به متغیر variable مقدار ۱ را نسبت می دهد (در مورد متغیرها در فصل بعدی صحبت خواهیم کرد.) و سپس به همان متغیر و این بار با نام VARIABLE مقدار ۲ را نسبت میدهد.اما نتیجه در هر دو پنجره ایجاد شده یکسان است :



شکل ۲-۳

و پس از فشردن دکمه OK دوباره ظاهر خواهد گشت:



شکل ۳-۳

که نشان دهنده آن است که هر دو متغیر در اصل یکی میباشند.

توضیحات:(Comments)

در VBScriptمانند هر زبان برنامه نویسی دیگر امکان در ج توضیحـات در برنامـه وجود دارد. توضیحات خوانایی و اصلاح برنامه را آسانتر میکنند. مترجم برنامـه را بـدون در نظر گرفتن توضیحات ترجمه و اجرا میکند. بر عکس آنچه که بعضی از برنامه نویسان مبتـدی فکر میکنند توضیحات هیچ تاثیری روی سرعت اجرا و یا ترجمه برنامه ها نـدارد و مـی تـوان بدون هیچ گونه دقدقه خاطر و نگرانی به هر میزان دلخواه در جهت افزایش خوانایی برنامه بـه آن توضیحات اضافـه نمـود. جـهت درج توضیحـات در VBScript از کلمـه کلیـدی **REM** استفاده میشود. یااینکه از علامت اختصاری ' (quote) استفاده مینماییم. و هر چه کـه از ایـن نقطه تا پایان سطر نوشته شوداز نظر مترجم توضیحات تلقی میشود.

به عنوان مثال:

```
<HTML>
  <HEAD>
    <TITLE>
      A test to see that there is no diffrence
      between capital letters and small letters
    </TITLE>
  </HEAD>
  <BODY>
     <SCRIPT LANGUAGE ="VBScript">
       variable = 1 ' This is a comment.
       VARIABLE = 2
       MSGBOX (variable) REM This is a comment too.
       MSGBOX (VARIABLE) ' But this format is more
                           comfortable.
     </SCRIPT>
  </BODY>
</HTML>
```

اجرای این کد نتیجه ای عیناً برابر با مثال قبل خواهد داشت. پس از اینجا به بعد میتوانیم هرجا و به هر میزان که لازم بدانیم توضیحات وارد برنامه نماییم البته با در نظر گرفتن این موضوع که منطقاً توضیحات باعث افزایش حجم برنامه می شود. (سعی کنید در تمام برنامه هایی که مینویسید توضیحات آنهم به میزان زیاد درج نمایید اما هنگامی که میخواهید این متن را به صورت یک صفحه وب منتشر نمایید توضیحات را در جهت کاهش حجم برنامه منتقل شده از طریق وب حذف نمایید.)

آرگومانهای VBScript:

آرگومانهای ورودی **MSGBOX** (که در درون پرانتز می آید) می توانند ساده و یا مرکب باشند. تا به حال مثالهای ما با آرگومانهای ساده کار میکردند حالا یک مثال از یک آرگومان مرکب شامل یک رشته متنی-عددی (Alphanumeric) و یک متغیر (به این عمل پیوند Concat می گوییم) می بینیم.

```
<HTML>
  <HEAD>
    <TITLE>Example of a compuest argument</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE ="VBScript">
     Variable = "Friend"
     MSGBOX ("Hello, " + friend)
    </SCRIPT>
  </BODY>
</HTML>
```

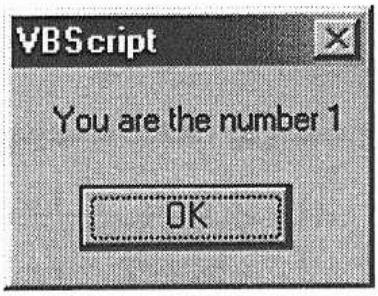
خروجی این مثال چنین است:

شکل ۴-۳

در مثال قبل یک رشته حرفی-عددی را با یک متغیر که محتوای آن هم یک رشته بـود توسط علامت + پیوند دادیم. اما اگر بخواهیم یک رشته و یک متغیر با محتوای عـددی را بـه هم پیوند بزنیم از علامت & استفاده می نماییم. مثال :

```
<HTML>
  <HEAD>
     <TITLE>Example of a complex argument</TITLE>
  </HEAD>
  <BODY>
     <SCRIPT LANGUAGE = "VBScript">
        variable = 1
        MSGBOX ("You are the number " & variable)

     </SCRIPT>
  </BODY>
</HTML>
```

حاصل چنین میشود:



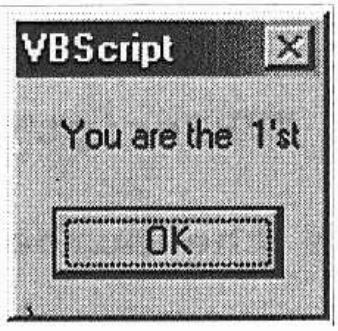
شکل ۵-۳

از علامت & همچنین میتوان جهت پیوند یک رشته به رشته یا متغیرهـای رشـته ای دیگر نیز استفاده نمود. البته واضح است که می توان در یـک خـط بیـش از یـک دسـتورپیونـد داشت و چندین رشته و متغیر را به هم ملحق نمود. مثال:

```
<HTML>
  <HEAD>
    <TITLE>Example of a compuest argument</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
       variable = 1
       Variable2 = "'st "
```

```
        MSGBOX ("You are the  " & variable &
                  variable2) ' Example of  complex
    </SCRIPT>
  </BODY>
</HTML>
```

حاصل چنین است :



شکل ۶-۳

فصل چهارم:

داده ها و متغیرها در VBScript

به عکس سایر زبانهای برنامه نویسی در .V.B تنها یک نوع داده ای عمومی وجـود د ارد که به آن Variant می گوییم در زبانهای دیگر نوع داده ای String جهت ذخیره سازی رشته ها وانواع داده ای دیگر برای ذخیره سازی اعـداد صحیـح و اعشـاری و ممـیز شـناور و نوع داده ای منطقی (Boolean) و غیره وجود دارد. این قـابلیت کـه در حـال حـاضر .V.B و Javascript تنها زبانهای سطح بالایی میباشند که از آن پشتیبانی میکنند قابلیتی فـوق العـاده بوده و باعث می شود تا پس از استفاده از یک متغیر به عنوان یک نوع داده ای در جای دیـگر حسب مورد نوع داده ای آن تغییر کند و لذا نوع داده ای متغیر در هر لحظه امکان تغیـیر دارد و در همان لحظه مشخص شده چک می شود. بدین ترتیب از تمام قابلیات متغیرها در زبانـهای سطح بالامی توان استفاده نمود ولی با این خاصیت مانور بیشتری روی متغیرها داریم. جـهت تغییر نوع داده ای یک متغیر کافیست نوع داده ای که درون آن ذخیره می گردد را تغیـیر دهیـم به عنوان مثال شبه برنامه زیر در زبانهایی نظیر C++ و Java باعث خطا میشود در حالی که در .V.B کاملاً درست است:

```
Variable = 1
' more code
' more code
' more code
Variable = "String"
```

کمی جلوتر با مثالهایی مواجه خواهیم شد که این خاصیت را نمایانتر می سازند. در ضمیمه ۱ فهرست کامل ساختارهای داده ای که .V.B (و بالطبع VBScript) به رسمیت میشناسد آورده شده است. در ادامه این کتاب مثالهایی از استفاده های گوناگون از انواع ساختارهای داده ای را خواهیم دید که روش استفاده از آنها را روشنتر خواهد ساخت .

از آنجایی که به محض اختصاص مقدار به یک متغیر این متغیر متولد میشود نوع داده ای متغیر هم درست در هنگام اختصاص مقدار معین می گردد. و این موضوع برای برنامه نویس بسیار ساده و شفاف خواهد بود چون احتیاجی به تعریف متغیر از یک نوع داده ای خاص یا نوع دیگر ندارد.

متغیرها (Variables):

تمام داده هایی که در یک زبان برنامه نویسی به کار میروند و مقدار آنها توانایی تغییر در حین اجرا را دارد با عنوان عمومی متغیر می شناسیم. می توان متغیر را یک جعبه فرض کرد که محتویات آن در طول اجرای برنامه تغییر میکند. این جعبه نامی دارد که بوسیله آن شناسایی میشود. و از همان طریق به محتویات آن دسترسی پیدا میکنیم.(۱) نام یک متغیر باید با یک حرف شروع شده و(۲) بعد آن میتوان حرف، عدد و یا علامت زیرخط(Underline) (به عنوان تنها کاراکتر نشانه گذاری مجاز برای نامگذاری متغیر) داشت.(۳) حروف بکار رفته در اسامی نیز باید جزو حروف زبان بین المللی (انگلیسی) باشد و نمیتوانند از حروف زبانهای دیگر نظیر اسپانیولی آلمانی و بالاخص فارسی باشد (در اصل تنها بعضی از کاراکتر هایی با کد اسکی (ASCII) کوچکتر از ۱۲۷ مجاز میباشد. و کاراکترهای ç و کاراکترهای فارسی چون کدی بالاتر از ۱۲۷ دارند قابل استفاده نیستند.) (۴) نام یک متغیر نمی تواند کاراکترهای خاصی نظیر ...,&,%,$داشته باشد(۵) ضمن آنکه مجاز به استفاده از فاصله، نقطه، کاما و یا هر گونه علامت نشانه گذاری دیگر هم نیستیم. البته(۶) محتوای یک متغیر حرفی – عددی میتواند هر چیزی که بخواهیم در خود جای دهد ضمناً باید مراقب باشیم از(۷) کلمات

کلیدی زبان هم به عنوان نام متغیر استفاده نکنیم. (در پیوست ۳ فهرست کامل کلمات کلیدی زبان آمده است.) کلمات ذیل به عنوان متغیر در VB و VBScript قابل قبول می باشند.

```
Variable = "This is a string."
Thing = "*&%$*/87          فارسی £"
Birthday2 = 74
Married_yes_or_no = true
```

ولی کلمات زیربه عنوان متغیر قابل استفاده نیستند.

```
72age = 28 ' با عدد آغاز شده است
MsgBox = false ' از کلمه کلیدی استفاده شده است
A?o = 1977 ' چون در کلمه از کاراکتر? استفاده شده است
Pit&sdss= "invalid "
' چون در کلمه از کاراکتر& و کاراکتری با کد اسکی بالاتر از ۱۲۷ استفاده شده
```

جهت استفاده از متغیر دو مرحله در پیش داریم ابتدا تعریف متغیر و سپس مقدار دهی اولیه آن. تعریف متغیر یعنی به زبان بگوییم که از یک متغیر استفاده میکنیم و این امر با کمک دستور **DIM** صورت می گیرد.

رسم الخط استفاده از این دستور به این صورت است:

```
DIM variable
```

این دستور مکانی را در حافظه جهت متغیر ما رزرو می کند ولی هنوز مقداری به آن نسبت داده نشده است در حقیقت مقدار متغیر ما در حال حاضر پوچ (Null) می باشد. (توجه نمایید که در زبانهای برنامه نویسی سطح بالا انواع مختلفی داده پوچ داریم که هرکدام مقدار پوچ دارند یعنی در اصل پوچ هم برای خود یک نوع مقدار می باشد و با معنی فارسی آن تناقض دارد.) مقدار اولیه متغیر همان مقداری است که دفعه اول به متغیر نسبت می دهیم که البته این مقدار می تواند از لحاظ محتوا و نوع در طول برنامه تغییر نماید و عمل ما صرفاً جهت مقداردهی اولیه میباشد.

```
Variable = "Anything"
```

تعریف یک متغیر امری اختیاری است یعنی اگر متغیری را تعریف ننماییم أین متغیر به محض مقداردهی تعریف خواهد شد. اما بهتر است که متغیرهایمان در ابتدا تعریف کنیم (با کمک دستور **DIM**) و بدین وسیله ساختیافتگی برنامه را افزایش داده و آن را خواناتر نماییم. تعریف متغیر باید همواره قبل از مقداردهی اولیه آن صورت گیرد وصحیحتر آن است که تعریف تمام متغیرها در ابتدای برنامه ما بیاید. روشی جهت اطمینان از تعریف تمام متغیرها در طول برنامه وجود دارد وآن هم استفاده از دستور **Option Explicit** در ابتدای برنامه میباشد اگر این دستور را به برنامه اضافه کنیم دیگر نمی توان از متغیرهایی که قبلاً تعریف نشده استفاده کنیم تا کنون در مثالهای قبل این دستور را در برنامه نداشتیم

چون هنوز آن را معرفی نکرده بودیم ولی از همکنون به بعد در تمام مثالها از آن استفاده خواهیم کرد . که این تمرین خوبی هم جهت برنامه نویسی ساخت یافته خواهد بود.

یک روش دیگر هم جهت آسان نمودن تعریف متغییرها وجود دارد :

```
DIM  var1,var2,var3
```

در این روش کار خاصی نکرده ایم تنها به جای اینکه هر متغییر را در یک سطر معرفی نماییم تمام آنها را با یک دستور معرفی نموده ایم از این روش زمانی استفاده میکنیم که بخواهیم متغییرهای مرتبط با هم را تعریف کنیم.

-گسترهٔ دید (Scope)متغیر ها :

مقصود از گستره دید محدوده أی از متن می باشد که متغیر در آن شناخته می شود . شاید برای این بحث کمی زود باشد برای همین پیشنهاد میکنیم در صورتی که با این مقوله آشنا نیستید ابتدا ادامه کتاب را مطالعه فرمایید و پس ازاتمام فصل توابع و روالها دوباره به این مطلب باز گردید دلیل اینکه این مطلب را در اینجا ذکر نموده ایم جلوگیری از پراکندگی مطالب می باشد.

در VB دو سطح گسترهٔ دید داریم یکی گسترهٔ عمومی (Global) و دیگری محلی (Local). متغییرهایی که خارج از توابع و روالها تعریف میشوند و ضمناً جزو خصوصیات شیی نباشند را به عنوان متغیر هایی با گستره دید عمومی می شناسیم و مابقی همگی محلیند با این حال میتوان با بهره گیری از دو دستور Private به معنی خصوصی و Public به معنی عمومی جهت تغییر گستره دید متغیر استفاده نمود ضمن آنکه این دو دستور این توانایی را برای متغیرها ایجاد میکنند تا بتوان از درون یک تابع به متغییرهای تابع دیگر دسترسی پیدا نموده یا این که این دسترسی را کلاً قطع نمایند. روش استفاده به صورت زیر است:

```
Public var1,var2
private var1,var2
```

حتی میتوان تابع و روالهایی را که بر روی دو فایل جدا ولی مرتبط با هم تعریف شده ان به طور کلی یا جزئی به هم پیوند داد و امکان دسترسی یک فایل به متقییر ها و توابع فایل دیگر را فراهم کرد. برای روشن تر شدن موضوع به مثالی که در همین زمینه در فصل ١٤ آمده است مراجعه نمایید.

ماتریسها:

تاکنون از متغیرهای ساده صحبت کردیم که به نوعی میتوان آنها را زوج **اسم-مقدار** نامید چون اساسا" نام یک داده میباشد که به آن مقداری نسبت داده شده است. ولی هیچ دلیلی وجود ندارد که نتوانیم انواع داده ای دیگری را که پیچییده تر هستند استفاده نماییم. به عنوان مثال فرض کنید میخواهیم مجموعه متغیرهایی داشته باشیم که سن دانش آموزان یک مدرسه را در آنها ذخیره نماییم این متغیرها همگی در مورد یک چیز مشابه هستند پس منطقی به نظر می رسد که بین آنها یک رابطه منطقی شباهت برقرار کنیم. کاری که انجام می دهیم ساختن یک جدول یا یک ماتریس از متغیرها میباشد. ماتریس مجموعه تعدادی متغیر است که تحت یک نام قرار دارند در مثال فوق می توان این نام مشترک را سن در نظر گرفت . خوب پس اگر تمام این متغیرها یک نام دارند پس چگونه بین آنها تمایز قایل شده و از هر کدام به صورت مجزا استفاده نماییم؟ برای حل مشکل از اندیس استفاده می نماییم اندیس شماره ترتیبی است که هر عنصر ماتریس را مشخص میکند و بدین

ترتیب هر متغیر را تنها با نام آن نمی شناسیم بلکه نام ماتریسی که بـه آن تعلـق دارد بـه انضمام شماره موقعیت آن درون ماتریس امکان دستیابی را فراهم مـی سـازد. اضافـه مـی شود که تمام این متغیرها به صورت یک صف و پشت سـر هـم قـرار مـی گیرنـد. اولیـن عنصر مکان صفرم بعدی یکم وبه همین منوال حافظـه را اشـغال کـرده و مـاتریس را پـر میکند. (شاید این موضوع برای کسانی که قبلاً با زبان Basic برنامه نویسی نموده اند عجیب باشد ولی جهت بهینگی در VBScript و کلاً مجموعهٔ VB از روش سایر زبانهای سطح بـالا استفاده شده و شروع ماتریس از صفر انتخاب گشته است.)

این تئوری ماتریس بود اما حالا ببینیم که چگونه در واقعیت مـی تـوان از ماتریسـها استفاده نمود . ابتدا ماتریسی با ٥ عضو تعریف می نمـاییم پـیرو مثـال قبـل ایـن مـاتریس جهت ورود سن دانش آموزان خواهد بود این ماتریس را ages مینامیم.

```
DIM ages (4)
```

حتماً متوجه شده ایـد کـه مـاتریس را بـا ٤ عضـو تعریـف نمـوده ایـم در حـالی کـه گفتیـم میخواهیم ٥ عضو داشته باشد. مساله اینجاست که اولین عنصر اندیـس صفـر را بـه خـود اختصاص می دهد ودستور قبل پنج مکان حافظه از صفر تا ٤ (شامل ٠ و ٤) را بـه مـاتریس ما اختصاص می دهد. البته باید متذکر شویم که نمی توان بعد از تعریف ماتریس ابعاد آنـرا در طول برنامه تغییر داد یعنـی اگر بـرای ٥ عنصـر ماتریسـی تشـکیل داده ایـم نمـی تـوان اعضای بیشتری درون آن چید. عکس این موضوع نیز صحیح نیست و میتوان تعداد عناصر کمتری از ظرفیت ماتریس درون آن جای داد چـون مـابقی اعضـا مقـدار Null را بـه خـود اختصاص می دهند ولی هرگز نمی توان بیش از ٥ عنصر درون این ماتریس جـای داد بـرای همین باید ماتریس را با حد اکثر ابعاد تعریف نمود.

حالا که با تعریف ماتریس آشنا شدیم به مقداردهی آن می پردازیم. نام هر عنصـر ماتریس متشکل از دو بخش است ابتدا نام ماتریس و سپس اندیس متغـیر کـه درون پرانـتز می آید. و مقداردهی اعضا ماتریس به روش ذیل صورت می پذیرد :

```
Matrix(index) =  value
```

مثلاً :

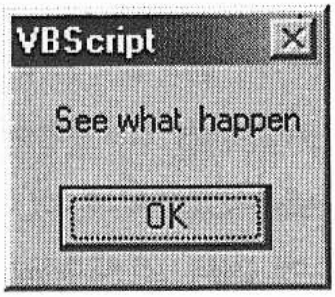
```
Ages(0) = 14
```

ماتریسها که به آنها آرایه نیز میگوییم ابزاری بسیار قوی و مطمئنی هستند که به عنوان مثال می توان به یک المان آن رشته ای حرفی-عددی نسبت داد در حالی که به المان دیگری از آن یک مقدار عددی نسبت دهیم. البته معمولاً از این قابلیت ماتریس به دلیل تعریف منطقی ماتریس استفاده نمی شود. چون همانطور که گفته شد ماتریس جهت نگهداری مجموعه ای از متغیرها میباشد که از جهاتی به هم شباهت داشته باشند وداده هایشان مرتبط با هم باشد.

به مثال زیر توجه فرمایید:

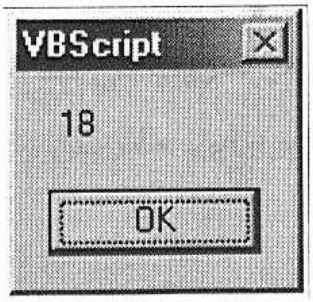
```
<HTML>
  <HEAD>
    <TITLE>
       An example of matrises with diffrent values
    </TITLE>
    <SCRIPT LANGUAGE = "VBScript">
       DIM matrix (1)
       Matrix (0) = "See what  happen"
       Matrix (1) = 6 * 3
      ' The asterix is used as multiplication sign.
       MSGBOX (matrix (0))
       MSGBOX (matrix (1))
    </SCRIPT>
  </HEAD>
  <BODY>
  </BODY>
</HTML>
```

حاصل این مثال:



شکل ۱-۴

و پس از فشار دادن کلید Ok



شکل ۲-۴

مشخص است که محتوای هر المان ماتریس جداگانه حفظ می شود.

نکته حائز اهمیت دیگر در باب ماتریسها آن است که تاکنون ماتریسهایی را معرفی کردیم که به صورت صف و تک بعدی بوده اند ولی می توان ماتریسهایی با ردیفهای متعدد و ابعاد متفاوت داشت مانند مربعها یا ماتریسهای دو بعدی همانطور که عناصرماتریسهای تک بعدی به کمک شماره ردیف آنها (اندیس) شناخته می شوند عناصر ماتریسهای دو بعدی نیز توسط دو اندیس مشخص میشوند یکی مربوط به ستون و دیگری مربوط به سطر محل قرار گرفتن ماتریس در جدول روش تعریف ماتریس دو بعدی به صورت زیر است:

```
DIM matrix_bidimensional (x,y)
```

تمام آنچه که در مورد ماتریسهای تک بعدی بیان نمودیم در مورد ماتریسهای چند بعدی نیز صادق است با این تفاوت که در ماتریسهای دو بعدی دو عدد مابین پرانتز قرار میگیرد که با کاما از هم جدا میشوند.

```
matrix_bidimensional (4,7) = "This an element of a
                             matrix"
```

واضح است که می توان ماتریسهایی با ابعاد بیشتر از دو نیز تعریف نمود (حداکثر تا ٦٠ بعد قابل تعریف است) که در آنها نیز هر بعد با یک اندیس داخل پرانتز نمایش داده میشود به طور عمومی معمولاً حداکثر ابعاد ماتریس را چهار فرض می کنند.

درست است که در ابتدای تعریف ماتریسها گفتیم که ماتریسها را نمیتوان ابعاد دهی مجدد نمود اما این موضوع به عنوان یک اصل پایه می باشد ولی در روایتهای جدیدتر VBScript این امکان بوجود آمده است تا بتوان ماترسیهایی با ابعاد متغیر تعریف نمود برای این کار از دو روش زیر استفاده مینماییم:

```
Dim MyArray()
ReDim AnotherArray()
```

در هر دو روش آرایه أی تعریف نموده ایم که ابعاد آن نامشخص می باشد. در این حالت آرایه ما هنوز هیچ بعدی ندارد و هنوز نمیتوان از آن استفاده نمود. برای آغاز استفاده باید برای بار اول و با کمک دستور **REDIM** آن را ابعاد دهی اولیه نماییم مثلاً:

```
REDIM MyArray(25)
```

باعث ابعاد دهی اولیه به تعداد ٢٦ عنصر می شود . حال هربار که بخواهیم ابعاد ماتریس را تغییر دهیم کافیست دستور بالا را با عدد جدید وارد کنیم. اگر میخواهید هنگام ابعاد دهی مجدد مقادیر داده هایی که از قبل ذخیره کرده بودید معتبر باقی بمانند باید از کلمه کلیدی **PRESERVE** به روش مثال زیر استفاده نمایید:

```
REDIM PRESERVE MyArray(30)
```

با این کار بدون آنکه مقادیر ذخیره شده در آرایه MyArray تغییر کنند ابعاد آن به ٣١ افزایش می یابد. البته بدیهی است که در صورت کوچک کردن ابعاد ماتریس بخشی از اطلاعات را از دست خواهیم داد.مثالی در همین زمینه در فصل ١٤ گنجانده شده است.

تمام اعمالی که بر روی متغیرها میتوان انجام داد بر روی ماتریسها نیز قابل استفاده میباشد که در فصلهای بعد بیشتر به آن خواهیم پرداخت.

ثبات ها:

ثبات ها نوعی عنصر تعریفی هستند که ظاهرشان بسیار به متغیر ها شبیه است اما نمیتوان در حین برنامه آنها را تغییر داد از ثبات جهت بالا بردن خوانایی برنامه و ساده نمودن اصلاحات استفاده می شود زیرا به عنوان مثال جمله ثابت Maximum_size بسیار با معنی تر است از عدد ۱۲۵۳ ضمن آنکه اگر این عدد بخواهد در جاهای مختلف در برنامه تکرار شود با کوچکترین تغییری بر روی آن مجبور خواهیم شد تا تمام برنامه را به دنبال موارد آن گشته و تغییر مورد نظر را اعمال نماییم در حالی که اگر به جای آن معادل ثبات تعریف شده را بگذاریم با تغییر مقدار نسبت داده شده به این ثبات به طور اتوماتیک این تغییر در کل برنامه تاثیر خواهد گذاشت. جهت تعریف ثبات از روش زیر استفاده مینماییم:

```
CONST constant_name = value
```

باز هم با توجه به این که در .V.B انواع دادهای مختلف تعریف نمیشوند برای تعریف ثباتهایی با انواع داده ای مختلف از روش مثالهای زیر استفاده می نماییم:

```
CONST const1 =  45                  برای تعریف ثبات عددی
CONST const2 = "String"             برای تعریف ثبات رشته أی
CONST const3 = #12-1-99             برای تعریف ثبات از نوع تاریخ یا ساعت
```

در VBScript تعداد بسیار زیادی ثبات داخلی تعریف شده است که در ادامه مباحث کم کم با آنها آشنا خواهیم شد. در پیوست ٤ لیست کامل این ثباتها را آورده ایم.

فصل پنجم:

ورود و خروج اطلاعات

یکی از مهمترین جلوه های یک زبان برنامه نویسی نحوه ارتباط آن با مخاطب میباشد این ارتباط در دو جهت اصلی صورت می پذیرد :

۱. **خروج اطلاعات روی صفحه**
۲. **ورود اطلاعات توسط صفحه کلید.**

در این فصل روشهای اصلی هر دو را بررسی خواهیم کرد به صورتی که بتوانیم یک کانال ارتباطی با مخاطب برقرار نماییم جلوتر روشهایی را بررسی خواهیم کرد که به کد برنامه ما خواص پویایی و جذابیت بیشتری خواهد بخشید.

خروج اطلاعات روی صفحه:

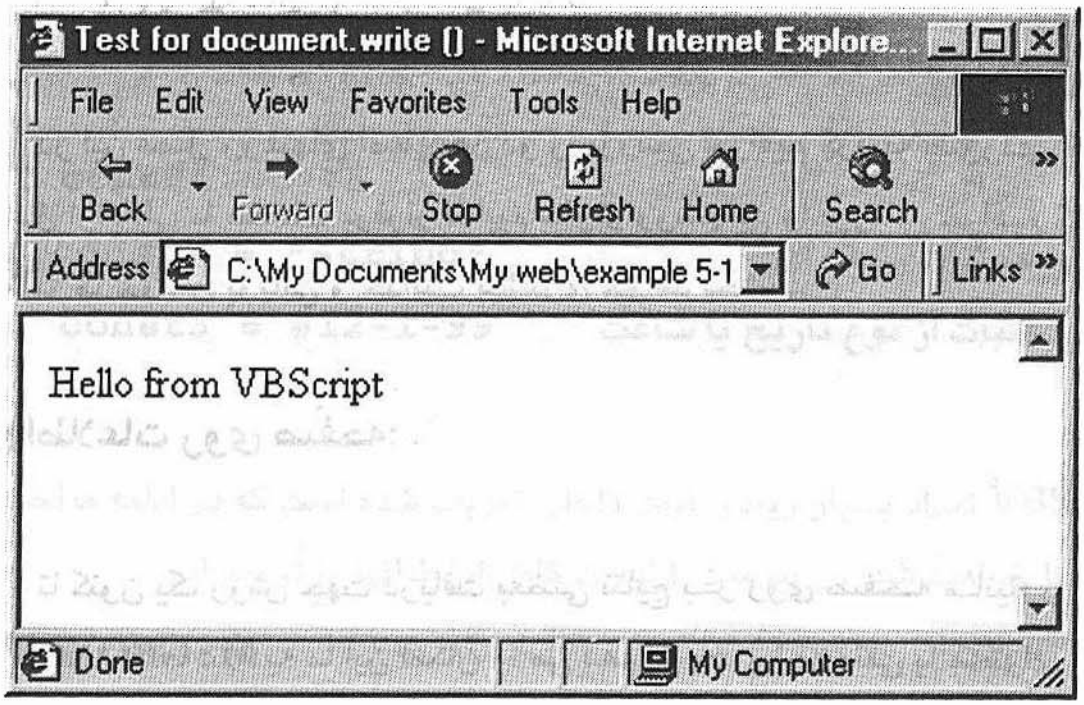
تا کنون یک روش جهت دریافت بعضی نتایج بر روی صفحه مانیتور را دیده ایم، دستور **MSGBOX( )** به ما این امکان را می دهد که یک رشته متنی یا محتوای یک متغیر یا تلفیقی از این دو را به صورت یک پنجره آشنای ویندوز نمایش دهیم. این پنجره یک کلید OK

در خود جای داده بود که کاربر باید روی آن فشار دهد تا ادامه اجرا از سر گرفته شود و همانطور که حتماً دیده اید با فشار این کلید پنجره حاوی پیغام ناپدید می شود. حالا فرض کنید که میخواهیم متنی را روی صفحه نمایش دهیم بدون آنکه احتیاجی به فشار دادن کلید و یا نمایش پنجره جدا و یا درنگ در اجرا داشته باشد ضمناً ناپدید هم نشود. چه باید کرد؟ گفتیم که VBScript زبانی شی گراست وبرنامه فعال (برنامه در حال اجرا) خود شیئی است که به آن Document می گوییم این شی متدهای مختلفی دارد که یکی از آنها متدی جهت نمایش متن بر روی صفحه میباشد.

اگر این مطلب کمی برایتان گنگ است سری به فصلهای دوم و دوازدهم کتاب که در مورد شیگرایی هستند بزنید. شاید مثال زیر کمی موضوع را روشنتر نماید.

```
<HTML>
  <HEAD>
    <TITLE>Test for document.write ()</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
       DOCUMENT.WRITE ("Hello from VBScript")
    </SCRIPT>
  </BODY>
</HTML>
```

حاصل کار مشابه زیر است:

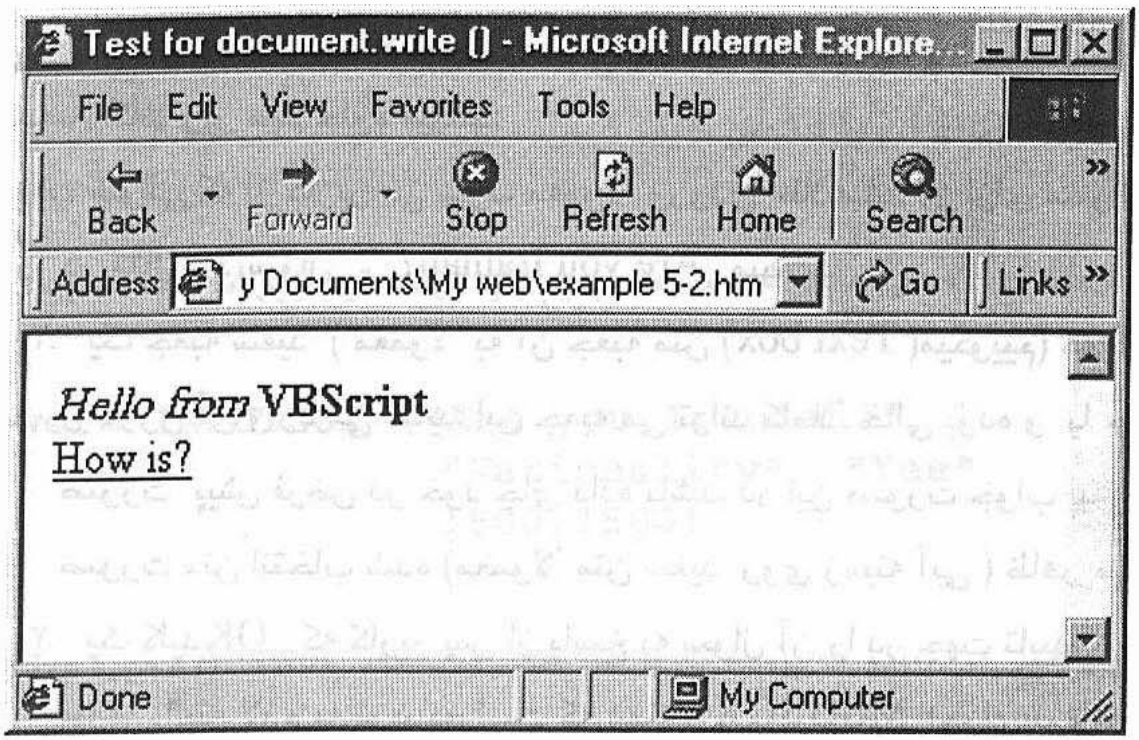


شکل ۱-۵

ورودی متد WRITE یک متن یا متغیر است که به جای متن ساده میتوان متن آرایش شده توسط دستورات HTML و یا حتی کل صفحه HTML مورد نظر را قرار داد به مثال زیر توجه فرمایید:

```
<HTML>
  <HEAD>
    <TITLE>Test for document.write ()</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
      DOCUMENT.WRITE ("<I>Hello from </I> <B>
                      VBScript </B><BR><U>How is?")
    </SCRIPT>
  </BODY>
</HTML>
```

نتیجه برنامه فوق چنین است:

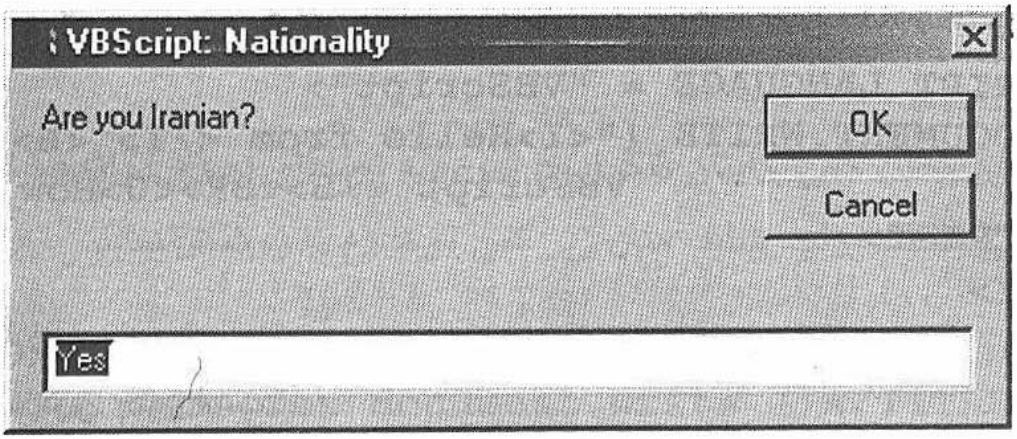


شکل ۲-۵

در حقیقت معمولا می خواهیم تا اطلاعاتمان را روی صفحه نمایش دهیم و لی نه روی یک صفحه سفید بلکه روی قطعاتی که درون صفحه تعریف نموده ایم. برای این کار احتیاج به شناخت بیشتر از اشیاء داریم که به این مبحث در فصول ۱۲ و ۱۳ خواهیم پرداخت.

ورود اطلاعات:

یکی از جالبترین روشهای ورود اطلاعات توسط صفحه کلید استفاده از تابع داخلی **InputBox** می باشد این دستور جعبه گفتگویی(Dialog box) مشابه زیر بوجود می آورد:



شکل ۳-۵

در این جعبه نکات زیر مهم جلوه میکنند:

۱. سوالی که از کاربر می شود. معمولاً ورودی اطلاعات در ازای سوالی میباشد در مثال بالا سؤال " Are you Iranian? " میباشد.

۲. یک جعبه سفید (معمولاً به آن جعبه متن (Text box)میگوییم) که کاربر پاسخ را درون آن وارد می نماید. این جعبه می تواند کاملاً خالی بوده و یا جوابی را به صورت پیش فرض در خود جای داده باشد. در این صورت جواب پیش فرض به صورت متن انتخاب شده (معمولاً متن سفید روی زمینه آبی) ظاهر می گردد.

۳. یک کلیدOK که کاربر پس از پاسخ به سوال آن را در جهت تایید پاسخ فشار میدهد. البته لازم به توضیح است که همانند MSGBOX اجرای برنامه تا زمان فشرده شدن کلید متوقف می شود.

٤. یک کلید Cancel کلیدی جدید که در مثالهای قبل ندیده بودیم به جهت آنکه کاربر بتواند به سوال پاسخ نداده و به ادامه برنامه بپردازد.

٥. در قسمت بالای جعبه نیز یک نوار پررنگ ظاهر شده که متنی درون آن است و به آن نوار عنوان (Title Bar) میگوییم .

حالا میتوانیم نحوه استفاده از **INPUTBOX** را ببینیم.

INPUTBOX(سوال ,عنوان , پاسخ پیش فرض , موقعیت x , موقعیت y)

همانطور که میبینیم این دستور می تواند با پارامترهایی مختلف که توسط علامت کاما از هم جدا شده اند را بپذیرد که توضیح هر کدام در ذیر آمده است:

۱. **سوال:** سوالی است که برای کاربر مطرح می شود و یا به آن پاسخ دهد.
۲. **عنوان:** عنوانی است که به عنوان سرتیتر در بالای جعبه ظاهر میشود.
۳. **پاسخ پیش فرض:** پاسخی است که می خواهیم به عنوان پیش فرض برای کاربر نمایش داده شود.
٤. **موقعیت x و موقعیت y:** موقعیت مکانی نقطه بالا سمت چپ جعبه را جهت ظاهر شدن در صفحه مشخص می سازد . این مختصات در مقیاس Twip که ازخصوصیات Windows است می باشد. جهت آشنا تر شدن با این مقیاس باید مقادیر مختلف آن را امتحان کنید.

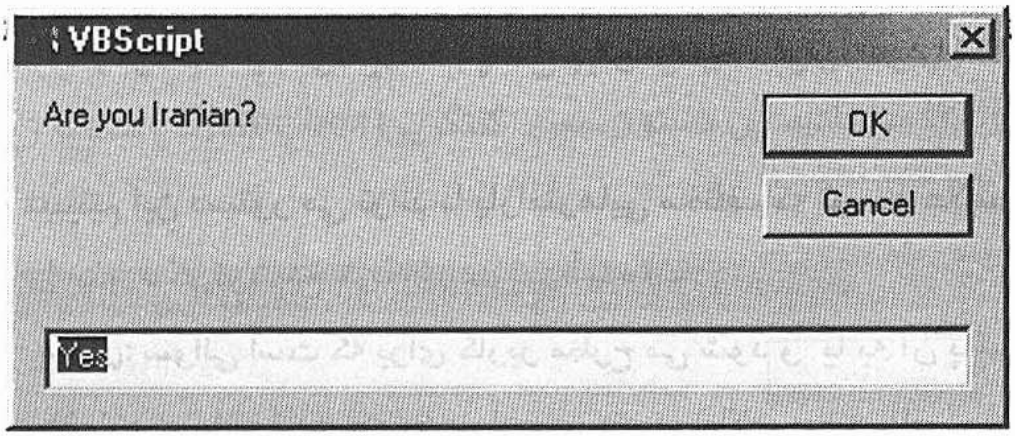
جهت ایجاد جعبه أی که در ابتدای این مبحث دیدیم باید از دستور زیر استفاده نماییم:

```
Answer= INPUTBOX ("Are you Iranian?",
                  "Nationality", "Yes",
                  1500,1500)
```

در این دستورالعمل می بینیم که پاسخ کاربر در متغیری به نام Answer ذخیره میشود. تا بعداً بتوان از آن استفاده کرد تنها پارامتر اجباری دردستور العمل فوق اولین پارامتر آن یعنی سوال می باشد و مابقی اختیارید به عنوان مثال فرض کنید می خواهیم سوال ما عنوان نداشته باشد . در این صورت دستور رابه صورت زیر تغییر میدهیم:

```
Answer= INPUTBOX ("Are you Iranian?", , "Yes",
                  1500,1500)
```

حاصل چنین است :



شکل ۴-۵

اگر کاربر کلید Cancel را فشار دهد متغیر بکار رفته مقدار Null را به خود اختصاص خواهد داد. فعلاً تمام پاسخهای کاربر به صورت رشته های متنی خواهد بود جلوتر خواهیم دید که چگونه می توان عدد یا حتی تاریخ درخواست کرد.

البته تنها راه گرفتن ورودی از کاربر استفاده از INPUTBOX نیست بلکه با بهره گیری از شی گرایی میتوان از تمام اشیایی که درون متن HTML تعریف میکنیم ورودی دریافت داریم.به این مبحث هم در فصو ل ۱۲ و ۱۳ خواهیم پرداخت.

فصل ششم:

دستورات شرطی

زمانی که پاسخی به یک سوال داده مـی شـود و یـا زمـانی کـه حـاصل یـک عبـارت محاسبه می شود یا در شرایط متفاوت دیگر کدی جهت تصمیم گیری لازم است تا یک دستور یا دستوری دیگر(یا هیچ کاری) با توجه به پاسخ دریـافتی اجـرا شـود. بـه عنـوان مثـال اگـر جواب سوال " آیا ایرانی هستید ؟ " صحیح باشد باید پس از آن شماره شناسنامه سوال شـود و در غیر این صورت شماره گذرنامه . موارد متعددی وجـود دارد کـه احتیـاج بـه دسـتورات شرطی پیدا می کنیم . در این فصل به شرح انواع دستورات شرطی می پردازیم.

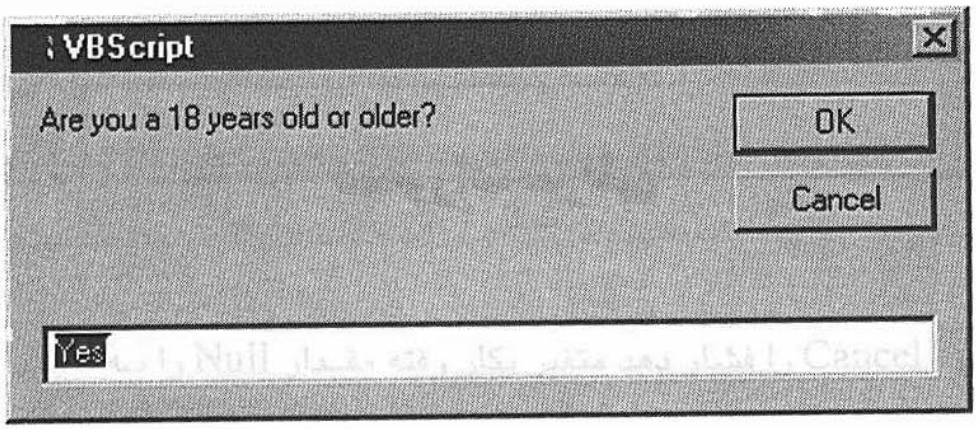
جملات IF... THEN...ELSE

کلمه IF به معنی اگر، THEN به معنی آنگاه و ELSE به معنی در غـیر اینصـورت می باشد. زمانی که می خواهیم برنامه یک یا چند دستورالعمل را با توجه به شـرط داده شـده انجام دهد معمولاً از این ساختار استفاده میشود . مثال:

```
<HTML>
  <HEAD>
    <TITLE>Test for a simple condition</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
      age = INPUTBOX ("Are you a 18 years old or
                       older?",,"Yes")
      IF (age = "Yes") THEN
```

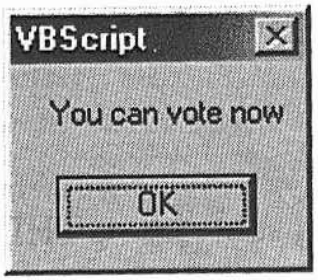
```
        MSGBOX ("You can vote now")
      END IF
    </SCRIPT>
  </BODY>
</HTML>
```

این کد در ابتدا جعبه زیر را نمایش می دهد:



شکل ۱-۶

حال اگر کاربر پاسخ " Yes " دهد (می توان این کار را صرفاً با فشار کلید OK به عنوان پیش فرض انجام داد) برنامه پنجره زیر را نمایش می دهد.



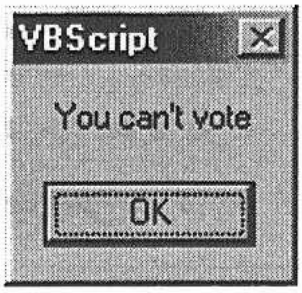
شکل ۲-۶

ودر صورتی که کاربر هر چیز دیگری را وارد نماید یا کلید Cancel را فشار دهد هیچ اتفاق خاصی رخ نخواهد داد. همانطور که می بینید شرط درون پرانتز آمده است این امر در VBScript ضروری نیست ولی خوانایی متن را افزایش می دهد. ما همواره از این روش استفاده خواهیم نمود.

کمی مثال را بهینه می نماییم :

```
<HTML>
  <HEAD>
    <TITLE> Test for a simple condition</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
      age = INPUTBOX ("Are you a 18 years old or
                       older?",,"Yes")
      IF (age = "Yes") THEN
         MSGBOX ("You can vote now")
      ELSE
         MSGBOX ("You can not vote")
      END IF
    </SCRIPT>
  </BODY>
</HTML>
```

در اینصورت پاسخ متفاوت از Yes و یا فشردن کلید Cancel منجر به ظاهر شدن پنجره زیرمی شود:



شکل ۳-۶

همانطور که می بینید جملات شرطی مابین IF و ENDIF محصور شده اند و در ادامه دستور IF و در همان خط شرط مربوطه و پس از آن THEN و سپس مجموعه دستوراتی که در صورت تحقق شرط اجرا می شوند می آید و پس از دستور ELSE مجموعه دستوراتی که در صورت عدم تحقق شرط اجرا می شوند و تمام بلوک شرط با کلمه کلیدی END IF خاتمه می یابد با این حال ساختار کلی دستورات شرطی به صورت زیر می باشد.

```
IF (condition) THEN
    Instruction block 1
ELSE
```

```
    Instruction block 2
END IF
```

شروط مضاعف :

ساختار قبلی حالتی را می سازد که در آن در صورت برقراری شرط یک امر و در صورت عدم برقراری امر دیگری اجرا می شود حال به ساختار زیر توجه فرمایید:

```
IF (condition_1) THEN
    Instruction block 1
ELSE
    IF (condition_2) THEN
       Instruction block 2
    ELSE
       Instruction block 3
    END IF
END IF
```

این ساختار کمی پیچیده تر به نظر می رسد . اگر شرط برقرار شود و بلوک ۱ دستورات اجرا شده و اگر نه شرط دوم چک می شود در صورت صحت بلوک دستورات ۲ و در غیر اینصورت بلوک دستورات ۳ اجرا می شود .

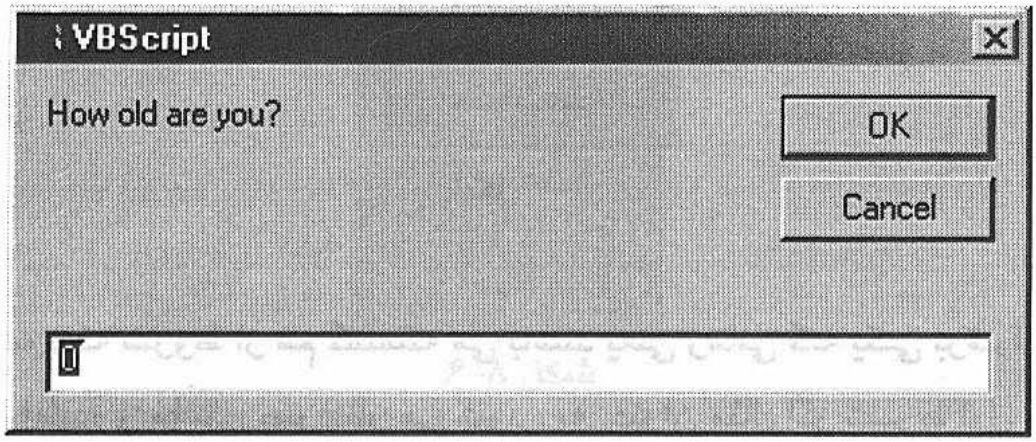
مثال:

```
<HTML>
  <HEAD>
    <TITLE>Use of a dual condition</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
      age = INPUTBOX ("How old are you?",,"0")
      IF (age < 18) THEN
        MSGBOX ("you have to go to School")
      ELSE
          IF (age > 65) THEN
              MSGBOX ("you are an old")
           ELSE
              MSGBOX ("you are young")
```

```
            END IF
        END IF
    </SCRIPT>
</BODY>
</HTML>
```

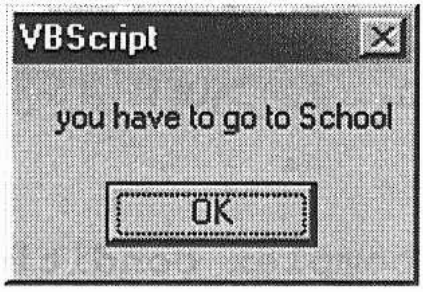
ابتدا" یک نکته آنهم اینکه در فصل قبل گفتیم INPUTBOX یک رشته برمیگرداند در حالی که در اینجا یک مقدار عددی دریافت نموده و با آن به صورت عددی هم برخورد میکند این امر به دلیل ساختار جالب داده ها در این زبان می باشد (فصل ۳ بخش متغیرها در VBScript را مطالعه نمایید.)

حالا ببینیم با اجرای کد قبل چه اتفاقی می افتد ؟ ابتدا از کاربر سن او سوال می شود. پاسخ پیش فرض صفر است.



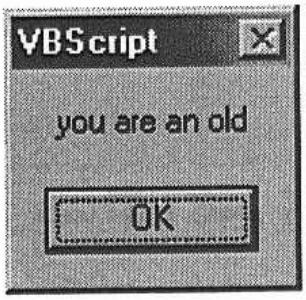
شکل ۴-۶

با ورود سن و فشار دکمه OK کوچکتر از ۱۸ بودن ورودی چک می شود در صورت صحت پنجره زیر ظاهر خواهد بود:



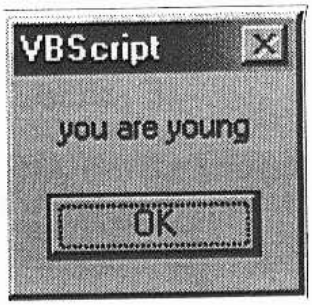
شکل ۵-۶

و در غیر این صورت عدد ورودی را با ٦٥ مقایسه مینماید در صورت بزرگتر بودن پیغام زیر را می دهد:



شکل ۶-۶

و تنها در صورتی که هر دو شرط نقض شود پنجره زیر ظاهر خواهد شد.



شکل ۶-۷

باید گفته شود که شروط از هم گسسته می باشند یعنی زمانی که یکی برقرار شد بلوک مربوطه اجرا شده و ما بقی صرفنظر می شود . به عنوان مثال اگر شرط اول برقرار شود مسلماً شرط دوم چک نمی شود.

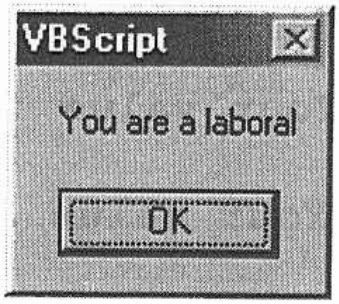
شروط مختلط:

فرض کنید تنها می خواهیم بفهمیم که آیا سن کاربر بین ۱۸ تا ٦٥ سال هست یا خیر و بالاتر یا پایینتر بودن آن برای ما تفاوتی ندارد. مثال :

```
<HTML>
  <HEAD>
    <TITLE>Use of a complex condition</TITLE>
  </HEAD>
  <BODY>
```

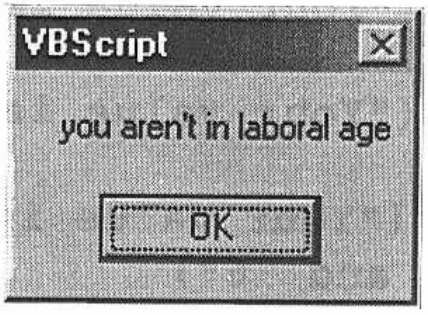
```
    <SCRIPT LANGUAGE = "VBScript">
      age = INPUTBOX ("How old are you?",,"0")
      IF (age > 18 AND age < 65) THEN
         MSGBOX ("You are a laboral ")
      ELSE

      END IF
    </SCRIPT>
  </BODY>
</HTML>
```

در این مثال می بینیم که شرط در اصل خود از دو شرط تشکیل شده که توسط عملگر منطقی **AND** به هم پیو سته اند (ضمیمه ۲ را مطالعه نمایید) این سطر را می توان به این صورت خواند :" اگر سن وارد شده از ۱۸ بیشتر بوده و از ٦٥ هم کمتر باشد آنگاه ..." به این صورت شروط مختلط پدید می آیند. پس اگر عدد ورودی مابین این دو عدد باشد حاصل چنین می شود :



شکل ۸-۶

و در غیر این صورت هر عدد دیگر:



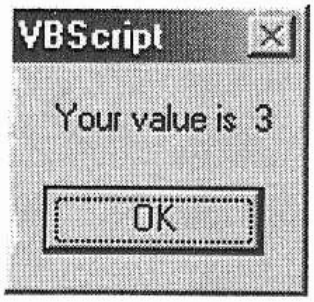
شکل ۹-۶

دستور SELECT CASE:

زمانی کـه شـرایط زیـادی جـهت تحقـق وجـود دارد نبـاید از دسـتورات شـرطی مضاعف یا مختلط استفاده کرد زیرا فوق العاده بی معنی و سخت خواهد شد . بـه جـای آن از مجموعه دستورات **SELECT CASE , ENDCASE** استفاده می نماییم .مثال:

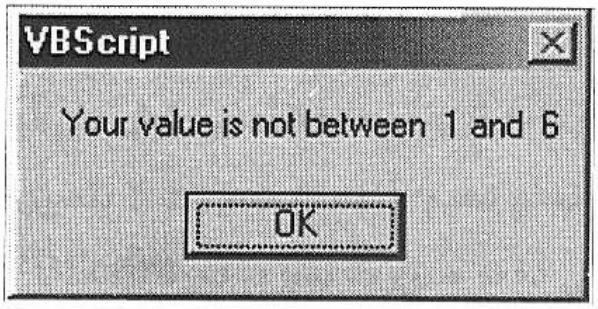
```
<HTML>
  <HEAD>
    <TITLE>Test of the  Select case</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
       OPTION EXPLICIT
       DIM quantity
       quantity = 3

      SELECT CASE quantity
        CASE 1:
           MSGBOX ("Your value is  1")
        CASE 2:
           MSGBOX ("Your value is  2")
        CASE 3:
           MSGBOX ("Your value is  3")
        CASE 4:
           MSGBOX ("Your value is  4")
        CASE 5:
           MSGBOX ("Your value is  5")
        CASE 6:
           MSGBOX ("Your value is  6")
        CASE ELSE:
           MSGBOX ("Your value is not between 1
                    and  6")
      END SELECT
    </SCRIPT>
  </BODY>
</HTML>
```

حاصل چنین است:



شکل ۱۰-۶

اما اگر فرض کنیم سطر quantity = 3 را با quantity = 7 جایگزین کنیم پاسخ چنین میشود:



شکل ۱۱-۶

فصل هفتم:

حلقه ها

در تمام زبانهای برنامه نویسی موقعیتهایی پیش می آید که می خواهیم یک مجموعه دستور را بیش از یک بار انجام دهیم آنهم به صورت لا یتغییر و مکرر واین عمل می تواند به تعداد متناهی یا نامتناهی دفعه صورت پذیرد. در VBScript ساختارهای متفاوتی برای این عملیات داریم تمام آنها را بررسی می کنیم تا در موقعیتهای متفاوت نوع مناسب را انتخاب نماییم.

حلقه FOR ... NEXT :

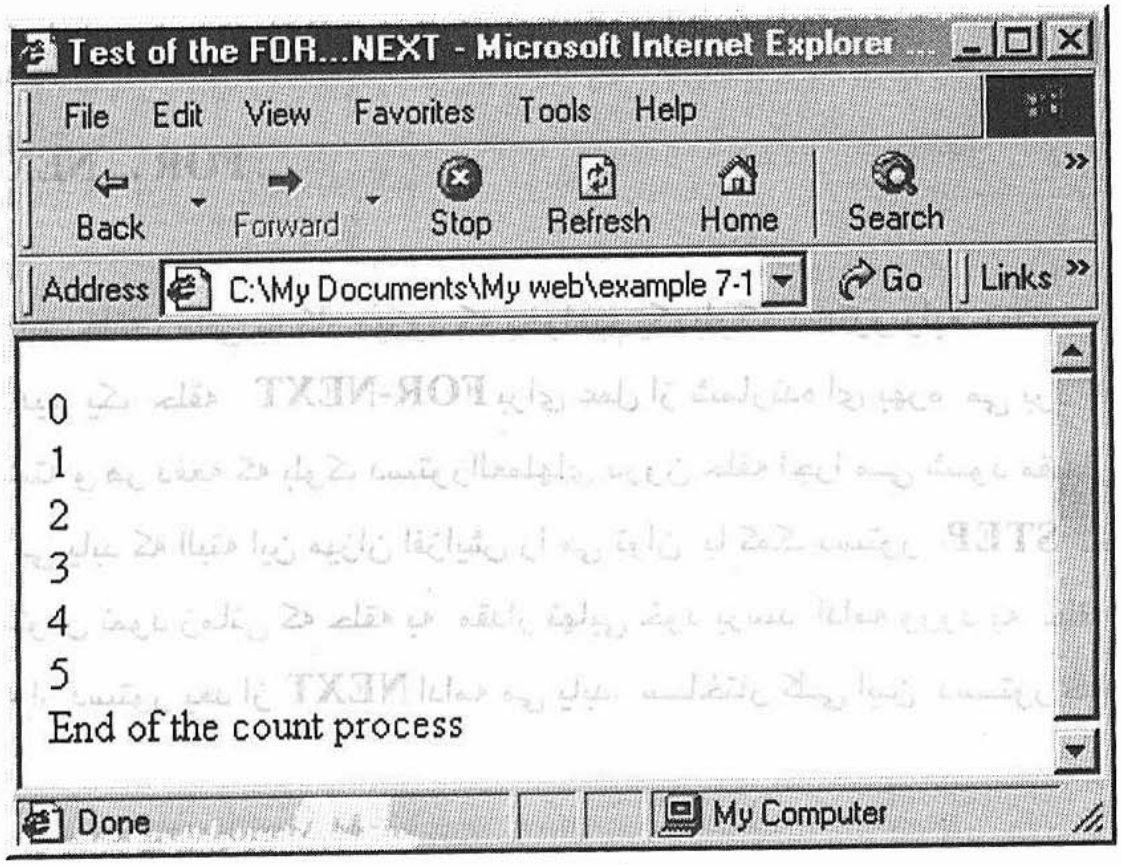
این حلقه زمانی به کار میرود که بخواهیم یک بلوک دستور رابه تعداد متناهی دفعه انجام دهیم. یک حلقه **FOR-NEXT** برای عمل از شمارنده ای بهره می برد که یک مقدار اولیه داشته و هر دفعه که بلوک دستورالعملهای درون حلقه اجرا می شود مقدارش یکواحد افزایش می یابد که البته این میزان افزایش را می توان با کمک دستور **STEP** تغییر داده و حتی معکوس نمود زمانی که حلقه به مقدار نهایی خود برسد ادامه ورود به حلقه پایان یافته و برنامه از دستور بعد از **NEXT** ادامه می یابد. ساختار کلی این دستور به شرح ذیل است:

```
FOR counter = Initial TO final STEP increment_step
      Instruction Block
NEXT
```

به عنوان مثال :

```
<HTML>
  <HEAD>
    <TITLE>Test of the FOR...NEXT</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
       OPTION EXPLICIT
       DIM counter
       FOR counter = 0 TO 5
         DOCUMENT.WRITE (counter & "<BR>")
       NEXT
       DOCUMENT.WRITE ("End of the count process")
    </SCRIPT>
  </BODY>
</HTML>
```

حاصل چنین می شود :

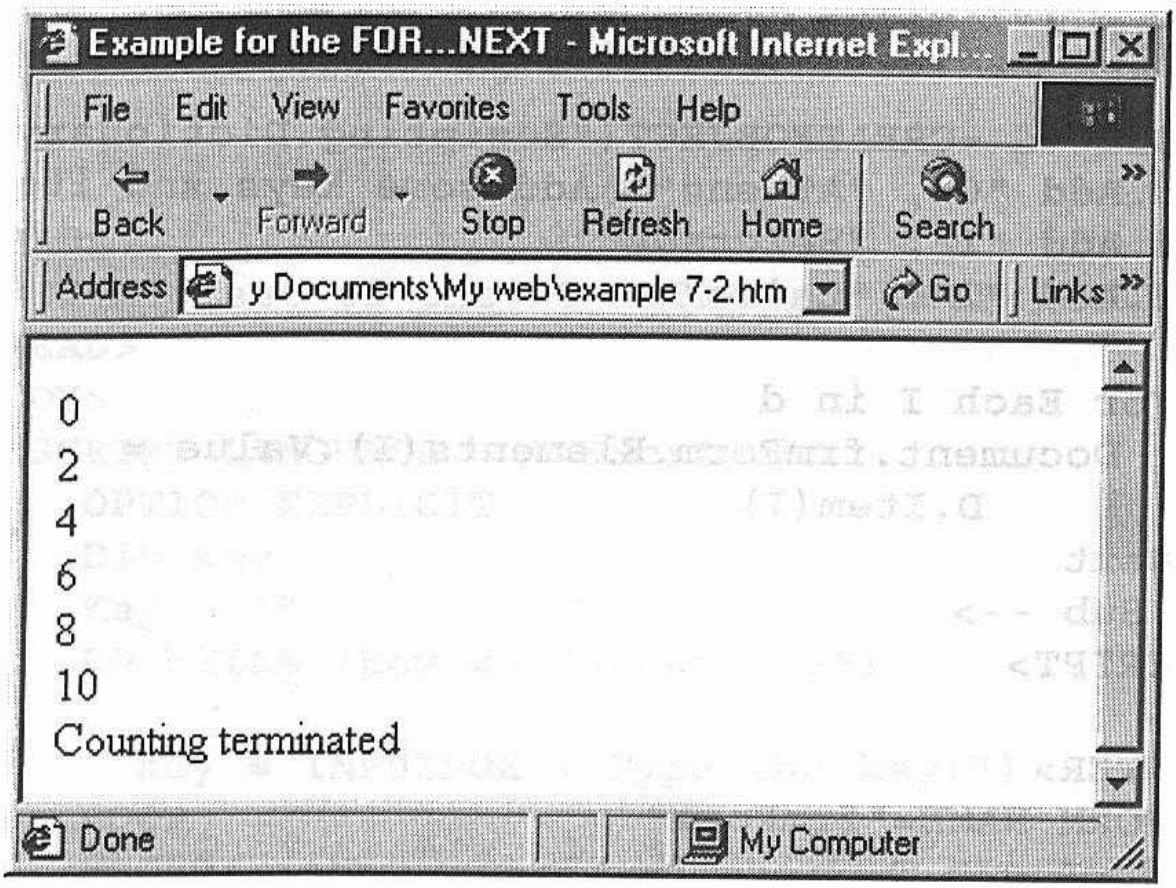


شکل ۱-۷

و حالا مثال بعدی :

```
<HTML>
  <HEAD>
    <TITLE>Example for the FOR...NEXT</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
       OPTION EXPLICIT
       DIM counter
       FOR counter = 0 TO 10 STEP 2
         DOCUMENT.WRITE (counter  & "<BR>")
       NEXT
       DOCUMENT.WRITE ("Counting terminated")
    </SCRIPT>
  </BODY>
</HTML>
```

حاصل مانند زیر است:



شکل ۲-۷

همانطور که ملاحظه می شود در مثال آخر عمل شمارش به صورت یکی در میان انجام میشود.

حلقه FOR EACH ... IN ... NEXT:

این حلقه برای بهره گیری از خواص شی گرایی .V.B استفاده می شود و معرف آن است که تا زمانی که متغیری عضوی از کلاس شی می باشد حلقه تکرار شود. این مبحث کمی پیچیده است در فصل ۱۳به طور مفصل در مورد آن صحبت خواهد شد. فعلاً در حد مثال زیر کافیست:

```
<HTML>
  <HEAD>
    <TITLE>Forms and Elements</TITLE>
  </HEAD>
  <SCRIPT LANGUAGE="VBScript">
  <!-- SUB cmdChange_OnClick
     DIM d              'Create a variable

     SET  d =CREATEOBJECT("Scripting.Dictionary")
     d.Add "0", "Athens" 'Add some keys and items
     d.Add "1", "Belgrade"
     d.Add "2", "Cairo"

     For Each I in d
       Document.frmForm.Elements(I).Value =
            D.Item(I)
     Next
  End Sub -->
  </SCRIPT>
<BODY>
  <CENTER>
    <FORM NAME="frmForm" >
       <Input Type = "Text"><p>
       <Input Type = "Text"><p>
       <Input Type = "Text"><p>
       <Input Type = "Text"><p>
       <Input Type = "Button" NAME="cmdChange"
              VALUE="Click Here"><p>
```

```
    </FORM>
   </CENTER>
  </BODY>
 </HTML>
```

حلقه DO WHILE ... LOOP:

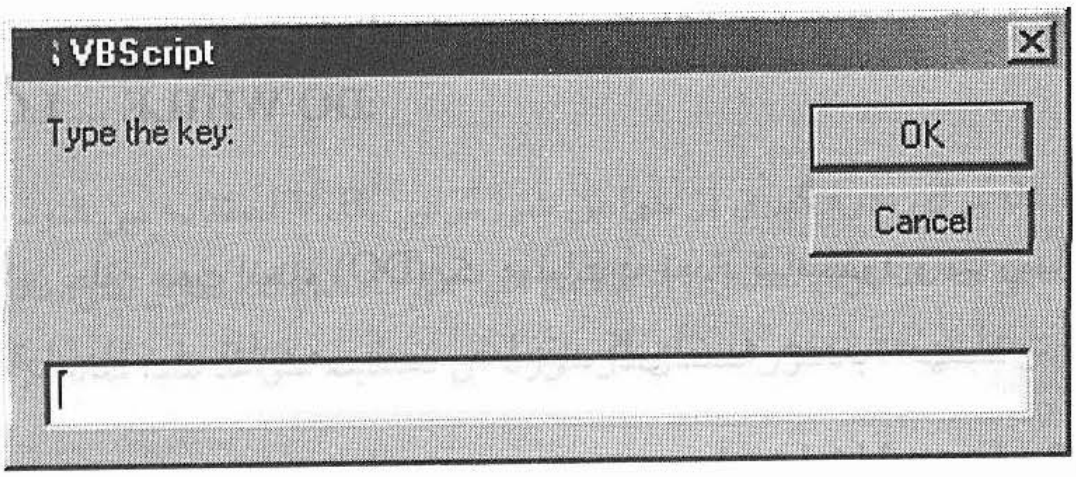
از این حلقه جهت انجام (DO) یک عملیات به تعداد نـامحدود و تـا زمـانی (WHILE) که شرط برقرار باشد استفاده می شود . ساختار کلی آن به صورت زیر است :

```
DO WHILE (condition)
     Instruction block
LOOP
```

بلاک دستوراتی که در میان حلقه آمده است تا زمانی کـه شـرط برقـرار باشد اجرا می شود. فرض نمایید می خواهیم از کاربر کلمه رمز عبور را دریـافت نمـاییم تـا برنامه ادامه یابد کلمه عبور به عنوان مثال می تواند Authority باشد. ببینیم چگونه میتوان این کار را انجام داد.

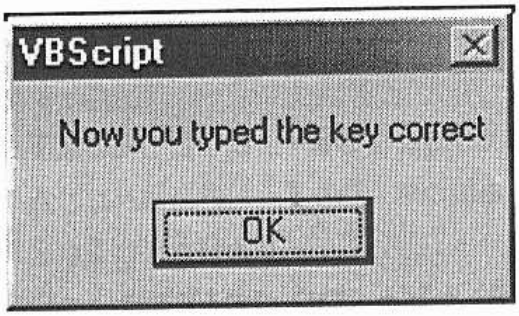
```
<HTML>
  <HEAD>
    <TITLE>Test of the DO WHILE ... LOOP</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
       OPTION EXPLICIT
       DIM key
       Key = ""
       DO WHILE (key <> "Authority")

         key = INPUTBOX ("Type the key:")
       LOOP
       MSGBOX ("Now you typed the key correct")
    </SCRIPT>
  </BODY>
</HTML>
```

این مثال کادر ذیل را نمایش می دهد و تا زمانی که رمز عبور درست وارد نشود یا حتی از ورود آن صرفنظر شود تا بینهایت این سوال تکرار خواهد شد.



شکل ۳-۷

و در صورت درستی پنجره زیر به نمایش در می آید :



شکل ۴-۷

حلقه های DO UNTIL ... LOOP:

این حلقه بسیار مشابه قبل عمل میکند با این تفاوت که تا زمانی اجرا می شود که شرط UNTIL برقرار نباشد پس از برقراری شرط از ادامه جلوگیری شده و از حلقه خارج می شود ساختار کلی آن به صورت زیر است:

```
DO UNTIL (condition)
      Instruction Block
LOOP
```

مثل همیشه بهترین راه درک روش کار این حلقه دیدن یک مثال می باشد ، فرض نمایید میخواهیم مسئله قبل را دوباره حل کنیم باید تا زمانی که رمز صحیح وارد نشده اسم رمز بخواهیم:

```
<HTML>
  <HEAD>
    <TITLE>
       Example of use of DO WHILE ... LOOP
    </TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
       OPTION EXPLICIT
       DIM key
       key = ""
       DO UNTIL ( key = "Authority")
          key= INPUTBOX ("Type the key:")
       LOOP
       MSGBOX ("Now you typed the key correct")
    </SCRIPT>
  </BODY>
</HTML>
```

نتیجه عیناً مطابق قبلی می باشد.

حلقه WHILE ... WEND:

این حلقه ها جزو اولین حلقه های دوران اولیه Basic می باشد و امروزه کمتر از آنها استفاده می شود و جای خود را به حلقه های داده اند چون مدل جدید حالت ساختیافته تری دارد ولی کماکان مترجم ها از این نوع حلقه به جهت سازگاری و عادت برنامه نویسان قدیمی پشتیبانی مینمایند . ساختار کلی آن به شرح زیر می باشد :

```
WHILE (condition)
      Instruction Block
WEND
```

به هر ترتیب اکیداً توصیه می نماییم که به این مدل آخر عادت نکنید چون ممکن است مترجمهای جدید از آن پشتیبانی ننمایند.

شکستن حلقه ها:

در بعضی مواقع لازمست که اجرای حلقه به صورت دستی پایان پذیرد به عنوان مثال زمانی که حالتی اتفاق می افتد که اجرای برنامه را نسبت به ادامه حلقه ارجح می داند. برای این امر از دستور **EXIT** استفاده می شود این دستور به سه صورت متفاوت وابسته به نوع حلقه مورد استفاده قرار میگیرد **EXIT FOR** برای شکستن حلقه های **FOR ... NEXT**، از **EXIT LOOP** برای شکستن حلقه های **DO ... LOOP** و از **EXIT WHILE** برای شکستن حلقه **WHILE ... WEND** استفاده می نماییم.

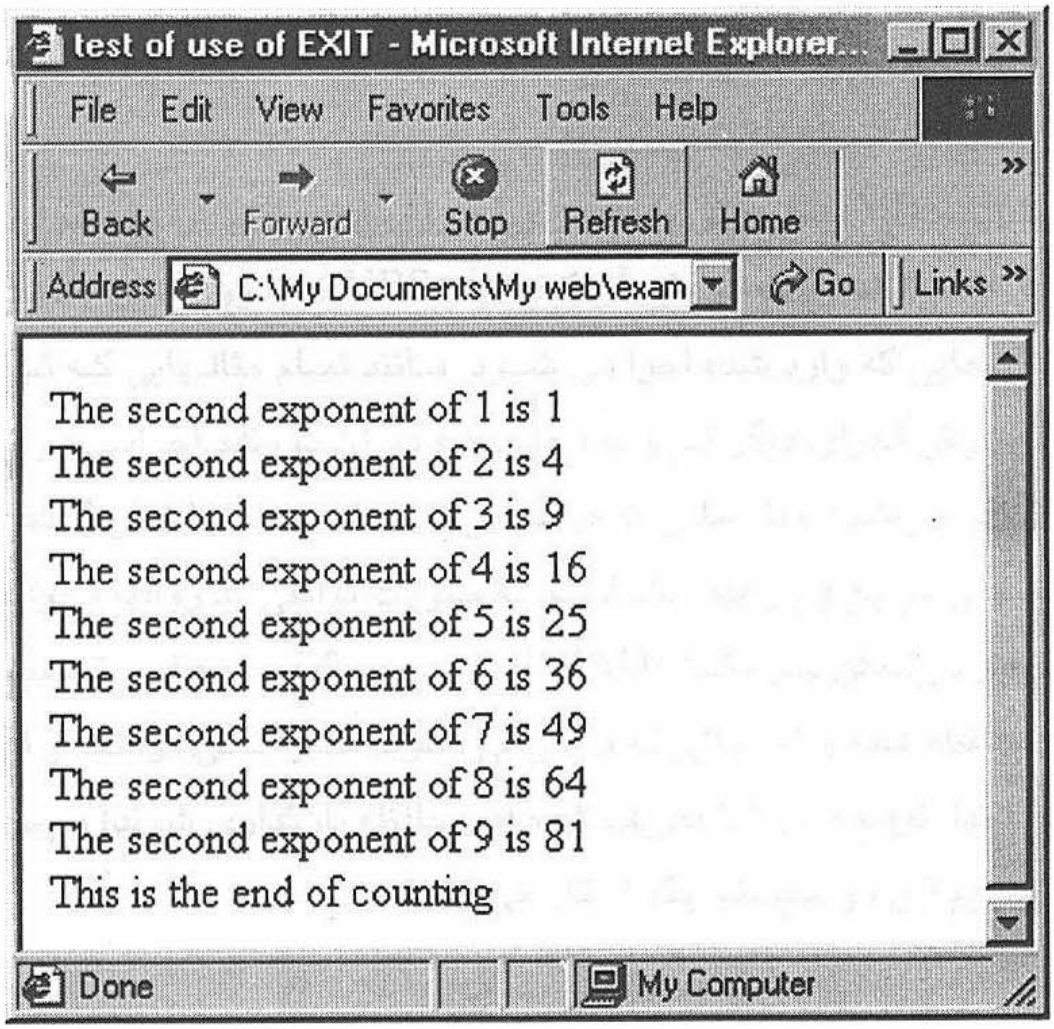
یک مثال کاربردی : فرض نمایید که در حال شمارش اعداد هستیم و می خواهیم که شمارش با رسیدن به عددی که مربع آن از دو بیشتر است متوقف گردد:

```
<HTML>
  <HEAD>
    <TITLE>test of use of EXIT</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
       OPTION EXPLICIT
       DIM num
       DIM quad
       num = 0
       quad = 0
       DO WHILE (num < 1000)
         num = num + 1
         quad = num * num
         IF (quad > 99) THEN
            EXIT DO
         END IF
         DOCUMENT.WRITE("The second exponent of " &
                        num & " is " & quad
                        &"<BR>")
```

```
        LOOP
        DOCUMENT.WRITE ("This is the end of
                         counting")
    </SCRIPT>
  </BODY>
</HTML>
```

حاصل مانند شکل ۵-۷ می شود.

این مثال نشان دهنده نحوه استفاده از **EXIT DO** بود سایر حلقه شکنها نیز از همین الگو استفاده مینمایند.



شکل ۵-۷

فرمان **EXIT** فرمان زیاد جالبی نیست چون معمولاً روشهای بسیار بهتری جهت حل اینگونه مسایل به صورت زیباتر و ساختیافته تر وجود دارد ولی شناخت این دستور و پدرش **GOTO** خالی ازلطف نیست (با توجه به تاکید بسیار زیادی که بر برنامه نویسی

ساختیافته وجود دارد از توضیح فرمان GOTO صرفنظر می کنیم علاقمنـدان میتواننـد بـه کتابهای آموزش زبان Basic مراجعه نمایند.)

فصل هشتم:

توابع و روالها

در VBScript سه روش پایه برای اجرا وجود دارد. اولی اجرای بدون وقفه (Immediate execution) که در طی آن کد VBScript که درون متن HTML وارد شده است در همان جایی که وارد شده اجرا می شود. مانند تمام مثالهایی که تاکنون مشاهده نموده ایم. دو روش اجرای دیگر نیز وجود دارد که در آن دو، کد اجرایی در مرحله دوم در حافظه بارگذاری می شود وتا زمانی که فراخوانی نشده است اجرا نمی گردد. این دو روش اجرا را به نامهای **توابع و روالها** میشناسیم. به صورت توافقی کد روالها و توابع معمولاً در ابتدای صفحه و در بخش سر متن (HEAD) وارد می گردد . به این ترتیب هنگام آمدن صفحه وارد حافظه شده و تا زمانی که فراخوانی نشوند اجرا نمی گردند. و این بدان علت است که باید ابتدا توابع و روالها تعریف شده (در حافظه بارگذاری شوند) و پس از فراخوانی اجرا شوند. ببینیم این دو سیستم چگونه کار می کنند؟

-توابع:

تابع یک مجموعه کد است که مقادیری را به عنوان پارامتر ورودی دریافت نموده و مقداری را به عنوان خروجی می دهد. کد تابع مابین کلمات کلیدی **FUNCTION,** **END FUNCTION** قرار میگیرد هر آنچه که مابین این دو دستور قرار بگیرد به عنوان کد تابع در نظر گرفته می شود. به تابع نامی اختصاص داده می شود که توسط آن نام

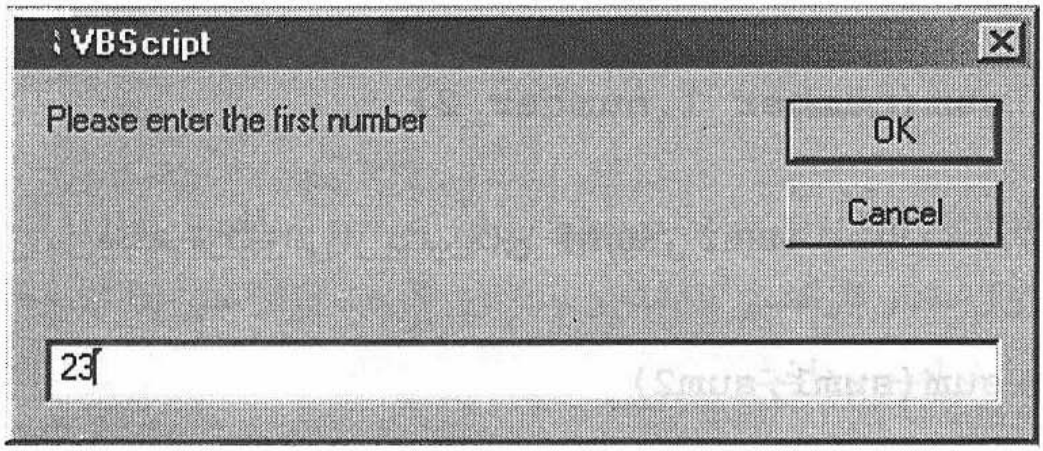
فراخوانی شده و مقدار بازگشتی را بر می گرداند. بـا ذکـر یـک مثـال موضـوع را روشـن تر میشود.

مثال: برنامه ای می نویسیم که دو عدد از ورودی دریافت داشته آنها را با هم جمع نماید.

```
<HTML>
  <HEAD>
    <TITLE>Example of use of FUNCTION</TITLE>
    <SCRIPT LANGUAGE = "VBScript">
    'In the next line begins the function
       FUNCTION add(number_1,number_2)
         add=(CLNG(number_1) + CLNG(number_2))
       END FUNCTION
    'The function terminates here
    </SCRIPT>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
       OPTION EXPLICIT
      'At next line we will define all 3 variables
       used in program.
       DIM    sum1
       DIM    sum2
       DIM    result

       sum1=0
       sum2=0
       result=0

      'The next block will give the first number
       that may be positive
       DO UNTIL (sum1>0)
          sum1=INPUTBOX(" Please enter the first
                        number")
       LOOP
        'The next block will give the second number
         that may be positive
       DO UNTIL (sum2>0)
```

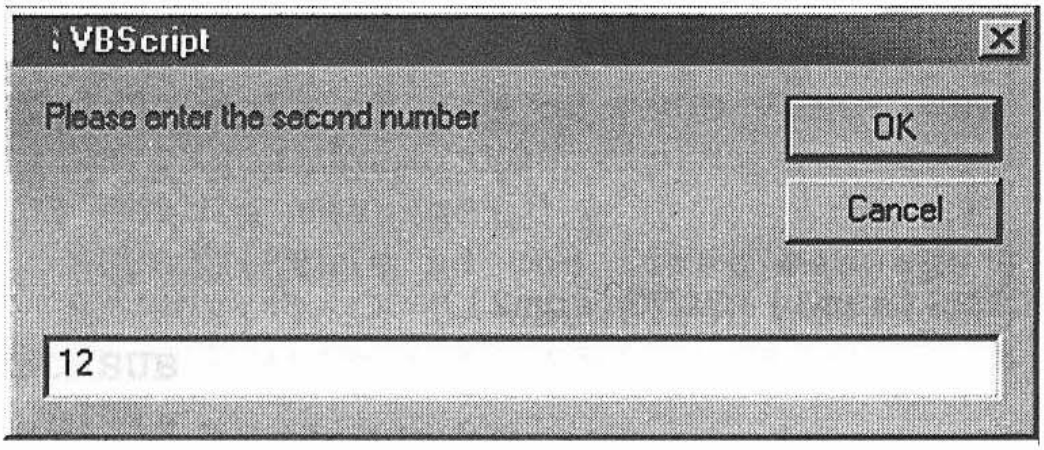
```
            sum2=INPUTBOX(" Please enter the second
                          number")
        LOOP
        result=add(sum1,sum2)
        MSGBOX("The result is " & result)
    </SCRIPT>
  </BODY>
</HTML>
```

حاصل برنامه فوق به این صورت است که از کاربر یک عدد بـه عنـوان عملونـد اول دریـافت میشود :



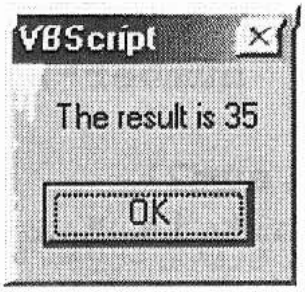
شکل ۱-۸

در این مورد ما عدد ۲۳ را وارد نموده ایم در ادامه عملوند دوم درخواست می شود.



شکل ۲-۸

همانطور که می بینید عدد ۱۲ را وارد نموده ایم. آنگاه برنامه تابع جمع را کـه add نـام دارد فراخوانی کرده این تابع مقدار بازگشتی را محاسبه و به صورت زیر نمایش می دهد:



شکل ۳-۸

حالا جزییات عمل را بررسی می کنیم . ابتدا نام متغیرهایی که به عنوان ورودی به تابع داده می شود. این نام ها دلیلی ندارد که در هنگام فراخوانی با نام آن درون تابع یکی باشد اگر به تعریف تابع دقت نمایید خواهید دید که از متغیرهای number_1,number_2 استفاده شده است.

```
FUNCTION   sum(number_1,number_2)
```

با این حال در هنگام فراخوانی از متغیرهای sum1 و sum2 استفاده شده است.

```
result   = sum(sum1,sum2)
```

و این بدان علت است که سیستم متغیر اول را در هنگام فراخوانی به اولی و دوی را به دومی و به همین منوال بعدیها را هم بر اساس شماره ترتیب جایگذاری مینماید. برای همین برای فراخوانییک تابع کافیست تعداد و ترتیب پارامترها ورودی آن را بدانیم پارامترها باید مابین پرانتز آمده و با کاما از هم جدا شوند چه در هنگام تعریف تابع و چه در هنگام فراخوانی.

در داخل تابع خط زیر نیز جلب توجه مینماید:

```
sum  =  (CLNG(sum1)+CLNG(sum2))
```

از سطر فوق نتیجه می گیریم که مقدار بازگشتی تابع همان مقداری است که به نام تابع نسبت می دهیم(در اینجا sum) با این که بعد از بازگشت این مقدار در داخل متغیرresult ذخیره می شود. همچنین کلمه کلیدی **CLNG( )** را می بینیم که شباهت زیادی به یک تابع دارد ولی داخل کد ما هرگز تعریف نشده است. علت این عدم تعریف آن است که **CLNG**

یک تابع داخلی بوده و احتیاجی به تعریف آن توسط برنامه نویس نیست. عملکرد این تابع توسط مترجم تعیین می شود.کمی جلوتر در مورد توابع داخلی صحبت خواهیم کرد. فعلاً کافیست بدانید که اگر این تابع را به کار نبریم پاسخ سیستم همان چیزی که انتظارش را داریم نخواهد بود و حاصل به جای جمع جبری دو عبارت الحاق آن دو به مشابه الحاق دو رشته خواهد شد.

نکته مهم در مورد توابع این است که توابع میتوانند ورودی نداشته باشند ولی حتماً باید خروجی داشته باشند.

-روالها :

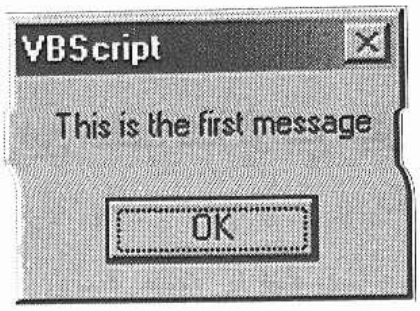
روال (Procedure) به جهت تعریف و روش استفاده شباهت زیادی به تابع دارد. با این تفاوت که روال مقدار بازگشتی نداشته و چند فرق کوچک دیگر نیز با تابع دارد که در این بخش به آنها خواهیم پرداخت.

در ابتدا روالها مابین کلمات کلیدی **SUB** و **ENDSUB** قرار میگیرد و برای فراخوانی کافیست تنها نام آنرا در یک سطر بنویسیم انگار که یک دستور همانند مابقی دستورات است.مثال:

```
<HTML>
  <HEAD>
    <TITLE>Example of use of  SUB FUNCTION</TITLE>
    <SCRIPT LANGUAGE = "VBScript">
      SUB double_message( )
         MSGBOX("This is the first message")
         MSGBOX("This is the first message")
      END SUB
    </SCRIPT>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
      OPTION EXPLICIT
      Double_message
```

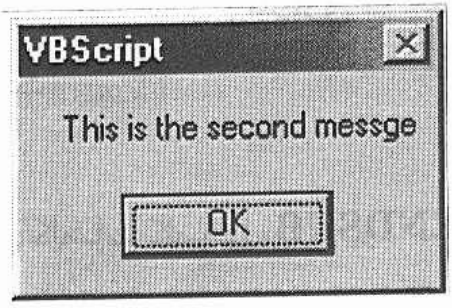
```
    </SCRIPT>
  </BODY>
</HTML>
```

حاصل همانطور که انتظار داشتیم ابتدا یک جعبه پیغام با پیغام زیر نمایش می دهد :



شکل ۴-۸

و سپس بعد از فشار دادن کلید OK خواهیم دید:



شکل ۵-۸

و با فشردن کلید OK برنامه خاتمه می یابد. همانطور که می بینید روال هیچ مقداری را باز نمیگرداند و تنها یک عملیات را انجام می دهد. در این مورد ضمناً روالی را دیدیم که هیچ پارامتر ورودی هم نداشت برای همین در تعریف روال با پرانتز باز و بسته خالی مواجه هستیم. در هنگام فراخوانی هم به همین دلیل بدون پرانتز ظاهر گشته است. اگر روال ما احتیاج به پارامترهای ورودی داشت دو روش برای نوشتن آنها وجود دارد. اولی که ساده ترین حالت هم هست به این صورت است که تمام پارامترها را پشت سر هم مینویسیم و با کاما از هم جدامی کنیم احتیاجی هم به پرانتز نیست :

```
Procedure  par1,par2,par3
```

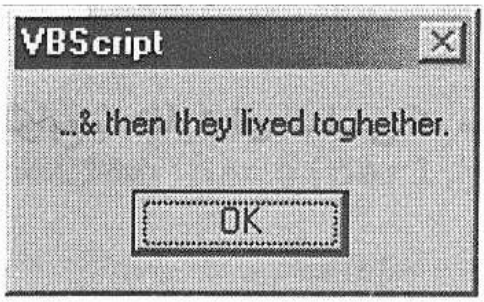
روش دوم که ظاهرش بهتر و ساختیافته تر است به این صورت است که قبل از نام روال کلمه کلیدی **Call** را می نویسیم و تمام پارامترها را با کاما از هم جدا کرده و در درون پرانتز میاوریم :

```
CALL Procedure (par1,par2,par3
```

مثال:

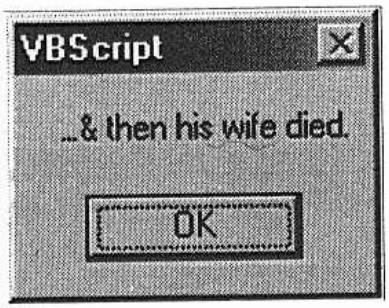
```
<HTML>
  <HEAD>
    <TITLE>
      Example of the models to call a procedure
    </TITLE>
    <SCRIPT LANGUAGE = "VBScript">
       SUB story(final)
        IF (final = "Good" ) THEN
          MSGBOX("... & then they lived
toghether.")
        END IF
        IF (final = "Good" ) THEN

        END IF
      END SUB
    </SCRIPT>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
       story = "Good"
       CALL (story)
    </SCRIPT>
  </BODY>
</HTML>
```

حاصل اجرای کد فوق چنین است:



شکل ۶-۸

و با فشرده شدن کلید OK :



شکل ۷-۸

و با فشار مجدد کلید OK برنامه خاتمه می یابد. با این مثال هردو روش فراخوانی روالها را دیدیم.

حلقه شکن ها:(Breakers)

درست به مشابه حلقه ها می توان اجرای یک روال یا تابع را هم متوقف کرد و کنترل اجرای برنامه را به برنامه اصلی منتقل نمود. برای این کار مجدداً از دستور **EXIT** استفاده می نماییم . اگر مورد یک تابع باشد از دستور **EXIT FUNCTION** و اگر از نوع روال باشد از دستور **EXIT SUB** استفاده می نماییم. به هر حال زیاد درست نیست که از این شکننده ها استفاده نماییم چون باعث به هم ریختگی و پیچیده شدن کد برنامه می شود. در حقیقت تقریباً می توان تمام حالات اتفاقی ممکن را بدون استفاده از این دستورات حل کرد. و تنها در شرایط بسیار خاص و به عنوان آخرین راه حل ومثلاً برای جایی که یک اشکال در برنامه ما را مجبور به خروج از روال طبیعی مینماید می توان استفاده کرد.

نکات مهم :

در هنگام استفاده از توابع به تفاوتهای اساسی که میان روالها و توابع وجود دارد دقت نمایید:

۱- توابعی که توسط برنامه نویس تعریف می شود مابین دستورات **IF , END IF** قرار می گیرند در حالی که روالها مابین دستورات **SUB , END SUB** میشوند

۲- به طور کلی تمام توابع (اعم از داخلی یا تعریفی برنامه نویس) مقداری را به عنوان خروجی برمی گردانند اما روالها به طور کلی مقدار بازگشتی ندارند (منظور مقدار بازگشتی مستقیم است و الا مانند اکثر زبانهای برنامه نویسی دیگر میتوان از متغیرهای ضمنی جهت بازگشت مقدار توسط روالها و یا بازگرداندن مقادیر بیش از یک متغیر توسط یک تابع استفاده نمود. اتفاقا یکی از مهمترین کاربردهای دستور DIM و تعریف متغیرها توسط آن همین است و بسته به موقعیت تعریف متغیر میتوان آن را به حالت عمومی (به عنوان یکی از روشهای ایجاد متغییر ضمنی) یامحلی تعریف نمود. به این معنی که در صورتی که متغیری را خارج از تمام حلقه ها تعریف نماییم آن متغیر به صورت عمومی (Global) تعریف شده و تمامی توابع به آن متغیر دسترسی دارند حال آن که اگر این متغییر درون یک تابع یا روال یا درون بخشی از برنامه اصلی تعریف گردد این متغییر تنها برای آن تابع معتبر بوده و طول عمر آن زمان اجرای این تابع است و پس از اتمام تابع این متغیر کلاً از بین می رود و حتی با ورود مجدد به تابع یا روال نمی توان مقدار قبل از خروج پیشین را بازیابی نمود. این مفهوم را گسترهٔ دید متغیر(Variable Scope) می نامیم. (برای اطلاعات بیشتر در زمینه انواع متغیرها میتوانید به کتابهای برنامه نویسی در VISUAL BASIC مراجعه نمایید.)

۳- در هنگام فراخوانی توابع باید پارامترهای آن را درون پرانتز نوشت و به کمک علامت کاما از هم جدا کرد و به دلیل بازگشت مقدار توسط تابع باید حتماً نام تابع را به مشابه یک متغیر یک طرفه استفاده کرد یعنی در یکی از دو حالت زیر:

```
variable = function_name(par1,par2
```

و یا:

```
comparator (logical  or mathematical operator (
            function_name () ))
```

که در حالت اول خروجی تابع را درون یک متغیر ریخته ایم و در حالت دوم با خروجی تابع به مشابه یک متغیر رفتار میکنیم و بر روی آن عملگرهای منطقی، ریاضی اعمال کرده و حاصل را در متغیر ریخته یا ازآن در دستورات مقایسه ای نظیر استفاده نماییم در حالی که برای فراخوانی روالها از دو روش ساده و یا با همراهی دستور CALL استفاده نمود.

حال که موضوع روالها و توابع را کاملاً متوجه شده ایم وقت آن است که به توابع داخلی بپردازیم.

فصل نهم:

توابع داخلی VBScript

در VBScript کتابخانه ای از توابع داخلی وجود دارد که کاربر می تواند در صورت نیاز از آنها استفاده نماید. و چون از قبل تعریف شده اند احتیاج به بارگذاری آنها در حافظه نیست و مترجم هرجا که لازم بداند آنها را وارد حافظه میکند. و برنامه نویس تنها باید آنها را با پارامترهای لازم فراخوانی نماید. این توابع با توجه به نوع داده ای که با آن کار میکنند به دو دسته تقسیم می شوند. در این فصل تمام این توابع را به ترتیب حروف الفبا معرفی می نماییم و کاربرد هرکدام را شرح می دهیم در پایان فصل هم مثالهایی از استفاده از تعدادی از آنها را آورده ایم.

۱- ABS (*number*)

به عنوان ورودی یک عدد (یا متغیری با محتوای عددی) دریافت نموده و به عنوان مقدار بازگشتی قدر مطلق عدد (عدد بدون علامت) را بازمیگرداند.

۲- ARRAY (*Element_1[,Element_2[...[,Element_n]]]*)

به عنوان خروجی ماتریس (آرایه) ای با المانهای ورودیش که توسط کاما از هم جدا شده اند بر می گرداند. و اگر پارامتر به آن داده نشود ماتریس صفر بعدی !!! برمی گرداند. (توجه نمایید که دو روش برای ساختن ماتریس وجود دارد یکی روشی که در فصل چهارم به آن پرداختیم و در طی آن از دستور DIM برای ایجاد یک ماتریس استفاده نمودیم و روش دوم استفاده از این تابع می باشد. اما در صورت استفاده از تابع ما در اصل شی آرایه را تعریف مینماییم که در اصل همان آرایه است ولی به دلیل داشتن خصوصیات شیی ظاهر مقبولتری دارد.)

۳-الف: ASC (*character*)

به عنوان خروجی کد اسکی کاراکتر ورودی را بر می گرداند. و اگر به عنوان آرگومان به آن رشته پاس داده شود کد اسکی اولین کاراکتر پاس داده شده را بر میگرداند. و اگر داده ای از نوع **NULL** بگیرد به عنوان خروجی هم مقدار **NULL** را بر می گرداند.

۳-ب: ASCB (*character*)

این تابع مشابه ASC می باشد ولی به جای کد اسکی اولین کاراکتر رشته اولین بایت رشته را برمی گرداند.

۳-ج: ASCW (*character*)

این تابع هم مشابه ASC می باشد ولی به جای کد اسکی اولین کاراکتر رشته اولین کلمه (دوبایت) رشته را برمی گرداند در اصل برای رشته هایی بکار میرود که از کاراکتر های ۳۲ بیتی کد ANSI تشکیل شده اند

۴- ATN (*number*)

به عنوان ورودی یک عدد (یا متغیری محتوی آن عدد) را به عنوان زاویه ای با مقیاس رادیان دریافت می نماید و آرکتانژانت (Arctangante) زاویه را بر می گرداند . عدد ورودی رنج خاصی ندارد ولی خروجی حتماً عددی مابین , pi/2 pi/2 - می باشد. برای تبدیل مقادیر زاویه هایی که در مقیاس درجه می باشند به مقیاس رادیان کافیست مقدار درجه را در pi/180 ضرب نمایید

۵- CBOOL (*number*)

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و مقدار منطقی مرتبط را برمیگرداند. **FALSE** اگر عدد مورد نظر صفر است و **TRUE** برای هر عدد غیر صفر. در صورتی که مقدار متغیر قابل تبدیل به عدد نباشد با پیغام خطا مواجه خواهید شد.

۶- CBYTE (*number*)

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و آن را به فرم بایتی تبدیل می نماید.(برای شرح بیشتر به مثالها مراجعه نمایید.)

۷- CCUR (*number*)

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و آن را به فرمت Currency تبدیل می نماید .

۸- CDATE (*String*)

به عنوان ورودی رشته ای حرفی- عددی نشان دهنده تاریخ دریافت می نماید و آن را به فرمت DATE تبدیل می نماید (به پیوست ۱ مراجعه نمایید)

۹- CDBL *(number)*

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و آن را به عددی از نوع Double تبدیل می نماید (به پیوست ۱ مراجعه نمایید)

۱۰-الف: CHR *(number)*

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و کاراکتر معادل کد اسکی داده شده را باز می گرداند.

۱۰-ب: CHRB *(number)*

این تابع مانند تابع **CHR** می باشد با این تفاوت که الزاماً یک عدد یک بایتی را به کاراکتر معادل تبدیل می نماید (برای کد اسکی)

۱۰-ج: CHRW *(number)*

همانند **CHRB** است ولی برای کد ANSI و ورودی آن یک عدد ۳۲ بیتی میباشد.

۱۱- CINT *(number)*

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و آن را به عددی با فرمت INT تبدیل می نماید. (به پیوست ۱ مراجعه نمایید)(این تابع با توابع **FIX** , **INT** و **ROUND** تفاوت دارد برای درک این تفاوت اعداد اعشاری مختلف را با این توابع تبدیل نمایید.)

۱۲- CLNG *(number)*

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و آن را به فرمت LONG تبدیل می نماید. (به پیوست ۱ مراجعه نمایید)

۱۳- COS (*number*)

به عنوان ورودی یک عدد (یا متغیری محتوی آن عدد) را به عنوان زاویه ای با مقیاس رادیان دریافت می نماید و کسینوس (Cosine) زاویه را بر میگرداند. برای تبدیل مقادیر زاویه هایی که در مقیاس درجه می باشند به مقیاس رادیان کافیست مقدار درجه را در pi/180 ضرب نمایید.

۱۴- CREATEOBJECT (*class*)

جهت ایجاد یک شی ازخانواده أی که میتواند شی سازی انجام دهد از این تابع استفاده می شود ورودی تابع نام متغییری است که میخواهیم از نوع آن Class شی به این نام ایجاد نماییم. به عنوان مثال برای ایجاد یک Document جدید از نوع Microsoft Excel میتوانیم از دستورات زیر استفاده نماییم:

```
Dim ExcelSheet
Set ExcelSheet = CreateObject ("Excel.Sheet")
```

در این مثال شیئی از نوع Excel به صورت یک Sheet جدید ایجاد نموده ایم (الزاماً باید بر روی سیستم کاربر نرم افزار Excel نصب باشد تا این دستور از Excel به عنوان سرور خود استفاده نماید.) حال درون برنامه به صورت زیر به اجزای شی ایجاد شده دسترسی خواهیم داشت:

```
' Make Excel visible through the Application
object.ExcelSheet.Application.Visible = True

' Place some text in the first cell of the sheet.
ExcelSheet.Cells(1,1).Value="column A, row 1"

' Save the sheet.
ExcelSheet.SaveAs "C:\DOCS\TEST.XLS"

' Close Excel with the Quit method on the
Application object.ExcelSheet.Application.Quit
```

```
' Release the object variable.
Set ExcelSheet = Nothing
```

پارامتر ورودی این تابع به صورت **Servername . typename** می باشد که در آن Servername نام برنامه أی است که میخواهیم شی را از آن مشتق نماییم. (در مثال فوق Excel)و typename نوع شیی است که میخواهیم از این خانواده ایجاد شود(در مثال فوق یک New Spred Sheet)

۱۵- CSNG (*number*)

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و آن را به فرمت SINGLE تبدیل می نماید. (به پیوست ۱ مراجعه نمایید)

۱۶- CSTR (*argument*)

به عنوان آرگومان ورودی یک نوع داده ای (یا متغیر محتوی آن) را دریافت می نماید و آن را به فرم رشته ای حرفی- عددی تبدیل می نماید. این تبدیل با توجه به جدول زیر انجام خواهد گرفت:

نوع داده ای	مقدار تبدیل شده توسط تابع
BOOLEAN	رشته أی محتوی کلمه TRUE یا FALSE
DATE	رشته أی به فرمت DATE در سیستم
NULL	خطای زمان اجرا.
EMPTY	یک رشته خالی ""
ERROR	رشته أی حاوی کلمه ERROR که در جلوی آن شماره خطا می آید.
اعداد دیگر	یک رشته با محتوای عددی.

DATE () -۱۷

ورودی نداشته و به عنوان خروجی تاریخ جاری سیستم (تاریخ روز) را بر می گرداند .

DATEADD (*intervals , value , date*) -۱۸

این تابع حاصل جمع یک مدت زمان با تاریخ داده شده را بر می گرداند وهر سه ورودی آن اجباری هستند. و در آن ***intervals*** رشته ای است که نوع عددی را که می خواهیم به تاریخ اضافه نماییم را مشخص می نماید که می تواند یکی از مقادیر زیر باشد :

رشته	معنی
yyyy	سال به صورت کامل
q	فصل
m	ماه
d	روز
w	هفته
ww	شماره هفته أی از سال
h	ساعت
n	دقیقه
s	ثانیه

Value مقدار عددی مورد نظر برای جمع شدن است این مقدار می تواند مثبت (برای زمانهای آینده) یا منفی (برای اشاره به ماضی) باشد.

Date تاریخی است که می خواهیم تابع عملیات جمع را روی آن انجام دهد.

به عنوان مثال اگر کد زیر را بنویسیم:

newyear = DATEADD ("yyyy",1,10-02-1996)

حاصل **10-02-1997** خواهد شد . این تابع خودش سالهای کبیسه را نیز در نظر می گیرد. (توجه نمایید که تاریخ محاسبه شده تنها برای سالهای میلادی می باشد و متاسفانه برای تاریخ شمسی و قمری کارایی ندارد حرفه ایها توجه نمایند که نباید برای کدی که قرار است بر روی وب و در سطح گسترده منتشر شود از ACTIVE X های فارسی و یا dll ها استفاده نمود. چون دلیلی برای اجرا شدن الزامی آنها روی هر سیستم ناشناخته ای وجود ندارد. در این گونه موارد تنها راه حل نوشتن تابع تبدیل به همراه کد است).

۱۹- **DATEDIFF (*intervals, date_1, date_2, [firstdayofweek, [firstweekofyear]]*)**

اختلاف میان دو تاریخ ورودی را محاسبه نموده و حاصل را با فرمت مشخص شده در پارامتر ***intervals*** برمیگرداند . به همین دلیل مقدار بازگشتی تنها یک عدد خواهد بود. مقادیر مجاز برای ***interval*** به مانند تابع قبلی (Dateadd) می باشد.

پارامتر ***firstdayofweek*** یک پارامتر اختیاری می باشد که روز اول هفته را مشخص می نماید در صورتی که داده نشود یکشنبه در نظر گرفته خواهد شد پارامترهای مجاز برای آن عبارتند از:

ثابت	معادل عددی	توضیحات
vbUseSystem	۰	مقدار پیش فرض از سیستم گرفته میشود
vbSunday	۱	یک شنبه
vbMonday	۲	دو شنبه
vbTuesday	۳	سه شنبه
vbWednesday	۴	چهار شنبه
vbThursday	۵	پنج شنبه

vbFriday	۶	جمعه
vbsaturday	۷	شنبه

پارامتر ***firstweekofyear*** هم پارامتری اختیاری می باشد که هفته اول سال را معین میسازد و در صورتی که داده نشود هفته اول سال هفته ای خواهد بود که اول ژانویه در آن قرار دارد. پارامترهای زیر برای آن قابل قبول میباشند.

ثابت	معادل عددی	توضیحات
vbUseSystem	۰	از حالت سیستم پیش فرض استفاده می نماید
vbFirstJan1	۱	با هفته أی که اول ژانویه در آن قرار دارد آغاز می شود
vbFirstFourDays	۲	با هفته أی که حداقل چهار روز آن در سال جدید واقع باشد.
vbFirstFullWeek	۳	با اولین هفته ای که تمام آن در سال جدید قرار دارد آغاز می شود.

۲۰-DATEPART*(interval, date[, firstdayofweek[, firstweekofyear]])*

این تابع بخش ***Interval*** از***date*** ورودی را به عنوان خروجی باز می گرداند تمامی پارامترها از جدول تابع **DATEDIFF** پیروی مینمایند

۲۱-DATESERIAL *(year, month, day)*

مقداری ازنوع **Date** با کمک پارامترهای ورودی ساخته و باز می گرداند و در آن ***year*** سالی مابین ۱۰۰ و ۹۹۹۹ می باشد. ***month*** ماهی مابین ۱ و ۱۲. ***day*** هم روزی بین ۱ و ۳۱.

۲۲-DATEVALUE *(string)*

از رشته ورودی بخش تاریخ و ساعت را جدا کرده و به صورت پارامتری از نوع Date بازمیگرداند. تمام فرمتهای تاریخ دهی به سیستم در رشته قابل قبول می باشد. اگر در رشته ورودی سال ذکر نشود سال جاری سیستم در نظر گرفته خواهد شد.

۲۳- DAY *(date)*

این تابع به عنوان ورودی یک تاریخ گرفته و عددی مرتبط با روز مشخص شده در ماهی که به عنوان تاریخ ورودی داده شده برمی گرداند. اگر date حاوی Null باشد مقدار Null بر می گرداند.

۲۴- EXP *(number)*

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و به عنوان خروجی معادل عدد نپر (e) به توان عدد داده شده را باز می گرداند . حداکثر مجاز برای عدد 709.782712893 می باشد.

۲۵-FILTER *(InputStrings, Value[, Include[, Compare]])*

این تابع رشته ورودی را به دنبال مصادیق Value جستجو میکند و با توجه به پرچم Include اگر این پرچم TRUE بود مصادیق را به صورت جداگانه در المانهای یک تابع(به عنوان خروجی باز خواهد گشت) ذخیره می نماید ودر صورت FALSE بودن کلماتی را که صدق نکند باز می گرداند پارامتر Compare از جدول زیر تبعیت می نماید:

ثابت	معادل	توضیحات
vbBinaryCompare	۰	(پیش فرض) برای مقایسه باینری
vbTextCompare	۱	مقایسه از نوع متنی

۲۶- FIX (number)

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و جزء صحیح عدد داده شده را با حذف مقدار اعشار باز می گرداند اگر عدد داده شده منفی باشد اولین عدد منفی مساوی یا بزرگتر از عدد داده شده را باز میگرداند. (به دستور INT هم توجه نمایید.)

۲۷- FORMATCURRENCY(number[, NumDigitsAfterDecimal [, IncludeLeadingDigit [, UseParensForNegativeNumbers [, GroupDigits]]]])

این تابع جهت تبدیل اعداد به فرم نمایش پول می باشد یعنی آنها را به فرم نمایشی اعداد در مسایل بازرگانی و بانکی تبدیل می نماید. آرگومانهای آن عبارتند از:

number: عددی (یا متغیر عددی) که میخواهیم تبدیل بر روی آن صورت پذیرد

NumDigitsAfterDecimal: تعداد ارقام اعشار را مشخص می نماید (اختیاری با مقدار پیش فرض ۱- به معنی تنظیمات سیستم کاربر.)

IncludeLeadingDigit: پارامتری منطقی سه حالته معرف آن که آیا می خواهیم عدد را با گذاشتن تعداد صفر لازم در پشت آن تکمیل کند یا خیر. به عنوان مثال اگر در پارامتر قبلی تعداد ارقام ٤ عنوان شود اما عدد داده شده تنها دو رقم غیر اعشار داشته باشد به تعداد دو عدد صفر در پشت آن قرار خواهد گرفت تا مجموع ارقام غیر اعشار ٤ شود. .(اختیاری بوده از جدول پیروی می نماید)

UseParensForNegativeNumbers: پارامتری منطقی سه حالته معرف آنکه آیا میخواهیم اعداد منفی درون پرانتز به نمایش در آیند یا خیر.(اختیاری بوده از جدول پیروی می نماید)

GroupDigits: پارامتری منطقی سه حالته معرف آنکه آیا میخواهیم اعداد به صورت سه تا سه تا برای هزارگان میلیونگان و ... از هم جدا شوند یا خیر؟ (اختیاری بوده از جدول تبعیت می نماید.)

هر سه پارامتر اخیر از جدول زیر برای مقدار منطقی استفاده مینمایند:

ثبات	معادل	توضیحات
TristateTrue	-۱	TRUE
TristateFalse	۰	FALSE
TristateUseDefault	-۲	مقدار پیش فرض سیستم کاربر

(توصیه می شود به دلیل آنکه کد شما قرار است بر روی کامپیوترهایی که از نظر شما ناشناخته هستند و سلیقه کاربران آنها را نمیدا نید همواره از حالت **TristateUseDefault** استفاده نمایید تا کاربر نهایی که صفحه شما را میبیند صفحه ای مطابق سلیقه خودش مشاهد ه نماید)

۲۸-FORMATDATETIME *(string [,NamedFormat])*

رشته (یا متغیر رشته أی) ورودی را به فرم Date تبدیل می نماید.

پارامتر ***NamedFormat*** اختیاری بوده و یکی از موارد زیر میتواند باشد:

ثبات	معادل	توضیحات
vbGeneralDate	۰	(پیش فرض) در صورت وجود تاریخ و ساعت را تواماً نمایش می دهد .
vbLongDate	۱	تاریخ را با فرمت Long تعریف شده در سیستم کاربر نمایش میذهد
vbShortDate	۲	تاریخ را با فرمت Short مطابق تعریف سیستم کاربر نمایش خواهد داد
vbLongTime	۳	ساعت را با فرمت تنظیم شده بر روی سیستم کاربر نمایش می دهد
vbShortTime	۴	ساعت را به صورت کوتاه شدهٔ ۲٤ ساعته به صورت hh:mm نمایش خواهد داد.

۲۹- **FORMATNUMBER(*number[, NumDigitsAfterDecimal [, IncludeLeadingDigit [, UseParensForNegativeNumbers [, GroupDigits]]]]*)**

این تابع عملکردی عیناً مشابه تابع **FORMATCURRENCY** دارد با این تفاوت که عدد داده شده را به صورت کلی به فرمت داده شده تبدیل می نماید بدون آنکه به فرمت پولی خاصی نسبت داده شود.

۳۰- **FormatPercent(*Expression[, NumDigitsAfterDecimal [, IncludeLeadingDigit [, UseParensForNegativeNumbers [, GroupDigits]]]]*)**

رشته به صورت درصد به همراه کاراکتر % باز می گرداند. و در آن:

Expression: ورودی است که میخواهیم به صورت درصد تبدیل شود.

NumDigitsAfterDecimal: تعداد ارقام اعشار را مشخص می نماید (اختیاری با مقدار پیش فرض ۱- به معنی تنظیمات سیستم کاربر.)

IncludeLeadingDigit: پارامتری منطقی سه حالته معرف آن که آیا می خواهیم عدد را با گذاشتن تعداد صفر لازم در پشت آن تکمیل کند یا خیر. به عنوان مثال اگر در پارامتر قبلی تعداد ارقام ٤ عنوان شود اما عدد داده شده تنها دو رقم غیر اعشار داشته باشد به تعداد دو عدد صفر در پشت آن قرار خواهد گرفت تا مجموع ارقام غیر اعشار ٤ شود. .(اختیاری بوده از جدول پیروی می نماید)

UseParensForNegativeNumbers: پارامتری منطقی سه حالته معرف آنکه آیا میخواهیم اعداد منفی درون پرانتز به نمایش در آیند یا خیر.(اختیاری بوده از جدول پیروی می نماید)

GroupDigits: پارامتری منطقی سه حالته معرف آنکه آیا میخواهیم اعداد به صورت سه تا سه تا برای هزارگان میلیونگان و ... از هم جدا شوند یا خیر؟ (اختیاری بوده از جدول تبعیت می نماید.)

هر سه پارامتر اخیر از جدول زیر برای مقدار منطقی استفاده مینمایند:

ثبات	معادل	توضیحات
TristateTrue	-۱	TRUE
TristateFalse	۰	FALSE
TristateUseDefault	-۲	مقدار پیش فرض سیستم کاربر

(توصیه می شود به دلیل آنکه کد شما قرار است بر روی کامپیوترهایی که از نظر شما ناشناخته هستند و سلیقه کاربران آنها را نمیدا نید همواره از حالت **TristateUseDefault** استفاده نمایید تا کاربر نهایی که صفحه شما را میبیند صفحه ای مطابق سلیقه خودش مشاهد ه نماید)

۳۱- HEX (*number*)

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و معادل تبدیل شده عدد در مبنای ۱۶ را باز می گرداند. این تبدیل با توجه به جدول زیر صورت می گیرد:

نوع متغیر	خروجی تابع HEX
Null	**Null**
Empty	صفر
هر عدد دیگر	معادل مبنای ۱۶ عدد.

۳۲- HOUR (*hour*)

به عنوان آرگومان ورودی یک متغیر محتوی ساعتی با فرمت hh:mm:ss می گیرد و ساعت را به صورت عددی مابین ۰ تا ۲۳ برمی گرداند . در صورتی که ورودی Null باشد. خروجی هم Null خواهد بود.

۳۳- INPUTBOX(*prompt[,title][,default][,xpos][,ypos][, helpfile, context]*)

در فصل ٦ به طور مفصل در مورد این تابع صحبت نمودیم اما به طور خلاصه:

PROMPT: پیغامی است که به عنوان سوال مطرح خواهد شد اندازه جعبه با توجه به طول این پیام تنظیم می شود میتوان با کمک ترکیب دستورات (**CHR(10) & CHR(13)**) میتوان پیغام های چند خطی ایجاد کرد.

Tittle:پیغامی است که در سر تیتر جعبه ظاهر می شود (اختیاری)

default: مقدار پیش فرضی است که در جعبه متن به نمایش در خواهد آمد(اختیاری)

xpos: محل x قرارگرفتن نقطه سمت چپ بالای جعبه به پیکسل (اختیاری پیش فرض قرارگرفتن جعبه در وسط صفحه)

ypos: محل y قرارگرفتن نقطه سمت چپ بالای جعبه به پیکسل (اختیاری پیش فرض قرار گرفتن جعبه در وسط صفحه)

HelpFile , context : پارامترهایی اختیاری که در صورت وجود هر دو باعث پدید آمدن یک دکمه Help جعبه خواهد شد و در آن ***Helpfile*** آدرس فایلی است که به عنوان کمک در نظر گرفته شده و ***context*** عبارت از شماره أی است که از فایل ***Helpfile*** برای این جعبه به عنوان کمک در نظر گرفته شده است.

۳۴- INSTR (*[start], string1, string2, [compare]*)

این تابع رشته اول را به دنبال اولین محل وقوع رشته دوم جستجو میکند پارامتر ***start*** اختیاری می باشد و مشخص کننده آن است که عملیات جستجو باید ازکدام کاراکتر رشته اول شروع شود اگر مشخص نشود به صورت پیش فزض ابتدای رشته در نظر گرفته می شود (برنامه نویسان جاوا توجه نمایند که در VBScript همانند اکثریت زبانهای دیگر با رشته به مشابه آرایه ای از کاراکترها برخورد می شود لذا می توان به هر کاراکتر رشته به صورت

مجزا دسترسی داشت ضمناً به همین دلیل هیچ تفاوتی میان کاراکترهای کد اسکی وجود ندارد و حتی اگر برنامه شما به فارسی نوشته شده باشد و بر روی سیستم کاربر به دلیل تفاوت Code page امکان دیدن صحیح آن نباشد این تابع کارش را به درستی انجام خواهد داد.)

پارامتر ***compare*** پارامتری اختیاری می باشد که نوع مقایسه را تعیین مینماید و یکی از دو حالت زیر است:

ثابت	معادل	توضیحات
vbBinaryCompare	۰	(پیش فرض) برای مقایسه باینری
vbTextCompare	۱	مقایسه از نوع متنی

خروجی تابع از جدول زیر پیروی می نماید:

ورودی	خروجی تابع
رشتهٔ String1 به طول صفر	صفر
رشته String1 Null باشد.	Null
رشته String2 به طول صفر	start
رشته String2 Null باشد.	Null
String2 پیدا نشود.	صفر
String2 درون String1 پیدا شود.	اولین نقطه شروع پیدایش رشته دوم.
طول String2 < start	صفر

۳۵- INSTRREV (*string1, string2, [start, [compare]]*)

این تابع همان عمل تابع بالا را انجام می دهد با این تفاوت که جستجورا از انتها به ابتدا آغاز میکند.

۳۶- INT (*number*)

به عنوان آرگومان ورودی عدد (یا متغیر محتوی عدد) دریافت می نماید و به عنوان خروجی جزء صحیح عدد را باز می گرداند و در صورتی که عدد ورودی منفی باشد بزرگترین عدد مساوی یا کوچکتر از عدد ورودی را بازمیگرداند. (با دستور **FIX** مقایسه نمایید)

۳۷- ISARRAY (*variable*)

به عنوان ورودی نام یک متغیر را می گیرد چک میکند که آیا متغیر داده شده آرایه است یا خیر. صحت یا عدم صحت را به عنوان یک مقدار **Boolean** بازمی گرداند.

۳۸- ISDATE (*variable*)

به عنوان ورودی نام یک متغیر را می گیرد چک میکند که آیا متغیر داده شده از نوع تاریخ است یا خیر. صحت یا عدم صحت را به عنوان یک مقدار **Boolean** بازمی گرداند.

۳۹- ISEMPTY (*variable*)

به عنوان ورودی نام یک متغیر را می گیرد چک میکند که آیا متغیر داده شده مقداردهی اولیه شده است یا خیر. صحت یا عدم صحت را به عنوان یک مقدار **Boolean** بازمی گرداند.

۴۰- ISNULL (*variable*)

به عنوان ورودی نام یک متغیر را می گیرد وخروجی آن یک مقدار **Boolean** است. **TRUE** اگر متغیر داده شده حاوی **NULL** باشد و **FALSE** در غیر این صورت.

۴۱- ISNUMERIC (*variable*)

به عنوان ورودی نام یک متغیر را می گیرد وبه عنوان خروجی مقدار منطقی **TRUE** را در صورتی که متغیر داده شده از نوع عددی باشد و در غیر این صورت **FALSE** برمی گرداند.

۴۲- ISOBJECT (*variable*)

به عنوان ورودی نام یک متغیر را می گیرد وبه عنوان خروجی مقدار منطقی **TRUE** را در صورتی که متغیر داده شده از نوع شیء باشد و در غیر این صورت **FALSE** برمی گرداند.

***** توجه نمایید که در V.B. نوع داده ای تعریف نمی شود و درست درهنگام مقداردهی یک متغیر نوع داده ای آن تعیین می شود که این نوع تعین شده هم می تواند در طول اجرای برنامه تغییر نماید لذا تنها راه تشخیص نوع داده ای فعلی متغیر استفاده از توابع فوق و تابع TYPENAME می باشد. (برای اطلاعات بیشتر به فصل سوم بخش متغیرها مراجعه نمایید.)**

۴۳- JOIN *(Array [, Delimiter])*

این تابع به عنوان ورودی یک آرایه (معمولاً متغیر آرایه أی) را گرفته و به عنوان خروجی رشته أی پیوسته از المانهای درون تابع بوجود می آورد. پارامتر ***Delimiter*** مبین نوع کاراکتر یا رشته أی است که میخواهیم درون رشته مابین المانهای تابع قرار گیرد مقدار پیش فرض آن یک کاراکتر خالی می باشد. در صورتی که به جای ***Delimiter*** یک رشته تهی "" قرار گیرد تمام المانهای تابع به هم چسبیده درون تابع ظاهر خواهند شد.

LCASE (*string*) -۴۴

به عنوان ورودی یک رشته (یا متغیر محتوی رشته) را می گیرد و آن را به صورت رشته ای با حروف کوچک تبدیل می نماید. (معمولاً از این تابع و برادرش **UCASE** برای مقایسه کلمات استفاده می شود زیرا بدون تغییر جمله اصلی جمله أی با تمام کاراکترهای کوچک (بزرگ) پدیدار میگردد. این تابع مخصوص کد اسکی کاراکترهای لاتین است و به صورت اتوماتیک بر روی سایر کاراکترها نمیتواند اثر داشته باشد.) ورودی Null خروجی Null در پی خواهد داشت.

LEFT (*string, length*) -۴۵

جزئی از رشته به طول **length** را از سمت چپ رشته جدا میکند و به عنوان خروجی بازمی گرداند . البته رشته اصلی دست نخورده باقی میماند .

LEN (*string*) -۴۶

به عنوان ورودی یک رشته می گیرد وطول رشته را به عنوان خروجی بازمی گرداند.

LOG (*number*) -۴۷

این تابع به عنوان آرگومان ورودی یک عدد (یا متغیر محتوی آن عدد) را که باید عددی بزرگتر از صفر باشد می گیرد و به عنوان مقدار بازگشتی لگاریتم آن عدد را در پایه عدد نپر (e) محاسبه می نماید. برای محاسبه لگاریتم اعداد در سایر پایه ها از فرمول زیر استفاده نمایید.

$$Log_n(x) = Log(x) / Log(n)$$

برای سادگی بهتر است برای پایه های مختلف تابع زیر را به برنامه اضافه نمایید:

```
Function Logn(X,n)
    Logn = Log(X) / Log(n)
End Function
```

LTRIM *(string)* -۴۸

به عنوان آرگومـان ورودی رشـته (یـا متغـیر حـاوی رشـته) گرفتـه و تمـام فواصل خالی کـه در سـمت راسـت آن باشـد را حـذف مـی نمـاید و بـه عنـوان خروجی بازمی گرداند (رشته اصلی دست نخورده باقی میماند)

MID *(string, start, length)* -۴۹

به عنوان آرگومان ورودی یک رشته (یا متغـیر حـاوی رشـته) و عـدد (یـا متغیر عددی) نقطه شروع وعددی (یا متغیر عددی) معرف طـول مـورد نظـر را می گیرد و از داخل رشته داده شده ازنقطه آغاز تعریف شده به طول داده شده جدا کرده و باز می گرداند. (رشته اصلی تغییر نخواهد کرد)

MINUTE *(hour)*-۵۰

ایـن تـابع بـه عنـوان ورودی یـک متغـیر بـا مقـدار سـاعت بـا فرمـت hh:mm:ss و دقیقه را به صورت عددی ما بین ۰ تا ۵۹ بازمی گرداند .

MONTH *(date)*-۵۱

این تابع به عنوان ورودی رشته تاریخ (یا متغیر حاوی تاریخ) را گرفته و مـاه مندرج در تاریخ ورودی را بازمی گرداند .

MONTHNAME *(mounth, [abreviation])*-۵۲

این تابع به عنوان ورودی عددی مابین ۱ تا ۱۲ (یـا متغـیر محتـوی آن عـدد) گرفته و نام مـاه ورودی را برمـی گردانـد. اگر بـه جـای پـارامتر ***abreviation*** مقدار منطقی **TRUE** را بگذاریم نام ماه بازگشـتی بـه صـورت مخفـف خواهـد بود. (نام ماه وابسته به سیستم عامل کاربر نهایی اجرا کننده برنامه می باشـد و اگر فرضاً سیستم عامل Windows وی برای نمـایش ماهـهای شمسـی تنظیـم شده باشد خروجی تابع ماههای شمسی بوده ولی مقدار بـازگشـتی همیـن تـابع

برای کاربری که از Internet exploroer فرانسوی استفاده می نماید به زبان فرانسه خواهد بود.)

۵۳- MsgBox(*prompt[, buttons][, title][, helpfile, context]*)

فصل دهم به طورکامل به این تابع اختصاص دارد. به صورت خلاصه:

PROMPT: پیغامی است که به نمایش در خواهد آمد اندازه جعبه با توجه به طول این پیام تنظیم می شود میتوان با کمک ترکیب دستورات **(CHR(10) & CHR(13))** پیغام های چند خطی ایجاد کرد.

buttons: نوع جعبه و کلیدها و احیاناً شکلهایی را که میخواهیم در جعبه ظاهر شود را بر طبق جدول زیر انتخاب می نماید:

ثابت	معادل	توضیحات
vbOkOnly	۰	(مقدار پیش فرض) تنها یک دکمه Ok نمایش می دهد .
vbOkCancel	۱	یک دکمه Ok و یک دکمه Cancel
vbAbortRetryIgnore	۲	سه دکمهٔ Retry , Abort Ignore را ظاهر میسازد.
vbYesNoCancel	۳	سه دکمهٔ yes, no , cancel را ظاهر میسازد.
vbYesNo	۴	دو دکمه yes , no
vbRetryCancel	۵	دو دکمه Retry , Cancel
vbCritical	۱۶	به فصل ۱۰ مراجعه نمایید.
vbQuestion	۳۲	به فصل ۱۰ مراجعه نمایید
vbExclamation	۴۸	به فصل ۱۰ مراجعه نمایید
vbQuestion	۶۴	به فصل ۱۰ مراجعه نمایید
vbDefaultButton1	۰	اولین کلید پیش فرض است

vbDefaultButton2	۲۵۶	دومین کلید پیش فرض است
vbDefaultButton3	۵۱۲	سومین کلید پیش فرض است
vbDefaultButton4	۷۶۸	چهارمین کلید پیش فرض است
vbApplicationModal	۰	(پیش فرض) کاربر باید قبل از ادامه اجرا به جعبه پاسخ دهد.
vbSystemModal	۴۰۹۶	تمام برنامه ها تا زمان پاسخ کاربر مختل میشوند

Tittle:پیغامی است که در سر تیتر جعبه ظاهر می شود (اختیاری)

HelpFile , context : پارامترهایی اختیاری که در صورت وجود هر دو باعث پدید آمدن یک دکمه Help جعبه خواهد شد و در آن Helpfile آدرس فایلی است که به عنوان کمک در نظر گرفته شده و context عبارت از شماره أی است که از فایل Helpfile برای این جعبه به عنوان کمک در نظر گرفته شده است.

خروجی های این تابع به شرح زیر است:

ثابت	معادل عددی	توضیحات
vbOk	۱	Ok
vbCancel	۲	Cancel
vbAbort	۳	Abort
vbRetry	۴	Retry
vbIgnore	۵	Ignore
vbYes	۶	Yes
vbNo	۷	No

کلید Help که در صورت ورود دو پارامتر آخر ظاهر می شود خود اجرایی می باشد لذا احتیاجی به بازگشت داده شدن آن نیست.

NOW () -۵۴

این تابع ورودی نداشته و به عنوان خروجی تاریخ و ساعت سیستم را بر میگرداند.

OCT (*number*) -۵۵

این تابع به عنوان آرگومان ورودی یک عدد (یا متغیر محتوی آن عدد) دهدهی گرفته و آن را به مبنای ۸ تبدیل می نماید. خروجی از جدول زیر پیروی می نماید:

نوع متغیر	خروجی تابع OCT
Null	Null
Empty	صفر
هر عدد دیگر	معادل مبنای ۸ عدد.

REPLACE (*string1, string2, replace_with, [start , [repeat, [compare]]]*) -۵۶

این تابع در داخل رشته ***string1*** به دنبال ***string2*** می گردد و هرکجا پیدا کرد آن را با رشته ***replace_with*** عوض میکند. همانند تابع **INSTR** آغاز جستجو از نقطه ***start*** بوده و اگر بیش از یک مورد از ***string2*** درون ***string1*** وجود داشته باشد این جایگذاری ***repeat*** دفعه تکرار می شود. برای این تابع هم هیچ تفاوتی میان کاراکترهای انگلیسی و فارسی وجود ندارد و حتی در صورت عدم تطابق Code page ها درست عمل می نماید. پارامتر ***compare***هم مانند تابع **INSTR ()** می باشد.

RGB(*red, green, blue*) -۵۷

این تابع عددی معادل ترکیب سه رنگ داده شده به آن را باز می گرداند. و در آن ***red*** معادل میزان قرمز در رنگ مورد نظر. ***green*** میزان سبز و ***blue*** میزان

آبی را مشخص می نماید برای دستورات و توابعی که احتیاج دارند تـا مکـان ایـن سه رنگ در عدد داده شده به صورت #RRGGBB باشد می توانیـد تـابع زیـر را در برنامه بنویسید:

```
Function RevRGB(red, green, blue)
 RevRGB= CLng(blue + (green * 256) + (red * 65536))
End Function
```

در صورتی که عدد نسبت داده شده به هر کدام از رنگهای فوق از ۲۵٦ بیشـتر باشد سیستم عدد ۲۵٦ را برای آن منظور خواهد کرد.

۵۸- RIGHT *(string, lenght)*

این تابع دقیقاً عملکردی مشابه تابع **LEFT (string, length)** دارد با این تفاوت که عمل جدا سازی را از سمت راست رشته انجام می دهد.

۵۹- RND ()

این تابع آرگومانی دریافت نمی کند ولی به عنوان خروجی یک عـدد تصـادفی برمـی گردانـد. بـرای اینکـه ایـن دسـتور درسـت کـار کنـد بـاید در ابتـدای کـد VBScript دستور **RANDOMIZE** را نوشت در غیر این صورت مادامـی کـه کاربر صفحه web شما را نبسته و مجـدداً بـاز ننمـاید (ایـن امـر مسـتلزم آن است که کاربر پنجره مربوط به سایت شما را به طور کامل ببندد و مجـدداً آن را download نماید) اعدادی تکـراری تولیـد خواهـد شـد. وایـن بـه دلیـل حفـظ سازگاری با نسخه های قدیم .V.B مـی باشـد کـه از تـایمر سیسـتم بـه عنـوان ورودی تابع hash تولید کننـده عـدد تصـادفی اسـتفاده میکردنـد و ایـن عـدد یکبار و در ابتدای درخواست برنامه برای اجرا از سیستم گرفته میشـد . امـا بـا دستور **RANDOMIZE** باعث میشویم تا مترجم هر دفعه در مواجهه بـا تـابع **RND** از سیستم تایمر جدید را گرفتـه و مجـدداً بـه محاسـبه مقـدار تصـادفی بپردازد.

۶۰- ROUND (*number, decimals*)

این تابع تعداد ارقام عدد (یا متغیر عددی) را به میزان ***decimals*** عـدد گرد مینماید . (اصل عدد تغییر نخواهد کرد)

۶۱- RTRIM (*string*)

این تابع رشته (یـا متغـیر رشـته ای) ورودی را بـدون کاراکترهـای خـالی در سمت راست آن باز می گرداند. (مشابه دستور **LTRIM**)

۶۲- SCRIPTENGINEBUILDVERSION ( )

از نام عجیب این تابع تعجب ننمایید نـام تـابع درسـت تـایپ شـده بـه زبـان آلمانی هم نیست. در اصل جمله می باشد که بـه جهت حفظ سازگاری با مترجم به صورت سر هـم نوشـته مـی شـود. ایـن تـابع پارامتر ورودی نداشته و به عنوان خروجی شـماره روایـت (version) مـترجمی که در حال حاضر برنامه با آن ترجمه می شود را باز میگرداند. توجـه نمـایید کـه روایت مـترجم ربطـی بـه مـترجمی کـه برنامـه نویـس بـا آن کـار میکـرده نـدارد ومقصود روایت مترجم کاربر نهایی است که برنامه بر روی سیسـتم وی اجـرا می شود از این تابع بیشتر زمانی استفاده می نمـاییم کـه مـی خواهیـم دسـتورات غیر استاندارد و مربوط بـه مـترجم هـای جدیـد را اجـرا نمـاییم در ایـن صـورت موظف هستیم چک کنیم و ببینیم که آیا کد مورد نظـر مـا بـر روی مـترجم کـاربر نهایی معتبر هست یا خیر (آیا روایت مترجم وی از روایت خاصی بالاتر هست یـا خیر)

۶۳- SECOND (*hour*)

این تابع به عنوان ورودی متغیری محتـوی سـاعت بـا فرمـت hh:mm:ss میگیرد و ثانیه آن را به صورت عددی مابین ۰ و ۵۹ بازمی گرداند.

برای تبدیل مقادیر زاویه هایی که در مقیاس درجه می باشند به مقیاس رادیان کافیست مقدار درجه را در pi/180 ضرب نمایید. برای محاسبه کتانژانت هم از فرمول :

```
cot(x) = 1/ tan(x)
```

قابل محاسبه می باشد.

TIME ( ) –۷۲

این تابع پارامتر ورودی نداشته و جمله أی معرف ساعت جاری سیستم با فرمت hh:mm:ss باز می گرداند.

TIMESERIAL *(number1, number2, number3)* –۷۳

این تابع سه مقدار عددی از ورودی دریافت نموده و آنها را به فرمت hh:mm:ss در کنار هم میچیند. ***number1*** باید عددی مابین ۰ و ۲۳ و معرف ساعت باشد.***number2*** و ***number3*** هم باید اعدادی مابین ۰ و ۵۹ وبه ترتیب معرف دقیقه و ثانیه باشد.

TIMEVALUE *(date)* –۷۴

این تابع یک پارامتر از نوع تاریخ دریافت می نماید و بخش ساعت آن را جدا کرده و باز پس می دهد.

TRIM *(string)* –۷۵

به عنوان ورودی یک رشته یا متغیر رشتهای دریافت نموده و کاراکترهای خالی موجود در دوطرف رشته را حذف می نماید وبه عنوان خروجی باز می گرداند (رشته اصلی بدون تغییر باقی خواهد ماند.)

۷۶- TYPENAME (*variable*)

این تابع نام یک متغیر را به عنوان ورودی دریافت داشته و نوع داده ای لـن را باز می گرداند. (به توضیحات ذیل تابع **ISNUMERIC** و فصـل سـوم بخـش متغیرها مراجعه نمایید.)

۷۷- UCASE (*string*)

عملکرد این تابع مشابه تابع **LCASE** می باشد تنها بـه جـای بـزرگ کـردن کاراکترهای رشته آنها را کوچک می نماید .

۷۸- VARTYPE(*variable*)

نوع داده أی متغییر به سادگی توسط این تابع مشخص می شود. ورودی تـابع نام یک متغییر و خروجی آن مطابق جدول زیر است:

ثابت	معادل	توضیحات
vbEmpty	۰	متغیری که هنوز مقدار دهی اولیه نشده باشد
vbNull	۱	متغیری با مقدار نامعتبر
vbInteger	۲	متغیری از نوع عددی دو بایتی
vbLong	۳	متغیری از نوع عددی ٤ بایتی
vbSingle	۴	متغیری از نوع اعشاری ساده
vbDouble	۵	متغیری از نوع اعشار با دقت مضاعف
vbCurrency	۶	متغیری از نوع Currency
vbdate	۷	متغیری از نوع تاریخ و ساعت
vbString	۸	متغیری از نوع رشته أی
vbObject	۹	متغیری که اشاره گری به یک شی باشد.
vbError	۱۰	متغیری با پیغام خطا
vbBoolean	۱۱	متغیری از نوع Boolean (منطقی)
vbVariant	۱۲	متغیری از نوع Variant (مخصوص آرایه های با ابعاد نامعین)

vbDataObject	۱۳	متغیری از نوع شی
vbByte	۱۷	متغیری از نوع بایت
vbArray	۸۱۹۲	متغیری از نوع آرایه های با ابعاد ثابت

۷۹- WEEKDAY *(date, [first_day])*

این تابع دو پارامتر دریافت می دارد. اولی یک تاریخ یا متغیر حاوی تاریخ میباشد و دومی یک ثابت می باشد که معرف روز اول هفته در کشور مربوطه می باشد و به عنوان خروجی عددی متناظر با روز از هفته وارد شده می باشد. ثابتهایی که می توان به عنوان روز اول هفته در کشورمان وارد نماییم می تواند یکی از مقادیر زیر باشد :

ثابت	معادل عددی	توضیحات
vbUseSystem	۰	مقدار پیش فرض از سیستم گرفته میشود
vbSunday	۱	یک شنبه
vbMonday	۲	دو شنبه
vbTuesday	۳	سه شنبه
vbWednesday	۴	چهار شنبه
vbThursday	۵	پنج شنبه
vbFriday	۶	جمعه
vbsaturday	۷	شنبه

۸۰- WEEKDAYNAME *(week_day, abreviation, first_day)*

این تابع به عنوان ورودی عدد روزی از هفته ، یک مقدار منطقی و ثابت نمایشگر روز اول هفته در کشور مربوطه و را دریافت می نماید و به عنوان خروجی نام روز هفته متناظر با آن عدد را برمی گرداند اگر مقدار ***abreviaton*** TRUE باشد آنگاه نام بازگشتی به صورت مخفف خواهد بود. ثابتهای معرف روز اول هفته هم مانند مثال قبل می باشد.

۸۱- YEAR(*date*)

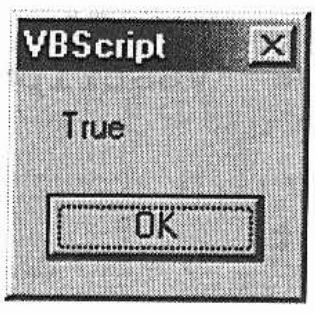
این تابع متغییری از نوع تاریخ گرفته و مقدار سال را از آن استخراج نمـوده و باز می گرداند.

این ۸۱ تابع تقریبا" کل توابعی بود که تاکنون به صورت داخلـی در VBScript معرفـی شده اند حال برای آشناتر شدن با آنها مثالهایی از کاربرد بعضی از آنها می آوریـم. مثلمـا" درک کامل و مهارت در استفاده از این توابع مستلزم برنامه نویسی زیاد است و هـرچـه بیشتر برنامه بنویسید بیشتر مهارت پیدا خواهید کرد.

استفاده از تابع CBOOL :

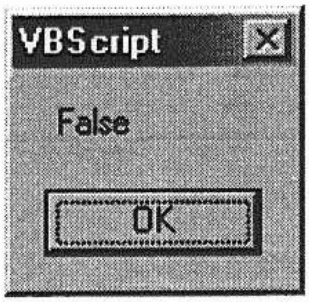
```
<HTML>
  <HEAD>
    <TITLE>Example of use of CBOOL</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
      OPTION EXPLICIT
      DIM variable
      variable = 1
      MSGBOX ( CBOOL( variable ))
    </SCRIPT>
  </BODY>
</HTML>
```

نتیجه اجرای کد فوق بر روی Internet explorer ای که به زبان انگلیسی تنظیم شده باشد چنین است:



شکل ۱-۹

و مثلما اگر به variable مقدار صفر را نسبت بدهیم حاصل چنین می شود :



شکل ۲-۹

البته توجه نمایید که در صورتی که Internet explorer کاربر به زبان دیگری باشد ترجمه دوپیغام فوق به زبان مربوطه را مشاهده خواهد نمود.

نکته ای در مورد تبدیل اعداد:

در هنگام تبدیل اعداد باید توجه داشت تا مبادا عدد بزرگی را بخواهیم به نوع عددی کوچکتری تبدیل نماییم چون این امر باعث خطا خواهد شد فرضا دستور زیر غلط می باشد :

```
number = CBYTE (567,384,866,456.29)
```

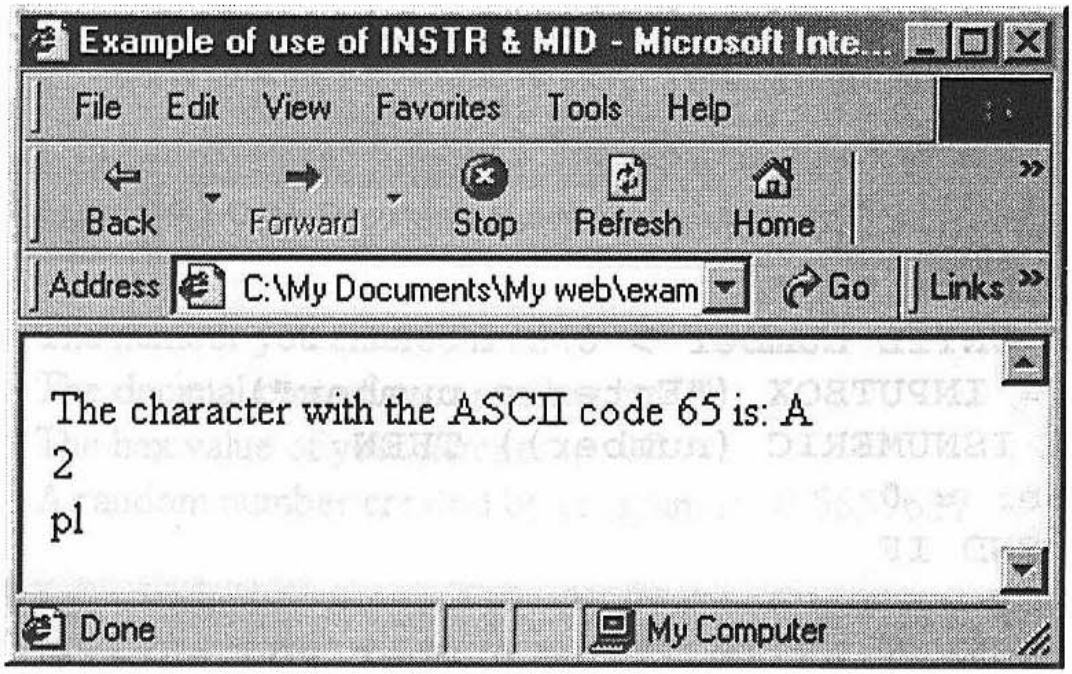
چون عدد داده شده در محدوده بایت جا نمی شود.

استفاده از توابع INSTR و MID

```
<HTML>
  <HEAD>
    <TITLE>Example of use of INSTR & MID</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
      OPTION EXPLICIT
      DIM variable
      DIM central_part

      variable = CHR(65)
```

```
        DOCUMENT.WRITE ("The character with  ASCII
                        code 65 is: " & variable &
                        "<BR>")
        Variable = "Alfa"
        Central_part = INSTR (variable, "1")
        DOCUMENT.WRITE (central_part & "<BR>")

        Variable = "Apple"
        Central_part = MID (variable , 3, 2)
        DOCUMENT.WRITE (central_part )
      </SCRIPT>
    </BODY>
</HTML>
```

حاصل کد فوق چنین است:



شکل ۳-۹

به خط دوم حاصل توجه نمایید. پاسخ تابع **INSTR** است اگر به کد برنامه نگاهی بیندازیم می بینیم که این تابع دو وردی دریافت داشته اولی یک رشته می باشد که باید درون آن به دنبال چیزی گشت و دومی رشته ای است که به دنبال آن هستیم عددی که برگشت داده می شود(در این مورد ۲) موقعیت مکانی اولین مورد یافت شده می باشد.

حالا به خط آخر توجه فرمایید همان جایی که نوشته شده pl . این نتیجه تابع MID است. به سطر مربوطه درون کد نگاه کنید. همانطورمیبنید که این تابع سه آرگومان ورودی دارد.

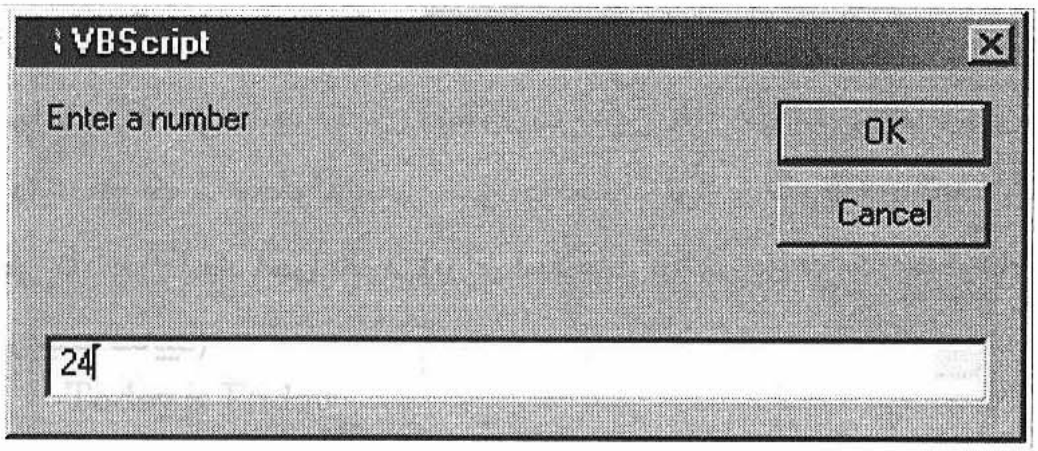
اولین آرگومان یک رشته است که قرار است از داخل آن زیر رشته ای را جدا نماییم. دومین آرگومان یک عدد است که مشخص کننده سرآغاز نقطه جدا یی زیر رشته میباشد . آرگومان سوم هم یک عدد است که معرف طول زیر رشته از نقطه جدا شده می باشد البته همانطور که در شرح تابع گفته شد اصل رشته دست نخورده باقی خواهد ماند.

مثالی از کاربرد بعضی توابع ریاضی:

```
<HTML>
  <HEAD>
    <TITLE>
       Example of use of some mathematic functions
    </TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
      OPTION EXPLICIT
      RANDOMIZE

      DIM number
      number= 0
      DO UNTIL number > 0
 number = INPUTBOX ("Enter a number")
 IF (NOT ISNUMERIC (number)) THEN
    number = 0
        END IF
      LOOP
      DOCUMENT.WRITE ("The number you entered is :
" & number & "<BR>")       number = INT ( number )
      DOCUMENT.WRITE ("The decimal part of your
number is : " & number & "<BR>")
      DOCUMENT.WRITE ("The hex value of your number
is : " & hex( number ) & "<BR>")
      DOCUMENT.WRITE ("A random number created by
program is : " & RND () & "<BR>")
    </SCRIPT>
  </BODY>
```

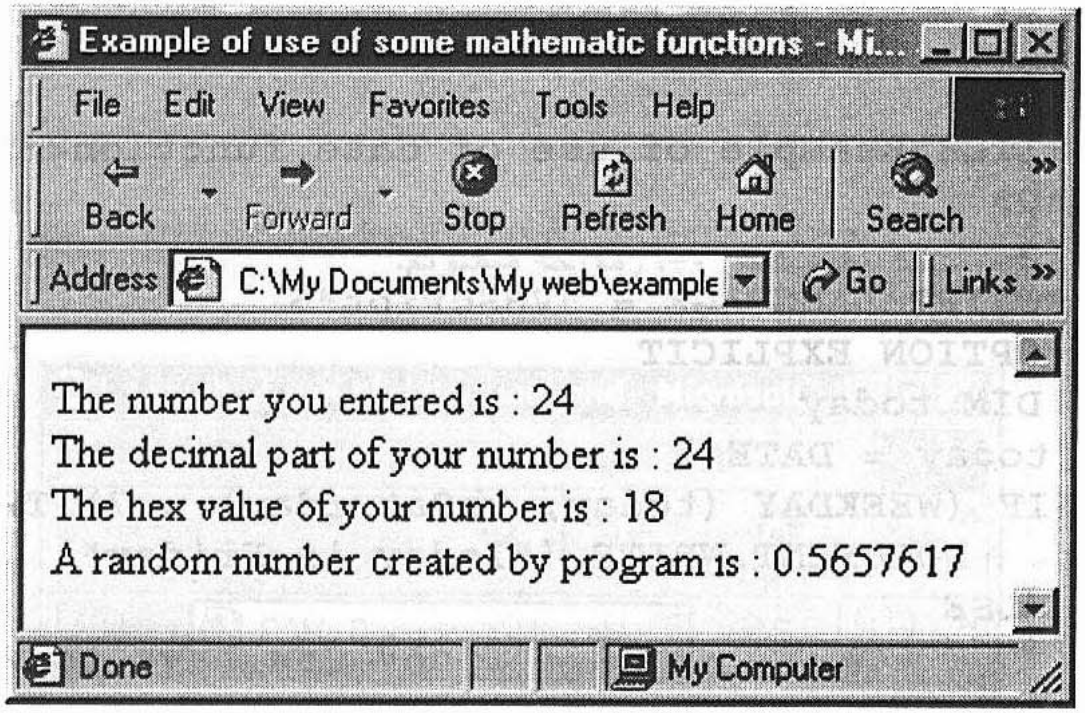
```
</HTML>
```

حاصل چنین است:



شکل ۴-۹

با توجه به عدد ورودی نتیجه شکل ۵-۹ پدیدار می شود.



شکل ۵-۹

سطر اول عددی را که وارد نموده ایم نمایش میدهد.

سطر بعدی نتیجه ای است که از اعمال این عدد در تابع INT حاصل میشود.

سطر سوم هم مقدار تبدیل شده عدد به مبنای ۱۶ را نمایش می دهد.

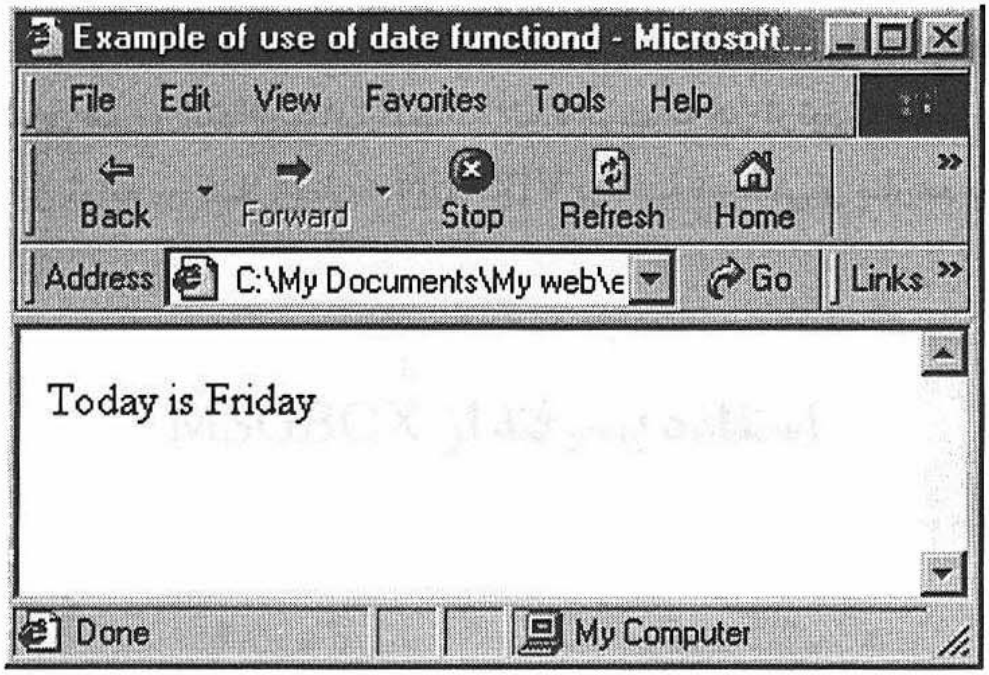
و بالاخره در سطر چهارم یک عدد کاملا تصادفی بین ۰ و ۱ ظاهر خواهد شد که مثلما اگر شما این کد را بر روی کامپیوتر خودتان اجرا نمایید با عدد دیگری مواجه خواهید شد. (البته به احتمال یک در ۳۶ ملیون امکان دارد عدد شما با مثال ما برابر باشد !!!.)

تابع دیگری هم در این مثال استفاده شده ISNUMERIC تا اگر کاربر چیزی بجز عدد وارد کرد از او پذیرفته نشده و مجدد سوال گردد. برای این امر در دستور IF از عملگر NOT استفاده شده است که باعث می شود به حاصل تابع را به صورت نقیض نگریسته شود زیرا همانطور که می دانید اجرای دستورات مابین IF , END IF منوط به صحت شرط است در حالی که ما میخواهیم در صورت عدم برقراری شرط و در صورتی که تابع به ما خروجی FALSE داد اجرا شود. (برای اطلاعات بیشتر در مورد عملگرهای منطقی به پیوست ۳ مراجعه نمایید)

مثالی از کاربرد توابع مربوط به تاریخ:

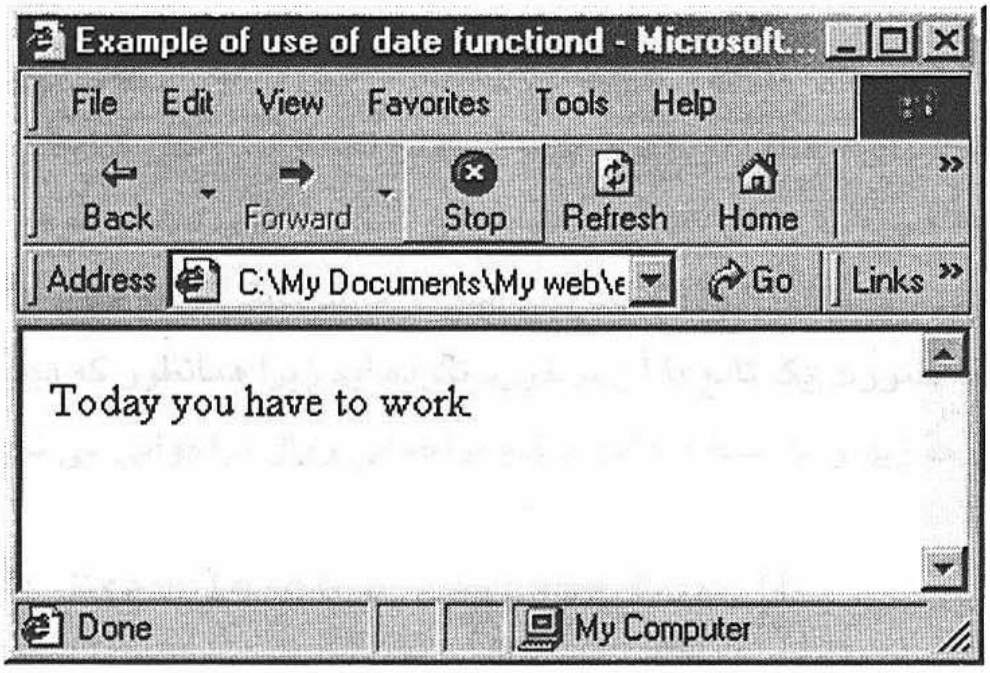
```
<HTML>
  <HEAD>
     <TITLE>Example of use of date function</TITLE>
  </HEAD>
  <BODY>
     <SCRIPT LANGUAGE = "VBScript">
       OPTION EXPLICIT
       DIM today
       today = DATE()
       IF (WEEKDAY (today, vbSaturday) = 7) THEN
          DOCUMENT.WRITE ("Today is Friday")
       ELSE
          DOCUMENT.WRITE ("Today you have to work")
       END IF
    </SCRIPT>
  </BODY>
</HTML>
```

کد فوق در ابتدا تاریخ روز را ازسیستم میگیرد و اگر جمعه باشد پیغام زیر را نمایش میدهد:



شکل ۹-۶

و در غیر اینصورت با پیغام شکل ۹-۷ مواجه میشویم:



شکل ۹-۷

برای این کار از تابع WEEKDAY استفاده می نماید و شنبه را به عنوان روز اول هفته به تابع معرفی نموده ایم.

تا اینجا برای آشنایی با کاربرد توابع همین تعداد مثال کافی به نظر میرسد. مثالـهای کاربردی دیگری هم در انتهای کتاب گنجانده شده است. به هر ترتیب همانطور که گفتیم تنها راه تسلط بر این توابع و کـلاً زبـان VBScript برنامـه نویسـی و بـازهم برنامـه نویسـی میباشد.

فصل دهم:

استفاده پیشرفته از MSGBOX

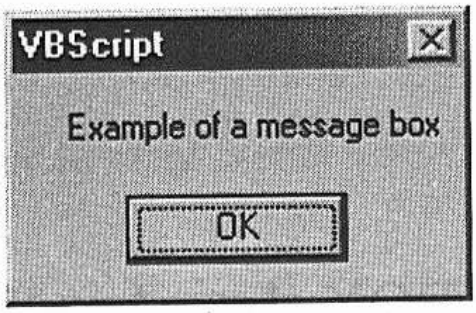
تا بکنون در مثالهای قبلی از تابع **MSGBOX** زیاد استفاده کرده ایم. در اصل این تابع هم یکی از توابع داخلی زبان می باشد ولی به دلیل رفتار خاصی که از خود بروز میدهد بهتر دیدیم تا یک فصل کامل را به آن اختصاص دهیم. این تابع امکانات بسیار بیشتری از آنچه تاکنون دیده ایم در اختیار ما میگذارد. وبه عنوان ابزاری قدرتمند جهت نمایش انواع پنجره های پیغام ، اخطار، سوال و تاییدیه قابل استفاده می باشد. برخی از این پنجره ها امکان انتخاب حالتهای مختلف از طرف کاربر را در اختیارمان می گذارد و حتی میتواند انتخاب کاربر را هم به ما منتقل کند تا بر اساس آن تصمیم گیری نماییم. به هر صورت باید توجه داشته باشیم که MSGBOX یک تابع است و مانند مابقی توابع مقدار باز می گرداند اما ما تاکنون به صورت یک تابع با آن برخورد نکرده ایم زیرا همانطور که دیده اید ما این تابع را به صورت زیر و به مشابه حالت ساده فراخوانی روال فراخوانی می نمودیم.

```
MSGBOX (" Example of a message box ")
```

در این حالت همان پنجره آشنا با یک دکمه OK ظاهر می شود. حالا یک حالت جدید و باز هم عجیب تر:

```
CALL MSGBOX ("Example of a message box ", VBOKONLY,
             "")
```

همانطور که میبینید به مشابه حالت درست فراخوانی روالها تابع را خوانده ایم. و اگر حاصل را ببینیم دقیقاً مشابه حالت قبل خواهد بود.



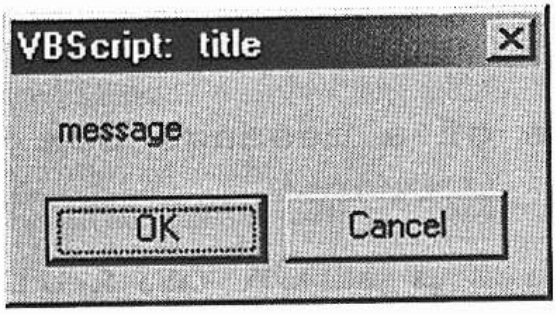
شکل ۱-۱۰

علت آن است که تابع MSGBOX حالت خاصی از توابع است که در شرایطی مانند بالا میتواند خروجی نداشته باشد. (البته سایر توابع را هم میتوان با کمک دستور CALLبه صورت روال فراخوانی نمود و مقدار بازگشتی آنها را دور ریخت ولی اولاً با این کار عملاً تابع ما کاری انجام نداده مگر اینکه تابع را خودمان نوشته باشیم و بخواهیم از تغییرات آن روی متغیر های ضمنی استفاده نماییم ثانیاً این نوع استفاده با تعریف تابع منافات دارد و به هیچ وجه توصیه نمیشود.)

و اما ببینیم حالات دیگر استفاده از این تابع به چه صورت است:

1- confirm = MSGBOX ("message", VBOKCANCEL, " title")

دستور بالا باعث پدید آمدن پنجره زیر می شود:

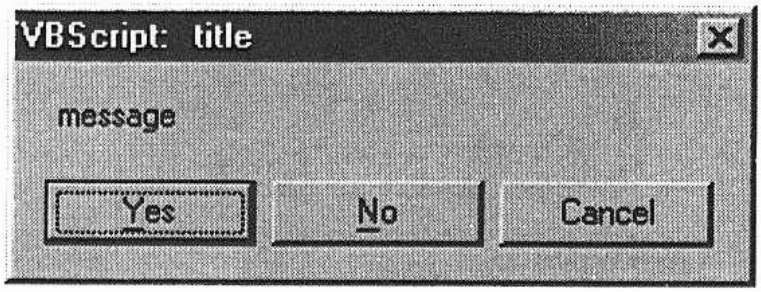


شکل ۲-۱۰

و منتظر خواهد شد تا کاربر یکی از دو کلید را فشار دهد. و اما درون متغیر confirm چه چیزی ذخیره میشود؟ عددی که معرف دکمه زده شده است. اگر کاربر دکمه YES را فشار دهد عدد ۱ و در غیر اینصورت و یا بسته شدن پنجره مقدار ۰ در درون این متغیر ذخیره خواهد شد. و به این ترتیب در ادامه برنامه خواهیم توانست بر اساس پاسخی که کاربر به پیغام ما می دهد تصمیم گیری نماییم. به تواناییهای دیگر این تابع توجه نمایید. در پایان فصل جدولی از کلیه حالات ممکن جهت کلید ها و اعداد مربوطه آورده شده است.

2- result = MSGBOX ("message", VBYESNOCANCEL, " title")

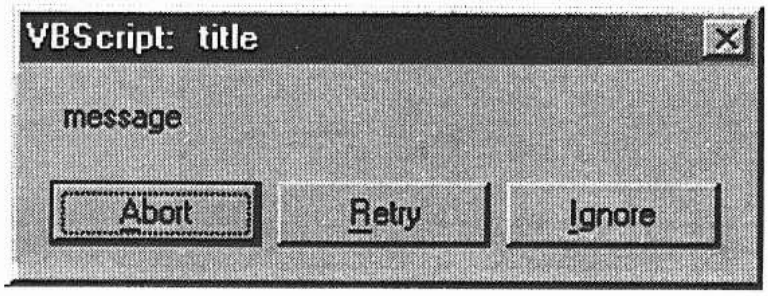
حاصل چنین است:



شکل ۳-۱۰

3- result = MSGBOX ("message", VBABORTRETRYIGNORE, " title")

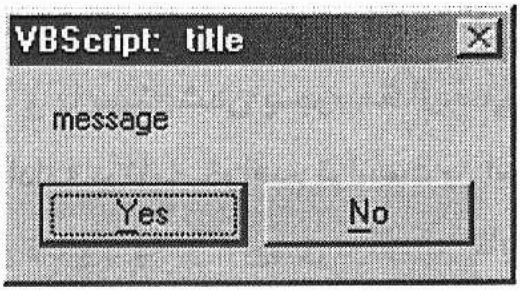
حاصل چنین است:



شکل ۴-۱۰

4- result = MSGBOX ("message", VBYESNO, " title")

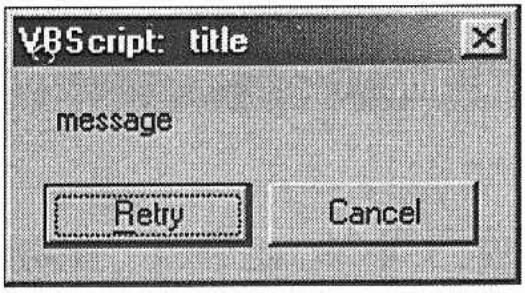
حاصل چنین است:



شکل ۵-۱۰

5- result = MSGBOX ("message", VBRETRYCANCEL, " title")

حاصل چنین است:



شکل ۶-۱۰

6- result = MSGBOX ("message", VBCRITICAL, " title")

حاصل چنین است:

شکل ۷-۱۰

7- result = MSGBOX ("message",VBQUESTION, " title")

حاصل چنین است:

8- result = MSGBOX ("message", VBEXCLAMATION, " title")

حاصل چنین است:

شکل ۹-۱۰

9- result = MSGBOX ("message", VBINFORMATION, " title")

حاصل چنین است:

شکل ۱۰-۱۰

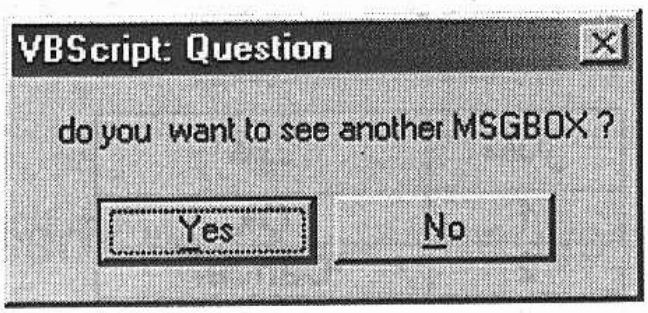
جدول زیر معادل عددی دکمه های زده شده می باشد.

کلید	مقدار
Ok	1
Cancel	2
Abort	3
Retry	4
Ignore	5
Yes	6
No	7

مثال :

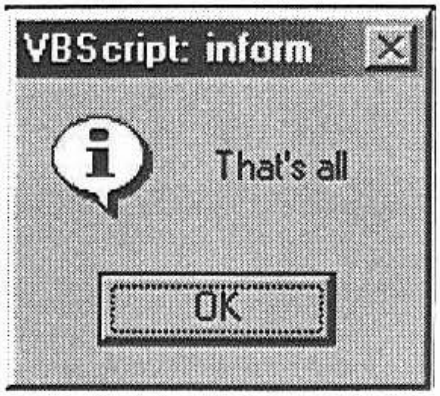
```
<HTML>
  <HEAD>
    <TITLE>Example of use MSGBOX function</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "VBScript">
      OPTION EXPLICIT
      DIM response

      response = 0
      response = MSGBOX ("do you  want to see
                         another MSGBOX ?",VBYESNO,
                         "Question")

      IF (response = 7 ) THEN
         DOCUMENT.WRITE("<H1>You selected to
                         continue<BR>")
      ELSE IF ( response = 6 ) THEN
         CALL M
                        "inform")
    </SCRIPT>
  </BODY>
</HTML>
```

حاصل این برنامه چنین است:



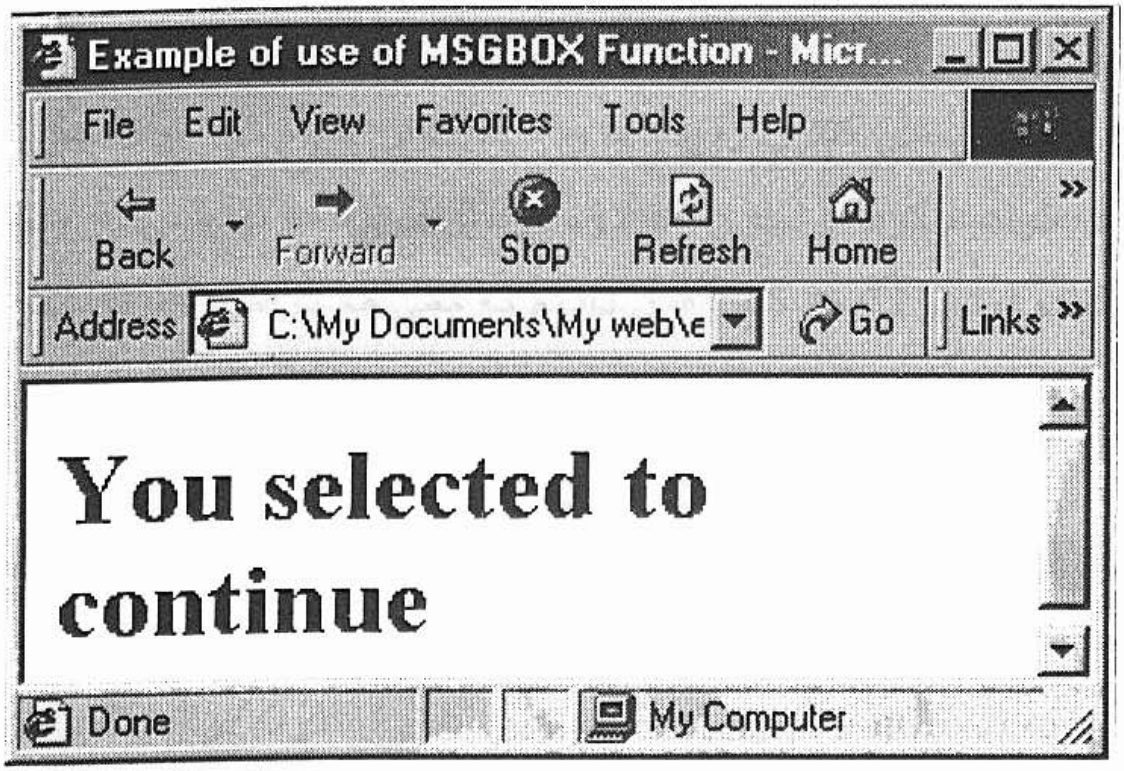
شکل ۱۱-۱۰

حال با فشردن کلید yes پنجره شکل ۱۲-۱۰ ظاهر می شود :



شکل۱۲-۱۰

و با فشردن کلید No پیغام زیر چاپ خواهد شد:



شکل ۱۳-۹

فصل یازدهم :

مقدمه ای بر ASP

تا اینجا به جز بخش کنترل رویدادها و استفاده از اشیاء که در فصول ۱۲ و ۱۳ در مورد آن صحبت خواهیم کرد هر چه را که لازم بود برای برنامه نو یسی در سطح کامپیوتر Clientبدانیم گفتیم اما در مقدمه کتاب قول دادیم که کمی هم در مورد برنامه نویسی برای server صحبت کنیم.

بسیاری از مواقع مواردی پیش می آید که لازم است برنامه ما کاری را روی سرور انجام دهد. به عنوان مثال زمانی که می خواهیم یک پایگاه داده را مدیریت نماییم لازم نیست و اصلاً نباید آن پایگاه داده را وارد کامپیوتر مشتری نمود. در این گونه موارد برنامه در سطح سرور اجرا می شود و تنها پاسخ و نتیجه اعمال است که به سیستم مشتری انتقال می یابد. مضاف بر این که در حین اجرای عملیات اگر مشکلی در سیستم وجود داشته باشد کافیست در سمت سرور این مشکل برطرف گردد و به صورت اتوماتیک تمام مشترکین از روایت جدید استفاده خواهند نمود بدون آنکه لازم باشد تا تمام آنها مجدداً اطلاعات و برنامه جدید را download نمایند و به دلیل اینکه این برنامه تنها بر روی سرور اجرا می شود امکان خوانده شدن سورس کد برنامه توسط کاربران به روشهای معمول و از طریق گزینه view source میسر نیست.(البته هکرها قدرت نفوذ در چنین سیستم هایی را دارند که صحبت در این زمینه از حوصلهٔ بحث خارج است.)

راه معمول برنامه نویسی بر روی سرور استفاده از تکنیک (CGI: common gateway interface) می باشد. و از این تکنیک کماکان در بعضی از صفحات وب استفاده می شود ولی به تدریج در حال انقراض بوده و امروزه کمتر به چشم می آید. روش جدیدی که هم اکنون به صورت گسترده در حال استفاده است به تکنیک ASP (Active server page) معروف می باشد . و در این فصل به طور مختصر به آن خواهیم پرداخت . ASP عبارت است از یک سری دستورات از زبانی خاص (مثلاً VBScript) که در هنگام فراخوانده شدن صفحه وب همراه مابقی متن به کامپیوتر مشتری منتقل نمی شود. بلکه بر روی کامپیوتر سرور اجرا شده و تنها نتایج آن را به کامپیوتر Client منتقل می نماید. اولین چیزی که باید در مورد این صفحات بدانیم آن است که این صفحات نباید پسوند **HTM**. یا **HTML**. داشته باشند بلکه پسوند آنها **asp**. می باشد. ضمناً باید بر روی یک سیستم عامل از نوع سرور نظیر Windows NT و یا UNIX اجرا شوند (باکمک ابزار کمکی PWS (Personal Web Server) که به همراه Windows CD اصل موجود است هم میتوان بر بر روی سیستم عامل windows 9x سرور مجازی ایجاد نمود.) و بر روی سیستم عاملهایی که برای استفاده در سطح کاربری باشند مانند Windows 9x و Windows NT workstation اجرا نخواهد شد. همچنین پوشه ای که برای نگهداری این فایلها در نظر گرفته می شود باید اجازه نوشتن برای آن صادر گردد (برای اطلاعات بیشتر در زمینه نحوه تخصیص این اختیار به پوشه ها به کتابهای مرجع سیستم عامل مربوطه مراجعه نمایید)

اگر تمام موارد بالا را فهمیده و رعایت کرده اید ببینیم که چگونه میتوان متن ASP به صفحه اضافه نمود. با یک مثال موضوع را شرح میدهیم.

```
<HTML>
  <HEAD>
    <TITLE> example of an asp page </TITLE>
  </HEAD>
  <BODY>
    <HR>
       This page is an asp test
       <%
        My_email = "myname@mydomain.com"
        Response.Write (my_email)
```

```
        %>
      <HR>
    </BODY>
</HTML>
```

در کد فوق دو دستور بسیار شبیه دستورات VBScriptمشاهده مینماییم.

```
My_email = "myname@mydomain.com"
Response.Write (my_email)
```

که اتفاقاً در صورت اجرا خواهیم دید که نتیجه أی بسیار مشابه نتیجه مـورد انتظـار دارد در اصل این هم همان VBScript است ولی در سمت Server اجرا شـده اسـت لـذا بـه جای تگهای <SCRIPT> </SCRIPT> دستورات مابین تگهای <% y%> نوشته شده اند البته همانطور که مثلماً هم می دانید أین دو تگ مربوط به زبان HTML نیسـتند پـس اشتباه هم نخواهند شد. شیء Response هم شیی متعلق به server مـی باشـد مـانند document در سمت client . با این وجود کاربر نخواهد توانست به راحتی متن اجرایی برنامه ما را ببیند.

هدف ما از این فصل آموزش ASP نبود بلکه تنها میخواستیم ایـده أی داده باشـیم و به اصطلاح جاده را برای علاقمندان بگشاییم . اگر به برنامـه نویسـی ASP علاقمندیـد ابتدا سعی کنید در برنامه نویسی در سطح Client مسلط شوید و سپس کم کم و پـس از آشنایی بیشتر با مباحث web و شبکه و بالاخص برنامه نویسـی شـی گرا بـه سـمت برنامه نویسی ASP بروید.(برای مطالعه بیشتر در زمینه ASP به کتاب " برنامه نویسی کاربردی روی سرویس دهنـده وب از طریـق ACTIVE SERVER PAGE " تـالیف بهزاد اکبری مراجعه نمایید.)

فصل دوازدهم:

اشیا و رویدادها

در این فصل از کتاب باشد تمام اشیایی را که در VBScript آنها را به رسمیت میشناسیم به همراه تمام خصوصیات و متدهایشان لیست نموده ایم البته منظور از تمام اشیاء اشیایی است که در برنامه نویسی در سمت کامپیوتر client به رسمیت میشناسیم. زیرا که برای سرور اشیاء بسیار بیشتر و گسترده تری را بکار می بریم. و سپس با مثالهای متعدد سعی در تفهیم موضوع خواهیم نمود. در ابتدا جهت یاد آوری چهار کلمه مهم در ارتباط با شی و شی گرایی را توضیح میدهیم.

شی : (object)

تمام المانهای موجود در یک صفحه web یا یک برنامه اجرایی که موجودیتی جداگانه داشته باشد. شی میتواند پنجره یک صفحه وب، برنامه در حال اجرا و یا یک جدول و ... باشد. لیست اشیایی که میتوانیم در VBScript استفاده نماییم در ادامه این فصل آمده است.

خصوصیات : (Properties)

هر کدام از خصوصیات یک شی می باشد و میتواند رنگ پس زمینه یک صفحه یا نوع خط یک نوشته باشد. همچنین خصوصیت میتواند خود شییی باشد که از شی بالاتر مشتق شده است. در این صورت می گوییم که این شی یکی از خصوصیات

پدرش می باشد . مانند Form که خد یک شی پدر بوده و تمام اشیایی نظیر متنها کلیدها و ورودیهای یک صفحه HTML در آن قرار می گیرند و در عین حال خود فرزند شی Document می باشد.

متد : (Methods)

عبارت از عملیاتی است که میتوان بر روی یک شی انجام داد . مثلاً بستن یک پنجره متدی از شی پنجره می باشد. یا فرستادن نشانگر(cursor) به منطقه یک جعبه متن(textbox) متدی از شی جعبه متن است.

رویداد :(Methods)

پیش بینی عملیاتی خاص که کاربر امکان انجام آن را دارد . در اصل کاربر این امکان را دارد تا این عملیات را انجام داده یا از آن صرفنظر نماید ولی به برنامه نویس این امکان داده می شود تا این عملیات را پیش بینی نموده و برای آن برنامه نویسی نماید در این صورت اگر آن عملیات بخصوص صورت بپذیرد کد مربوطه اجرا خواهد شد. این عملیات خاص میتواند فرضاً کلیک کردن بر روی یک منطقه خاص از صفحه نمایش و یا حتی گذشتن نشانگر ماوس از روی یک منطقه باشد. حتی أین امکان وجود دارد تا رفتارهای سیستم هم در اینگونه موارد پیش بینی شود رفتار سیستم می تواند بار نمودن یک صفحه یا بستن آن و یا حتی گزارش خطا باشد که برای هر کدام میتوان جداگانه دستوراتی نوشت و در اصل کنترل اوضاع پیش آمده را به دست بگیریم. به طور خلاصه یک رویداد زمانی اجرا می شود که شرایط پیش بینی شده برای آن تحقق یابد. به مثال زیر توجه فرمایید :

```
<HTML>
 <HEAD>
   <TITLE>Use of the MOUSEOVER event</TITLE>
   <SCRIPT LANGUAGE = "VBScript">
     FUNCTION  message ()
       MSGBOX ("You can never click on the Link  ")
     END FUNCTION
   </SCRIPT>
```

```
  </HEAD>
  <BODY>
    <H1>Try to click on the link ...</H1><BR>
    <A HREF = "HTTP://WWW.YAHOO.COM"  ONMOUSEOVER=
             "message ()" > Click here to go to the
             YAHOO main page </A>
   </BODY>
</HTML>
```

همانطور که می بینید از یک دستور جدید در متن HTML استفاده نموده ایم **ONMOUSEOVER** که به همراه تنظیمات مربوط به یک لینک آمده است با اجرای این برنامه خواهید دید که به محض آنکه می خواهید نشانگر ماوس را به متن لینک شده نزدیک نمایید بلافاصله پنجره أی ظاهر می شود و نمیگذ ارد شما بر روی متن کلیک نمایید هر چه قدر هم که سعی نمایید و فرض باشید امکان ندارد بتوانید بر روی این متن کلیک نمایید زیرا به محض آنکه ماوس به این متن نزدیک می شود رویداد MOUSEOVER اتفاق می افتد و چون نتیجه این رویداد را اجرای تابع message قرار داده ایم لذا به محض تحقق رویداد این تابع اجرا شده و با نمایش یک جعبه پیغام شما را مجبور به کلیک بر روی دکمه O K خود می نماید و بدینوسیله امکان ادامه عملیات را از شما سلب می شود.

برای اشاره به خصوصیات و متدهای یک شی از نقطه استفاده مینماییم :

Object . Property

و یا :

Object . Method ()

که البته برای متدها باید پارامترهای مربوطه را نیز در صورت وجود میان پرانتز بنویسیم.

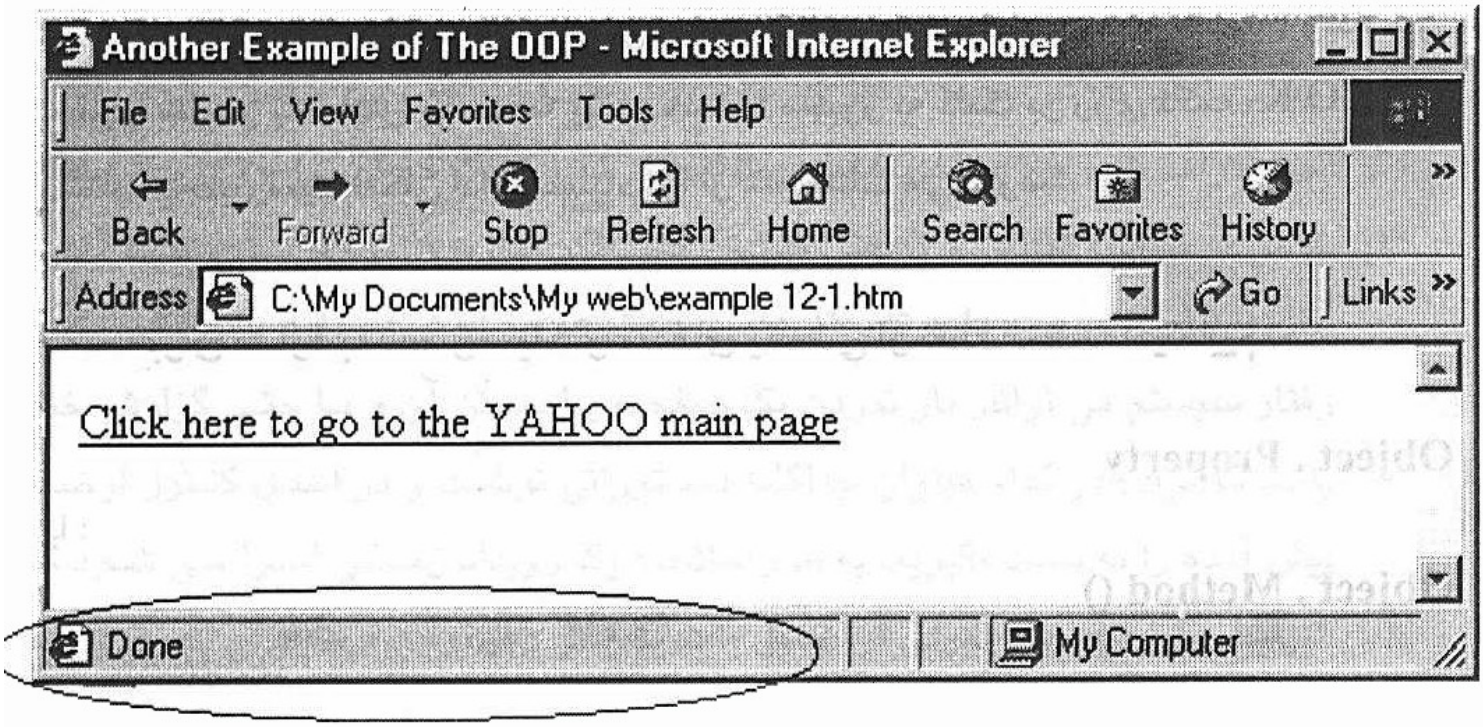
برای روشنتر شدن موضوع به مثال زیر توجه نمایید:

```
<HTML>
  <HEAD>
    <TITLE>Another Example of The OOP</TITLE>
```

```
</HEAD>
<BODY>
   <A HREF = "HTTP://WWW.YAHOO.COM"  ONMOUSEOVER=
             "window.status = ' Click to go to
             the yahoo official web site';return
             true"> Click here to go to the YAHOO
             main page
   </A>
 </BODY>
</HTML>
```

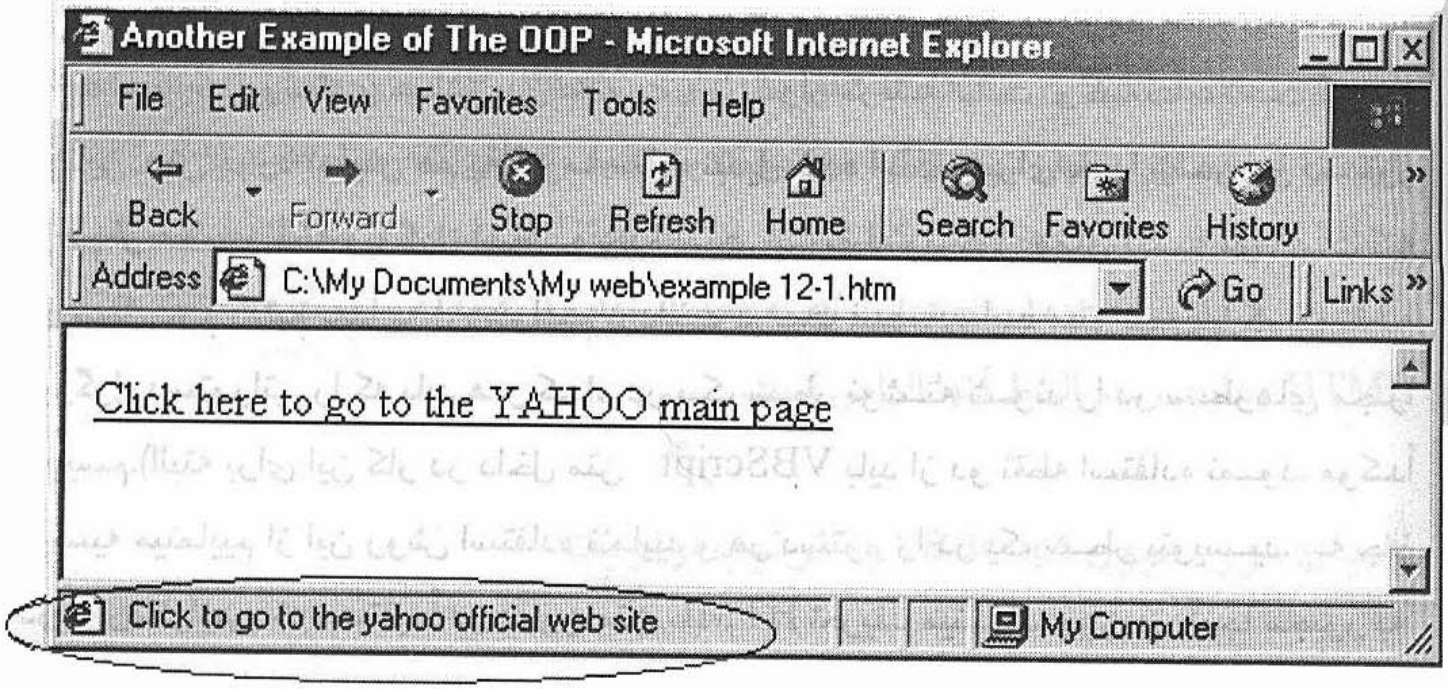
این مثال تقریباً هیچ شباهتی به تمام مسایلی که تاکنون آموخته ایم ندارد. ابتدا ببینیم حاصل این کد چگونه است؟ این کد ابتدا یک متن لینک به نمایش می گذارد و زمانی که با ماوس از روی آن میگذریم در قسمت پایین صفحه در قسمت توضیحات متنی به نمایش در می آید.

قبل از گذشتن از روی متن :



شکل ۱-۱۲

بعد از ورود به محدوده متن شکل ۲-۱۲ حاصل می شود :



شکل ۲-۱۲

تمام موضوع در یک سطر خلاصه شده :

```
<A HREF = "HTTP://WWW.YAHOO.COM"  ONMOUSEOVER=
          "window.status = ' Click to go to the
          yahoo official web site';return true">
          Click here to go to the YAHOO main page
          </A>
```

در حقیقت أین یک سطر کل برنامه ماست که در آن واحد دو کار انجام می دهد اولاً یک متن از نوع متنهای لینک بوجود می آورد و ثانیاً تابعی را برای زمانی که از روی متن میگذریم تعریف می نماید. همانطور که متوجه شده اید باز هم برای تشخیص عبور نشانگر ماوس از روی متن از رویدادنمای **ONMOUSEOVER** استفاده نموده ایم ولی این بار به جای آنکه این رویداد باعث اجرای تابعی از پیش تعریف شده شود تابع را مستقیماً در ادامه رویداد تعریف نموده ایم در اصل این تابع عبارت است از :

```
Window.status = "Click to go to the yahoo official
                 web site"
 return true
```

که به صورت پیوسته و مـابین دو علامـت "" قـرار گرفتـه اسـت. و ضمنـاً علامـت گیومـه دوتایی متن مورد نمایش هم به گیومه ساده تبدیل شده است و برای جـدا شـدن دو دسـتور از هم از علامت نقطه ویرگول استفاده شده است. این نقطه ویرگول همان نکتـه أی اسـت که در فصل دوم گفتیم بعداً به آن خواهیم پرداخت درست فـهمیده ایـد میتوانیـم بـا کمـک نقطه ویرگول دستوراتی را که باید هـر کـدام در یـک سـطر نوشـته شـوند را در سـطرهای مجزا بنویسم.(البته برای این کار در داخل متن VBScript باید از دو نقطه استفاده نمـود. موکداً توصیه مینماییم از این روش استفاده ننمایید و هر دستور را در یک سـطر بنویسـید. بـه جز در مورد توابعی که مستقیماً در درون متن HTML تعریف مینمایید چون در آنجا مجبور به این کار هستید.)

اما یک نکته دیگر در مورد این مثال باقی مانده. اگر یادتان باشد گفتیم در این کتـاب تمام کلمات کلیدی و توابع داخلی را با حروف بزرگ خواهیم نوشت ولی همانطور که میبینید این قانون را در أین مثال رعایت نکرده ایم و کلمات کلیدی window, status, return را بـا حروف کوچک نوشته ایم چرا؟ این بدان دلیل است کـه مـا أیـن برنامـه را مسـتقیماً در درون تگهای VBScript ننوشته ایم تا از قواعد VBScript پیروی نماییم در ایـن شـرایط زمـانی که ما تابعمان را مستقیماً در درون متن HTML تعریف می نماییم در زمان اجـرا مترجم VBScript را نمی شناسد بلکه دستورات JAVAScript را اجرا میکنـد و JAVAScript نسبت به حروف بزرگ و کوچک فوق العاده حساس می باشد و هر دسـتور و کلمـه أی بـاید دقیقاً به همان صورتی نوشته شود که گفته شده. (علت اینکه در متن بالا هم به جای دو نقطـه از نقطه ویرگول استفاده نموده ایم همین موضـوع مـی باشـد و در حقیقـت در مثـال بـالا مـا JAVAScript نوشته ایم و اثری از VBScript دیده نمی شود. دلخـور نشـوید ولـی بـاید بپذیریم که VB در مقابل JAVA حرفی برای گفتن ندارد مضاف بر اینکه پیشکسوت بـودن JavaScript باعث می شود تا اکثر رویدادها و اتفاقات ابتدا برای Java تعریـف شـوند. بـا این حال VBScript برای برنامه نویسی ساده و سریع و با بهره گیـری از امکانـات از پیـش تعرف شده Windows و ضمناً برای کسانی که میخواهند برای اولین بار به برنامـه نویسـی بر روی WEB روی آورند بهترین گزینه می باشد.)

در ادامه لیست تمام شی هایی را که در VBScript به رسمیت شناخته میشوند را به همراه تمام خصایص و متدهای متعلق به آن و رویدادهایی که بر روی آن امکان تعریف دارند را آورده ایم . یکبار دیگر تاکید میکنیم که بسته به مکانی که میخواهید از این اشیاء استفاده نمایید رعایت حروف کوچک و بزرگ میتواند واجب و یا اختیاری باشد. و مخصوصاً برای رویدادها دقت نمایید که به دلیل استفاده از آنها به صورت مستقیم در متن HTML رعایت رسم الخط آنها الزامی است.

اشیای مورد قبول VBScript به ترتیب اولویت:

1-window
2-location
3-document
4-navigator
5-frame
6-history
7-link
8-anchor
9-form

۱- Window

این شی پدر تمام اشیای دیگر در یک صفحه web می باشد. معرف پنجرهٔ فعالی است که برنامه در آن در حال اجراست.

خصوصیات:

defaultstatus	متنی که به صورت پیش فرض در قسمت توضیحات صفحه نمایش داده می شود.
document	عبارت است از برنامه HTML فعالی که در حال حاضر در این پنجره اجرا می شود.(مقصود برنامه اصلی است و برنامه مربوط به فریمها جداگانه تعریف می شود.)
frames [ ]	ماتریس فریمهای موجود در صفحه که از فریم شماره صفر آغاز میشود.

history	لیست تمام صفحات مشاهده شده از طریق أین مرورگر در روز جاری شامل صفحاتی که از اینترنت یا اینترانت دیده شده و در مورد Interet explorer به انضمام تمام صفحه هایی که در روز جاری حتی از روی هارد ورق زده ایم. مانند تمام پرونده ها و پوشه های عادی.
length	تعداد کل فریمهای موجود در صفحه.
location	نشانی با فرمت Protocol ://URL/path/filename صفحه جاری را نمایش می دهد . نظیر http://www.yahoo.com یا file://C:/My document/examples/example2.htm
name	نام پنجره فعال جاری را در بر دارد
navigator	معین کننده نوع مرورگر أی است که در حال حاضر کاربر از آن استفاده می نماید.
self	به خود همین پنجره اشاره می نماید.
status	مشخص کننده پیغام موقتی است که به ازای یک رویداد باید در Statusbar نمایش داده شود.
window	مشخص کننده پنجره فعال جاری یا یک پنجره دیگر یا یک زیر پنجره می باشد.

متدها:

close ()	در صورتی که به شکل self.close() استفاده شود پس از کسب اجازه از کاربر پنجره را میبندد.(این اجازه گرفتن از خواص مرورگر می باشد و با برنامه نویسی عادی نمیتوان این قابلیت را از آن سلب نمود.)
open()	باعث باز شدن یک پنجره جدید می شود و در اصل شی پنجره جدیدی خلق می نماید. روش استفاده از آن به صورت زیر است: new_w=window.open ("URL" ,"Target", "options")

	که در آن **new_w** نامی است که میخواهیم به این زیر پنجره بدهیم. **URL** آدرس فایلی است که میخواهیم در این صفحه به نمایش در بیاید. **TARGET** نام فریمی است که در صورت وجود میخواهیم زیر پنجره در آن باز شود. و بالاخره **options** خصوصیات اولیه أی است که میخواهیم این پنجره جدید داشته باشد و میتواند از جدول بعد انتخاب شود.

جدول انتخابهای ممکن برای **option** در دستور **window.open**

انتخاب	نوع داده	توضیحات
toolbar	Boolean	وجود یا عدم وجود نوار ابزار در بالای پنجره
location	Boolean	وجود یا عدم وجود نوار نشانی
directories	Boolean	وجود یا عدم وجود directories
Status	Boolean	وجود یا عدم وجود نوار توضیحات Status bar
menubar	Boolean	وجود یا عدم وجود نوار منوها
scrollbars	Boolean	وجود یا عدم وجود ابزار حرکتی برای پنجره
resizable	Boolean	آیا کاربر اجازه دارد پنجره را تغییر اندازه بدهد؟
width	Pixel	عرض پیش فرض صفحه
height	Pixel	ارتفاع پیش فرض صفحه
top	Pixel	موقعیت Y صفحه (موقعیت نقطه بالا سمت چپ صفحه)
left	Pixel	موقعیت X صفحه (موقعیت نقطه بالا سمت چپ صفحه)

رویدادها :

onLoad	به محض بار شدن صفحه این رویداد رخ میدهد .
onUnload	زمانی که صفحه بسته می شود این رویداد اتفاق می افتد.

Document -۲

به برنامه در حال اجرا اشاره دارد.

خصوصیات :

alinkColor	رنگ لینکهایی مشاهده نشده.
bgColor	رنگ پس زمینه صفحه را مشخص می نماید.
fgColor	رنگ پیش فرض متنهای نمایش داده شده را تعیین می نماید.
lastModified	تاریخ آخرین تغییرات اعمال شده روی صفحه را نمایش می دهد .
linkColor	رنگ لینکهای موجود در صفحه.
location	آدرس URL صفحه را در بر دارد.
title	سر تیتر صفحه را نمایش می دهد.
vlinkColor	رنگ لینکهایی مشاهده شده.

متدها:

Write ()	متنی را بر صفحه نمایش می دهد.
Writeln ()	متنی را بر صفحه نمایش می دهد و به سطر بعد میرود.(کسانی که با پاسکال برنامه نویسی کرده باشند به تشابه این دو دستور با دستورات پاسکال توجه نمایند.)

رویدادها:

هیچ رویدادی ندارد.

Form-۳

این شی به فرمهای موجود در صفحه اشاره می نماید. به هر فرم میتوان به صورت زیر دست پیدا کرد.

document.form [index]

که index در اصل شماره ترتیبی است که به هنگام ایجاد فرم ها به وجود می آید.

خصوصیات :

action	آدرس برنامه أی است که وظیفه پردازش فرم در هنگام زدن کلید submit را بر عهده می گیرد.
length	تعداد المانهای بکار رفته در فرم مورد نظر را نمایش می دهد .
method	روش submit را که یکی از روشهای **GET** یا **POST** می باشد را مشخص میسازد.

متدها :

submit	باعث فرستادن فرم به پردازنده server که وزیفه پردازش فرم را بر عهده دارد می شود.

رویدادها :

onSubmit	زمانی که دکمه submit فشار داده می شود روی می دهد .

Location -۴

آدرس URL صفحه جاری

خصوصیات:

href	نشان دهنده اصل URL صفحه می باشد
pathname	مسیر دیسک سروری که صفحه در آن قرار دارد.

متدها:

متدی ندارد.

رویدادها:

رویدادی بر روی آن تعریف نشده است.

navigator -۵

معرف مرورگر أی است که کاربر در حال استفاده از آن می باشد..

خصوصیات:

appName	نام مرورگر در حال استفاده
appVersion	شماره روایت مرورگر

متدها:

متدی ندارد.

رویدادها :

رویدادی برای آن تعریف نشده.

history -۶

لیست صفحات مشاهده شده را در بر دارد.

خصوصیات :

lengtht	تعداد کل صفحه های مشاهده شده را در بر دارد.

متدها :

back ()	به صفحه قبل می رود
forward ()	به صفحه بعد میرود
go (n)	به تعداد n صفحه به جلو می رود و یا در صورتی کـه عـدد n منفـی باشد به عقب می رود.

رویدادها :

برای آن رویدادی تعریف نشده است.

سایر رویدادهایی که بر اشیای روی صفحه تعریف میشوند :

در زیر سایر رویدادهایی را که بر روی انواع کـه بـر روی یک صفحه HTML قابل تعریف است را لیست کرده ایم البته در اینجا تمـام رویدادهـا را نیاورده ایم و تنها آن بخش از رویدادها کهک کاربرد بیشتری دارند ذکر شده اسـت. زیـرا که سایر رویدادها همیشه درست جواب نمیدهند و بیشتر کاربردهای خاص دارند.

۱-رویدادهای مربوط به ماوس:

ONCLICK	به محض کلیک بر روی شی مربوطه رخ می دهد .
ONDBLCLICK	در صورتی که بر روی آن شی کلیک مضاعف نماییم رخ میدهد
ONMOUSEDOWN	در صورت فشردن کلید سمت راست ماوس اتفاق می افتد.
ONMOUSEMOVE	در صورت حرکت دادن ماوس واقع می شود .
ONMOUSEOVER	در صورتی که نشانگر ماوس بر روی شی مربوطـه نشـانه رود اتفاق می افتد.
ONMOUSEOUT	در صورتی که نشانگر ماوس از محدوده شی کـه رویـداد بـرای آن تعرف شده خارج گردد اتفاق می افتد.
ONMOUSEUP	در صورتی که کلیک ماوس بر روی أین شی رها گردد اتفاق می افتد .(عکس رویداد ONCLICK)

ONDRAGSTART	زمانی که عملیات بـه همـراه کشـیدن شـی اتفـاق مـی افتـد روی میدهد .
ONSELECTSTART	زمانی که شروع به انتخاب شی مینماییم اتفاق می افتد.
ONSELECT	زمانی که شی توسط ماوس انتخاب شد أین رویداد رخ می دهد .

۲-رویدادهای مربوط به صفحه کلید:

ONKEYDOWN	زمانی که کلیدی در حالت فشار داده شده نگاه داشته می شود.
ONKEYPRESS	زمانی که کلیدی فشار داده شده و رها میگردد.
ONKEYUP	زمانی که پس از نگاه داشتن کلید آن را رها میکنیم.
ONHELP	زمانی که کلید کمک (معمولاً F1) را فشار دهیم.

۳-رویدادهای مربوط به **Focus**:

ONFOCUS	زمانی که Cursor به محدوده شی مورد اتفاق منتقل می شود.
ONBLUR	زمانی که Cursorاز محدوده شی مورد اتفاق خارج میگردد.

۴-رویدادهای مربوط به **Form** :

ONRESET	زمانی که کلید RESET زده می شود فعال میگردد.
ONSUBMIT	زمانی که دکمه SUBMIT زده می شود فعال میگردد.

۵-رویدادهای مربوط به بارگذاری صفحه :

ONABORT	زمانی که به صفحـه خاتمـه داده مـی شـود روی مـی دهـد .(بـا فشردن کلید Cancel)
ONERROR	زمانی که ایرادی در روند اجرای صفحه پیش می آید رخ میدهد .

ONLOAD	زمانی که صفحه شروع به بارگذاری میکند اتفاق می افتد.
ONUNLOAD	زمانی که به بارگذاری صفحه خاتمه داده می شود. (صفحه دیگری شروع به بارگذاری در همین پنجره می نماید یا کاربر سعی در خروج از مرورگر می نماید.)
ONAFTERUPDATE	در صورت Updateنمودن یا Refresh صفحه رخ می دهد .

جدول تطابق رویدادهای ماوس با اشیا:

در این جدول رویدادهایی که بر روی اشیای مختلف قابل تعریف میباشند را با علامت ۞ مشخص شده اند.

	Blur	Click	Change	Focus	Load	Mouse over	select	Submit	Unload
Button		۞							
Checkbox		۞			۞				۞
Document								۞	
Form						۞			
Link		۞							
Radio		۞							
Reset		۞							
Selection	۞		۞	۞					
Submit		۞							
Text	۞		۞	۞			۞		
Textarea	۞		۞	۞			۞		

حال که با اشیاء و خصوصیات و متدهای آنها آشنا شدیم و کل آنها را شناختیم میتوانیم به برنامه نویسی حرفه أی بپردازیم. اول ببینیم که چگونه میتوانیم در درون یک متن HTML شیء تعریف نمود؟

فصل سیزدهم:

تعریف و استفاده از اشیاء

حالا دیگر کاملاً آماده ایم تا به برنامه نویسی حرفه أی و خلق صفحات پویای وب بپردازیم در این فصل میخواهیم به تعریف اشیاء و انواع روشهای کدنویسی برای آنها بپردازیم.

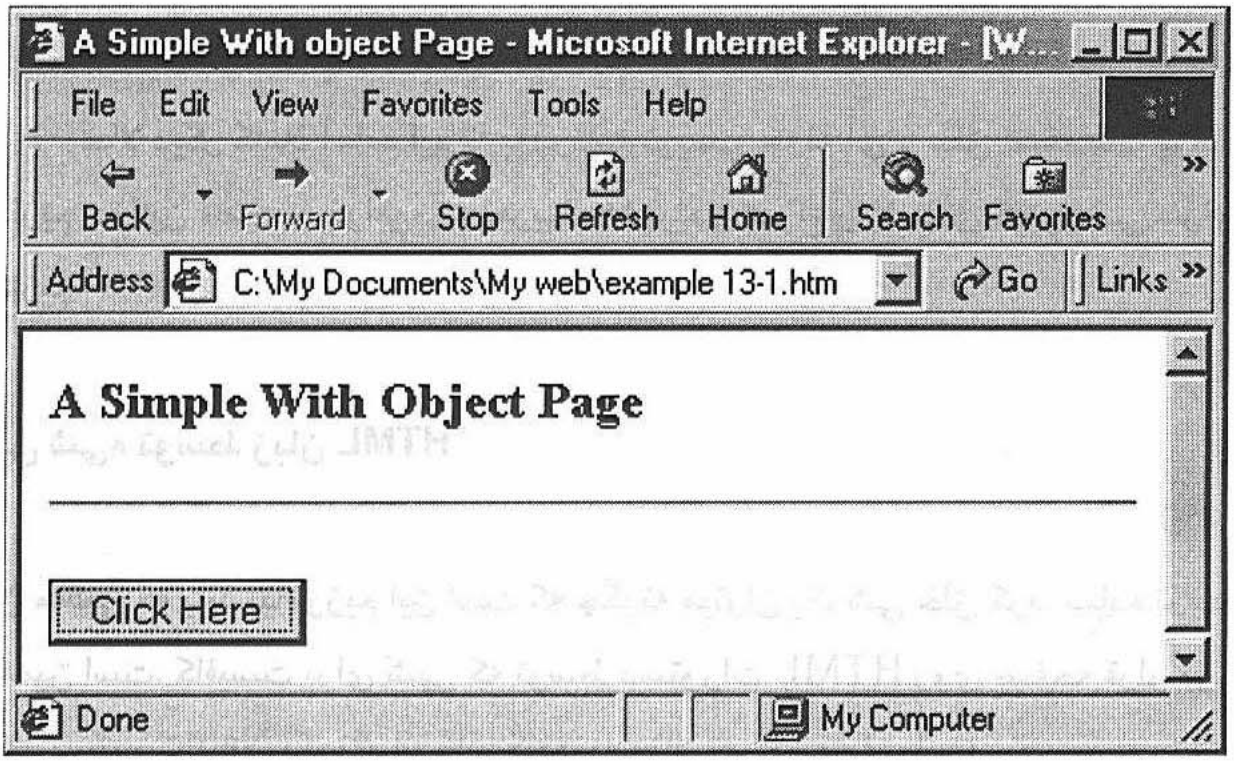
-خلق شیء توسط زبان HTML

اولین مطلبی که باید بیاموزیم این است که چگونه میتوان یک شی خلق کرد. ساده ترین کار هم همین است. کافیست برای شیی که توسط دستورات HTML روی صفحه قرار می دهیم توسط دستور Name یک نام انتخاب کنیم. به مثال زیر توجه فرمایید:

```
<HTML>
  <HEAD>
    <TITLE>A Simple With object Page</TITLE>
    <SCRIPT LANGUAGE="VBScript">
    <!--
       Sub Button1_OnClick
         MsgBox "Hey! it Functioned!!!."
       End Sub
    -->
    </SCRIPT>
  </HEAD>
  <BODY>
    <H3>
```

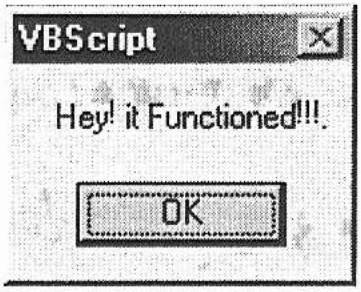
```
      A Simple With Object Page
    </H3>
    <HR>
    <FORM>
      <INPUT NAME="Button1" TYPE="BUTTON"
                    VALUE="Click Here">
    </FORM>
  </BODY>
</HTML>
```

ابتدا حاصل را ببینیم:



شکل ۱-۱۳

و پس از فشردن کلید Click Here پنجرهٔ زیر ظاهر می شود:



شکل ۲-۱۳

حال ببینیم که چه مطالبی را می توانیم در این مثال ببینیم . از ساده ترین قسمت شروع میکنیم. همانطور که میبینیم متن تابع را درون تگهای <-- --!> آورده ایم علت اینکار که اجباری هم نیست بهینه کردن هرچه بیشتر کد برنامه است زیرا به این ترتیب در صورتی که مرورگر کاربر نهایی از کدهای VBScript پشتیبانی ننماید در این صورت بدون اینکه کاربر با پیغام خطا مواجه شود مرورگر با این کد به عنوان توضیحات برخورد خواهد کرد. مطلب دوم آن است که ظاهراً در هیچ جای کد روالی را که Message Box را نمایش نمایش می دهد را فراخوانی ننموده ایم پس چگونه است که با فشار دادن کلید Button این روال فراخوانی می شود؟ اشتباه نکنید این مطلب اتفاقی نیست برای اطمینان نام روال را تغییر دهید. همانطور که میبینید دیگر درست اجرا نمی شود کمی به نام روال دقت نمایید میبینید که در اصل از دو بخش تشکیل شده است بخش اول Button1 و بخش دوم OnClick است احتمالاً اکنون موضوع روشن شده است بله ما در اصل درون متن HTML شیء کلید را ایجاد نموده ایم و به آن نام Button1 را نسبت داده ایم با رویداد OnClick هم که قبلاً در فصل ۱۲ آشنا شدیم.

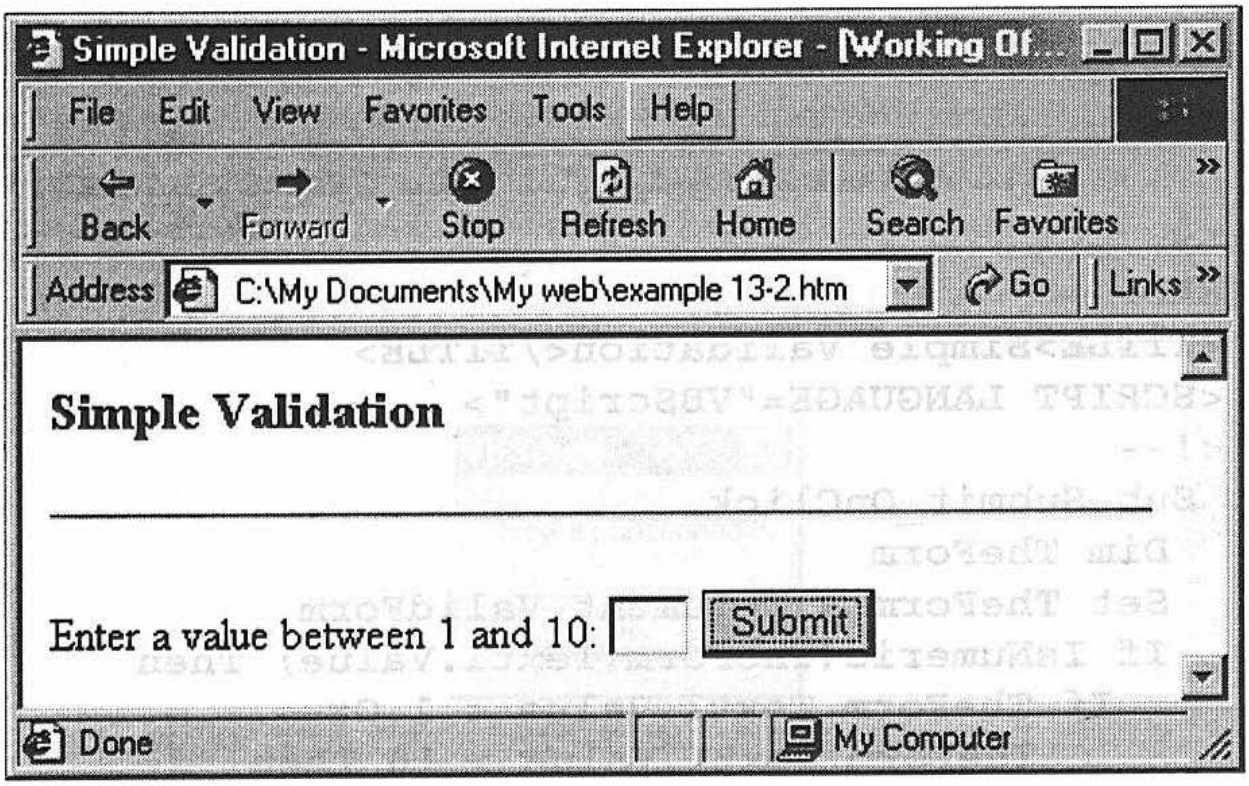
این بهترین روش استفاده از اشیاء و برنامه نویسی می باشد زیرا تنها مرورگرهایی آن را اجرا خواهند کرد که توانایی آن را داشته باشند و سایر مرورگرها هیچ مشکلی نخواهند داشت. روشی که در فصل قبل بکار بردیم نیز صحیح است ولی تنها روی مرورگرهایی درست اجرا می شود که تواماً از VBScript و JavaScript پشتیبانی نمایند.

حالا ببینیم که چگونه میتوان اشیای دیگری در صفحه ایجاد نمود.

```
<HTML>
  <HEAD>
    <TITLE>Simple Validation</TITLE>
    <SCRIPT LANGUAGE="VBScript">
    <!--
     Sub Submit_OnClick
      Dim TheForm
      Set TheForm = Document.ValidForm
      If IsNumeric(TheForm.Text1.Value) Then
        If TheForm.Text1.Value < 1 Or
           TheForm.Text1.Value > 10 Then
```

```
            MsgBox "Please enter a number
                    between 1 and 10."
          Else
             MsgBox "Thank you."
          End If
        Else
          MsgBox "Please enter a numeric value."
        End If
      End Sub
    -->
    </SCRIPT>
  </HEAD>
  <BODY>
    <H3>Simple Validation</H3><HR>
    <FORM NAME="ValidForm">
      Enter a value between 1 and 10:
      <INPUT NAME="Text1" TYPE="TEXT" SIZE="2">
      <INPUT NAME="Submit" TYPE="BUTTON"
             VALUE="Submit">
    </FORM>
  </BODY>
</HTML>
```

حاصل اجرای کد فوق چنین است :

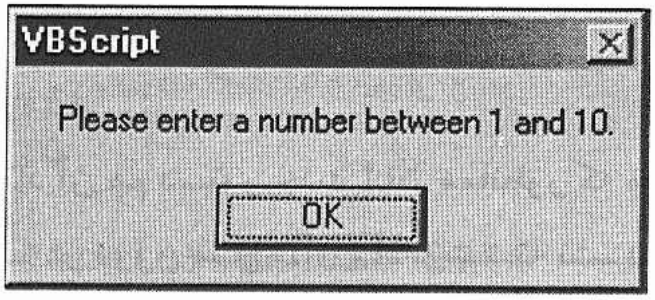


شکل ۳-۱۳

حال با ورود عددی مابین ۰ و ۱۰ با جعبه پیغام زیر مواجه میشویم:

شکل ۴-۱۳

و در صورتی که عددی خارج از این محدوده وارد نماییم با جعبه پیغام زیر مواجه میشویم.



شکل ۵-۱۳

در این مثال شیء جدیدی ایجاد نموده ایم این شیء جدید همان Text Box می باشد که از آن برای ورود اطلاعات استفاده نموده ایم قبلاً هم در فصل پنجم قول داده بودیم تا روش ساده تری نسبت به Input Box برای ورود اطلاعات ارائه نماییم. از این ساده تر می خواهید؟ به راحتی به هر تعدادی که بخواهیم میتوانیم Text Box در صفحه بگذاریم و ورودی بگیریم و حتی خروجی را نیز بر روی همانها نمایش دهیم. اما حالا بیایید ببینیم که چگونه این کار را به این سادگی انجام داده ایم؟

همانطور که میبینید مانند مثال قبل شی را با اختصاص نام به آن ایجاد نموده ایم و در هنگام استفاده در دستور If به این صورت به نوشتهٔ درون Text Box دسترسی پیدا کرده ایم:

```
TheForm.Text1.Value
```

مشاهده می نمایید که مطابق اصولی که در فصل قبل گفتیم ترتیب اشیاء را رعایت کردیم و با توجه به این که در window و Document جاری هستیم به شی Form به نام TheForm اشاره نموده ایم و از طریق آن به شی Text box به نام Text1 که درون این فرم تعریف شده است دسترسی پیدا کرده ایم اما مشکل اینجاست که خصوصیت Value در کجا تعریف شده است ؟ این خصوصیت جزو خصوصیات داخلی Text Box می باشد. زیرا همانطور که می دانید در هنگام تعریف هرکدام از این اشیاء می توانیم در متن HTML می توانیم خصوصیات آن را تعیین نماییم این خصوصیا ت همان خصوصیاتی است که به شی تعلق دارد و می توانیم از آنها در کد Script استفاده نماییم. تمام اعمالی را که در هنگام نوشتن عادی متن HTML میتوان انجام داد توسط VBScript هم به کمک خاصیت شی گرایی آن قابل انجام است.

نکته أی ازقلم افتاد آن هم دستور جدید Set. همانطور که میبینید نامی که به شی Form اختصاص داده ایم در درون متن HTML نوشته نشده است بلکه در متن VBScript و به طریق زیر آن را نامیده ایم:

```
Dim TheForm
Set TheForm = Document.ValidForm
```

ابتدا متغییر TheForm را تعریف نموده ایم سپس با کمک دستور Set از این متغییر به عنوان اشاره گری به شیء ValidForm استفاده نموده ایم. این شیء در اصل فرم فعالی است که با آن برخورد نموده ایم. میتوانستیم نام فرم را مستقیماً در درون متن HTML به آن اختصاص دهیم ولی در این صورت به جای اشاره گر به فرم به خود فرم دسترسی پیدا میکردیم در مثال فوق شاید فرق بین این دو روش احساس نشود ولی زمانی که ما به شی Form اشاره گر داشته باشیم این اشاره گر در اصل خود فرم است و هر تغییری بر روی آن مستقیماً بر روی صفحه اعمال خواهد شد در حالی که اگر شیی خلق نماییم وآن را مساوی شی Form قرار دهیم در این صورت تغییرات بر روی این شی جدید باعث تغییر Form نمی شود و میتوان به این طریق هر تغییری را بر روی فرم اعمال نمود بدون آنکه فرم اصلی تاثیر بپذیرد مثالی در همین زمینه در فصل ١٤ آمده است.

-خلق اشیایی با کنترل های Active X

به جز اشیایی که مستقیماً توسط متن HTML تعریف می شوند می توان از اشیایی که توسط کنترلهای Active X شناسایی می شوند هم استفاده نمود به عنوان مثال میتوان توسط شبه کد زیر یک شی جدید از نوع کنترلهای Active X تعریف نمود:

```
<OBJECT
        classid="clsid:99B42120-6EC7-11CF-A6C7-
                 00AA00A47DD2"
        id=lblActiveLbl
        width=250
        height=250
        align=left
        hspace=20
        vspace=0 >
  <PARAM NAME="Angle" VALUE="90">
  <PARAM NAME="Alignment" VALUE="4">
  <PARAM NAME="BackStyle" VALUE="0">
  <PARAM NAME="Caption" VALUE="A Simple Desultory
                Label">
  <PARAM NAME="FontName" VALUE="Verdana, Arial,
                Helvetica">
  <PARAM NAME="FontSize" VALUE="20">
  <PARAM NAME="FontBold" VALUE="1">
  <PARAM NAME="FrColor" VALUE="0">
</OBJECT>
```

در دستورات فوق ابتدا شیئی با Class ID مشخص تعریف نموده ایم و سپس به کمک دستور < Param> خصوصیات آن را مقدار دهی اولیه می نماییم.

به هر حال در خصوص روش استفاده از اشیاء در VBScript تغییری حاصل نمی شود همانطور که گفتیم بهتر است از Active X ها استفاده ننمایید زیرا با توجه به گسترش روز افزون ویروسها بر روی وب اکثر کاربران فعالیت Active X ها را محدود و یا به طور کلی منحل می کنند.با این حال دو مثال مفید در فصل بعد در همین زمینه گنجانده شده است.

کل VBScript در همین جا خاتمه می یابد ولی این پایان تازه آغاز راه است زیرا بدون حل تمرین و مثال نمی توان نه بر این زبان و نه هیچ مقولهٔ دیگری مسلط شد به همین دلیل فصل آخر را به طر ح مثال های متعدد اختصاص داده ایم.

فصل چهاردهم:

مثالها

همانطور که می دانید بدون حل مثال و تمرین نمی توان بر هیچ زبان برنامه نویسی مسلط شد .در این فصل به طرح مثالهای متعدد می پردازیم که در هرکدام از آنها نکات متعددی نهفته است. برخی از این مثالها حاوی عکس و یا لینک هستند که برای اجرا میتوانید عکس یا لینک مورد نظر خود را به جای آن بگذارید. و یا اینکه این مثالها را از روی دیسکت همراه کتاب اجرا نمایید. بر روی این دیسکت مثالهایی بیش از مثالهای کتاب گنجانده شده که می توانید از آنها نیز استفاده نمایید.

۱-تاریخ و ساعت

در این مثال کاربردی از تابع Now گفته می شود. ضمن آنکه تمرین خوبی برای استفاده از اشیا می باشد.

```
<HTML>
  <HEAD>
    <TITLE>Date/Time Example</TITLE>
  </HEAD>
  <BODY>
    <CENTER>
    <H3> Welcome to my Web page! </H3>
    <FORM NAME="frmMyForm">
      <INPUT TYPE=text  NAME="timeinfo">
      <INPUT TYPE=button VALUE="Update Time"
                    NAME="btnTime">
    </FORM>
    <SCRIPT LANGUAGE="vbscript">
       frmMyForm.timeinfo.value = Now
```

```
      Sub btnTime_OnClick
        frmMyForm.timeinfo.value = Now
      End Sub
    </SCRIPT>
 </BODY>
</HTML>
```

۲- محاسبهٔ هزینه خرید کالا:

در این مثال با استفاده از شی های آرایـه أی آشـنا خواهیـد شـد. ضمـن آنکـه ایـن صفحه میتواند ایدهٔ مناسبی جهت طراحی صفحات محاسباتی بدهد.

```
<HTML>
 <HEAD>
  <TITLE>
    Manipulating Data from the Server
  </TITLE>
  <SCRIPT LANGUAGE="vbscript">
    OPTION EXPLICIT
    Dim dblPrices(2)
    Dim dblShipping

    dblPrices(0) = 25.95
    dblPrices(1) = 30.50
    dblPrices(2) = 50.55
    dblShipping = 10.99

    Sub cmdCalc_OnClick
      Dim intSelected
      Dim dblTotal
      Dim dblTot

      intSelected = frmForm1.cboProducts.SelectedIndex
      dblTot = dblPrices(intSelected) *
                        frmForm1.txtQty.Value
      dblTot = dblTot + dblShipping
      frmForm1.txtTotal.Value = "$" & dblTot
   End Sub

  </SCRIPT>
</HEAD>
<BODY BGCOLOR="white">
  <FONT FACE="arial" SIZE=2><B>
```

```
    <P>
     Select the product you require from the list, then
     simply enter the quantity, and click the Calculate
     button to see the total cost.
    </P>
    <FORM NAME="frmForm1">
      <SELECT NAME="cboProducts">
        <OPTION>Product A
        <OPTION>Product B
        <OPTION>Product C
      </SELECT>
        Qty
      <INPUT TYPE="text" NAME="txtQty" SIZE=10>
      <P>
      <INPUT TYPE="button" NAME="cmdCalc"
                          VALUE="Calculate">
      <P>
      Total Cost including Shipping and Handling
      <INPUT TYPE="text" NAME="txtTotal">
  </FORM>
 </BODY>
</HTML>
```

۳-مثالی از ارتباط میان دو فریم :

متاسفانه به دلیل چند فایل بودن امکان انتشار نیست میتوانید از روی دیسک ببینید.

در این مثال روش ارتباط میان دو فریم مجزا بررسی می شود در فریم سمت راست رشته أی تایپ نمایید و با زدن کلیدی که در فریم سمت چپ قرار دارد متن تایپ شده را ببینید. و مثال خوبی برای درک مفاهیم Private و Public و متغیرهای عمومی و محلی می باشد.

۴- مثال دیگری از ارتباط میان دو فریم:

متاسفانه به دلیل چند فایل بودن امکان انتشار نیست میتوانید از روی دیسک ببینید.

در این مثال علاوه بر ارتباط میان فریم ها از توابع رشته أی نیز استفاده شده است.

۵- مثالی از استفاده از JavaScript در درون VBScript:

در این مثال از تابع Alert که تابعی مربوط به زبان JavaScript می باشد در درون کد VBScript استفاده نموده ایم.

```
<HTML>
```

```
  <HEAD>
   <SCRIPT LANGUAGE="vbscript">
     Sub MyButton_OnClick
         Alert "Hello World"
     End Sub
   </SCRIPT>
  </HEAD>

  <BODY BGCOLOR="white">
    <FORM>
     <INPUT TYPE="Button" NAME="myButton">
    </FORM>
  </BODY>
</HTML>
```

۶- مثالی از کاربرد توابع چند بعدی:

```
<HTML>
 <HEAD>
  <TITLE>Product Shipping Data</TITLE>
   <SCRIPT LANGUAGE="vbscript">

    OPTION EXPLICIT          'instruct vbscript to check all
                             variable names
     Dim strShippingData(3,2) 'declare a multidimension array
     Dim COUNTRY, WEIGHT, HAZARDS, CRTN     'declare the
                                            constants

     'assign values to the constants
     COUNTRY =0
     WEIGHT=1
     HAZARDS=2
     CRTN=Chr(10) & Chr(13)
     'assign values to the array
     strShippingData(0, COUNTRY)="Finland"
     strShippingData(1, COUNTRY)="Malawi"
     strShippingData(2, COUNTRY)="USA"
     strShippingData(3, COUNTRY)="Outer Mongolia"
     strShippingData(0,WEIGHT)="Weight = 34 Kilos"
     strShippingData(1,WEIGHT)="Weight = 17 Kilos"
     strShippingData(2,WEIGHT)="Weight = 10 Kilos"
     strShippingData(3,WEIGHT)="Weight = 15 Kilos"
     strShippingData(0,HAZARDS)="No Hazard"
     strShippingData(1,HAZARDS)="Highly Inflammable"
     strShippingData(2,HAZARDS)="No Hazard"
     strShippingData(3,HAZARDS)="Highly Inflammable"

     'declare a sub routine that can be used by all the
buttons
```

```
      Sub Shipping(Index)
         'declare variables to be used in this sub
         Dim x, y, strMessage, strCaption
         'we want a line for each data item - use the
          constants
         For x = COUNTRY TO HAZARDS
            strMessage = strMessage & strShippingData(Index,x)
            strMessage = strMessage & CRTN
         Next
         'construct a caption from the index number
         strCaption = "Shipping Data for Product No." &
                      CStr(Index + 1)
         'finally, show a message box to the user
         y = MsgBox(strMessage,0,strCaption)
      End Sub
    </SCRIPT>
  </HEAD>

  <BODY BGCOLOR="white">
   <FONT FACE="ARIAL" SIZE=3>
     <B>Product No.1</B><BR>
     Details of the first product go here
     <INPUT TYPE="BUTTON" NAME="cmdShipping"
OnClick="Shipping(0)"
          VALUE="Click for Shipping Data"><P>
     <B>Product No.2</B><BR>
     Details of the second product go here
     <INPUT TYPE="BUTTON" NAME="cmdShipping"
OnClick="Shipping(1)"
            VALUE="Click for Shipping Data"><P>
     <B>Product No.3</B><BR>
     Details of the third product go here
     <INPUT TYPE="BUTTON" NAME="cmdShipping"
OnClick="Shipping(2)"
          VALUE="Click for Shipping Data"><P>
     <B>Product No.4</B><BR>
     Details of the fourth product go here
     <INPUT TYPE="BUTTON" NAME="cmdShipping"
OnClick="Shipping(3)"
          VALUE="Click for Shipping Data"><P>
   </FONT>
  </BODY>
</HTML>
```

۷- مثالی از متغیرهای Private و Local, Global, Public

متاسفانه به دلیل چند فـایلی بـودن امکـان انتشـار نیسـت میتوانیـد از روی دیسـک ببینید. در این مثال متغیرهای Public و Private تعریف شده اند و بـا اسـتفاده از کلیـد بـه آنها دسترسی پیدا میکنیم مورد خطا هم مثال زده شده.

۸-مثالی ازآرایه های با طول متغیر

در این مثال با روش تعریف مقدار دهـی و تغییـر ابعـاد ماتریسـهای بـا ابعـاد شـناور آشـنا میشویم:

```
<HTML>
  <HEAD>
    <TITLE>Dynamic Array Application</TITLE>
    <SCRIPT LANGUAGE="vbscript">
       OPTION EXPLICIT 'require all variables to be
                       declared
       ReDim myArray(0)     'create a dynamic array with
                       1 elemen
       Dim intIndex          'declare variable to track
                       the array index number
       Dim CRTN         'variable for a neat Carriage
                       Return trick in VBScript 1
       CRTN = Chr(10) & Chr(13) 'assign carriage return
                       to CRTN
       intIndex = 0       'assign the first index number
                       to our counter

       Sub cmdButton1_OnClick
          myArray(intIndex) =
                       Document.frmForm1.txtText1.Value
                       'Store the user input in the
                       array
          intIndex = intIndex + 1      'increment the
                       array counter by one
          ReDim Preserve myArray(intIndex)
                            'increase the size of the
                       array
          Document.frmForm1.txtText1.Value = ""
          'Empty the text box again
       End Sub
```

```
        Sub cmdButton2_OnClick
          Dim x, y, strArrayContents    'declare some
                        variables we're going to need

          'repeat this process as many times as there
          'are array elements
          'note: the last element will always be empty
          'because we've incremented the counter
          '*after* the assignment.
          'try changing the above sub so that we always
          'fill every element
           For x = 0 to intIndex - 1
            'assign a short description and the element
            'no to the variable
            strArrayContents =  strArrayContents &
                        "Element No." & CStr(x) & " = "
            'add to this the contents of the element and
            'our carriage return
            strArrayContents =  strArrayContents &
                        myArray(x) & CRTN
            'go back and do it again for the next value
            'of x
          Next
          'when we've done show the result in a
          'message box
          y = MsgBox(strArrayContents,0,"Dynamic Array
                        Application")
        End Sub
      </SCRIPT>
    </HEAD>
    <BODY BGCOLOR="white">
      <FORM NAME="frmForm1">
        <INPUT TYPE="text" NAME="txtText1"><BR>
        <INPUT TYPE="button" NAME="cmdButton1"
                        VALUE="Add to array"><P>
        <INPUT TYPE="button" NAME="cmdButton2"
                        VALUE="Show Array Contents">
      </FORM>
    </BODY>
</HTML>
```

۱۰-مثالی از رویداد های مربوط به Focus و Change

در این مثال با کاربرد بعضی از رویدادها آشنا میشویم.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="vbscript">
Sub cmdButton1_OnClick
 Dim CRLF
 Dim strProperties
 CRLF = Chr(13) & Chr(10)
 strProperties = "Name: "
 strProperties = strProperties & Document.frmMyForm.txtTestText1.Name &
CRLF
 strProperties = strProperties & "Form Elements: "
 If InStr(1, navigator.useragent, "MSIE 3") > 0 Then
  strProperties = strProperties &
Document.frmMyForm.txtTestText1.Form.Elements.Count & CRLF
 Else
  strProperties = strProperties &
Document.frmMyForm.txtTestText1.Form.Elements.Length & CRLF
 End If
 strProperties = strProperties & "Value: "
 strProperties = strProperties & Document.frmMyForm.txtTestText1.Value
 x = MsgBox(strProperties,0,"Test Box 1 Properties")
End Sub

Sub cmdButton2_OnClick
 Document.frmMyForm.txtTestText2.Focus
End Sub

Sub cmdButton3_OnClick
 Document.frmMyForm.txtTestText1.Select
End Sub

Sub txtTestText1_OnChange
 Alert "I've changed"
End Sub

Sub txtTestText1_OnBlur
 Document.frmMyForm.txtTestText2.Value =
Document.frmMyForm.txtTestText1.Value
End Sub

Sub txtTestText1_OnSelect
 Alert "I'm selected"
End Sub
```

```
  Sub txtTestText1_OnFocus
    Alert "I've got the Focus"
  End Sub
 </SCRIPT>
</HEAD>

   <BODY BGCOLOR="white">
 <FORM NAME="frmMyForm">
  Test Text 1<INPUT TYPE="text" NAME="txtTestText1"><BR>
  Test Text 2<INPUT TYPE="text" NAME="txtTestText2"><BR>
  <INPUT TYPE="button" NAME="cmdButton1" Value="Display Text Box 1
Properties"><P>
  <INPUT TYPE="button" NAME="cmdButton2" Value="Focus to Text Box
1"><P>
  <INPUT TYPE="button" NAME="cmdButton3" Value="Select Text Box
1"><P>
 </FORM>
</BODY>
</HTML>
```

۱۱- مثالی از کاربرد VBScript برای تغییر رنگهای صفحه :

در این مثال کاربرد VBScript در تغییر رنگها را خواهیم دید ضمن آنکه با تغییر مشخصات متن بدون نوشتن کد HTML جدید آشنا خواهید شد.

```
<HTML>
  <HEAD>
     <TITLE>
          This an Example of how to change window's
color
     </TITLE>
  </HEAD>
  <BODY>
  <H1>
    Example 2
  </H1>
    Clicking a button executes a subroutine that
randomly changes
    the background or foreground color of this page.
    <INPUT name=btnFG type=button value=Foreground>
    <INPUT name=btnBG type=button value=Background>
    <SCRIPT language=VBScript>
      Sub btnFG_OnClick
```

```
        Color = hex(rnd()*16777215)
        Document.FGColor = Color
        btnFG.Value = Color
      End Sub
      Sub btnBG_OnClick
        Color = hex(rnd()* 16777215)
        Document.BGColor = Color
        btnBG.Value = Color
      End Sub
  </SCRIPT>
 </BODY>
</HTML>
```

۱۲- مثالی از کاربرد تابع DateAdd

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="vbscript">
      OPTION EXPLICIT
      dim vDateOption
      Sub cmdButton1_onClick
                    frmForm1.txtText1.Value =
                    dateAdd(vDateOption,
                    frmForm1.txtDiffValue.Value,
                    Now())
      End Sub
    </SCRIPT>
  </HEAD>
  <BODY BGCOLOR="white">
    <FONT FACE="arial" SIZE=2>
       <CENTER>
         <H3>DateAdd</H3>
         <FORM NAME="frmForm1">
            Enter a value to add (or use a negative
                    value to subtract)<BR>
            from today's date.<BR>
            <INPUT TYPE="text"
                    NAME="txtDiffValue"><BR><BR>
            Then select a date/time unit to perform
                    the calculation on<BR>
            <TABLE>
```

```
<TR>
  <TD><INPUT LANGUAGE="vbscript"
        TYPE=radio VALUE="yyyy"
        ONCLICK="vDateOption =
        &quot;yyyy&quot;"
        NAME="rdoOption"></TD>
  <TD><FONT SIZE=2>Year</TD>
  <TD><INPUT LANGUAGE="vbscript"
        TYPE=radio VALUE="q"
        ONCLICK="vDateOption =
        &quot;q&quot;"
        NAME="rdoOption"></TD>
  <TD><FONT SIZE=2>Quarter</TD>
</TR>

<TR>
  <TD><INPUT LANGUAGE="vbscript"
        TYPE=radio VALUE="m"
        ONCLICK="vDateOption =
        &quot;m&quot;"
        NAME="rdoOption"></TD>
  <TD><FONT SIZE=2>Month</TD>
  <TD><INPUT LANGUAGE="vbscript"
        TYPE=radio VALUE="y"
        ONCLICK="vDateOption =
        &quot;y&quot;"
        NAME="rdoOption"></TD>
  <TD><FONT SIZE=2>Day of year</TD>
</TR>

<TR>
  <TD><INPUT LANGUAGE="vbscript"
        TYPE=radio VALUE="d"
        ONCLICK="vDateOption =
        &quot;d&quot;"
        NAME="rdoOption"></TD>
  <TD><FONT SIZE=2>Day</TD>
  <TD><INPUT LANGUAGE="vbscript"
        TYPE=radio VALUE="w"
        ONCLICK="vDateOption =
```

```
                &quot;w&quot;"
                NAME="rdoOption"></TD>
            <TD><FONT SIZE=2>Weekday</TD>
        </TR>

        <TR>
          <TD><INPUT LANGUAGE="vbscript"
                TYPE=radio VALUE="ww"
                ONCLICK="vDateOption =
                &quot;ww&quot;"
                NAME="rdoOption"></TD>
          <TD><FONT SIZE=2>Week of year</TD>
          <TD><INPUT LANGUAGE="vbscript"
                TYPE=radio VALUE="h"
                ONCLICK="vDateOption =
                &quot;h&quot;"
                NAME="rdoOption"></TD>
          <TD><FONT SIZE=2>Hour</TD>
        </TR>

        <TR>
          <TD><INPUT LANGUAGE="vbscript"
                TYPE=radio VALUE="n"
                ONCLICK="vDateOption =
                &quot;n&quot;"
                NAME="rdoOption"></TD>
          <TD><FONT SIZE=2>Minute</TD>
          <TD><INPUT LANGUAGE="vbscript"
                TYPE=radio VALUE="s"
                ONCLICK="vDateOption =
                &quot;s&quot;"
                NAME="rdoOption"></TD>
          <TD><FONT SIZE=2>Second</TD>
        </TR>
  </TABLE>
  <INPUT TYPE=button VALUE="Click me"
                NAME="cmdButton1">
  <BR><BR>
  Result.... <INPUT TYPE=text NAME="txtText1">
  <BR><BR><BR>
</FORM>
```

```
 </BODY>
</HTML>
```

۱۳- مثالی جهت تعین سرعت VBScript بر روی سیستم

این مثال در ابتدا عددی را از کاربر می گیرد و سپس تمام اعداد اول قبل ازآن را محاسبه و در آرایه أی با طول متغیر می ریزد در پایان اعداد و زمان انجام محاسبه را نمایش می دهد.

```
<HTML>
  <HEAD>
    <TITLE>Timer</TITLE>
      <SCRIPT LANGUAGE="vbscript">
  Sub cmdCommand1_OnClick
          Dim t
          Dim sSeive

          t = Timer
          sSeive = doSeive(frmForm1.
                        txtMaxNumber.Value)
          Alert sSeive & " - processed in " &
                        (Timer-t) & " seconds"
        End Sub

        Function doSeive(iMax)
          ReDim iNumbers(iMax)
          Dim i, j
          Dim sReturn

          For i = 2 to iMax
            For j = 2 to iMax
             If cInt((i * j)) > CInt(iMax) Then
                  Exit For
             Else
                  iNumbers(cInt(i * j)) = 1
             End if
            Next
          Next

          For i = 1 to iMax
            If iNumbers(i) <> 1 Then
              If sReturn = "" Then
```

```
                    sReturn = i
                Else
                    sReturn = sReturn & "," & i
                End If
              End if
            Next
            DoSeive = sReturn
          End Function
      </SCRIPT>
  </HEAD>
  <BODY BGCOLOR="white">
    <CENTER>
      <H3>The Sieve of Eratosthenes</H3>
      <BR>
      <FORM NAME="frmForm1">
          Find Prime Numbers up to
          <INPUT TYPE="text" NAME="txtMaxNumber">
            
          <INPUT TYPE="button" NAME="cmdCommand1"
                    VALUE="Run">
      </FORM>
    </CENTER>
  </BODY>
</HTML>
```

۱۴- مثالی جهت استفاده از کنترلهای Active X

در این مثال که به دلیل داشتن شکل بهتر است از روی دیسک اجـرا شـود بـا کمـک کنترلهای Active X اطلاعات مربوط به لینکهای فعال را نمایش می دهیم.

```
<HTML>
  <HEAD>
    <TITLE>Link Descriptions</TITLE>
      <SCRIPT language="vbscript">
        Dim DescribeLink(5)
        DescribeLink(0) = "Find out all about us"
        DescribeLink(1) = "So good you'll want to
                    stay for life"
        DescribeLink(2) = "The under 17's stay
                    longer for free"
        DescribeLink(3) = "Our new Hotel now open
                    in Communicado"
```

```
DescribeLink(4) = "Frequently Asked
          Questions about Desert
          Holidays"
DescribeLink(5) = "All the lastest news
          and gossip"

Sub ShowLink(LinkNo)
  Label1.Caption =
          DescribeLink(CInt(LinkNo))
  Status = DescribeLink(CInt(LinkNo))
  tID = setTimeout("ClearDesc()",4000)
End Sub

Sub Link4_MouseMove(s,b,x,y)
   Label1.Caption = DescribeLink(3)
   Status = DescribeLink(3)
   tID = setTimeout("ClearDesc()",4000)
End Sub

Sub Link4_onMouseMove(s,b,x,y)
   Label1.Caption = DescribeLink(3)
   Status = DescribeLink(3)
   tID = setTimeout("ClearDesc()",4000)
End Sub

Sub Link5_MouseMove(s,b,x,y)
   Label1.Caption = DescribeLink(4)
   Status = DescribeLink(4)
   tID = setTimeout("ClearDesc()",4000)
End Sub

Sub Link5_onMouseMove(s,b,x,y)
   Label1.Caption = DescribeLink(4)
   Status = DescribeLink(4)
   tID = setTimeout("ClearDesc()",4000)
End Sub

Sub Link6_MouseMove(s,b,x,y)
   Label1.Caption = DescribeLink(5)
   Status = DescribeLink(5)
```

```
      tID = setTimeout("ClearDesc()",4000)
    End Sub

    Sub Link6_onMouseMove(s,b,x,y)
      Label1.Caption = DescribeLink(5)
      Status = DescribeLink(5)
      tID = setTimeout("ClearDesc()",4000)
    End Sub

    Sub ClearDesc()
      Status = ""
      Label1.Caption = ""
    End Sub
  </SCRIPT>
</HEAD>

<BODY BGCOLOR="#80FFFF" TEXT="#FF0000">
  <P>
  <FONT FACE="arial" SIZE=2>
    <CENTER>
      <H2>Khalifa Desert Holidays</H2>
      <P>
      <OBJECT ID="Label1" WIDTH=439 HEIGHT=24
      CLASSID="CLSID:978C9E23-D4B0-11CE-BF2D-
               00AA003F40D0">
        <PARAM NAME="ForeColor" VALUE="33023">
        <PARAM NAME="BackColor"
               VALUE="16777088">
        <PARAM NAME="Size" VALUE="11610;635">
        <PARAM NAME="FontEffects"
               VALUE="1073741827">
        <PARAM NAME="FontHeight" VALUE="240">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily"
               VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="700">
      </OBJECT>
      <P>
      <TABLE WIDTH=80%>
        <TD ALIGN=LEFT>
```

```
                <B>
                <A LANGUAGE="vbscript" NAME="Link1"
                        OnMouseOver="call showLink(0)"
                        HREF="">About Khalifa Desert
                        Holidays</A><BR>
                <A LANGUAGE="vbscript" NAME="Link2"
                        OnMouseOver="call showLink(1)"
                        HREF="">Rates and
                        Tariffs</A><BR>
                <A LANGUAGE="vbscript" NAME="Link3"
                        OnMouseOver="call showLink(2)"
                        HREF="">Special Deals for
                        Children</A><BR>
                <A NAME="Link4" HREF="" VALUE=3>Some of
                        Our Fine Hotels</A><BR>
                <A NAME="Link5" HREF="" VALUE=4>The
                        Desert Holiday FAQ</A><BR>
                <A NAME="Link6" HREF="">What's
                        New</A><BR>
                <TD ALIGN=CENTER>
                <IMG SRC="desert.gif">
                <TR>
            </TABLE>
    </BODY>
</HTML>
```

۱۵-مثالی در مورد نحوه استفاده از Error handler ها

در این مثال استفاده از Error handler ها جهت عدم مواجههٔ کـاربر بـا خطاهـای سیسـتمی مشاهده میگردد مخصوصاً در کد اشتباه قرار داده شده تا با پیغام خطا مواجه شـوید.ضمنـاً قواعد برنامه نویسی دندانه أی هم رعایت نشده تا به عینه مشـکلات خوانـدن چنیـن متنـی را ببینید.

```
<HTML>
<HEAD>
<TITLE>Using the Err Object</TITLE>
<SCRIPT LANGUAGE="vbscript">

Function DoErrMsg()

Dim Msg, MsgBoxType
```

```
  Dim CRLF
  CRLF = Chr(13) & Chr(10)
  Msg = "A serious error - "
  Msg = Msg & Err.Description & CRLF
  Msg = Msg & "has occured in "
  Msg = Msg & Err.Source & CRLF
  If Err.Number = 11 Then
  Msg = Msg & "Halting Execution"
  MsgBoxType = 16
  DoErrMsg = "Fatal"
  Else
  Msg = Msg & "Continuing Execution"
  MsgBoxType = 48
  DoErrMsg = "OK"
  End If
  MsgBox Msg, 48, "My Script Error"
  Err.Clear
  End Function
Sub cmdButton1_OnClick
Dim x, y ,z
On Error Resume Next
x = 10
y = 0
z = x / y
If Err.Number <> 0 Then
If DoErrMsg() = "Fatal" Then
Exit Sub
End IF
End If

Alert z

End Sub

</SCRIPT>
</HEAD>
<BODY BGCOLOR="white">
<CENTER>
<H2>An Error Handler</H2>
<P>
<INPUT TYPE="button" NAME="cmdButton1" VALUE="OK">
```

```
</CENTER>
</BODY>
</HTML>
```

۱۶-طراحی یک سایت فروش گل به طور کامل.

به دلیل تعداد زیاد عکسها و فایلهای مورد نیاز این مثال متن آن تنها بر روی دیسک موجود می باشد . مثالی فوق العاده که علاوه بر آموزش نکات بسیار زیاد در خصوص VBScript ایدهٔ بسیار مناسبی هم در مورد طراحی سایتهای تجاری با ساختار Frame های نامریی می دهد. از روش ارایه شده در این مثال میتوان جهت طراحی هر نوع سایت دیگر به کمک فریم ها بهره جست.

۱۷- بازی !!!!

تعجب نکنید این مثال طراحی یک بلزی قدیمی می باشد توضیحات بازی در زیر صفحهٔ آن آمده است. البته کمتر کسی از این زبان که برای برنامه نویسی روی وب می باشد برای بازی کردن استفاده می نماید ولی نکات بسیار زیادی در این مثال نهفته است که مهمترین آنها تعریف اشیاء خارج از HTML و استفاده گسترده از توابع و دستورات می باشد. هرچند که متن این برنامه طولانی است ولی توصیه می نماییم ابتدا متن آن را تا پایان خوانده پس از اینکه سرتان گیج رفت با آن بازی کنید!!!! (هرچند که این بازی فکری است نه سر گرمی)

```
<HTML>
 <HEAD>
  <TITLE>Light-It-Up!</TITLE>
 </HEAD>
 <BODY bgColor=#ffffff>
  <SCRIPT language=VBScript>
  <!--
    Option Explicit

    Sub window_onLoad()
      Set Grid(1,1) = Button1
          Set Grid(1,2) = Button2
          Set Grid(1,3) = Button3
          Set Grid(1,4) = Button4
          Set Grid(1,5) = Button5
```

```
        Set Grid(2,1) = Button6
        Set Grid(2,2) = Button7
        Set Grid(2,3) = Button8
        Set Grid(2,4) = Button9
        Set Grid(2,5) = Button10
        Set Grid(3,1) = Button11
        Set Grid(3,2) = Button12
        Set Grid(3,3) = Button13
        Set Grid(3,4) = Button14
        Set Grid(3,5) = Button15
        Set Grid(4,1) = Button16
        Set Grid(4,2) = Button17
        Set Grid(4,3) = Button18
        Set Grid(4,4) = Button19
        Set Grid(4,5) = Button20
        Set Grid(5,1) = Button21
        Set Grid(5,2) = Button22
        Set Grid(5,3) = Button23
        Set Grid(5,4) = Button24
        Set Grid(5,5) = Button25

        White = RGB(255,255,255)
        Black = RGB(0,0,0)
    end sub
    -->
</SCRIPT>

<SCRIPT language=VBScript>
<!--
    Sub Flip(GridX, GridY)
            If GridX >= 1 And GridX <= 5 And _
              GridY >= 1 And GridY <= 5 Then
                If Grid(GridX, GridY).BackColor = White Then
                    Grid(GridX, GridY).BackColor = Black
                Else
                    Grid(GridX, GridY).BackColor = White
                End If
            End If
    end sub
    Sub Splash(GridX,GridY)
            Flip GridX,GridY
            Flip GridX+1,GridY
            Flip GridX-1,GridY
            Flip GridX,GridY+1
            Flip GridX,GridY-1
            ClickLabel.Caption = CStr(Int(ClickLabel.Caption)+1)
            CheckForWin
    end sub
    dim Grid(5,5)
```

```
-->
</SCRIPT>

<H1>Light-It-Up!</H1>
<P>
<SCRIPT language=VBScript>
<!--
      Sub Button1_Click()
            Splash 1,1
      end sub
-->
</SCRIPT>

<OBJECT align=baseline
      Classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
      height=32 id=Button1 width=49 border="0">
      <PARAM NAME="BackColor" VALUE="0">
      <PARAM NAME="Size" VALUE="1305;847">
      <PARAM NAME="FontCharSet" VALUE="0">
      <PARAM NAME="FontPitchAndFamily" VALUE="2">
      <PARAM NAME="ParagraphAlign" VALUE="3">
      <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
 Sub Button2_Click()
            Splash 1,2
     end sub
-->
</SCRIPT>

<OBJECT align=baseline
      Classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
      height=32 id=Button2 width=49 border="0">
      <PARAM NAME="BackColor" VALUE="0">
      <PARAM NAME="Size" VALUE="1291;846">
      <PARAM NAME="FontCharSet" VALUE="0">
      <PARAM NAME="FontPitchAndFamily" VALUE="2">
      <PARAM NAME="ParagraphAlign" VALUE="3">
      <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
     Sub Button3_Click()
            Splash 1,3
     end sub
-->
</SCRIPT>
```

```
<OBJECT align=baseline
    classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
    height=32 id=Button3 width=49 border="0">
    <PARAM NAME="BackColor" VALUE="0">
    <PARAM NAME="Size" VALUE="1291;846">
    <PARAM NAME="FontCharSet" VALUE="0">
    <PARAM NAME="FontPitchAndFamily" VALUE="2">
    <PARAM NAME="ParagraphAlign" VALUE="3">
    <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
    Sub Button4_Click()
        Splash 1,4
    end sub
-->
</SCRIPT>

<OBJECT align=baseline
    Classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
    height=32 id=Button4 width=49 border="0">
    <PARAM NAME="BackColor" VALUE="0">
    <PARAM NAME="Size" VALUE="1291;846">
    <PARAM NAME="FontCharSet" VALUE="0">
    <PARAM NAME="FontPitchAndFamily" VALUE="2">
    <PARAM NAME="ParagraphAlign" VALUE="3">
    <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
    Sub Button5_Click()
        Splash 1,5
    end sub
-->
</SCRIPT>

<OBJECT align=baseline
    classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
    height=32 id=Button5 width=49 border="0">
    <PARAM NAME="BackColor" VALUE="0">
    <PARAM NAME="Size" VALUE="1291;846">
    <PARAM NAME="FontCharSet" VALUE="0">
    <PARAM NAME="FontPitchAndFamily" VALUE="2">
    <PARAM NAME="ParagraphAlign" VALUE="3">
    <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<OBJECT align=baseline
    classid=CLSID:978C9E23-D4B0-11CE-BF2D-00AA003F40D0
```

```
        height=39 id=ClickLabel width=127 border="0">
        <PARAM NAME="BackColor" VALUE="16777215">
        <PARAM NAME="Caption" VALUE="0">
        <PARAM NAME="Size" VALUE="3351;1023">
        <PARAM NAME="FontEffects" VALUE="1073741825">
        <PARAM NAME="FontHeight" VALUE="480">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="2">
        <PARAM NAME="FontWeight" VALUE="700">
</OBJECT>
<BR>
<SCRIPT language=VBScript>
<!--
      Sub Button6_Click()
            Splash 2,1
      end sub
      -->
</SCRIPT>

<OBJECT align=baseline
      classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
      height=32 id=Button6 width=49 border="0">
      <PARAM NAME="BackColor" VALUE="0">
      <PARAM NAME="Size" VALUE="1291;846">
      <PARAM NAME="FontCharSet" VALUE="0">
      <PARAM NAME="FontPitchAndFamily" VALUE="2">
      <PARAM NAME="ParagraphAlign" VALUE="3">
      <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
      Sub Button7_Click()
            Splash 2,2
      end sub
-->
</SCRIPT>
<OBJECT align=baseline
      classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
      height=32 id=Button7 width=49 border="0">
      <PARAM NAME="BackColor" VALUE="0">
      <PARAM NAME="Size" VALUE="1291;846">
      <PARAM NAME="FontCharSet" VALUE="0">
      <PARAM NAME="FontPitchAndFamily" VALUE="2">
      <PARAM NAME="ParagraphAlign" VALUE="3">
      <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
```

```
        Sub Button8_Click()
                Splash 2,3
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button8 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
        Sub Button9_Click()
                Splash 2,4
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button9 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
        Sub Button10_Click()
                Splash 2,5
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button10 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
```

```
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
    </OBJECT>
    <BR>
    <SCRIPT language=VBScript>
    <!--
        Sub Button11_Click()
            Splash 3,1
        end sub
    -->
    </SCRIPT>

    <OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button11 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
    </OBJECT>
    <SCRIPT language=VBScript>
    <!--
        Sub Button12_Click()
            Splash 3,2
        end sub
    -->
    </SCRIPT>

    <OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button12 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
    </OBJECT>
    <SCRIPT language=VBScript>
    <!--
        Sub Button13_Click()
            Splash 3,3
        end sub
    -->
    </SCRIPT>

    <OBJECT align=baseline
```

```
      classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
      height=32 id=Button13 width=49 border="0">
      <PARAM NAME="BackColor" VALUE="0">
      <PARAM NAME="Size" VALUE="1291;846">
      <PARAM NAME="FontCharSet" VALUE="0">
      <PARAM NAME="FontPitchAndFamily" VALUE="2">
      <PARAM NAME="ParagraphAlign" VALUE="3">
      <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
      Sub Button14_Click()
            Splash 3,4
      end sub
-->
</SCRIPT>

<OBJECT align=baseline
      classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
      height=32 id=Button14 width=49 border="0">
      <PARAM NAME="BackColor" VALUE="0">
      <PARAM NAME="Size" VALUE="1291;846">
      <PARAM NAME="FontCharSet" VALUE="0">
      <PARAM NAME="FontPitchAndFamily" VALUE="2">
      <PARAM NAME="ParagraphAlign" VALUE="3">
      <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
      Sub Button15_Click()
            Splash 3,5
      end sub
-->
</SCRIPT>

<OBJECT align=baseline
      classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
      height=32 id=Button15 width=49 border="0">
      <PARAM NAME="BackColor" VALUE="0">
      <PARAM NAME="Size" VALUE="1291;846">
      <PARAM NAME="FontCharSet" VALUE="0">
      <PARAM NAME="FontPitchAndFamily" VALUE="2">
      <PARAM NAME="ParagraphAlign" VALUE="3">
      <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<BR>
<SCRIPT language=VBScript>
<!--
      Sub Button16_Click()
```

```
                Splash 4,1
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button16 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
        Sub Button17_Click()
                Splash 4,2
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button17 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
        Sub Button18_Click()
                Splash 4,3
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button18 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
```

```
    <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
    Sub Button19_Click()
        Splash 4,4
    end sub
-->
</SCRIPT>

<OBJECT align=baseline
    classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
    height=32 id=Button19 width=49 border="0">
    <PARAM NAME="BackColor" VALUE="0">
    <PARAM NAME="Size" VALUE="1291;846">
    <PARAM NAME="FontCharSet" VALUE="0">
    <PARAM NAME="FontPitchAndFamily" VALUE="2">
    <PARAM NAME="ParagraphAlign" VALUE="3">
    <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
    Sub Button20_Click()
        Splash 4,5
    end sub
-->
</SCRIPT>

<OBJECT align=baseline
    classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
    height=32 id=Button20 width=49 border="0">
    <PARAM NAME="BackColor" VALUE="0">
    <PARAM NAME="Size" VALUE="1291;846">
    <PARAM NAME="FontCharSet" VALUE="0">
    <PARAM NAME="FontPitchAndFamily" VALUE="2">
    <PARAM NAME="ParagraphAlign" VALUE="3">
    <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<BR>
<SCRIPT language=VBScript>
<!--
    Sub Button21_Click()
        Splash 5,1
    end sub
-->
</SCRIPT>

<OBJECT align=baseline
    classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
```

```
        height=32 id=Button21 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
        Sub Button22_Click()
                Splash 5,2
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button22 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
        Sub Button23_Click()
                Splash 5,3
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button23 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
```

```
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button24 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
        Sub CheckForWin()
                Dim Rows,Cols, Win
                Win = True
                For Rows = 1 To 5
                  For Cols = 1 To 5
                    If Grid(Rows,Cols).BackColor = Black Then Win = False
                  Next
                Next

                If Win = True Then
                  MsgBox "You Win! And it only took you " & ClickLabel.Caption & "
clicks!"
                End If
        end sub
        Sub Button25_Click()
                Splash 5,5
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=Button25 width=49 border="0">
        <PARAM NAME="BackColor" VALUE="0">
        <PARAM NAME="Size" VALUE="1291;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
</P>
<P>
<SCRIPT language=VBScript>
<!--
```

```
        Sub AllOutButton_Click()
                ColorGrid(Black)
                ClickLabel.Caption = "0"
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=AllOutButton width=96 border="0">
        <PARAM NAME="Caption" VALUE="All Lights Out">
        <PARAM NAME="Size" VALUE="2540;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
<!--
        Sub RandomStartButton_Click()
                Dim Rows,Cols

                Randomize
                For Rows = 1 To 5
                  For Cols = 1 To 5
                    If Rnd > .5 Then
                      Grid(Rows,Cols).BackColor = Black
                    Else
                      Grid(Rows,Cols).BackColor = White
                    End If
                  Next
                Next

                ClickLabel.Caption = "0"
        end sub
-->
</SCRIPT>

<OBJECT align=baseline
        classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
        height=32 id=RandomStartButton width=96 border="0">
        <PARAM NAME="Caption" VALUE="Random Start">
        <PARAM NAME="Size" VALUE="2540;846">
        <PARAM NAME="FontCharSet" VALUE="0">
        <PARAM NAME="FontPitchAndFamily" VALUE="2">
        <PARAM NAME="ParagraphAlign" VALUE="3">
        <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
<SCRIPT language=VBScript>
```

```
<!--
    Sub ColorGrid(Color)
        Dim Rows,Cols

        For Rows = 1 To 5
          For Cols = 1 To 5
            Grid(Rows,Cols).BackColor = Color
          Next
        Next

    end sub
    dim White

    dim Black

    Sub EasyStartButton_Click()
        Dim Rows, Cols, Choice

        Randomize

        Choice = Int(Rnd * 5)+1
        Select Case Choice
            Case 1
              ColorGrid(Black)
              For Rows = 1 To 5
                Grid(Rows, 3).BackColor = White
              Next
              For Cols = 1 To 5
                Grid(2, Cols).BackColor = White
                Grid(4, Cols).BackColor = White
              Next
            Case 2
              ColorGrid(White)
              Grid(1,3).BackColor = Black
              Grid(3,3).BackColor = Black
              Grid(5,3).BackColor = Black
            Case 3
              ColorGrid(White)
              For Rows = 1 To 5
                If Rows <> 3 Then
                  Grid(Rows, 2).BackColor = Black
                  Grid(Rows, 5).BackColor = Black
                End If
              Next
              Grid(1,4).BackColor = Black
              Grid(5,4).BackColor = Black
            Case 4
              ColorGrid(White)
              For Rows = 2 To 4
```

```
                    Grid(Rows, 1).BackColor = Black
                    Grid(Rows, 5).BackColor = Black
                  Next
                  For Cols = 2 To 4
                    Grid(1, Cols).BackColor = Black
                    Grid(5, Cols).BackColor = Black
                  Next
                  Grid(3,3).BackColor = Black
                Case 5
                  ColorGrid(Black)
                  For Rows = 1 To 3
                    Grid(Rows,2).BackColor = White
                    Grid(Rows,4).BackColor = White
                  Next
                  For Rows = 3 To 5
                    Grid(Rows,1).BackColor = White
                    Grid(Rows,5).BackColor = White
                  Next
                  For Cols = 2 To 4
                    Grid(1,Cols).BackColor = White
                    Grid(5,Cols).BackColor = White
                  Next
        End Select

        ClickLabel.Caption = "0"
    end sub
-->
</SCRIPT>

<OBJECT align=baseline
    classid=CLSID:D7053240-CE69-11CD-A777-00DD01143C57
    height=32 id=EasyStartButton width=96 border="0">
    <PARAM NAME="Caption" VALUE="Easy Start">
    <PARAM NAME="Size" VALUE="2540;846">
    <PARAM NAME="FontCharSet" VALUE="0">
    <PARAM NAME="FontPitchAndFamily" VALUE="2">
    <PARAM NAME="ParagraphAlign" VALUE="3">
    <PARAM NAME="FontWeight" VALUE="0">
</OBJECT>
</P>
<CENTER>
 <Input Type="Button" name="EXIT_B" VALUE="  Exit  "
    align="baseline" OnCLick="window.close()">
</CENTER>
<H2>Turn All The Lights On...</H2>
<P>
    Light-It-Up! is a strategy game where you try to light up a building with a
    very strange electrical system. As you can see, the building is divided up into
    five floors of five rooms each. You can light up any room you want just by
```

clicking on it. This will also light up all the rooms around it (above, below and to the left and right). That is, UNLESS they are already lit up. If they are, it will turn their lights out! You can also click on rooms that are already lit to turn their lights out. But whatever you do, it always affects all the rooms around the room you clicked on, too. So be careful! And keep on clicking until all the lights are on.

```
</P>
</BODY>
</HTML>
```

اگر حوصله کرده اید و این برنا مه را مطالعه فرمودید خسته نباشید همـانطور کـه مـی بینید در این مثال با انواع مختلف داده أی و تعاریف اشیاء بر می خوریم تمام برنامه به جـز دکمهٔ Exit و متنهای نوشته شده بر صفحه از خواص شی گرایی VBScript استفاده مـی نمـاید البته دکمهٔ Exit هم از این قضیه مستثنی نیست ولی بـه روش عـادی و بـه عنـوان یـک شـی HTML تعریف شده است.

مثلماً نمی توانستیم از تمام موارد مثـالی ذکـر کنیـم ولـی ایـن مثالـها تقریبـاً تمـام مباحث مربوط به VBScript را می پوشانند برای مثالهای بیشتر میتوانید به دیسکت همـراه کتاب و منابع و مراجعی که در پایان کتاب ذکر شده مراجعه نمایید.

پیوست ۱ :

انواع داده ای در VBScript

انواع داده أی زیر در VBScript تعریف شده اند :

String	داده هایی از نوع رشته أی (حرفی – عددی).
Byte	اعدادی بین ۰ تا ۲۵۶.
Integer	اعدادی بین ۳۲,۷۶۸- تا ۳۲,۷۶۷
Long	اعدادی بین ۲,۱۴۷,۴۸۳,۶۴۸- تا ۲,۱۴۷,۴۸۳,۶۴۷
Single	اعداد اعشاری با دقت ساده.
Double	اعداد اعشاری با دقت مضاعف.
Currency	اعداد اعشاری بین ۹۲۲,۳۳۷,۲۰۳,۶۸۵,۴۷۷،۵۸۰۸- و ۹۲۲,۳۳۷,۲۰۳,۶۸۵,۴۷۷،۵۸۰۸
Boolean	داده أی از نوع منطقی (TRUE , FALSE).
Null	داده أی با یک مقدار نامعتبر.
Date	داده أی از نوع تاریخ و ساعت.
Object	اشاره گری به یک شیء
Error	شماره یکی از خطاهای اتفاق افتاده
Array	داده أی از نوع آرایه (مخصوص آرایه هایی با ابعاد ثابت)
Variant	داده أی مخصوص آرایه هایی که ابعاد نا معین دارند.

پیوست ۲ :

عملگر ها در VBScript

عملگر های زیر در VBScriptتعریف شده اند:

۱-" + "

روش استفاده :

```
Result = expresion_1 + expresion_2
```

جدول صحت :

اگر	حاصل
هر دو expression عدد باشند.	جمع عددی
هر دو expression رشته باشند	الحاق دو رشته به هم
یکی از دو expression عدد و دیگری رشته باشد:	جمع عددی
یکی از دو expression از نوع Null می باشد.	Null
یکی از دو expression از نوع Empty می باشد.	حاصل عدد دیگر می شود.

۲-" - "

روش استفاده :

Result = expresion_1 - expresion_2
Result = - number

جدول صحت :

همانند جدول صحت جمع.

در حالت دوم عدد منفی می شود.

۳-" × "

روش استفاده :

Result = number_1 * number_2

جدول صحت :

اگر	حاصل
هر دو number عدد باشند.	ضرب دو عدد
یکی از دو number از نوع Null می باشد.	Null
یکی از دو number از نوع Empty می باشد.	صفر.

۴-" / "

روش استفاده :

Result = number_1 / number_2

جدول صحت :

اگر	حاصل
هر دو number عدد باشند.	تقسیم number_1 بر number_2
یکی از دو number از نوع Null می باشد.	Null
یکی از دو number از نوع Empty می باشد.	با آن عدد به عنوان صفر برخورد خواهد شد.

xxx ***نوجه کنید که تقسیم بر صفر باعث وقوع خطا خواهد شد.***

۵-" \ "

روش استفاده :

Result = expresion_1 \ expresion_2

جدول صحت :

همانند جدول عملگر تقسیم با این تفاوت که حاصل عددی غیر اعشاری خواهد بود

۶-" ^ "

روش استفاده :

Result = number ^ exponent

جدول صحت :

اگر	حاصل
هر دو عدد باشند.	number به توان exponent خواهد رسید.
یکی از دو number از نوع Null می باشد.	Null

۷-" MOD "

روش استفاده :

Result = number_1 Mod number_2

جدول صحت :

اگر	حاصل
هر دو عدد باشند.	بخش صحیح باقیمانده تقسیم number1 بر number2
یکی از دو number از نوع Null می باشد.	Null
یکی از دو number از نوع Empty می باشد.	با آن عدد به عنوان صفر برخورد خواهد شد.

xxx نوجه کنید که تقسیم بر صفر باعث ونوع خطا خواهد شد.

۸-" = "

روش استفاده :

Variable = value

جدول صحت :

مقداردهی Variable با مقدار value که میتواند به صورت مقدار مستقیم، متغییر ، ثبات و یا خروجی یک تابع باشد. تمام انواع داده ای معتبر می باشند.

۹-" <> "

روش استفاده :

Condition_check_operator (stat_1 <> stat_2)

جدول صحت :

در دستورات شرطی به علامت مخالف بودن دو stat

۱۰-" > "

روش استفاده :

Condition_check_operator (stat_1 > stat_2)

جدول صحت :

در دستورات شرطی به علامت بزرگتر بودن stat_1 از stat_2

۱۱- " < "

روش استفاده :

Condition_check_operator (stat_1 < stat_2)

جدول صحت :

در دستورات شرطی به علامت کوچکتر بودن stat_1 از stat_2

۱۲- " >= "

روش استفاده :

Condition_check_operator (stat_1 >= stat_2)

جدول صحت :

در دستورات شرطی به علامت بزرگتر یا مساوی بودن stat_1 از stat_2

۱۳- " <= "

روش استفاده :

Condition_check_operator (stat_1 <= stat_2)

جدول صحت :

در دستورات شرطی به علامت کوچکتریا مساوی بودن stat_1 از stat_2

۱۴- " & "

روش استفاده :

Result = expresion_1 & expresion_2

جدول صحت :

اگر	حاصل
هر دو expression رشته باشند	الحاق دو رشته به هم
یکی از دو expression عدد و دیگری رشته باشد یا هر دو عدد باشند :	تبدیل عدد به رشته و الحاق دو رشته به هم
یکی هر دو expression از نوع Null می باشد.	Null
یکی از دو expression از نوع Empty و یا Null می باشد.	رشتهٔ دیگر که معتبر می باشد.

۱۵- " And "

روش استفاده :

Result = expresion_1 And expresion_2

جدول صحت :

اگر expression ها غیر عددی باشند:

اگر expression_1	اگر expression_2	حاصل
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
TRUE	NULL	NULL
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

FALSE	NULL	FALSE
NULL	TRUE	NULL
FALSE	FALSE	NULL
NULL	NULL	NULL

اگر هر دو expression عددی باشند به صورت بیت به بیت جدول زیر بر روی آن دو اعمال خواهد شد:

اگر بیت متناظر در expression_1	اگر بیت متناظر در expression_2	حاصل
0	0	0
0	1	0
1	0	0
1	1	1

۱۶- " Or "

روش استفاده :

Result = expresion_1 Or expresion_2

جدول صحت :

اگر expression ها غیر عددی باشند:

اگر expression_1	اگر expression_2	حاصل
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	NULL	TRUE
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE
NULL	NULL	FALSE
TRUE	TRUE	NULL
NULL	FALSE	NULL
NULL	NULL	NULL

اگر هر دو expression عددی باشند به صورت بیت به بیت جدول زیر بر روی آن دو اعمال خواهد شد:

حاصل	اگر بیت متناظر در expression_2	اگر بیت متناظر در expression_1
0	0	0
1	1	0
1	0	1
1	1	1

۱۷- " Xor "

روش استفاده :

```
Result = expresion_1 Xor expresion_2
```

جدول صحت :

اگر expression ها غیر عددی باشند:

حاصل	اگر expression_2	اگرexpression_1
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
NULL	NULL	TRUE
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE
NULL	NULL	FALSE
NULL	TRUE	NULL
NULL	FALSE	NULL
NULL	NULL	NULL

اگر هر دو expression عددی باشند به صورت بیت به بیت جدول زیر بر روی آن دو اعمال خواهد شد:

حاصل	اگر بیت متناظر در expression_2	اگر بیت متناظر در expression_1
0	0	0
1	1	0
1	0	1
0	1	1

۱۸- " Not "

روش استفاده :

Result = Not expresion

جدول صحت :

اگر expression ها غیر عددی باشند:

اگرexpression	حاصل
TRUE	FALSE
FALSE	TRUE
NULL	NULL

اگر هر دو expression عددی باشند به صورت بیت به بیت جدول زیر بر روی آن دو اعمال خواهد شد:

اگر بیت متناظر در expression	حاصل
0	1
1	0

۱۹- " Eqv "

روش استفاده :

Result = expresion_1 Eqv expresion_2

جدول صحت :

اگر expression ها غیر عددی باشند:

اگرexpression_1	اگر expression_2	حاصل
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
TRUE	NULL	NULL
FALSE	TRUE	FALSE
FALSE	FALSE	TRUE
FALSE	NULL	NULL
NULL	TRUE	NULL
NULL	FALSE	NULL
NULL	NULL	NULL

اگر هر دو expression عددی باشند به صورت بیت به بیت جدول زیر بر روی آن دو اعمال خواهد شد:

اگر بیت متناظر در expression_1	اگر بیت متناظر در expression_2	حاصل
0	0	1
0	1	0
1	0	0
1	1	1

۲۰- " Imp "

روش استفاده :

Result = expresion_1 Xor expresion_2

جدول صحت :

اگر expression ها غیر عددی باشند:

اگر expression_1	اگر expression_2	حاصل
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
TRUE	NULL	NULL
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE
FALSE	NULL	FALSE
NULL	TRUE	TRUE
NULL	FALSE	NULL
NULL	NULL	NULL

اگر هر دو expression عددی باشند به صورت بیت به بیت جدول زیر بر روی آن دو اعمال خواهد شد:

اگر بیت متناظر در expression_1	اگر بیت متناظر در expression_2	حاصل
0	0	1
0	1	1
1	0	0
1	1	1

۲۱- "" Is ""

روش استفاده :

```
Result = Object_1 Is Object_2
```

جدول صحت :

أین عملگر برای مقایسهٔ دو شیء بکار می رود و در صورتی که هر دو بـه یک شیء اشاره نمایند مقدار **True** به **result** نسبت داده می شـود و در غـیر ایـن صورت **False** .

پیوست ۳:

لیست تمام کلمات کلیدی
VBScript

در این پیوست لیست تمام کلمات کلیـدی کـه در VBScript تعریـف شـده انـد را منتشر ساخته ایم. اهمیت این لیست در نـامگذاری متغییرهـا مشـخص مـی شـود زمـانی کـه متغییری را تعریف می نمایید ولی با پیغام خطای Unexpected Variable مواجه میشـوید حتماً سری به این پیوست بزنید شاید نام متغییر شما یکی از این کلمـات کلیـدی باشـد. البتـه ممکن است در این لیست با توابع و یا دستوراتی مواجه شوید که قبلاً در مورد آنـها توضیـح نداده ایم این بدان علت است که بعضی از آنها اولاً توسط تمام مرورگرها پشتیبانی نمیشـوند ثانیاً برخی برای VBScript در سطح Client کاربرد ندارنـد بـه جـهت سـهولت در یـافتن نامها آنها را در دوگروه کلمات کلیدی زبان و ثباتهای تعرف شده منتشر ساخته ایم:

الف: کلمات کلیدی به ترتیب الفبایی:

در جلوی هر کلمه کلیدی نوع آن نیز ذکر شده است.

Abs Function
Add Method
AddFolders Method
Addition Operator (+)
And Operator
Array Function
Asc Function
Assignment Operator (=)
AtEndOfLine Property

Drive Property
DriveExists Method
DriveLetter Property
Drives Collection
Drives Property
DriveType Property
Empty
Eqv Operator
Erase Statement
Err Object
Exists Method
Exit Statement
Exp Function
Exponentiation Operator (^)
False
FileExists Method
File Object
Files Collection
Files Property
FileSystemObject Object
FileSystem Property
Filter Function
Fix Function
Folder Object
Folders Collection
FolderExists Method
For...Next Statement
For Each...Next Statement
FormatCurrency Function
FormatDateTime Function
FormatNumber Function
FormatPercent Function
FreeSpace Property
Function Statement
GetAbsolutePathName Method
GetBaseName Method
GetDrive Method
GetDriveName Method
GetExtensionName Method
GetFile Method
GetFileName Method
GetFolder Method
GetObject Function
GetParentFolderName Method
GetSpecialFolder Method
GetTempName Method
Hex Function
HelpContext Property
HelpFile Property

Hour Function
If...Then...Else Statement
Imp Operator
InputBox Function
InStr Function
InStrRev Function
Int Function
Integer Division Operator (\)
Is Operator
IsArray Function
IsDate Function
IsEmpty Function
IsNull Function
IsNumeric Function
IsObject Function
IsReady Property
IsRootFolder Property
Item Property
Items Method
Join Function
Key Property
Keys Method
LBound Function
LCase Function
Left Function
Len Function
Line Property
LoadPicture Function
Log Function
LTrim Function
Mid Function
Minute Function
Mod Operator
Month Function
MonthName Function
Move Method
MoveFile Method
MoveFolder Method
MsgBox Function
Muliplication Operator (*)
Name Property
Negation Operator (-)
Not Operator
Now Function
Nothing

OpenAsTextStream Method
OpenTextFile Method
Operator Precedence
Option Explicit Statement
Or Operator
ParentFolder Property
Path Property
Private Statement
Public Statement
Raise Method
Randomize Statement
Read Method
ReadAll Method
ReadLine Method
ReDim Statement
Rem Statement
Remove Method
RemoveAll Method
Replace Function
RGB Function
Right Function
Rnd Function
RootFolder Property
Round Function
RTrim Function
ScriptEngine Function
ScriptEngineBuildVersion Function
ScriptEngineMajorVersion Function
ScriptEngineMinorVersion Function
Second Function
Select Case Statement
SerialNumber Property
Set Statement
Sgn Function
ShareName Property
ShortName Property
ShortPath Property
Sin Function
Size Property
Skip Method
SkipLine Method
Source Property
Space Function
Split Function
Sqr Function
StrComp Function
String Function
StrReverse Function
Sub Statement

SubFolders Property
Subtraction Operator (-)
Tan Function
Time Function
TextStream Object
TimeSerial Function
TimeValue Function
TotalSize Property
Trim Function
True
Type Property
TypeName Function
UBound Function
UCase Function
VarType Function
VolumeName Property
Weekday Function
WeekdayName Function
While...Wend Statement
Write Method
WriteBlankLines Method
WriteLine Method
Xor Operator
Year Function

ب:لیست ثباتهای زبان به ترتیب موضوعی:

۱-ثباتهای رنگ :

ثبات	معادل در مبنای ۱۶	توضیحات
vbBlack	#00	سیاه
vbRed	#FF	قرمز
vbGreen	#FF00	سبز
vbYellow	#FFFF	زرد
vbBlue	#FF0000	آبی
vbMagenta	#FF00FF	صورتی
vbCyan	#FFFF00	نیلی
vbWhite	#FFFFFF	سفید

۲-ثباتهای عملیات مقایسه أی:

ثابت	معادل	توضیحات
vbBinaryCompare	۰	برای مقایسه باینری
vbTextCompare	۱	مقایسه از نوع متنی

۳- ثباتهای مربوط به تاریخ و ساعت :

رشته	معنی
yyyy	سال به صورت کامل
q	فصل
m	ماه
d	روز
w	هفته
ww	شماره هفته أی از سال
h	ساعت
n	دقیقه
s	ثانیه

ثابت	معادل	توضیحات
vbUseSystem	۰	مقدار پیش فرض سیستم
vbSunday	۱	یک شنبه
vbMonday	۲	دو شنبه
vbTuesday	۳	سه شنبه
vbWednesday	۴	چهار شنبه
vbThursday	۵	پنج شنبه
vbFriday	۶	جمعه

vbsaturday	۷	شنبه

ثابت	معادل	توضیحات
vbUseSystem	۰	از حالت سیستم پیش فرض استفاده مینماید
vbFirstJan1	۱	با هفته أی که اول ژانویه در آن قرار دارد آغاز می شود
vbFirstFourDays	۲	با هفته أی که حداقل چهار روز آن در سال جدید واقع باشد.
vbFirstFullWeek	۳	با اولین هفته ای که تمام آن در سال جدید قرار دارد آغاز می شود.

۴- ثباتهای در مورد نوع تاریخ :

ثبات	معادل	توضیحات
vbGeneralDate	۰	(پیش فرض) در صورت وجود تاریخ و ساعت را تواماً نمایش می دهد .
vbLongDate	۱	تاریخ را با فرمت Long تعریف شده در سیستم کاربر نمایش میذهد
vbShortDate	۲	تاریخ را با فرمت Short مطابق تعریف سیستم کاربر نمایش خواهد داد
vbLongTime	۳	ساعت را با فرمت تنظیم شده بر روی سیستم کاربر نمایش می دهد
vbShortTime	۴	ساعت را به صورت کوتاه شدهٔ ۲٤ ساعته به صورت hh:mm نمایش خواهد داد.

۵- ثباتهایی در مورد نوع درایو ها:

(این ثباتها برای استفاده در سطح Client نیستند.)

ثابت	معادل	توضیحات
Unknown	0	درایو تشخیص داده نشد.
Removable	1	درایو از نوع سیار می باشد .Floppy disk
Fixed	2	درایو از نوع ثابت می باشد مانند Hard disk
Remote	3	درایو از نوع شبکه می باشد.
CDROM	4	درایو از نوع CD-Rom می باشد.
RAMDisk	5	درایو از نوع Ramdrive می باشد.

۶- ثباتهایی در مورد نوع فایل :

(این ثباتها برای استفاده در سطح Client نیستند.)

ثبات	معادل	توضیحات
Normal	0	فایل معمولی
ReadOnly	1	فایل فقط خواندنی
Hidden	2	فایل مخفی
System	4	فایل سیستمی
Volume	8	نام Drive volume Label
Directory	16	پوشه
Archive	32	فایل پس از آخرین بار گرفتن نسخه پشتیبان تغییر کرده است
Alias	64	فایل میانبر
Compressed	128	فایل Compress شده

۷- انواع ثباتهای دستیابی به فایل:

(این ثباتها برای استفاده در سطح Client نیستند.)

ثبات	معادل	توضیحات
ForReading	1	فایل را برای فقط خواندن باز میکند.
ForWriting	2	فایل را برای نوشتن باز می کند اگر فایل از قبل موجود باشد محتویات آن پاک خواهد شد.
ForAppending	8	به فایل موجود اضافه می شود. در صورت عدم وجود فایل ایجاد خواهد گشت.

۸-ثبات خطای اشیاء :

ثبات	معادل	توضیحات
vbObjectError	-2147221504	تمام کد خطاهایی که توسط برنامه نویس تعریف میشوند باید از این عدد بزرگتر باشند

۹-ثبات های مربوط به MsgBox

ثابت	معادل	توضیحات
vbOkOnly	۰	(مقدار پیش فرض) تنها یک دکمه Ok نمایش می دهد .
vbOkCancel	۱	یک دکمه Ok و یک دکمه Cancel
vbAbortRetryIgnore	۲	سه دکمهٔ Abort , Retry , Ignore را ظاهر میسازد.
vbYesNoCancel	۳	سه دکمهٔ yes, no , cancel را ظاهر

		میسازد.
vbYesNo	۴	دو دکمه yes , no
vbRetryCancel	۵	دو دکمه Retry , Cancel
vbCritical	۱۶	به فصل ۱۰ مراجعه نمایید.
vbQuestion	۳۲	به فصل ۱۰ مراجعه نمایید
vbExclamation	۴۸	به فصل ۱۰ مراجعه نمایید
vbQuestion	۶۴	به فصل ۱۰ مراجعه نمایید
vbDefaultButton1	۰	اولین کلید پیش فرض است
vbDefaultButton2	۲۵۶	دومین کلید پیش فرض است
vbDefaultButton3	۵۱۲	سومین کلید پیش فرض است
vbDefaultButton4	۷۶۸	چهارمین کلید پیش فرض است
vbApplicationModal	۰	(پیش فرض) کاربر باید قبل از ادامه اجـرا به جعبه پاسخ دهد.
vbSystemModal	۴۰۹۶	تمام برنامه ها تا زمان پاسخ کـاربر مختـل میشوند

ثابت	معادل	توضیحات
vbOk	۱	Ok
vbCancel	۲	Cancel
vbAbort	۳	Abort
vbRetry	۴	Retry
vbIgnore	۵	Ignore
vbYes	۶	Yes
vbNo	۷	No

۱۰-ثباتهای مربوط به رشته ها:

ثابت	معادل	توضیحات
vbCr	CHR(13)	معادل کلید Enter
vbCrLf	CHR(13) & CHR(10)	ترکیب خط جدید وابتدای خط
vbFormFeed	CHR(12)	حرکت کل صفحه به بیرون از پرینتر (برای Windows کاربرد ندارد)
vbLf	CHR(10)	معادل حرکت یک سطر
vbNewLine	CHR(13) & CHR(10) or CHR(10) only	معادل خط جدید با در نظر گرفت پیش فرض سیستم
vbNullChar	CHR(0)	معادل کاراکتر صفر
vbNullString	String having nothing	مخصوص فراخوانی توابع خارجی
vbTab	CHR(9)	معادل یک Tab
vbVerticalTab	CHR(11)	معادل Tab در جهت عمودی (در Windows کاربرد ندارد)

۱۱- ثبات های دستورات منطقی سه گانه

ثبات	معادل	توضیحات
TristateTrue	-۱	TRUE
TristateFalse	۰	FALSE
TristateUseDefault	-۲	مقدار پیش فرض سیستم کاربر

۱۲- ثباتهای انواع داده ای

ثابت	معادل	توضیحات
vbEmpty	۰	متغیری که هنوز مقدار دهی اولیه نشده باشد
vbNull	۱	متغیری با مقدار نامعتبر
vbInteger	۲	متغیری از نوع عددی دو بایتی
vbLong	۳	متغیری از نوع عددی ٤ بایتی
vbSingle	۴	متغیری از نوع اعشاری ساده
vbDouble	۵	متغیری از نوع اعشار با دقت مضاعف
vbCurrency	۶	متغیری از نوع Currency
vbdate	۷	متغیری از نوع تاریخ و ساعت
vbString	۸	متغیری از نوع رشته أی
vbObject	۹	متغیری که اشاره گری به یک شی باشد.
vbError	۱۰	متغیری با پیغام خطا
vbBoolean	۱۱	متغیری از نوع Boolean (منطقی)
vbVariant	۱۲	متغیری از نوع Variant (مخصوص آرایه های با ابعاد نامعین)
vbDataObject	۱۳	متغیری از نوع شی
vbByte	۱۷	متغیری از نوع بایت
vbArray	۸۱۹۲	متغیری از نوع آرایه های با ابعاد ثابت

پیوست ۴ :

جدول جستجوی سریع توابع

در این پیوست جدولی را ارائه می نماییم تا بتوانید تابع و دستو رات مـورد نیـاز را سریع پیدا نمایید در این جدول توابع با توجـه بـه موضوعـی کـه بـا آن در ارتبـاطند مرتـب گشته اند. اکثریت این توابع و دستورات در فصول قبلی توضیح داده شده اند. آنهایی هـم کـه توضیح داد هنشده اند در سطح Client قابل استفاده نمی باشند و یـا مربـوط بـه روایتـهای خاصی می باشند و به علت عدم عمومیت استفاده از آنها توصیه نمیگردد.

موضوع	توابع و دستورات
آرایه ها	Array Dim, Private, Public, ReDim IsArray Erase LBound, Ubound
انتساب	Set
توضیحات	Rem
ثوابت	Empty Nothing Null True, False
کنترل روند اجرا	Do .. Loop

Select Case	
Abs Asc, AscB, AscW Chr, ChrB, ChrW CBool, CByte CCur, CDate CDbl, CInt CLng, CSng, CStr DateSerial, DateValue Hex, Oct Fix, Int Sgn TimeSerial, TimeValue	تبدیل مقادیر
Date, Time DateAdd, DateDiff, DatePart DateSerial, DateValue Day, Month, MonthName Weekday, WeekdayName, Year Hour, Minute, Second Now TimeSerial, TimeValue	تاریخ و ساعت
Const Dim, Private, Public, ReDim Function, Sub	تعریف
FormatCurrency FormatDateTime FormatNumber FormatPercent	فرمت بندی رشته ها
On Error Err	مرتبط با خطاها
InputBox LoadPicture MsgBox	ورودی / خروجی
Empty False Nothing Null True	مشخصه ها
Abs Int, Fix, Round Sgn Atn, Cos, Sin, Tan Exp, Log, Sqr	ریاضی

Randomize, Rnd	
RGB Function	کمکی
CreateObject Dictionary Drive Object Drives Collection Err File Object Files Collection FileSystemObject Folder Object Folders Collection GetObject TextStream	اشیاء
Addition (+), Subtraction (-) Exponentiation (^) Modulus arithmetic (Mod) Multiplication (*), Division (/), Integer Division (\) Negation (-) String concatenation (&) Equality (=), Inequality (<>) Less Than (<), Less Than or Equal To (<=) Greater Than (>), Greater Than or Equal To (>=) Is And, Or, Xor Eqv, Imp	عملگرها
Option Explicit	فرضیات
Call Function, Sub	روالها
ScriptEngine ScriptEngineBuildVersion ScriptEngineMajorVersion ScriptEngineMinorVersion	شناخت روایت مرورگر
Asc, AscB, AscW Chr, ChrB, ChrW Filter, InStr, InStrB InStrRev Join Len, LenB LCase, Ucase Left, LeftB Mid, MidB Right, RightB	رشته ها

Replace Space Split StrComp String StrReverse LTrim, RTrim, Trim	
IsArray IsDate IsEmpty IsNull IsNumeric IsObject TypeName VarType	متغيرها

پیوست ۵:

پیشنهادهایی جهت نامگذاریها

به پیشنهاد شرکت **Microsoft** خالق زبان **VBScript** بهتر است برای نامگذاری متغییر ها اشیاء و ثبات ها از جداول زیر استفاده نمایید البته بدیهی است که این امر الزامی نیست و جهت هم الگو شدن برنامه های نوشته شده به این زبان ارائه می شود.

نامگذاری متغییرها:

نوع داده ای	پیشوند پیشنهادی	مثال
Boolean	bln	blnFound
Byte	byt	bytRasterData
Date(Time)	dtm	dtmStart
Double	dbl	dblTolerance
Error	err	errOrderNum
Integer	int	intQuantity
Long	lng	lngDistance
Object	obj	objCurrent
Single	sng	sngAverage
String	str	strFirstName

گسترهٔ دید(Scope) متغییر:

گستره دید	پیشوند پیشنهادی	مثال
سطح روال و تابع	هیچ	dblValue

sdblValue	s	سطح کل Script

اشیای موجود در صفحه HTML:

مثال	پیشوند پیشنهادی	نوع شیء
pnlGroup	pnl	3D panel
aniMailBox	ani	Animated Button
chkReadOnly	chk	Check box
cboEnglish	cbo	Combo box
cmdExit	cmd	Command button
dlgFileOpen	dlg	Common dialog
fraLanguage	fra	Frame
hsbVolume	hsb	Horizontal scroll bar
imgIcon	img	Image
lblHelpMessage	lbl	Label
linVertical	lin	Line
lstPolicyCodes	lst	List Box
spnPages	spn	Spin
txtLastName	txt	Text box
vbsRate	vsb	Vertical scroll bar
sldScale	sld	Slider

مثالی ازنحوه نوشتن توابع :

```
'*************************************************************
' Purpose:          در این قسمت وظیفه تابع نوشته می شود
' Inputs:           در این قسمت پارامترهای ورودی تابع نام برده میشوند.
' Return:           نوع نحوه و مقادیر بازگشتی تابع در این قسمت معین می گردد.

'*************************************************************
'
Function intFindUser (strUserList(),
                      strTargetUser)
    Dim i                           ' Loop counter.
```

```
    Dim blnFound              ' Target found flag
    intFindUser = -1
    i = 0                   ' Initialize loop counter

    Do While i <= Ubound(strUserList) and
                    Not blnFound
      If strUserList(i) = strTargetUser Then
        blnFound = True     ' Set flag to True
        intFindUser = i     ' Set return value
      End If
      i = i + 1             ' Increment loop counter
   Loop
End Function
```