

[www.kandoo.cn](http://www.kandoo.cn)

# طراحی وب سایت

(ASP)

[www.kandoo.cn](http://www.kandoo.cn)

[www.kandoo.cn](http://www.kandoo.cn)

[www.kandoo.cn](http://www.kandoo.cn)

## مقدمه

امروزه وب به عنوان یک واقعیت انکارناپذیر سایه خود را در تمامی زندگی اجتماعی، اقتصادی و بشریت انداخته و دارای رشد بسیار سریع در زمینه بکارگیری تکنولوژی های جدید است.

همزمان با این رشد اعجاز گونه وب سایتها از حالت اولیه خود که کوچک و عمدتاً ایستا بودند خارج شده و به سمت حجیم شدن و پویایی حرکت کردند.

امروزه روی یک وب سایت نرم افزارهای متعددی اجرا می شوند تا پاسخ وی نیاز کاربران خود بوده و در این دنیای رقابت، دلیلی برای بودن خود داشته باشند.

در این مسیر تکنولوژی های متعددی نظیر تکنولوژی Dynamic Scripting مانند ASP بوجود آمد تا طراحان را جهت طراحی یک وب سایت پویا یاری کند. استفاده از مزایای کلاس های Vb script، استفاده از متدها و نمونه هایی از این تدابیر جهت طراحی صفحات ASP ماجولار با قابلیت استفاده مجدد Reusable است.

شرکت ماکروسافت در سال ۱۹۹۶ تکنولوژی (Active Server page) را معرفی نمود. تکنولوژی فوق اولین بار به همراه سرویس دهنده اطلاعاتی اینترنت (IIS: Internet Information server) نسخه 3.0 ارائه شد. با بکارگیری تکنولوژی فوق امکان طراحی و پیاده سازی نرم افزارهای بزرگ و بانک های اطلاعاتی به همراه صفحات وب

فراهم شد، هسته اولیه ASP زبان Vbscript است. صفحات ASP همراه کدهای نوشته شده توسط زبان فوق ایجاد می شود. Vbscript به عنوان زبان استاندارد برای پیاده سازی صفحات ASP در نظر گرفته شده است. بنا به اظهارات شرکت مایکروسافت در ماه مه سال ۲۰۰۰، قریب به ۸۰۰۰۰۰ نفر در این زمینه فعالیت داشتند و به طراحی صفحات وب با بهره گیری از تکنولوژی فوق مشغول بودند. آمار فوق نشان دهنده سرعت رشد تصاعدی و استقبال عموم از تکنولوژی فوق در سطح جهان دارد. قریب به ۵۰۰ شرکت کامپیوتری در سطح جهان بصورت کاملا تخصصی صرفا به تولید قطعات و اجزای متفاوت بصورت گیت های آماده نرم افزاری به کمک ASP فعالیت داشته و محصولات آنها در وب سایت های متعدد عرضه می شود.

ASP تقریبا هر کاری می تواند انجام دهد. مثلا دستیابی و جستجو در بانک های اطلاعاتی متفاوت مستقل از بانک، شخصی کردن صفحات وب، نمایش صفحات متفاوت برای مرورگرهای متفاوت، حفاظت رمز عبور، بازیهای کامپیوتری، پردازش فرم ها و ...

### برای شروع به چه چیزی نیاز است؟

جهت استفاده کامل از توانایی های ASP، می بایست سیستم عامل ویندوز NT 4.0 را به همراه IIS 4.0 استفاده کرد. از ویندوز Server 2000 همراه IIS 5.0 نیز می توان بهره برد. جهت استفاده از ASP روی کامپیوترهایی که بر روی آنها ویندوز ۹۸ یا ۹۵ نصب شده

می توان سرویس دهنده شخصی وب یا (PWS: Personel Web Server) را روی کامپیوتر نصب کرد.

### Microsoft Internet Information Server 2.0

Windows NT Server 4.0 با سرویس دهنده وب، سرویس دهنده FTP و حتی سرویس دهنده گوفر کامل می شود. تمام سرویسهای عمده اینترنت به سادگی از طریق این سرویس نصب می شوند فقط از برنامه کاربردی Network control panel برای افزودن Microsoft Internet Information Server 2.0 استفاده کنید و قادر خواهید بود که حضور وب جهانی را احساس کنید.

### استفاده از ASP بدون بکارگیری IIS یا PWS :

به دلیل اینکه سرویس دهندگان وب تنها بر روی محصولات ویندوز شرکت مایکروسافت اجرا می گردند. شما ممکن است فکر کنید که برای استفاده از صفحات ASP حتما باید از IIS یا PWS روی سرویس دهنده وب خود استفاده نمایید. این درست نیست. زیرا که شرکت های زیادی نرم افزارهایی را ایجاد کرده اند که به شما اجازه می دهند صفحات ASP را روی سرویس دهنده های متنوع و کامپیوترهای مختلف ایجاد نمایید.

یکی از محصولات Holycon Softwares Instant ASP است که به صورت مخفف ASP می گویند. یکی دیگر از این محصولات توسط شرکت Chili !soft ایجاد

ASP Chili! نامیده می شود این محصولات می توانند روی سرویس دهنده هایی که IIS

ندارند اجرا شوند به علاوه محصولات زیر:

. Net scape Eterprise server . Apache . Sun Web Server . Jave Web Server

این محصولات روی کامپیوترهای زیر نیز می توانند نصب شوند.

Linux Sun Solaris Apple Mac os IBM / AIX

**ASP چیست؟**

ASP مخفف کلمات Active server page است. ASP برنامه ای است که به

کمک IIS اجرا می شود. IIS یکی از عناصری است که همراه ویندوز ۲۰۰۰ آورده شده.

IIS یکی از اجزای Win NT 4.0 option pack نیز می باشد.

PWS یک نسخه خاص و کوچکتر از برنامه IIS است. PWS را می توان از روی

CD حاوی ویندوز ۹۸ نصب کرد.

**چگونه می توان یک فایل ASP را روی کامپیوتر شخصی اجرا کرد؟**

با نصب یکی از نرم افزارهای PWS و یا IIS (با توجه به سیستم عامل نصب شده)

بدون نیاز به یک سرویس دهنده کمکی دیگر.

**نحوه نصب PWS و اجرای فایل های ASP روی Win 98**

**مرحله اول:** برنامه Setupexe را از مسیر Add ons \ PWS موجود در CD حاوی Win

98 اجرا کنید.

**مرحله دوم:** یک فولدر به برنامه Inetpub روی هارد ایجاد و در آن یک فولدر به نام Wwwroot بسازید.

**مرحله سوم:** یک فولدر جدید با نام دلخواه مثل My web در فولدر Wwwroot بسازید.

**مرحله چهارم:** با استفاده از یک ادیتور مبتنی مثل Notepad کدهای دلخواه Asp خود را تایپ کرده و فایل فوق را با نام دلخواه Test . asp ولی با انشعاب Asp در فولدر My web ذخیره کنید.

**مرحله پنجم:** اطمینان داشته باشید که PWS در حالت اجرا است.

**مرحله ششم:** مرورگر خود را فعال کرده و در محل مربوطه آدرس زیر را تایپ کنید.

<http://localhost/MyWeb/Test.asp>

#### 4.0 NT Server چیست؟

همه چیز از سال ۱۹۹۲ با نشر نگارش ۳/۱ Windows Advanced Server آغاز

شد در ابتدا موفقیت Windows NT 4.0 سرویس دهنده برنامه کاربردی بود که به عنوان

سیستمی برای برنامه‌های کاربردی مبتنی بر سرویس دهنده میکروسافت مانند

SQL Server، در داخل ناول یا سایر سیستمهای عامل شبکه موجود استفاده می‌شد NT 4.0

در بسیاری از شبکه های محلی شرکتها که قبلا دقیق میکروسافت را به عنوان استاندارد

برای محصولات رومیزی انتخاب کرده بودند جای پای محکمی پیدا کرد و به عنوان

سیستمی مفید برای برنامه‌های کاربردی Back office مایکروسافت مطرح شد. این جای پای محکم با معرفی Windows NT Server 3.5 در سال ۱۹۹۳ به راه‌حل کامل و قدرتمندی برای سیستم عالم شبکه تبدیل گردید.

NT Server برخلاف سایر سرویس دهنده‌های سیستم، ایستگاه کاری نیز می‌باشد. NT 4.0 رابط کاربر تقریباً با رابط کاربر ویندوز ۹۵ و برنامه‌های رومیزی معمولی مانند Word، Excel و هزاران برنامه کاربردی دیگر ویندوز دارد که به خوبی بر روی NT Server 4.0 اجرا می‌شوند. اما، در زیر این رابط ساده‌میز کار، ابزارهای پیچیده مدیریت و سرپرستی شبکه و سیستم عامل شبکه قدرتمندی با پشتیبانی چندین پروتکل و سیستم می‌باشد این رابط آشنا و ساده یکی از جذاب‌ترین ویژگی‌های NT Server 4.0 و همچنین مشکلی عمده برای سرپرستان مبتدی می‌باشد. نصب NT 4.0 ساده است. NT 4.0 رابط ساده‌ای دارد که با سبک اشاره و کلیک عملیات ابتدایی را فوراً برای افراد آشنا با ویندوز (که هم اکنون با ۱۰۰ میلیون نسخه فروش تقریباً برای همه آشنا می‌باشد) قابل درک و دسترسی می‌سازد.

## نیازهای سخت‌افزاری

حداقل نیازهای سخت‌افزاری برای IIS به صورت زیر است:

CPU: 33 / 486 یا بالاتر

حافظه: ۱۶ مگا بایت برای سرویس دهنده NT و ۱۲ مگا بایت برای ایستگاه کاری NT.

فضای دیسک: ۱۲۵ مگا بایت برای سرویس دهنده NT و ۱۱۰ مگا بایت برای ایستگاه

کاری NT.

صفحه نمایش: VGA یا بالاتر.

CD – ROM.

دیسکت گردان فلاپی ظرفیت بالا.

### نرم افزارها و ابزارهای لازم:

نرم افزارها و زبانهای برنامه نویسی که در این پروژه استفاده شده اند به شرح زیر می باشد:

– زبانهای HTML و ASP و VBScript

– پایگاه داده Access

– سیستم عامل ویندوز XP و وب سرور IIS



[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

## فصل اول

### مروری بر ASP

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

ASP راه حل شرکت مایکروسافت برای ایجاد صفحات پویای وب می باشد. با گسترش اینترنت و WWW در زندگی امروزه، ایجاد وب سایت یکی از بخش هایی است که به سرعت در حال رشد است. در روزهای اولیه WWW، HTML به منظور ایجاد صفحات وب ایستا استفاده می شد. اما امروزه، صفحات وب ایستا به سرعت در حال منسوخ شدن است. راه حل مایکروسافت برای ساختن صفحات وب پویا استفاده از ASP است. ASP شامل دو بخش است. کدهای برنامه نویسی و Embedded HTML.

کدهای برنامه نویسی به وسیله تعدادی از زبانهای دست نویس (Script language) نوشته می شوند. زبانهای اسکریپت نویسی دست نویس مربوط به وب که محبوب می باشند عبارتند از VBScript و JavaScript، هنگامی که یک صفحه ASP درست می کنیم می توان از یکی از چهار زبان برنامه نویسی استفاده کرد: VBScript، Jscript، Perlscript، Pythen.

### آشنایی با مدل Client – server

یک مدل Client – server، دو کامپیوتر می باشند که با هم برای انجام یک کار همکاری می نمایند. یک کامپیوتر سرویس گیرنده اطلاعات مورد نیاز خود را از کامپیوتر سرویس دهنده تقاضا می کند. کامپیوتر سرویس دهنده اطلاعات درخواستی را به Client برمی گرداند و کامپیوتر Client روی آن اطلاعات کار می کند.

همانطور که قبلا بیان شد صفحات ASP شامل ترکیبی از HTML و کدهای برنامه نویسی می باشند، این کد که با زبانهای مختلف می تواند نوشته شود. به صفحات ASP اجازه می دهد که به صورت پویا عمل کنند. هر چند سرویس دهنده وب مجبور است کدهای برنامه نویسی را قبل از فرستادن HTML پردازش کند. وقتی که مرورگر وب درخواست یک صفحه ASP را می کند مراحل زیر اتفاق می افتد.

۱- سرویس گیرنده محل سرویس دهنده را به وسیله بخش اول URL تعیین می کند.

(www. Something . com)

۲- سرویس گیرنده درخواست صفحه ASP را به وسیله بخش دوم URL انجام می دهد.

(default. Asp)

۳- سرویس دهنده وب فایل ASP را می خواند و کدها را پردازش می کند.

۴- بعد از اینکه صفحه ASP بطور کامل به وسیله سرویس دهنده پردازش شد خروجی به فرمت HTML به سرویس گیرنده فرستاده می شود.

۵- سرویس گیرنده HTML فرستاده شده به وسیله سرویس دهنده را دریافت کرده و آن را

برای کاربر اجرا می کند.

### توجه

توجه کنید که کد ASP به وسیله علامت `< % % >` احاطه شده است. وقتی یک

صفحه ASP از یک سرویس دهنده وب درخواست می گردد سرویس دهنده وب بطور

کامل همه کدهای مابین دو علامت `< % % >` را قبل از فرستادن خروجی برای سرویس گیرنده پردازش می کند.

### فرق بین ASP با تکنولوژی Client – side scripting

وقتی از ASP استفاده می شود مهم است بدانید که کدهای ASP فقط روی سرویس دهنده وجود دارند. کد ASP که توسط محدود کننده های `< % % >` محصور می شوند توسط سرویس دهنده کامل پردازش می گردند. سرویس گیرنده به این کدهای ASP دسترسی ندارد. Client – side scripting کدهای برنامه نویسی است که در یک فایل HTML روی مرورگر اجرا می گردد.

کدهای اسکریپت نویسی Client – side شبیه کدهای HTML است و به وسیله تگ `< Script >` HTML فهمیده می شود. فهمیدن اینکه اسکریپت های Client – side و اسکریپت های Server – side دو موجودیت کاملا مجزا هستند بسیار مهم است. برای بحث و توضیح بیشتر به آدرس زیر مراجعه کنید.

<http://www.4guysformolla.com/Webtech/0823921.Shtml>

قابلیت های ASP

استقلال از مرورگر

ASP از مروگر کاملاً مستقل است زیرا تمامی دستورالعمل‌ها بر روی سرور اجرا می‌شود و مروگر تنها صفحات HTML معمولی را به عنوان حاصل اجرای دستورالعمل‌ها دریافت می‌کند.

### استفاده از زبان دستورالعمل نویسی

این بدین معنی است که یک برنامه ASP به راحتی قابل تغییر است چرا که زبانهای دستورالعمل نویسی نیاز به کامپایل نداشته و به راحتی می‌توان در یک ویرایشگر مثل Notepad برنامه نوشت و آن را تغییر داد.

### دسترسی آسان و سریع به انواع بانک‌های اطلاعاتی

اعمال پیچیده مربوط به پردازش بانک‌های اطلاعاتی در ASP بسیار ساده است و به سرعت می‌توان برنامه دلخواه را نوشته یا تغییر داد.

همان روش مورد استفاده در VB یا C++ را می‌توان در ASP استفاده کرد.

(ADO)

### ایجاد صفحات ASP

صفحات ASP، همانند صفحات وب HTML، فایل‌های متنی ساده بر روی سرور دهنده وب می‌باشند و برای ایجاد صفحه ASP بعد از اینکه سرور دهنده وب را نصب کردید تمام چیزی که شما احتیاج دارید یک ویرایشگر متن همانند Notepad است. برنامه Notepad به صورت معمول برای ایجاد و اصلاح صفحات ASP استفاده می‌شود.

ایجاد ASP توسط Notepad دارای مزایا و معایبی می باشد ابتدا مزایای این کار را بیان می کنیم.

### مزایا

سرعت: Notepad یک ویرایشگر کوچک و کاراست سادگی به کارگیری: چون Notepad گزینه های زیادی ندارد.

### معایب

نداشتن پشتیبانی سایت وب: با Inter Der یا Front page شما می توانید سراسر سایت وب را مدیریت کنید. ولی با Notepad شما فقط می توانید یک فایل در یک زمان را ویرایش کنید.

- کدهای دستور زبان غیر رنگی: وقتی از Front page و یا Inter Der استفاده می کنیم قطعات مختلف کدهای رنگهای متفاوت خواهد داشت.

- نداشتن ابزار صفحات وب & - drop - Drog: هم Inter Der و هم Front page به ما

اجازه می دهند که ظواهر صفحه وب را به وسیله کشیدن و رها کردن اشیاء یا عناصر داخل صفحه وب ایجاد نمائید.

بعد از اینکه یک وب سرور را نصب کردیم می توان صفحات ASP را در دایرکتوری ریشه سایت وب خود ایجاد کنید یا آن را در زیر فهرستهای دایرکتوری ریشه قرار دهیم و صفحات ASP حاصله را توسط مرورگر وب استاندارد مشاهده نماییم.

### Delimiter

یک کاراکتر یا رشته ای از کاراکترها که ابتدا و انتهای هر واحد را نشان می دهد، ASP برای نشان دادن ابتدا و انتهای هر واحد از `< % % >` استفاده می کند و دستورات اسکرپت بیان این علامت ها قرار می گیرند. متنی که بین دو عبارت `< % % >` قرار می گیرد به عنوان کدهای ASP در نظر گرفته می شود به این خاطر که سرویس دهنده قبل از فرستادن صفحه برای مرورگر وب آنها را مور پردازش قرار می دهد.

```
<HTML>
```

```
<BODY>
```

```
This page was last refreshed on
```

```
<% = now%>
```

```
</BODY>
```

```
<HTML>
```

تابع Now در VBScript زمان و تاریخ جاری را برمی گرداند و زمانی که سرویس دهنده وب این صفحه را پردازش می کند به جای تابع Now زمان و تاریخ جاری را، می گذارد و نتیجه را به مرورگر می فرستد دستوراتی که به وسیله `< % % >` احاطه می شوند

فرمانهای اولیه اسکریپت نامیده می شوند این فرمانها بوسیله زبان اسکریپت ابتدایی<sup>۱</sup> پردازش می شوند. دستوراتی که بین Delimiter نوشته می شود باید برای زبان اسکریپت ابتدایی شناخته شده باشند که این زبان بطور پیش فرض VBScript است.

### نوشتن کدهای ASP بدون استفاده از <%... %>

اسکریپت های Client – Side بر روی کامپیوتر کاربر بعد از اینکه صفحه دریافت شد اجرا می شود. اگر به یک اسکریپت Client – Side نگاهی بیندازیم احتمالا Tag های HTML، <Script> و </Script> را تشخیص می دهیم. آنها به منظور درج کردن کد اسکریپت Client – Side در داخل یک صفحه HTML استفاده می شود.

برای مثال ممکن است شما چیزی شبیه به این را ببینید:

```
<Script Language = Java Script>
d = new Date ( )
if (d.getHours ( ) > 12)
document. Write ("Good evening") ;
else
document. Write ("Good Morning") ;
</Script>
```

---

<sup>1</sup> Primary Scripting Language (P.S.L)



اگر شما از <Script> برای اسکریپت های Server – Side استفاده نمایید، یک پیچیدگی اضافی وجود خواهد داشت. حتی می توانید <Script> و <%... %> را در یک صفحه به صورت ترکیبی به کار ببرید.

تابع write در Response می تواند به جای document.write استفاده شود. علت استفاده این تابع این است که ASP موجود در روی سرویس دهنده را به شکل HTML که مرورگر وب کاربران نشان می دهد، تبدیل نماید. برای مثال:

```
<B>  
<%  
Response . write "Hello"  
%>  
</B>
```

متن Hello به خروجی فرستاده می شود.

اگر متن نوشته شده با تابع write در Response شامل Tag های HTML نیز باشد آنها نیز توسط مرورگر وب کاربران تفسیر خواهد شد.

## استفاده از دستور دهنده های ASP<sup>۲</sup>

دو نوع دایرکتیو در ASP وجود دارد که عبارتند از

۱- ASP output directive ها. که بصورت زیر استفاده می شوند.

<% expression %> که مقدار expression را نشان می دهد و این معادل . Response write است.

۲- ASP Processing directive ها که بصورت زیر استفاده می شوند.

<% a key word %>

این دسته اطلاعات لازمی را که برای پردازش صفحات ASP لازم است را در اختیار

ما قرار می دهند. مثلا <% a language = VBScript %> یعنی زبان اولیه برای اسکریپت ها VBScript می باشد.

این دایرکتیوها باید در خط اول فایل ASP باشند و نمی توان آن را به عنوان یک

فایل دیگر به برنامه اضافه کرد همچنین لازم است یک فضای خالی بین @ و کلمه کلیدی وجود داشته باشد.

**کلمات کلیدی که در Processing directive استفاده می شوند، عبارتند از:**

- LANGUAGE، زبان مورد نیاز برای صفحات را مشخص می کند.

- CODEPAGE، کد صفحه لازم را برای صفحات مشخص می کند.

- LCID، شناسه های محلی برای صفحه را مشخص می کند.

- TRANSACTION، مشخص می کند که صفحه تحت چه تراکنشی اجرا خواهد شد.

- ENABLE SESSION STATE

می توان از بیش از یک کلمه کلیدی در یک Directive استفاده کرد هر کلمه کلیدی و تعداد آن توسط >= از هم جدا می شوند و هیچ فضای خالی دیگری نباید بین آنها باشد.

```
<% a LANGUAGE = Jscript CODE PAGE = 932 %>
```

### کار کردن با زبانهای اسکریپت

بطور کلی در زبانهای اسکریپت واسطی بین HTML و زبانهای برنامه نویسی مانند جاوا و C++ و ویژوال بیسیک وجود دارد HTML برای فرمت کردن متنها و لینک کردن صفحات، مورد استفاده قرار می گیرد و زبانهای برنامه نویسی برای دادن دستورات پیچیده به کامپیوتر استفاده می شود.

زبانهای اسکریپت می توانند همچنین دستورات را به کامپیوتر بدهند و قوانین و فرمهای آنها آسان تر از زبانهای برنامه نویسی دیگر می باشد. زبانهای اسکریپت روی فرمت متن متمرکز می شوند و از اجزای کامپایل شده زبانهای برنامه نویسی استفاده می کند.

ASP این امکان را می دهد که رویه های کاملی را بوسیله استفاده از انواع مختلف

زبانهای اسکریپت بنویسیم. در حقیقت می توان از چندین زبان اسکریپت در یک فایل ASP استفاده کرد بعلاوه به دلیل اینکه روی سرویس دهنده خوانده و پردازش می شود مرورگر CLIENT که فایل ASP را درخواست می کند نیازی نیست که اسکریپت ها را ساپورت

کند. و هر زبان اسکریپت را که مناسب با Scripting engine است می توان روی سرویس دهنده وب نصب کرد.

### استفاده از VB و JS روی سرویس دهنده

Java Script معمولاً برای اسکریپت های Server - Side مورد استفاده قرار نمی گیرد. VBScript به خاطر راحتی در یادگیری، بیشتر در نظر گرفته می شود. بنابراین ما از Java Script برای اسکریپت های Server - Side استفاده نمی کنیم. زبان Java Script سخت تر از C و C++ و زبان برنامه نویسی Java نمی باشد، اولین مزیت VBScript راحتی یادگیری آن است. VBScript بر پایه VB است. VBScript همچنین زبان پیش نویس فرض برای ASP می باشد. این بدان معنی است که اگر شما قصد دارید از VBScript استفاده کنید می توانید از خط

<% @ Language ... %> استفاده نکنید.

### فرمانهای VBScript

در VBScript، توضیحات بوسیله (‘) شروع شده و برخلاف HTML زمانی که اسکریپت پردازش می شود آنها حذف شده به مرورگر فرستاده نمی شوند. در جملات خروجی نباید یک توضیح قرار داد به عنوان مثال عبارت زیر صحیح نیست.

<% = Name this prints the variahe name %>

اما عبارت زیر کار می کند.

```
<% i = i+1 this incrementi %>
```

## فرمانهای Jscript

در Jscript توضیحات با (" شروع می شوند VBScript به حروف بزرگ و کوچک حساس نیست اما نمی توان دو متغیر همنام داشت در حالیکه Jscript به این مسئله حساس است و اسم اشیاء باید با حروف بزرگ نوشته شود اما متدها و خاصیت ها می توانند با حروف کوچک هم نوشته شوند و ASP از Jscript تبعیت می کند.

## استفاده از متغیرها و ثابت ها

### اعلان متغیرها در VBScript:

در VBScript، همه متغیرها به وسیله کلمه کلیدی Dim تعریف می شود. این بدین خاطر است که در VBScript از Variant variables ها استفاده می گردد. بنابراین، جمله معادل برای آن چیزی که در C++ توصیف شده عبارت زیر است. استفاده کردن از عبارت %< Option Explicit باعث می شود که همه متغیرها از نوع توصیف صریح تعریف شود.

## دامنه متغیرها

متغیرهای سراسری و متغیرهای محلی می توانند همنام باشند به مثالهای زیر توجه کنید.

```
<%
```

Dim y

Y=1

Call set laocal Variable

Response. Write y

Sub set Local Variable

Dim y

Y=2

End sub

مثال فوق مقدار Y را ۱ برمی گرداند.

اما در مثال زیر مقدار ۲ برگردانده می شود. زیرا متغیر Y صریحا اعلان شده است.

<%

Y=1

Call setlaocal Variable

Response. Write y

Sub setlocal Variable

Y=2

End sub

%>

ولی باید تا حد امکان سعی کنیم که متغیرها را اعلان کنیم بخصوص زمانی که

بخواهیم از جملات `#include`<sup>۳</sup> برای صفحه Asp استفاده کنیم.

**تعیین دامنه متغیرها**

---

۳ - شامل يك فایل جداگانه است که در هر کجای برنامه که صد ا شود اجرا می شود.

برای اینکه متغیرهای ما در تمام صفحات قابل دسترسی باشند از دو روش استفاده کنیم.

۱- Session Scope variable متغیرهایی هستند که در تمام صفحات یک کاربرد ASP موجودند و توسط یک کاربر درخواست می شوند.

۲- Application Scope variable متغیرهایی که در تمام صفحات وجود دارند و بوسیله هر کاربری می توانند درخواست شوند.

نوع اول یک روش خوب برای ذخیره سازی اطلاعات برای یک کاربر می باشد اما روش دوم برای ذخیره سازی اطلاعات برای همه کاربران در یک کاربرد بخصوص است. ASP از دو شیء برای موارد فوق استفاده می کند.

۱- Session object

۲- Application object

برای اینکه به یک متغیر دامنه Session داده شود آن متغیر را در شیء اول ذخیره

کنیم و یک مقدار به اسم داخل شیء نسبت می دهیم مثال ۱:

```
<%
```

```
Session ("first name") = "Jeff"
```

```
%>
```

و برای بدست آوردن اطلاعات ذخیره شدن از دستور زیر استفاده کنیم.

```
Wellcome <% = Session ("first name") %>
```

مثال ۲:

<% Application ("Greeting"); Well come to exploration Air" %>

برای بدست آوردن اطلاعات ذخیره شده از Response. Write و یا از

Output directive استفاده می کنیم.

Response. Write "session ("first name") %>

و یا روش

<% = Application ("first name") %>

<% = Session ("first name") %>



## استفاده از ثابت‌ها

ثابت‌ها مکانی از اعداد یا رشته‌ها را بوسیله اسمی مشخص می‌کنند برخی از اجزای اساسی تهیه شده با ASP مانند ADO ثابت‌های ما را که می‌توانیم در اسکریپت‌هایمان استفاده کنیم را تعریف می‌کنند.

یک جزء می‌تواند اجرای داخل یک کتابخانه اجزا را اعلان کند که فایلی است شامل اطلاعاتی درباره‌ی شیء و نوع آنها که بوسیله یک جزء Activex پشتیبانی می‌شوند. از این‌رو می‌توان یک کتابخانه را در داخل یک Global.asa اعلان کرد و می‌توان اجزاء تعریف شده را در داخل هر اسکریپتی در برنامه به کار برد. برای اعلان یک نوع کتابخانه می‌توان از تگ <METADATA> در فایل Global.asa استفاده کرد.

مثال:

```
<!-METADATA type = "typelib"  
File = "C:/ program files \ common files\  
System ado \ msado  
-->
```

پس می‌توان ثابت‌های ADD را در هر اسکریپتی در برنامه مشابه بکار برد در مثال زیر Ado penkeyset و Adlock optimistic دو ثابت ADO هستند.

```
Create and open Recordset object  
Set Rs Customer list = server. Create object ("ADO DB. Recordset")  
RS Customer list. Active connection = OB odb Connection
```

RS Customer list. Cursortype = adopenkey set

RS Customer list. Lock type = adlock optimistic

برای استفاده از ثابت‌ها در VBScript از جمله Const و در Jscript از Var استفاده

می‌کنیم و برای استفاده از یک ثابت در چندین صفحه آن را در فایل جداگانه قرار داده

آنگاه به صفحات ASP که آن را نیاز دارند Include می‌کنیم.

## نوشتن رویه‌ها

ساب‌روتین‌ها (رویه‌ها Procedure) عملیاتی را انجام می‌دهند. یک نمونه ساده از

ساب‌روتین در زیر بیان شده است.

Subname

Code block

End sub

برای ساختن یک ساب‌روتین به صورت نمونه به شکل زیر عمل می‌شود.

Sub subHeading

Response. Write ("<p align = right>")

Response. Write ("Your name <BR>")

Response. Write ("Your name <HR>")

حال اگر این ساب‌روتین را داخل صفحه خود قرار دهید برای شما سر تیترا چاپ

می‌کند.

صدا زدن رویه‌ها

برای فراخوانی رویه اسم آن را داخل دستوراتمان می آوریم اگر رویه های Jscript را در Vbscript فراخوانی کنیم بعد از اسم رویه باید پرانتز بگذاریم و اگر رویه ای پارامتر ندارد از پرانتز خالی استفاده کنیم اگر رویه های Jscript و Vbscript را از Jscript فراخوانی کنیم بعد از اسم رویه همیشه از پرانتز استفاده می کنیم.

برای Vbscript می توان از لغت کلیدی Call نیز استفاده کرد و اگر رویه ما دارای پارامتری نیز هست لیست آرگومانهای آن باید در پرانتز قرار بگیرد. و اگر از فرم Call استفاده کنیم برای هر تابعی، مقدار بازگشتی تابع رد می شود.

```
<%@ LANGUAGE = VBSCRIPT %>
```

```
<HTML>
```

```
<BODY>
```

```
<%Call Echo %>
```

```
<BR>
```

```
<%Call print Data %>
```

```
</BODY>
```

```
</HTML>
```

```
<%
```

```
Sub Echo
```

```
Response. Write "<TABLE>"
```

```
Response. Write "<TR> <TH> Name </TH> <TH> value
```

```
</TH> </TR>
```

```
Set params = Request. Query string
```

```
For Wach p in params
```

Response. Write "<TR> <TD>" & p &" </TD> <TD>"

& params (p)

& "</TD></TR>"

Next

Response. Write "</Table>"

End sub

%>

<Script LANGUAGE = Jscript Runat = server

Function print Data

{

Var x

X = new Data ( )

Response. Write (x. tostring ( ) )

}

</Script>

برای انتقال آرایه یک رویه در VB از اسم و آرایه همراه پرانتزهای خالی استفاده

می کنیم و در Jscript از براکت خالی استفاده می کنیم.

### استفاده از کامپونت ها و اشیاء

استفاده کردن از کدهایی که قبلا نوشته شده اند کاری ضروری می باشد. اگر هر فرد

مجبور باشد که تمامی چیزهایی که لازم دارد را خودش بساز کار زیادی نمی تواند انجام

دهد. خوشبختانه، راههای زیادی برای استفاده مجدد از کدها وجود دارند. عمل Copy و

Paste برای کارهای کوچک مورد استفاده قرار می گیرند، اما برای قطعاتی بزرگ از کدها کاربرد ندارند. به همین دلیل است که از کامپونت ها استفاده می شود. کامپونت ها کدهایی که مورد احتیاج می باشند را بسته بندی و از هم جدا می کند تا برای شما قابل دسترس شوند. هنگامی که یک کار بصورت یک کار عادی در آمد، بعضی از افراد برای آن کار یک کامپونت می نویسند. بعضی از کامپونت ها را می توان بطور رایگان از طریق وب دریافت کرد ولی بعضی دیگر را باید خرید.

### نمونه سازی کامپونت ها

کامپونت های پیش ساخته را با استفاده از متدهای Set و Creat object از آبجکت Server، نمونه سازی می نمایند. اولین قدم، تعریف متغیری است که آبجکت Server را نگهداری می نماید. معمولاً نام این متغیر را با پیشوند "Obj" شروع می کنند.

#### Dim objwhatever

Set objwhatever = Server. Creat object ("classname. Componentnan")

Classname عبارت MSWC می باشد که بسته ای است که کامپونت ها در آن قرار

دارند.

**آزاد کردن حافظه از کامپونت ها:** آزادسازی حافظه ای که در اختیار کامپونت است

هنگامی که کار با کامپونت تمام می شود دارای اهمیت می باشد. اگر چه خود ASP باید این

کار را برای شما انجام دهد، اما عاقلانه نیست که خیلی به این مسئله اطمینان نمایید.

اگر ASP در انجام این کار با شکست مواجه شود، شما به انوعی از گرفتاریها دچار خواهید شد.

برای اینکه حافظه را به صورت صریح از کامپونت های Objwhatever پاک کنید باید به صورت زیر عمل کنید.

Set objwhatever = Nothing

**Ad Rotator:** تبلیغات در همه جای وب وجود دارد. خوب یا بد. آرمهای تبلیغاتی 460

x60 تبدیل به بخش ثابتی از وب، تحت عنوان [www.com](http://www.com) شده اند. مایکروسافت برای

اینکه بتوانید آرمهای سایتان را به سادگی تغییر دهید کامپونت Ad Rotator را ساخته است.

### استفاده از ASP برای ساختن اشیاء (Object)

خود ASP نیز یکسری از اشیاء را تهیه می کند برای مثال Request که برای داده های عبور داده شده به یک سرور را ذخیره می کند برای فرستادن اطلاعات به سرور بر خلاف موارد گذشته دیگر نیازی به تهیه نمونه از شیء نیست.

### صدا زدن یک متد شیء

یک متد یک عمل می باشد که می توان روی یک شیء یا با یک شیء انجام داد.

فرم مربوطه بصورت زیر است.

Object. Method parameters

مثال: می توان از متد Write برای فرستادن اطلاعات به مرورگر استفاده کرد.

```
<%Response.Write "Hello world">
```

### هماهنگ کردن خاصیت های یک شیء (Object)

می توان مقادیر خاصیت ها را خواند و یا این خاصیت ها را مقداردهی کرد و یا خواص جدید اضافه کرد.

فرم کلی آن بصورت زیر است.

Object property

به عنوان مثال AD Rotator Component یک خاصیت به نام Border دارد.

```
<% my Ads.Border = 0%.>
```

برای برخی از خاصیت ها می توان مقادیر جاری را نشان داد این کار را می توان به کمک خروجی مستقیم ASP انجام داد مثال زیر چنانچه مرورگر هنوز به Server وصل باشد مقدار درست را برمی گرداند.

```
<% = Response.IS client connected %>
```

### استفاده از کلکسیون ها (Collection)

اغلب اشیا (object) های ساخته شده در ASP، کلکسیون را پشتیبانی می کنند یک کالکشن مکانی برای ذخیره سازی رشته ها، اعداد، شیء ها و دیگر مقادیر می باشد در واقع شبیه یک آرایه است با این تفاوت که به آسانی می تواند توسعه پیدا کند یعنی زمانی که عناصری به آن اضافه می شود بطور اتوماتیک توسعه می یابد.

برخلاف یک آرایه موقعیت یک عنصر بر حسب تغییرات انجام شده روی کلکسیون عوض می شود و می توان به یک عنصر بوسیله اسم یا اندیس و یا حرکت مکرر میان عناصر دسترسی پیدا کرد.

### چرا آبجکت ها می توانند در صفحه هایی که می نویسید مفید باشند؟

آبجکت ها به افزایش سطح تجرد در صفحه های تان کمک می کند. به شما گفته می شود که بصورت تصادفی یک پرچم انتخابی را نمایش دهید. می توانید در میان لیست پرچم ها یکی را انتخاب کنید به این صورت که تابع Random که اعداد تصادفی تولید می کند را اجرا کرده و بعد کدی را برای `<Img SRC = ...>` Tag بنویسید. می توانید تمام این کارها را هر زمانی که می خواهید یک پرچم را نمایش دهید انجام دهید. اما ساده تر آن است که تمام کارهایی را که می خواهید انجام شود در داخل یک آبجکت که یک پرچم را بصورت تصادفی نمایش می دهد بنویسید بعد از این کار احتیاج دارید به اینکه فقط چیزی مثل Random Banner را بنویسید. پس از آن اگر بخواهید به سیستم نمایش پرچم خود، چیزی را اضافه یا تغییر دهید، فقط احتیاج دارید که یک تکه از کد را تغییر دهید.

### Cookies

کوکی ها فایل های کوچکی می باشند که سایت های وبی که از آنها بازدید می کنید بر روی کامپیوترتان ذخیره می کند. کوکی ها به سایت وب چیزهایی مثل اینکه چه کسی هستید با چه کاری را وقتی صفحه را قبلا دیدید انجام دادید را می گویند. Response شامل



یک مجموعه به نام Cookies می باشد که می تواند برای نوشتن کوکی ها مورد استفاده قرار گیرد.

کوکی ها دارای دو دسته دستور می باشند.

Request. Cookies (Cooki Name) [ (KeyName) ]

این دستور برای خواندن کوکی ها مورد استفاده می باشد.

دستور دوم Request. Cookies است که برای نوشتن کوکی ها مورد استفاده است.

## هماهنگ کردن دامنه اشیا

دامنه یک شیء تعیین می کند که کدام اسکریپت ها می توانند به آن دسترسی داشته باشند بطور پیش فرض دامنه، یک صفحه ASP می باشد و تمام اسکریپت های نوشته شده در آن صفحه می توانند به شیء دسترسی داشته باشند اما می توان این دامنه را تغییر داد.

## دادن دامنه سی شن (Session) به یک شیء

اگر دامنه یک شیء از نوع Session Scope باشد آن شیء در پایان سی شن پاک می شود دامنه سی شن برای اشیائی استفاده می شود که بوسیله چندین اسکریپت فراخوانی می شوند اما فقط روی یک کاربر اثر می گذارند برای داشتن دامنه سی شن باید شیء را در سی شن ذخیره کنیم و یا از تگ <Object> در فایل Global.asa استفاده کنیم و یا از متد Server.CreateObject در یک صفحه ASP استفاده کنیم.

مثال:

```
<OBJECT Runat Server Scope = Session  
ID = my AD  
PROGID = "mswc. Ad rotator">  
</OBJECT>
```

بدین ترتیب شیء ذخیره شده در سی شن می تواند توسط دیگر صفحات قابل

دسترسی باشد.

مثال زیر شیء ای را که در مثال قبل ایجاد شده را استفاده می کند.

```
<% = myAd. Get Advartis ement ("Ad Rotator. Txt") %>
```

مثال فوق با استفاده از متد Server. Create object

```
<% = Set Session ("myAd") = Server. Create object ("mswc. Adrotator  
%>
```

حال برای نمایش Ad می نویسیم

```
<% = Set myAd = Session ("my Ad") %>
```

آنگاه از متدی برای نمایش Ad استفاده می کنیم.

استفاده از تگ </OBJECT> اجرای بهتری را نسبت به متد Server. Create object عرضه می کند.

با استفاده از دو مجموعه Static object و Contents می توان در داخل متغیرهای Session ذخیره شده در یک Session کاربر (بجز در آجکت های متغیر Session ایجاد شده در یک صفحه Asp) وقتی از آجکت Session برای پایدار نگهداشتن سایت وب خود استفاده می کنید فکر کنید آجکت Session یک انبار می شود و متغیرهای Session کالاهای داخل انبار می باشند. چون هر کاربر، Session مختص به خود را دارد هر کاربر می تواند مقادیر متغیر Session منحصر به فردی را داشته باشد.

### دامهای متغیرهای Session:

وقتی از متغیرهای Session در نرم افزار وب خود استفاده می کنید از چند تله عادی اجتناب کنید:

- دام ۱- قرار دادن آبجکت ها در یک Session کاربر
  - دام ۲- تنظیم خاصیت Timeout به یک مقدار غیر بهینه
  - دام ۳- ایجاد متغیرهای Session غیر ضروری
- چون هر کاربر برای خودش نمونه‌ای از آبجکت Session را تخصیص می‌دهد، هر چه تعداد زیادی کاربر بطور همزمان از سایت تان بازدید کنند، نمونه‌های بیشتری از Session مورد نیاز می‌باشد. اگر شما آبجکت‌های بزرگی را داخل Session کاربر قرار دهید هر آبجکت Session رشد خواهد کرد و به حافظه بیشتری از سرویس دهنده وب نیاز پیدا می‌کند.

### دادن دامنه کاربردی به یک شیء

همانطور که آبجکت‌های Session بزرگتر و بیشتر نمونه‌سازی می‌شوند سرویس دهنده وب شما کندتر خواهد شد. به همین دلیل عاقلانه است که آبجکت‌ها را از Session خارج کنید.

**تله دوم** وقتی است که هنگام استفاده از آبجکت Session، خاصیت Timeout را یک مقدار غیر بهینه قرار می‌دهند. وقتی کاربر برای مدت زمان معینی سایت شما را مورد بازدید قرار نداد، Session کاربر از حافظه آزاد می‌شود. این طول از زمان قبل از اینکه

Session پایان پذیرید را اصطلاحاً Session Timeout گویند. این مقدار را خیلی بزرگ

نگیرید.

**تله سوم:** اگر شما در حال استفاده از متغیرهای Session برای ذخیره اطلاعاتی

هستید که کاربر معینی ندارند به جای آن از آبجکت Application استفاده کنید.

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

## فصل دوم

شی‌ها در ASP

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

ASP شامل ۶ شیء مخصوص است و به دلیل اینکه آنها در Active server page

ایجاد و کامپایل شده اند دیگر نیاز به ایجاد دوباره آنها نمی باشد و می توان آنها را در

اسکرپت ها استفاده کرد.

در این بخش به شرح هر یک از این شیء ها و خاصیت های آنها و متدهای آنها

خواهیم پرداخت.

### Application object

از این شیء برای به اشتراک گذاردن اطلاعات بین همه کاربرها استفاده می شود و

بدلیل اینکه این شیء چندین کاربر به اشتراک گذاشته می شود برای اطمینان از اینکه چند

کاربر بطور همزمان روی یک مقدار کار انجام ندهند از متدهای Lock و Unlock استفاده

می شود.

فرمت این شیء بصورت زیر می باشد.

Application. Method

این شیء شامل متدهای Lock و Unlock و واقعه های Application – onstart و

Application – onend می باشد.

مثالهایی از این شیء در زیر آورده شده است.

<%

Application (“num”) = 25

%>

<%

Set Application (“obj”) = Server Create object (“mycomponent”)

%>

اگر یک آرایه را در این شیء ذخیره کنید نمی توانید بطور مستقیم یک عنصری را به

آن اضافه کنید مثلا اسکریپت زیر اجرا نمی شوند.

<%

Application (“Store Array”) (3) = “new valve”

%>

### Request object

این شیء برای انتقال اطلاعات به سرور استفاده می شود.

فرمت آن بصورت زیر می باشد.

Request [ . Collection ] (Variable)

این شیء دارای کلکسیون های<sup>۴</sup> زیر می باشد:

Clientcertificate: مشخصات کاربر که در Http درخواست شده را به سرور انتقال

می دهد.

Cookies: مقادیری از Cookie که به سرور فرستاده می شود در این کلکسیون مشخص

می گردد.

From: مقادیری از عناصر فرم که به سرور فرستاده می شود.



Query String: مقادیری از متغیرها که به سرور فرستاده می شود.

Server Variables: متغیرهای محیطی که برای سرور تعریف شده را مشخص می کند.

اکنون برخی از این کلکسیونها را شرح می دهیم.

### **:Clientcertificate**

این کلکسیون مشخصات فیلدهایی که از طرف مرورگر وب صادر می شود را نگه

می دارد قبل از استفاده از این کلکسیون باید وب سرور را پیکربندی کنید.

فرمت این کلکسیون بصورت زیر می باشد.

Request. Clientcertificate (key [ subfield ])

برای مشاهده همه کلیدهای این کلکسیون از اسکریپت زیر استفاده کنید:

```
<% For each key in Request. Clientcertificate
```

```
Response. Write (key &": "& Request. Clientcertificate (key) & "<BR>")
```

```
Next
```

```
%>
```

### **:Cookies**

این کلکسیون امکان دسترسی به مقادیر Cookie های فرستاده شده که یک آدرس

http درخواست شده را فراهم می کند فرمت این کلکسیون بصورت زیر می باشد.

Request. Cookies (Cookie) [ (key) 1. Attribute

در این قسمت پارامترهای این کلکسیون را شرح می دهیم:

Cookie: اسم Cookie را مشخص می کند.

Key: یک پارامتر اختیاری می باشد.

Attribute: خاصیت هایی که Cookie می تواند دارا باشد و از جمله این خاصیت ها

Haskeys می باشد که مشخص می کند آیا Cookie دارای کلید است یا خیر.

### :Session object

از این شی برای ذخیره اطلاعات مورد نیاز یک کاربر خاص استفاده می گردد.

هنگامی که کاربر یک صفحه وب را درخواست می کند وب سرور بطور اتوماتیک یک

Session object ایجاد می کند و زمانی این شی از بین می رود که آن Session پایان

پذیرد. از این شی برای ذخیره کردن سلیقه های کاربران روی صفحاتی که قبلا مشاهده کرده

اند استفاده می شود البته از این خاصیت زمانی می توان استفاده کرد که مرورگر کاربر،

Cookie را پشتیبانی می کند.

فرمت این شیء بصورت زیر می باشد:

Session. Property | method

این شی دارای خاصیت های زیر می باشد.

:Session ID

این خاصیت مشخصات یک Session را برای این کاربر برمی گرداند. در اصل هر

Session یک ID منحصر به فرد دارد و در زمانی که این Session ایجاد می گردد توسط

سرور تولید می شود.

## :Time Out

مدت زمانی که یک Session می تواند باقی بماند را به دقیقه مشخص می کند اگر در این فاصله زمانی کاربر صفحه را بازبایی نکند و یا صفحه دیگری را درخواست نکند این Session پایان می پذیرد که System آن بصورت زیر می باشد:

Session Time Out [ = nminutes ]

این شیء شامل متد Abandon می باشد که در ادامه توضیح خواهیم داد.

از این شیء برای ذخیره اطلاعات نیز استفاده می گردد به مثالهای زیر توجه کنید.

```
<% Session ("name") = "Ali"
```

```
Session ("age") = 24
```

```
%>
```

برای ذخیره کردن یک شیء از دستور زیر استفاده کنید.

```
<% Set Session ("obj") = server.Createobject ("my component") %>
```

آنگاه می توانید متد یا خاصیت های این شیء را در صفحات وب فراخوانی کنید.

همچنین می توانید یک آرایه را در این شیء ذخیره کنید در این حالت اگر

می خواهید تغییراتی در عناصر آرایه ایجاد کنید یا عنصری را اضافه کنید ابتدا باید یک کپی

از این آرایه تهیه کرده و تغییرات را روی آن اعمال گردانید آنگاه دوباره آرایه را در

Session object ذخیره کنید.

## :Response object

این شیء برای انتقال اطلاعات خروجی به کاربر استفاده می شود و فرمت این شیء بصورت زیر می باشد.

Response. Collection | property | method

کلکسیون این شیء Cookie می باشد که به شرح آن می پردازیم

## :Cookies

بطور کلی این کلکسیون برای انتقال یک سری مقادیر و اطلاعات از صفحه ای به صفحه ای دیگر مورد استفاده قرار می گیرد. در صفحه ای ابتدا بوسیله Response مقدار لازم را در Cookies قرار می دهیم آنگاه در صفحه مقصد بوسیله Request مقدار را بازیابی می کنیم. فرمت آن بصورت زیر می باشد.

sResponse. Cookies (Cookie) [ (Key) | Attribute ] = Value

پارامترهای این کلکسیون در زیر شرح داده شده است.

Cookie: اسم کلکسیون مورد نظر می باشد.

Key: یک پارامتر اختیاری می باشد.

Attribute: اطلاعاتی درباره Cookie مشخص می کند و می تواند یکی از مقادیر زیر باشد.

Exprise: که تاریخ انقضاء این کلکسیون را مشخص می کند.

Domain: تعیین می کند این کلکسیون تنها در به درخواستهای آن Domain انتقال یابد.

Path: مشخص می کند کلکسیون تنها به مسیر درخواست شده فرستاده می شود.

Haskeys: معین می کند آیا این کلکسیون شامل کلید است یا خیر.

Value: مقدار نسبت داده شده به کلید یا ویژگی را نشان می دهد.

مثال زیر یک Cookie با کلید را مشخص می کند.

<%

Response. Cookies ("mycookie") ("Type1") = "Suger"

Response. Cookies ("mycookie") ("Type2") = "giger snap"

برای مشخص شدن اینکه یک Cookie دارای کلید است یا خیر از اسکریپت زیر

استفاده کنید.

<% Response. Cookies ("mycookie"). Haskeys %>

اگر Cookie دارای کلید باشد مقدار TRUE بر گردانده می شود در غیر این صورت

مقدار FLASH برمی گرداند.

**:End**

این متد باعث می گردد که وب سرور پردازش همه اسکریپت ها را متوقف کند و

خروجی را به Client بر گرداند و ادامه فایل پردازش نشود.

**نکته:**

اگر صفحه ای بافر شده باشد این متد همانند Flash عمل می کند و اگر قصد دارید

خروجی به کاربر بر گردانده شود بصورت زیر عمل کنید.

<%

Response. Clear

Response. End

%>

### **:Flash**

این متد باعث می گردد که خروجی بافر شده فوراً به کار برگردد اما اگر خاصیت

Buffer صفحه ای True نباشد این متد خطا برمی گرداند.

### **:Redirect**

به کمک این متد می توان در هر نقطه از برنامه بک یک آدرس URL دیگری

متصل شد فرمت این متد بصورت زیر می باشد.

Response. Redirect URL

### **:Write**

این متد یک رشته را به خروجی می فرستد و فرمت آن بصورت زیر می باشد.

Response. Write Variant

Variant شامل رشته ای است که باید به خروجی فرستاده شود این رشته هر نوع داده

که در زبان VBScript پشتیبانی می شود را شامل می شود.

مثال:

I just want to say <% Response. Write "Hello worls" %>

Your name is <% Response. Write Request. Form ("name") %>

همچنین به کمک این متد می توان یک تگ Html را به صفحه وب خروجی اضافه

کرد.

### Server object

این شیء برای دسترسی به متدها و خاصیت های سرویس دهنده استفاده می شود.

فرمت آن به صورت زیر است.

Server. Method

این شیء شامل خاصیت زیر می باشد.

### :Script time out

مدت زمانی که یک اسکریپت می تواند اجرا شود را مشخص می کند.

مدت زمان در نظر گرفته شده نمی تواند کمتر از مقدار پیش فرض باشد یعنی کمتر از زمانی

که در Registry setting مشخص شده است.

مثال:

<% Server. Script time out = 30%>

این شیء دارای متدهای زیر می باشد.

### :Createobject

این متد برای ایجاد شیء از یک کامپونت می باشد فرمت آن بصورت زیر است.

Server. Create object (prog ID)

**:Prog ID**

نوع شیء را مشخص می کند. این متد بطور پیش فرض دارای حوزه صفحه می باشد

یعنی بطور پیش فرض بعد از تمام شدن پردازش صفحه خراب می شود.

**:Mappath**

این متد مسیر مجازی یا نسبی را به یک دایرکتوری فیزیکی روی سرور دهنده

مطابقت می دهد فرمت آن در زیر مشخص شده است.

Server. Mappath (path)

این متد نمی تواند تشخیص دهد که مسیر برگردانده شده روی سرور دهنده

موجود است یا خیر.

ASP Error object: آجکت ASP Error یک ویژگی جدید در ASP می باشد. این

آجکت به شما اجازه می دهد تا در مورد خطاهای اسکرپت ایجاد شده در صفحه هایتان

اطلاعاتی را بدست بیاورید.

برای اطلاعات بیشتر در مورد این آجکت به صفحه وب MSDN مراجعه کنید.

(<http://www.Msdn.Microsoft.Com>)



[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

## فصل سوم

نوشتن اسکریپت های ساده

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

این بخش شامل موارد زیر است:

۱- فرستادن کاربر به صفحه دیگر

۲- فرستادن نوع فایل ها

۳- بافر کردن صفحات ASP

۴- Chaching صفحات ASP

۵- فرستادن فایل ها به Server

۶- دریافت اطلاعات از کاربر.

۷- پردازش فرم ورودی

۸- اشکال زدایی از ASP

۹- استفاده از بانک های اطلاعاتی

**فرستادن کاربر به صفحه دیگر**

احتمالا با صفحات وبی روبرو شده اید که روی صفحه نمایش فقط برای چند ثانیه

نمایش داده می شوند و بعد ناگهان شما را به صفحه دیگر می برند. یکی از راههای انجام این

کار استفاده از META tag می باشد.

```
<META HTTP – EQUIV = REFRESH CONTENT = "2; URL = http: //
```

```
www. Mamillon. Com">
```

این Tag سبب می شود که مرورگر به آدرس [www. Macmillun](http://www.Macmillun) بعد از ۲ ثانیه

ارسال گردد. همچنین این کار می تواند با استفاده از اسکریپت های Client – side مانند

Java Script انجام شود.

### فرستادن نوع فایل ها:

Response. Redirect URL کاربر را به صفحه ای که آرگومان URL معین

می کند می برد. اگر آدرس مزبور در همان سایت باشد یک URL وابسته کار خواهد کرد،

اگر آدرس مربوط در سایت دیگری باشد باید آدرس کامل شامل عبارت `http://` به کار

برده شود.

هر زمانی که سرویس گیرنده یک تقاضا برای صفحه ASP به خصوص را می کند،

یک آبجکت Context ایجاد می شود. یک آبجکت Context چیزهایی مثل آبجکت های

Session و Request و تعدادی متغیرهای سرور را در خود نگهداری می کند. Response.

Redirect به این صورت کار می کند که به سرویس گیرنده می گوید که تقاضای یک

صفحه جدید نماید.

سرویس گیرنده این کار را انجام می دهد و یک آبجکت جدید Content برای

صفحه جدید ایجاد می شود و فقط وقتی کار می کند که چیزی برای سرویس گیرنده ارسال

نشده باشد. این به آن معنی است که باید قبل از اینکه هر خروجی با استفاده از بافرینگ به

Client فرستاده شود مقداری را در Response.Redirect قرار دهید.

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

## بافر کردن صفحات ASP

علاوه بر فرستادن خروجی به سرویس گیرنده آبجکت Response می تواند چگونگی و زمان فرستادن خروجی به سرویس گیرنده را نیز کنترل کند. خروجی می تواند به دو شکل مختلف ارسال شود. شیوه بافر شده و بافر نشده. خروجی بافر نشده فوراً ارسال می شود. خروجی بافر شده تا زمانی که اسکرپت تمام نشده یا تا زمانی که یک دستور خاص دریافت نشده ارسال نمی شود. شما می توانید با استفاده از یکی از خصوصیات Response به نام Buffer، تعیین کنید که عمل بافر کردن انجام شود.

`Response.Buffer = True` عمل بافر کردن را فعال می کند. اگر می خواهید که عمل بافر کردن را تنظیم کنید قبل از اینکه خروجی فرستاده شود این کار را انجام دهید. ممکن است صفحه ای داشته باشید که زمان زیادی برای تمام شدن پردازش هایش لازم است و در عین حال می خواهید بافرینگ را غیرفعال کنید تا اینکه کاربران بتوانند شروع به خواندن بخشی از خروجی تان نمایند تا بقیه عمل پردازش انجام شود. برعکس ممکن است نخواسته باشید که کاربران هیچ چیزی را ببینند تا تمام پردازش انجام گیرد. ممکن است خواسته باشید که به کاربران اجازه دهید تا یک بخش از خروجی را ببینند و منتظر بمانند تا صفحه، قبل از فرستاده شدن مابقی آن انجام شود.

```
<% @Language = VBScript %>
<% Option Explicit %>
<% Response. Buffer = True %>
<HTML>
<BODY>
<%
Dim Ingcounter, Ingtimewaster
&. Response. Write (" It is now" "& now)
Response. Write ("<BR>")
For Ingcounter = 1 To 5000000
Ingtimewaster = Ingtimewaster + 1
Next
Response. Write ("IT is now: "& Now)
%>
</BODY>
</HTML>
```

## تجزیه و تحلیل

در خط ۸ زمان فعلی در داخل بافر نوشته می شود. خطوط ۱۰ تا ۱۲ یک حلقه

For – next را نشان می دهد که فقط وقت تلف می کند. بعد در خط ۱۳ زمان فعلی

جدید در بافر می رود. اسکریپت بعد از خط ۱۶ تمام شده و خروجی به سرویس گیرنده

ارسال می شود. با اینکه هر دو خط خروجی در یک زمان به مرورگر رسیده اند در زمان های

متفاوتی تولید شده اند.

**Chaching صفحات ASP:**

ممکن است توجه کرده باشید که وقتی یک صفحه وب را مشاهده می کنید و برای چند لحظه آن را ترک می کنید و دوباره به صفحه برمی گردید مرورگر تا آن صفحه را دوباره دریافت نمی کند. این بخاطر عمل کاشه کردن صفحه است.

صفحات HTML و تصاویر می توانند در کاشه ذخیره شوند.

کاشه سرعت حرکت صفحات را بالا می برد اما باعث بروز پاره مشکلات می شود. اگر یک سرویس گیرنده یک صفحه را از کاشه خود به جای سرویس دهنده دریافت کند، ممکن است نسخه آن صفحه روی سرویس دهنده تغییر کرده باشد. ASP بصورت پویا عمل می کند. اغلب محتویات یک صفحه ASP که در حال مشاهده می باشد توسط سرویس گیرنده تغییر می کند در حالی که کد ASP همان مد قبلی می باشد. کد ASP وابسته به زمان است. با استفاده از خصوصیت Exprise معین می کنید که یک صفحه چه مدتی در داخل کاشه نگهداری شود؟

Response. Exprise = number

یعنی مرورگری که از نسخه کاشه شده یک صفحه استفاده می کند باید بعد از، Number دقیقه، صفحه را دوباره دریافت کند.

Response. Epires Absovte = Date item به مرورگر می گوید که نسخه کاشه شده صفحه باید در تاریخی که آرگومان Date و زمانی که Time نشان می دهد دوباره از سرور گرفته شود. اگر Date حذف شود تاریخ فعلی در نظر گرفته می شود.

## فرستادن فایلها به سرور

یک مرورگر می تواند از برنامه Posting Acceptor برای فرستادن فایلها به سرور استفاده کند. برای مثال می توان اسکریپتی نوشت که بطور اتوماتیک به اسمها و موقعیتهایی از فایلها، Email بفرستد.

هر چند ASP بیشتر برای پردازش صفحات وب روی سرور مورد استفاده قرار می گیرد اما می توان کارایی آن را توسعه داد بدین ترتیب که از آن برای ایجاد سایت کاربرهایی که روی مرورگر کاربر پردازش می شوند استفاده کرد روش ایجاد بدین صورت است که اسکریپت های سایت کاربر را که بوسیله تگ های HTML محدود شده اند را با اسکریپت های سایت سرور که با `<% %>` محدود شده اند ترکیب می کنیم.

```
<Script Language = "VB Script">
```

```
<! .....
```

```
Client Script
```

```
<% Server Script%>
```

```
Client Script
```

```
<% Server Scripts>
```

```
.
```

```
.
```

```
.
```

```
</ Script>
```

برای اضافه کردن یک فایل ASP از فرم زیر استفاده می کنیم.



```
<! ..... #include Virtual file = "File name" ....>
```

لغات کلیدی Virtual و File نوع مسیری که شما برای Include کردن فایل استفاده

می کنید را مشخص می کنند و File Name مسیر و اسم فایلی است که شما برای Include

کردن بخواهید.

این فایلها نیاز به پسوند و اسم مشخص ندارند اما اصولاً برای تشخیص آنها از دیگر

فایلها از پسوند Inc استفاده می کنیم لغت کلیدی Virtual برای مشخص کردن مسیر شروع

با یک دایرکتوری مجازی می باشد.

مثال:

```
<! ..... #include Virtual = "my app / footer. inx" ....>
```

مثال فوق محتوایی از footer. Inc را در فایلی که شامل این خط شود قرار می دهد.

لغات کلیدی file برای مشخص کردن یک مسیر نسبی می باشد یک مسیر نسبی، با

دایرکتوری شامل فایل include شده شروع می شود برای مثال اگر شما یک فایل در

دایرکتوری my app داشته باشید و فایل Header 1. Inc در My app \ Headers باشد

خط زیر فایل مربوطه را در فایل ما قرار می دهد.

```
<! ..... #include file = "Headers / header |. inc ....>
```

دریافت اطلاعات از کاربر

تمامی کاربران تنها یک چیز را مشاهده می کنند. هیچ نوع سفارشی کردن صفحات وجود ندارد. برای قرار دادن این نمونه ها از خروجی باید سناریوی زیر را دنبال کنید.

وقتی شما صفحه وب را بازدید می کنید مرورگر تان بر روی Client یک درخواست برای سرویس دهنده وب برای درخواست یک صفحه وب بخصوص ارسال می کند.

سرویس دهنده وب در جواب سند درخواست شده را برای سرویس گیرنده ارسال می کند.

وقتی درخواست مربوط به یک صفحه ASP می باشد سرویس دهنده وب ابتدا کدهای موجود در صفحه ASP را قبل از ارسال صفحه وب به سرویس گیرنده مورد پردازش قرار می دهد. حال اگر بخواهیم که صفحه ASP ما، بر پایه ورودی کاربر تصمیم گیری کند نیاز به فرم ها داریم. یک فرم دو وظیفه دارد: جمع آوری اطلاعات از کاربر و فرستادن آن اطلاعات به یک صفحه وب دیگر، برای انجام عمل پردازش روی آنها. بنابراین با استفاده از فرم، یک صفحه ASP می تواند ورودیهای کاربر را دریافت کند و تصمیم های برنامه ریزی شده ای روی آنها اعمال کند.

### پردازش فرمهای ورودی با Asp

با استفاده از یک مرورگر وب استاندارد، یک کاربر می تواند در داخل صفحه وب که دارای یک فرم می باشد با وارد کردن اطلاعات که گشت و گذار پردازد. وقتی کاربر این کار را انجام می دهد اطلاعاتی که او در حال تایپ کردن می باشد هنوز برای سرویس دهنده

وب ارسال نشده است. این اطلاعات برای سرویس دهنده وب تا زمانی که کاربر به وسیله کلیک کردن بر روی دکمه Submit، فرم اطلاعات را تایید نکنید قابل دسترس نمی باشد. کارتان وقتی جالب است که قادر باشید تا از روی اطلاعات ارسال شده به یک صفحه ASP، تعیین کنید که کدام کاربر اطلاعات را داخل فرم وارد کرده و بعد روی اطلاعات پردازش انجام دهید. تگ <FORM> دو خصوصیت دارد که به شما اجزا می دهد تا روی اطلاعات فرستاده شده به یک صفحه ASP عمل پردازش انجام دهید. خصوصیت ACTION و METHOD، با استفاده از خصوصیت Action یک فرم، می توان هر URL معتبری را تنظیم کرد. وقتی کاربر فرم را Submit می کند URL تعیین شده در Action فراخوانی می شود. و مقادیر موجود در فیلدهای فرم به آن URL ارسال می شود.

### تفاوت بین GET و Post

دو راه برای عبور اطلاعات از یک فرم به یک صفحه ASP وجود دارد. اولین متد استفاده از Querystring می باشد، این متد وقتی مورد استفاده قرار می گیرد که خصوصیت METHOD فرم به GET تنظیم شده باشد. متد دیگر، Post، اطلاعات کاربر را با استفاده نکردن از Querystring پنهان می کند.

Querystring اطلاعات اضافه ای است که با چسبیدن به انتهای URL برای صفحه وب ارسال می شود.

## تعیین درستی فرم ورودی

یک فرم پردازش اسکریپت خوب باید قبل از پردازش، صحت و سقم اطلاعات وارد شده را بررسی کند یک اسکریپت معتبر می تواند چک کند که کاربر اطلاعات را به صورت صحیح در فرم وارد کرده یا خیر یعنی در واقع آیا نوع داده ها درست هستند. مثلا اگر سایت وب ما شامل یک فرم برای محاسبه اطلاعات ما می باشد باید اطلاعات وارد شده توسط کاربر چک شود که حتما عددی باشند (این کار باید قبل از پردازش داده ها انجام شود). یک روش مخصوص برای معتبر ساختن اطلاعات ورودی ایجاد فرمی است که اطلاعات را به خودش می فرستد. در این حالت فایل asp شامل فرمی است که اطلاعات را جبران می کند.

برای مثال اسکریپت زیر تعیین می کند که آیا کاربر یک عدد در فیلد سن وارد کرده

یا خیر.

```
<% if Isnumeric (Request. Query string ("Age")) then  
%>
```

```
<p> Hello your 5 age is <% = Request. Query string  
("Age") %>
```

```
<% Else %>
```

```
<p> Please enter a numerical age
```

```
<% End if %>
```

```
<FORM METHOD="Post" Action = "Verigy. Asp">
```

```
Name: <INPUT Type = "Text" Name = "Name">
```

```
Age: <INPUT Type = "Text" Name = "Age">
```

```
<INPUT Type = "Submit" Value, "Eter">
```

```
</FORM>
```

در این مثال اسکریپت در فایل Verigy. Asp می باشد که شبیه فایلی است که فرم را

شامل می شود و در واقع فرم اطلاعات را با مشخص کردن Verify. Asp در Action به خودش می فرستد.

همچنین می توان اسکریپت های سایت کاربر را برای چک کردن اطلاعات کاربر

استفاده کرد، در این حالت برای اخطار کردن به کاربر در مورد داده های ورودی می توان از

مرورگر وب به منظور کاهش بار شبکه استفاده کرد. مثال زیر روی مرورگر وب کاربر اجرا

می شود و قبل از زدن کلید Submit اطلاعات وارد شده را چک می کند.

```
<SCRIPT LANGUAGE = "VB Script".
```

```
<! - -
```

```
Sub btn Enter – on elick
```

```
Dim the form
```

```
Set the form = Document. My form
```

```
If Isnumeric (the form. Age Value) then
```

```
The form. Submit Else
```

```
Msg box "please enter a numerical age"
```

```
End sub
```

```
// - ->
```

```
</SCRIPT>
```

```
<FORM METHOD> = "Post" Name = my form Action = "my file . asp">
```

```
Name: <INPUT Type = "Text" Name = "Name">
Age: <INPUT Type = "Text" Name = "Age">
<INPUT Type = "button" Name = "btn Enter" Value = "Eter">
</FORM>
```

## اشکال زدائی

وقتی یک صفحه ASP را می نویسد، ممکن است اشکالاتی در داخل کدها وجود داشته باشد، خطاها همیشه با اشتباهات می آیند و نتیجه موارد خارجی هستند که به سختی می توان از آنها آگاهی یافت. اشکال زدایی و کنترل خطاها دو موضوع مهم در هنگام استفاده از یک برنامه هستند. با وجود اهمیت حیاتی اشکال زدایی و کنترل خطا، تعدادی از توسعه دهندگان وقت زیادی را صرف برنامه نویسی واقعی می نمایند. در هنگام اشکال زدایی نرم افزار ASP به دو نوع باگ برخورد خواهید کرد. اولین نوع باگ Fatal bug است.

**(خطای مخرب):** Fatal bug وقتی ایجاد می شود که برنامه در حال اجرا به صورت ناگهانی به پایان برسد.

**(خطای غیر مخرب):** خطایی است که اجرای برنامه را متوقف نمی کند بلکه باعث می شود که برنامه خروجی اشتباهی را برای ورودی های داده شده تولید کند. خطاهای مخرب که کدهایتان را خراب می کند سبب خروجی ناگهانی اجرای برنامه هایتان می شوند. خطاهای غیر مخرب لزوماً مشکل تر از خطاهای مخرب درست نمی شوند اما همیشه سخت تر مکان یابی می شوند زیرا در خطاهای مخرب شماره خط دقیقاً فهرست می شود.

## کنترل دقیق خطاهای ASP

### استفاده از آجکت Err:

در ASP 2.0 کنترل خطا به وسیلهٔ VBScript در دسترس قرار می‌گرفت. دستور زبان مشابه با کنترل خطای استفاده شده در برنامه VB بود. در هر صفحه‌ای که می‌خواستید توانایی کنترل خطا را داشته باشید مجبور بودید خط `On Error Resume Next` را وارد کنید که این خط به موتور VBScript می‌گوید که اگر با یک خطا مواجه شد، به خط بعدی پرش کند. پس هرگاه در جاهای گوناگون در کدهایتان نیاز به تست کردن داشتید تا ببینید آیا یک خطا اتفاق افتاده یا نه، این کار توسط آجکت Err انجام می‌شد.

### استفاده از آجکت ASP Error

هنگامی که توسعه دهندگان شرکت مایکروسافت در حال کار کردن روی ASP 3.0 بودند. کاری بزرگ را با آجکت جدید ASP Error انجام دادند. این آجکت پیش ساخته جدید در ASP 3.0 کنترل خطا را شبیه به آجکت Err انجام می‌دهد. ASP Error، معایب آجکت Err را برطرف کرده است.

چون ASP Error یک آجکت داخلی ASP است اهمیتی ندارد که از چه زبان اسکریپت Server – side استفاده می‌کنید. این کار اولین عیب آجکت Err را حل می‌کند، راه حل عیب دوم در این است که ASP Error به همراه IIS 5.0 هر زمانی که یک

خطا در صفحات ASP اتفاق می افتد خطای HTTP ۱۰۰ و ۵۰۰ اتفاق می افتد. یک خطای HTTP، خطای سطح پایینی است که روی سرویس دهنده وب در پاسخ به رفتارهای غیر منتظره اتفاق می افتد.

### استفاده از بانک های اطلاعاتی

در حال حاضر از ActiveX data object (ADO) برای دسترسی به پایگاه داده استفاده می شود.

### ایجاد یک ODBC

قبل از اینکه اسکریپت های پایگاه داده را ایجاد کنید لازم است روشی را بر این ADO تهیه کنید. مشخص کردن روش مناسب برای ADO به منظور مشخص کردن، شناساندن و ارتباط برقرار کردن با پایگاه داده خودتان لازم است. برنامه های راه انداز پایگاه داده اطلاعات را از وب شما به یک پایگاه داده که از DSN به منظور مشخص کردن و شناساندن یک ODBC مخصوص استفاده می کنند، عبور می دهد.

DSN شامل مشخصات پایگاه داده و امنیت کاربر می باشد و می تواند فرم را از بیرون بگیرد یا یک فایل متنی یا ODBC می تواند نوع DSN که شما می خواهید ایجاد کنید را

انتخاب کند، انواع DSN عبارتند از File DSN System DSN ' user DSN



User / System DSN ها در رجیسترهای ویندوز NT قرار دارند، System DSN

این امکان را فراهم می کند که همه کاربرها روی یک سرور مخصوص برای دسترسی به پایگاه داده اقدام کنند. در شرایطی که User DSN ارتباطات پایگاه داده را به یک کاربر مخصوص با وسایل امنیتی محدود کرده است.

File DSN که فرم را از یک فایل متنی می گیرد امکان دسترسی را برای چندین

کاربر تهیه می کند و خیلی راحت انتقال از یک سرور به سرور دیگر است.

برای ایجاد یک File DSN ابتدا در Control pannel روی شمای ODBC دو بار

کلیم کنید سپس خاصیت File DSN را انتخاب کنید، روی Add کلیک کنید و درایو

پایگاه داده خودتان را انتخاب کنید. و سپس روی Next کلیک کنید.

پیشرفت دستورات برای پیکربندی یک DSN به پایگاه داده مخصوص شما بستگی

دارد.

۱- داخل جعبه Driver Microsoft Access create New data source را انتخاب کنید

و کلید Next را بفشارید.

۲- یک اسم را برای DSN فایل تایپ کنید و Next را بفشارید.

۳- برای ایجاد یک منبع داده روی پایان کلیک کنید.

۴- روی جعبه ODBC microsoft Access 97 setup کلیک کنید و یک فایل پایگاه داده

Access (\*. Mdb) را انتخاب کنید و OK.

ASP فایل‌های به اشتراک گذارده شده را پشتیبانی می کند هر چند مثال‌های ASP از

فایل‌های اشتراکی پایگاه داده استفاده می کند اما توصیه می شود که این نوع از Engein‌های پایگاه داده تنها برای هدف‌های خاصی استفاده شود.

### مراحل پیکربندی یک DSN SQL Server Dat base file

۱- در جعبه Create New data source to SQL Server را انتخاب کنید و به مرحله بعد بروید.

۲- اسم مورد نظر را برای فایل DSN بنویسید و به مرحله بعد بروید.

۳- روی دکمه پایان کلیک کنید.

۴- کلمه شناسایی و کلمه عبور خودتان را وارد نمایید.

۵- در جعبه Create New data source to SQL Server اسم سروری را که شامل پایگاه SQL Server می شود را بنویسید و به مرحله بعد بروید.

۶- متدی را برای اینکه صحت و درستی کلمه شناسایی را بررسی کنید انتخاب کنید.

۷- اگر SQL Server authentication را انتخاب می کنید لازم است یک کلمه عبور و ID وار کنید و به مرحله بعد بروید.

۸- در جعبه Create New data source to SQL یک متد انتقال را انتخاب کنید.

۹- در همان جعبه روش رمزگشایی را انتخاب کنید.

۱۰- در جعبه ODBC Microsoft server setup روی Test data source کلیک کنید.

اگر DSN بطور صحیح انتخاب شده باشد نتیجه با موفقیت اعلام خواهد شد.

توجه: فایل های DSN دارای پسوند DSN هستند و در دایرکتوری زیر قرار دارند.

\ Programs \ Common Files \ ODBC \ Data Source

## اتصال به یک پایگاه داده

در وهله اول برای دسترسی به اطلاعات پایگاه داده باید ارتباط با منبع پایگاه برقرار کنیم. ADO شی لازم برای اتصال را تهیه می کند که شما از آن برای برقراری و مدیریت ارتباط بین برنامه خودتان و ODBC پایگاه داده استفاده می کنید.

شیء اتصال ۵ خاصیتها و متدهای مختلفی دارد که می توان از آن برای باز و بسته کردن ارتباط پایگاه داده و به روز درآوردن اطلاعات استفاده کرد.

سپس در قدم اول با یک شیء اتصال ایجاد کنید مثال زیر این شیء را ایجاد کرده و پردازش می کند و از آن برای باز کردن یک ارتباط استفاده می کند.

<%

Create a Connection object

Set Cn = Server.Creat object ("ADODB.Connection")

Open a Connection, the string oefers to the DSN

Cn. Open "FILE DSN = My data base. DSN"

%>

در رشته DSN نباید قبل و بعد از مساوی (=) فضای خالی وجود داشته باشد.

---

5- Connection object

6-Connection

7-Recordset

در این مثال متد Open به یک فایل DSN که شامل اطلاعاتی در مورد موقعیت و

پیکربندی پایگاه است اشاره می کند.

نیز می توان بطور صریح به سرور، کلمه عبور، کلمه شناسایی و غیره اشاره کرد.

## اجرای درخواست با شیء اتصال

با متد Executive می توان از SQL برای درخواست زدن روی منبع پایگاه داده و

اصلاح نتایج استفاده کرد در مثال زیر از این متد برای زدن درخواست در یک فرم بوسیله

SQL استفاده شده که داده ای را در یک جدول پایگاه داده بخصوص وارد می کند.

```
<% Define file based DSN
```

```
Str DSN = "“FILE DSN = My data base. DSN”
```

```
Str Cn = Server.Creat object (“ADODB. Connection”)
```

```
En. Open = Str DSN
```

```
Define SQI SELECT Statement
```

```
Str SQL = INTER INTO customer (First name, last name) ralues (jose,  
luge)
```

```
Use 4e the Execute method to is sue a SQL puery database
```

```
Cn. Execute (Str SQL)
```

در مثال فوق Jose در یک پایگاه داده به نام Customer وارد می شود.

همچنین می توان از دستورات UP date و Delete نیز استفاده کرد.

مثال زیر از دستور UP date برای تغییر اسم Jose به Jeff برای تمام فیلدهایی که

اسم فایل آنها Smith می باشد استفاده می کند.

<%

Set Cn = Server.Creat object ("ADODB. Connection")

Cn. Open "FILE DSN = My data base. DSN"

Cn. Execute "UPDATE customer SET First name = Jeff"

Where last name = "Smith"

%>

مثال استفاده از Delete

<%

Set Cn = Server.Creat object ("ADODB. Connection")

Cn. Open "FILE DSN = My data base. DSN"

Cn. Execute "DELETE From customer

Where last name = "Smith"

%>

### استفاده از شیء Recordset برای دستکاری نتایج

شیء اتصال یکسری محدودیت دارد، مخصوصا اینکه نمی توان با آن اسکرپت هایی

را ایجاد کرد که اطلاعات پایگاه داده را نشان بدهد و اصلاح کند.

دقیقا باید بدانید که چه تغییراتی می توانید روی پایگاه داده ایجاد نمایید برای اصلاح

داده و نتایج را امتحان کنید و تغییراتی را روی پایگاه داده تان ایجاد کنید.

ADO یک شیء دیگر به نام Recordset ایجاد می کند که دارای ویژگی هایی است

که بسته به محدودیت درخواست می توانید از آن برای نشان دادن یک دسته سطر و یا

رکورد پایگاه داده استفاده کنید این شیء موقعیت هر رکوردی را که بوسیله درخواست برگردانده می شود را نگه می دارد.

بسته به اینکه چگونه خاصیت کرزر این شیء را پیکربندی کنید می توان بین پایگاه داده اسکرول کنید و رکورد را به روز در آورید.

کرزر پایگاه داده شبیه اشاره گر ها عمل می کنند و اجازه می دهند که یک عنصر بخصوص را در یک دسته از رکوردها پیدا کنید این شیء ویژگیهایی دارد که شما را قادر می سازد که بطور صریح رفتار کرزر را کنترل کنید و نتایج را به روز در آورید.

کرزرها مخصوص برای اصلاح و بازرسی رکوردها مفید هستند. از ویژگیهای موقعیت کرزر و نوع کرزر می توان برای مشخص کردن یک نوع کرزر استفاده کرد که نتایج را به یک متد از برنامه کاربر شما برمی گرداند و آخرین تغییرات پایگاه داده که نتایج را به یک متد از برنامه کاربر شما برمی گرداند و آخرین تغییرات پایگاه داده که بوسیله دیگر کاربران ایجاد شده را نشان می دهد.

[www.kandoo.cn.com](http://www.kandoo.cn.com)

[www.kandoo.cn.com](http://www.kandoo.cn.com)

## اصلاح یک Recordset

در یک برنامه پایگاه داده موفق از هر دو شیء اتصال<sup>۶</sup> و دستکاری<sup>۷</sup> داده‌های برگردانده شده استفاده می‌شود در مثال زیر اسکرپت سمت سرویس دهنده از شیء Recordset برای اجرای دستور Select استفاده می‌کند.

```
<% Establish a Connection with data source
Str DSN = "FILE DSN = My data base. DSN"
Set Cn = Server.Create object ("ADODB. Connection")
Cn. Open Str DSN
Set Rscustomers = Server.Create object ("ADODB. Recorderst")
Strs QI = "Select First name, Last name FROM Customers
Where last name = "Smith"
Rscustomers. Open StrsQI, cn
Set obj first name = Rscustomers ("Last name")
Do Until Rscustomers. EofResponse. Write objfirst name & " " & obj last
name &
"<BR>"
Rscustomers. Move Next
Loop
%>
```



توجه کنید در مثال فوق شیء اتصال از تباط با پایگاه داده را برقرار می کند و شیء دستکاری داده های برگردانده شده از همان ارتباط برای اصلاح نتایج از پایگاه داده استفاده می کند.

اگر بخواهید تنها یک ارتباط برقرار کنید و از خاصیت های پیش فرض ADO استفاده کنید می توانید از متد `Recaedset, open` استفاده کنید.

یعنی در مثال قبل خطوط زیر می توانید حذف شوند.

```
Set Cn = Server.Creat object ("ADODB. Connection")
```

```
Cn. Open StrDSN
```

و خط زیر اضافه شود.

```
Rscustomers. Open StrSQL, StrDSN
```

### گسترش درخواستها با شیء Command

یکی دیگر از اشیای ADO و Command که همانند دو شیء قبلی عمل می کند به

استثنای اینکه با این شیء می توان درخواست را روی منبع پایگاه داده کامپایل کرد و سپس

مکررا درخواست را با مقادیر مختلف هماهنگ کرد و مورد استفاده قرار داد.

کامپایل کردن بدین صورت مفید است زیرا شما می توانید زمان درخواست شده

برای اصلاح دوباره یک درخواست موجود را بطور قابل ملاحظه ای کاهش دهید.

**ترکیب فرمهای HTML و پایگاه داده**

صفحات وب شامل فرمهای HTML، کاربر را قادر می سازند که به پایگاه داده  
دور دست درخواست برند و اطلاعات مخصوص را اصلاح کند.

به کمک ADO می توان اسکریپت های بسیار ساده ای برای جمع آوری اطلاعات،

فرمهای کاربر، ایجاد یک درخواست پایگاه داده و برگرداندن اطلاعات به کاربر نوشت.

به کمک Request object می توانید اطلاعات وارد شده به فرم HTML را اصلاح

کنید و این اطلاعات را با جملات SQL مطابقت دهید.

مثال زیر اطلاعات پاسخ داده شده بوسیله یک فرم HTML را به یک جدول اضافه

می کند. اسکریپت ها اطلاعات کاربر را بوسیله Request object form Collection

جمع آوری می کنند.

مثال:

```
SteDSN = "FILE DSN = My data base. DSN"
```

```
Set Cn = Server.Creat object ("ADODB. Connection")
```

```
Cn. Open StrDSN
```

```
Set Cm = Server.Creat object ("ADODB. Command")
```

```
cm. Active Connection = Cn
```

```
cm. Command text = "INSRT INTO myseedstable (type) Value (?)"
```

```
cm. parameters. Apped cm. Create parameters ("type". 200.225)
```

```
cm ("type) = Request ("Seed type")
```

```
cm. Execute
```