

رد: تحويل مشاركتك الي كتاب

الرسائل الخاصة

السلام عليكم

لا مانع لدي أخي الكريم من عملية النشر طالما فيها خيراً لأخواننا المسلمين والعرب، وأرجوا أن يكون الكتاب مجاناً ومتاحاً للجميع.

وأرجوا أن لا تنسى من وضع اسمك بآئك من قام بعملية ترتيب الدروس للـ asp وجمعها من المشاركات وأن تضع حقوق النشر لديك أيضاً.

مع التحية

عبدالله حابر شقيله

الأوس، ٢٢:٠٢:٩٠ مساءً

مدير منتديات المحيط العربي
الرتبة في المنتدى: عماد

المجموعة: إدارة المنتدى
المشاركات: ٢٧٥٢
العضو من: ٢٢
سبتمبر ٢٠٠٤، ١٢:٥٧:٢٧ مساءً
البلد: فلسطين
رقم العضوية: ١

[إضافة لقائمة الأصدقاء]

الإنتقال إلى: البريد الوارد

هذا الكتاب هو مشاركة العضو الكريم عبد الله وأرجوا من الله أن يوفقه للمزيد

قد طلبت منه أن يوافق علي تحويل مشاركته إلي كتاب وهذه الصورة للرد

فارجوا منكم الدعاء له ولي بظهر الغيب

الدرس الأول

مقدمة عن الـ asp

ما الذي يجب أن تعرفه؟

قبل أن أبدأ في شرح دروس الـ ASP يجب أن تعرف بعض القواعد الهامة في بناء الـ ASP، HTML-1- www، وقواعد بناء صفحات الشبكة.
2- يجب أن تعرف لغات السكريبات مثل: الجافا سكريبت والف بي سكريبت.

ما هي الـ ASP؟

- 1- ASP هي اختصار لـ (Active server pages) وهي عبارة عن صفحات الخادمة النشطة.
- 2- ASP هو برنامج ينفذ داخل الـ IIS.
- 3- IIS هي اختصار (Internet Information services) وهي بيانات أو معلومات خدمات الانترنت.
- 4- IIS تأتي مع اعدادت مركبات windows2000 و windows NT.4.0 option pack.

Option pack 5- تستطيع تحميلها من ميكروسوفت. PWS 67- تستطيع إيجادها في windows95,98 CD

ما هو ملف ال ASP ؟

-ملف ال ASP يستطيع أن يحتوي على نصوص، مثل أوسمة ال HTML و XML و السكريات، والسكريات في ملف ال ASP تنفذ على السيرفر، ملفات ال ASP امتدادها.(asp)

ماذا يستطيع ال ASP أن يقدم لك؟ -1 محرر ديناميكي، تستطيع أن تغير أو إضافة أي محتوى لصفحة الموقع.

2-يستجيب لاستعلامات المستخدم أو البيانات المرسله من خلال نماذج ال HTML.

3-الدخول إلى أي بيانات أو قواعد بيانات وإرجاع النتيجة إلى المتصفح.

4-يجعل صفحة الويب ملائمة لجميع المستخدمين.

5-إن من مميزات استخدام ال ASP بدلاً من لغة CGI و PERL هو السهولة والسرعة.

6-يضمن السريه والأمان بحيث كود ال ASP لأنه لا يظهر عند قيامك بعرض مصدر الصفحة على المصدر، لأنك الكود يكون مشفر.

7-بما أن ملفات ASP ترجع للمتصفح كصفحات HTML عاديه تستطيع عرضها من أي متصفح.

8-برامج ال ASP الذكيه تقلل من بطئ الشبكة.

ما هو الفرق بين ال ASP و HTML ؟

1-عندما يطلب المتصفح ملف ال HTML فإن الخادم (السيرفر) يرجع نفس الملف.

2-عندما يطلب المتصفح ملف ال ASP فإن ال IIS يمرر هذا الطلب إلى محرك ال ASP ، محرك ال ASP يقرأ ملف ال ASP سطر بعد سطر، وينفذ السكريت الموجود داخل الملف، في النهايه فإن ملف ال ASP يرجع إلى المتصفح كملف HTML.

صيغ ال:asp

أنت لا تستطيع أن ترى كود ال ASP عن طريق عرض المصدر في متصفحك، أنت الوحيد الذي تستطيع أن ترى كود ال ASP عند تصميمك لصفحات الخادمة النشطة، وذلك لأن السكريات تنفذ على السيرفر قبل أنت تظهر النتائج وترسل إلى المتصفح.

المفاهيم الأساسية -:ملفات ال ASP بشكل طبيعي تحتوي على أوسمة ال HTML ، وتحتوي أيضاً على سكريات السيرفر، ونصوص ال ASP تكتب دائماً بين هاتين العلامتين **<% %>** ، وسكريات السيرفر تنفذ على السيرفر، وتحتوي سكريات السيرفر على المعادلات، والجمل، والمعاملات الصحيحة للغات السكريت التي أنت تستخدمها.

ملاحظة :عندما تريد الكتابة باللغة العربية يجب أن تجعل اتجاه اللغة من اليمين إلى اليسار، وإذا لم تفعل ذلك سوف تلاحظ بأن النتيجة مكتوبة باللغة اللاتينية، فلذلك يجب أن تجعل اللغة باللغة العربية.

وبهذا نكون قد انتهينا من شرح المقدمة عنها.

وان شاءالله في الدروس القادمة سوف أوضح كيف يتم التعامل مع هذه اللغة

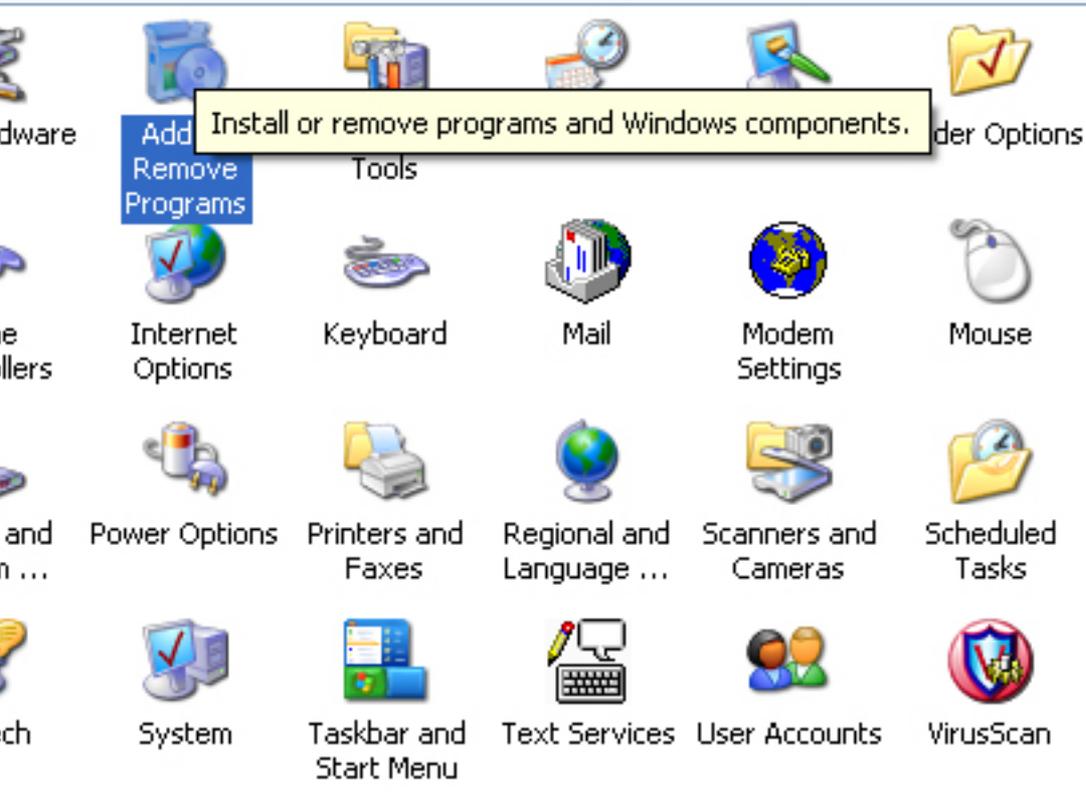
الدرس الثاني

تركيب سيرفر ال IIS

- أنت تستطيع تشغيل ال ASP على جهازك الشخصي بدون سيرفر خارجي، لعمل ذلك، أنت تحتاج إلى تنزيل PWS (Internet Information Server (IIS على جهازك الخاص بك

كيفية تنزيل سيرفر ال IIS على جهازك الخاص بك على (Windows 2000 و Windows XP ، سوف أقوم الآن بشرح كيفية تنزيل سيرفر ال IIS على جهازك الخاص بك على (Windows 2000 و Windows XP وهي نفس الطريقة متبعة في تنزيلها على Windows 2000.

أولاً: اذهب إلى (Control panel لوحة التحكم) ومن اضغط على (Add or Remove Programmes إضافة

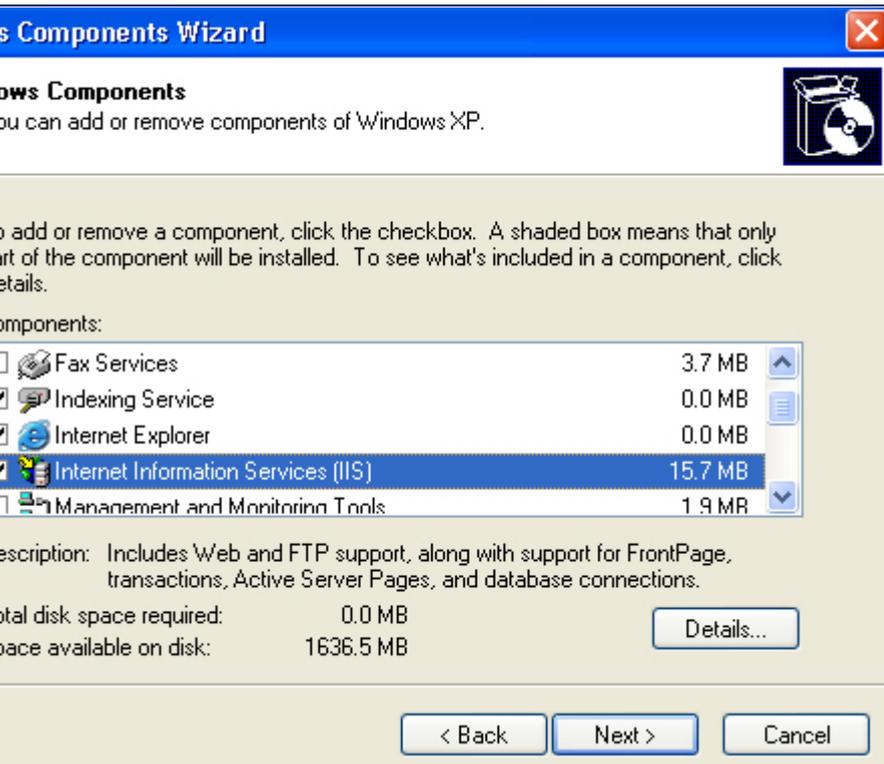


ثانياً: سوف تظهر لك النافذة التالية واختر منها (Add or Remove Windows Components

Programs: Sort by:

Program Name	Size	Used	Last Used On	Action
Audio Mixer 2.02	2.63MB	rarely	05/03/1424	Change/Remove
...	15.19MB			
...	89.99MB			
...	2.07MB			
...	3.58MB			
...	2.38MB			
...	0.90MB			
...	57.45MB			
...	1.36MB			
...	32.41MB			
...	38.24MB			
...	38.32MB			
...	57.07MB			
...	243.00MB			

ثالثاً: سوف تظهر لك النافذة التاليه، واختر منها: **Infomation server**



ضع عليها إشارة صح كما هو موضع في الشكل أعلى، ويجب أن ترى أن كل ما بداخل IIS موضوع عليه إشارة صح، و

يطلب منك اسطوانة الوندوز كما بإدخال الاسطوانه ومن ثم اضغط على OK

بعد أن قمت بعمل جميع هذه الإجراءات اذهب إلى القرص الصلب الذي قمت بتنزيل الوندوز عليه، سوف تجد ملف اسمه asp

عليه جميع العمل هو wwwroot حيث تخزن عليه جميع صفحات ال asp وأنا أنصح أن تضع جميع الصفحات

لتشغيل صفحات ال ASP قم بوضع على عنوان المتصفح التالي:

<http://localhost/test.asp>

أو

http://computer_name/test.asp

تستخدم هذه الطريقة عندما تكون صفحة test.asp موجوده داخل المجلد root

المقصود بـ Computer_name اسم الجهاز الخاص بك الذي قمت بتعريفه عندما قمت بتنزيل

أما إذا كانت صفحة ال `test.asp` موضوعة داخل مجلد فرضاً اسمه `MyWeb` داخل مجلد ال `wwwroot` فسوف

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

أما بالنسبة لتنزيلها على وندوز ٩٨ ، فيمكنكم متابعة شرح لتنزيلها على موقع عالم النور

www.alnoor-world.com/asp/topicbody....D=2&SectionID=1

وذلك لأنني لا يوجد على جهازي الخاص بـ وندوز ٩٨ ، لكي أشرح لكم كيفية

وبهذا نكون قد انتهينا من كيفية تثبيت سيرفر ال `iis` على جهازك

الدرس الثالث

كائن

Response Object

كائن - Response object: تستخدم لإرسال النتائج إلى المستخدم من السيرفر (الخادم).

الآن سوف أقوم بشرح كيفية التعامل مع هذا الكائن والتعرف على خصائصه عن طريق الأمثلة والتطبيقات.

كيف يتم كتابة النصوص داخل ASP ؟

- عن طريق المنهج `response.write` يتم كتابة النصوص داخل `ASP` ، وفي المثال التالي يوضح ذلك:-

كود

```
<html dir="rtl">
<body>
```

```
<%  
response.write ("العربي طريقك لاحتراف علوم أهلا وسهلا بيك في منتديات المحيط"  
البرمجة"  
%>  
</body>  
</html>
```

فسوف يكون ناتج المثال التالي:

كود

البرمجة أهلا وسهلا بيك في منتديات المحيط العربي طريقك لاحتراف علوم

شرح الكود:

- عند البدء في كتابة ملف ال ASP يجب تضع وسم البداية في ال ASP وبعدها تكتب الكود وبعد الإنتهاء من الكود تقوم بغلقه بوسم النهاية.

- أمر الكتابة في ملف ال ASP هو (" response.write. النصوص ")

ما هي لغات السكريبت المستخدمة مع ال ASP ؟

- أنت تستطيع أن تستخدم نوعين مختلفين من السكريبات:

النوع الأول: VBscript :

- لكي تضع مجموعة من سكريبات الفي بي سكريبت في الصفحة، يجب أن تعرف اللغة المستخدمة في أعلى الصفحة، والمثال التالي يوضح ذلك:

كود

```
<%@ language= "VBscript" %>  
<html dir="rtl">  
<body>  
<%  
العربي طريقك لاحتراف علوم أهلا وسهلا بيك في منتديات المحيط"
```

```
("البرمجة")
%>
</body>
</html>
```

النوع الثاني: Javascript :

-لكي تضع مجموعة من سكريبات الجافا سكربت في الصفحة، يجب أن تعرف اللغة المستخدمة في أعلى الصفحة، والمثال التالي يوضح ذلك:

كود

```
<%@ language= "Javascript" %>
<html dir="rtl">
<body>
<%
response.write("العربي طريقك لاحتراف علوم أهلا وسهلا بيك في منتديات المحيط")
البرمجة
%>
</body>
</html>
```

ملاحظة:

عند كتابة كود الجافا سكربت يجب أن تكتب الكود كما هو إذا كان الحرف كبير أو صغير لأنها حالة حساسة، أما كودالفي سكربت وال ASP على عكسها فتستطيع كتابته كما تشاء بحرف كبير أو صغير.

إرشاد: ليس من الشرط أن تقوم بتعريف لغة السكربت المستخدمة في الـ بي سكربت، ولكن يجب تعريفها في الجافا سكربت .

مثال: كيفية إضافة نصوص ال HTML إلى كود ال ASP ؟

كود

```
<html dir="rtl">
<head>
</head>
<body>
<%
response.write("<center><h2> أهلا وسهلا بيك في منتديات المحيط العربي طريقك
علوم البرمجة لاحتراف علوم
%>
<%
response.write("<center><p style='color:#33cc44'> أهلا وسهلا بيك في منتديات
علوم البرمجة المحيط العربي طريقك لاحتراف
%>
</body>
</html>
```

وسوف يكون الناتج كالتالي:

كود

أهلا وسهلا بيك في منتديات المحيط
لاحتراف علوم العربي طريقك
البرمجة

منتديات المحيط العربي طريقك لاحتراف علوم البرمجة أهلا وسهلا بيك في

تعريف المتغيرات -: المتغيرات تستخدم لتخزين البيانات.

المثال التالي سوف يوضح كيف يتم تعريف المتغيرات في ال.ASP

كود

```
<html dir="rtl">
<head>
</head>
<body>
<%
dim name
name="طريقك لاحتراف علوم البرمجة منتديات المحيط العربي"
response.write("<center><B>سهلا وسهلا" &name&"</b></center>")
%>
</body>
</html>
```

سوف يكون الناتج كالتالي:

كود

البرمجة أهلا وسهلا بيك في منتديات المحيط العربي طريقك لاحتراف علوم

شرح الكود:

dim-تستخدم لتعريف المتغيرات .

name-اسم المتغير ونقوم بتحديد القيمة المخزنة بداخله.

-في أمر الطباعة تلاحظ أننا لم نضع المتغير بين علامتي التنصيص " " وذلك لأننا نريد طباعة قيمة هذا المتغير وليس طباعة المتغير نفسه.

-عندما نريد أن نقوم بطباعة قيمة المتغير ونريد أن ندرج عناصر HTML معه يجب أن تفصل بينهما بإشارة & كما هو موضح في الأعلى.

ملاحظة: عندما نقوم بطباعة أكثر من متغير أو طباعة جملة ومتغير في نفس الأمر يجب أن نفصل بينهما بإشارة & ، وليس من الشرط أن نضع dim لتعريف المتغيرات.

مثال: كيفية إنشاء مصفوفة عندما تكون البيانات مخزنة داخل متغير؟

كود

```
<html dir="rtl">
<head>
</head>
<body>
<%dim fname(5),i
fname(0)="وسهلا أهلا"
fname(1)="بك"
fname(2)="في موقع"
fname(3)="العربي منتدى المحيط"
fname(4)="طريقك لاحتراف علوم البرمجة"
for i=0 to 4
response.write("<center><b>"&fname(i)&"</b><center>")
next
%>
</body>
</html>
```

فسوف يكون النتاج عن ذلك التالي:

أهلا وسهلا
بك
في موقع
منتدى المحيط العربي
لاحتراف علوم البرمجة طريقك

شرح الكود:

fname(5) - متغير من نوع مصفوفة وهي تحتوي على خمس مصفوفات.

fname(0) - متغير من نوع مصفوفة، وهو الصف الأول من المصفوفة التي قمنا بتعريفها.

i - متغير.

for: سوف نستخدم loop وقيمة المتغير i وقيمه تبدأ من صفر إلى 4.

next: تعني دوران، أي سوف تعيد الحلقة مرة أخرى.

-سوف يخرج عن loop عندما القيمة التالية لا تساويها.i

مثال: كيفية إظهار الوقت والتاريخ عن طريق استخدام لغة **VBscript** ؟

```
<html dir="rtl">
<head>
</head>
<body>

<%
h=hour(now())
response.write("<center><b>")
response.write(now())&"<br/>")
```

```
if h >12 then
response.write("<br>صباح الورد")
else
response.write("<br>")
response.write("بيك في منتديات المحيط العربي طريقك لاحتراف علوم أهلا وسهلا  
")
end if
response.write("</b></center>")
%>
</body>
</html>
```

سوف يكون ناتج الكود كالتالي:

كود

5/8/2004 2:38:33 PM

طريقك لاحتراف علوم البرمجة أهلا وسهلا بيك في منتديات المحيط العربي

شرح الكود:

now(): دالة من دوال الفي بي سكريبت ، تدل على التاريخ والوقت الحالي.

hour(now()): هنا سوف يتم اخذ قيمة الساعة الحالية من دالة التاريخ والوقت

h - متغير قيمته تساوي قيمة الساعة من التاريخ.

if h>12 then - جملة **if** الشرطية أي إذا كانت **h** قيمتها أقل من ١٢ اطبع الجملة التالية وهي

(صباح الورد).

Else - إذا لم يتحقق الشرط داخل جملة ال **if** قم بتنفيذ الجملة التاليو (أهلا وسهلاً بك من جديد في

منتدى الفريق العربي للبرمجة.)

end if: نهاية جملة **if** الشرطية.

مثال: كيفية إظهار الوقت والتاريخ عن طريق استخدام لغة Javascript ؟

كود

```
<%@ language="javascript" %>
<html dir="rtl">
<head>
</head>
<body>
<%
var d=new Date()
var h=d.getHours()

Response.Write("<p>")
Response.Write("<center><b>"+d+"</b></center>" )
Response.Write("</p>")
if (h>12)
{
Response.Write("الخير صباح")
}
else
{
Response.Write("وسهلاً أهلاً")
}
%>
</body>
</html>
```

المثال واضح ولا يختلف عن المثال أعلاه، ولكن الاختلاف هو عندما نريد أن نطبع متغيرات من لغة الجافا سكربت وفي نفس جملة الطباعة يجب أن نضع الإشارة + وليس الإشارة & ، وتلاحظ أيضاً أن الأمر Response يبدأ بحرف كبير.

-مثال: كيفية استخدام procedure في ASP بواسطة VBscript ؟

كود

```
<html dir="rtl">

<head>
<%
sub VBproc(a,b)
response.write(a*b)
end sub
%>
</head>
<body>
<%response.write("<center><B>")%>
result: <%call VBproc(5,4)%>
result: <% vbproc 5,4%><br>
<% response.write("</b></center>")%>
</body>
</html>
```

سوف يكون الناتج كالتالي:

كود

result: 20

شرح الكود:

sub: تعريف `procedur` واسمها `VBproc` تحتوي على متغيرين كما موضح في الأعلى.

`(response.write (a*b):` -سوف يطبع نتيجة عملية الضرب.

`end sub:` -تعني الإنتهاء.

`<% (call VBproc (5*4 %)>:` -تقوم باستدعاء `procedure` عن طريق `call` وتضع قيمة

للمتغيرين.

`<% vbproc 5,4 %>:` -طريقة أخرى لـ استدعاء ال `procedure` هكذا أيضاً.

مثال: كيفية استخدام دالة ال `function` مع ال `ASP` بواسطة الجافا سكريبت؟

كود

```
<%@ language="javascript" %>
<html dir="rtl">

<head>
<%
function jsfunction(a,b)
{
Response.write(a*b)
}
%>

</head>
<body>
<%Response.write("<center><B>")%>
result:<% jsfunction(5,4)%>
<% Response.write("</b></center>")%>
</body>
</html>
```

نتج المثال كما في المثال أعلاه.

شرح الكود - نفس شرح الكود السابق ولكن باستخدام **Function** بدل من **sub**.
-ملاحظة: عند استدعاء **Function** لا تكتب **call**.

مثال: كيفية استخدام لغة الجافا سكريبت والفي بي سكريبت معاً؟

كود

```
<html dir="rtl">

<head>
<%
sub VBproc(a,b)
response.write(a*b)
end sub
%>
<script language="javascript" runat="server" %>

function JSfunction(a,b)
{
response.write(a*b)
}

</script>
</head>
<body>
<%response.write("<center><B>")%>
result: <%call vbproc(5,4)%><br>
```

```
result:<% call JSfunction(5,4)%>
<% response.write("</b></center>")%>
</body>
</html>
```

النتائج:

كود

```
result VBscript: 20
result Javascript:20
```

كود

شرح الكود:

- عملها نفس عمل المثالين السابقين، ولكن الاختلاف بذلك عندما نقوم باستخدام لغة الفي بي سكريبت و لغة الجافا سكريبت عند كتابة نصوص الجافا سكريبت، يجب أن تضع أعلام نصوص الجافا سكريبت اللغة المستخدمة وتضع تعريف اخر للجافا سكريبت عندها، وهو `<%language="javascript" runat="server" %>`، وعند استدعاء كل من `function` و `procedure` يجب أن تضع عند استدعاء `function` كلمة `call` وتستطيع كتابة أمر `response` سواء كان الحرف الأول كبير أو صغير.

في الدرس القادم سوف أقوم بشرح لكم خصائص كائن ال `Response`

هناك خصائص لكائن `Response Object` وهي كالتالي:-

`Buffer`

`CacheControl`

`Charset`

`ContentType`

`Expires`

`ExpiresAbsolute`

IsClientConnected

والآن سوف نقوم بشرح كل واحد منها على حده.

الخاصية

Response.Buffer

الخاصية - Buffer: تحدد إذا كان الناتج خزن داخل **buffer** أو لا، في حالة عندما يخزن الناتج داخل ال **buffer** ، فإن الخادم سوف يحمل الرد إلى المتصفح حتى يتم معالجة كل سكربتات الخادم، أو يتم استدعائها بواسطة المنهجين **flush** أو **end**.
ملاحظة: هذه الخاصية يجب أن تضع قبل الوسم **<HTML>**

الصيغة لها:

كود

```
response.buffer=(flag)
```

flag:

1-قيمة Boolean التي تحدد إلى **buffer** بيان يرسل النتائج أو لا.

2-إذا كانت خطأ لا يوجد **buffer**.

3-إذا كانت **buffer** صحيحة، فإنه السيرفر لن يرسل النتائج حتى يتم معالجة السكريبات على

الصفحة، أو حتى يتم استدعاء المنهجين **flush** أو **end**.

المثال التالي، لن يرسل النتائج إلى المتصفح قبل أن ينتهي من **loop** ، إذا كانت ال **buffer** خطأ

فإنها سوف تقوم بكتابة سطر كل وقت فإنها سوف تذهب من خلال **loop**

كود

```
<%response.Buffer=true%>
<html dir="rtl">
<body>
<%
for i=1 to 10
response.write(i & "<br />")
next
%>
</body>
</html>
```

ونائج الكود سوف يكون التالي:

كود

1
2
3
4
5
6
7
8
9
10

Cachecontrol الخاصية

الخاصية - **Cachecontrol**: تتحكم فيما إذا كانت السيرفرات الوكيله يجب أن تنقل الصفحة إلى الذاكرة.

الصيغة:

كود

```
response.cachecontrol=(control_header)
```

المتغير **control_header**: تستطيع أن تضع مكانها "public" أو "private".
Public: تخزن داخل السيرفر الوكيل عندما تكون الصفحات لا تحتاج أن يتم تغييرها.
private: لا تخزن داخل سيرفر الوكيل، لأن صفحاتها تتغير من من طلب إلى طلب آخر.

مثال:

كود

```
<% response.cachecontrol="public"%>
```

أو

```
<% response.cachecontrol="private"%>
```

الخاصية **charset**

الخاصية **charset**:

-تتحكم بسلسلة من الأحرف التي يستخدمها برنامج الاستعراض للمتصفح
المجموعة الافتراضية هي ISO=LATIN-1.

على سبيل المثال لتغيير سلسلة الأحرف إلى ISO=LATIN-7 ، استخدم التالي:

كود

```
<% response.charset="ISO=LATIN-7"%>
```

أو ممكن أن استخدام كود ال HTML بدل من ذلك وهو أن تضعه داخل محتوى صفحات ال
HTML، وهو كالتالي:

كود

```
<meta http-equiv="Content-Type" content="text/html;  
charset=ISO=LATIN-7">
```

الخاصية Contenttype

الخاصية Contenttype:- تقوم بتحديد محتوى الصفحة، هل تحتوي على صفحات html أو
تحتوي على صور وغيره، وهي تكتب أعلى نصوص ال.html

الصيغة:

كود

```
<%response.contentType=conotenttype%>
<html>
<body>
</body>
</html>
```

وسوف أقوم بكتابة لكم الآن بعض القيم التي تكتب داخل: **contentType**

كود

```
<%response.contentType="text/html"%>
تحتوي على الصفحة نصوص
html
<response.contentType="image/jpeg"%>
على صور من نوع تحتوي الصفحة
jpeg
<%response.contentType="image/gif"%>
من نوع تحتوي الصفحة على صور
gif
<%response.contentType="application/vnd.ms-excel"%>
الصفحة على تحتوي
Excel
```

وممكن أن تحدد ذلك عن طريق نصوص ال HTML ، على النحو التالي:-

كود

```
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1256">
```

الخاصية Expires

الخاصية - Expires: تعد الفترة الزمنية، بالدقائق، التي تحدد زمن إنتهاء فعالية الصفحة .

الصيغة:

كود

```
response.expires=number
```

Number: -الفترة الزمنية بالوقت التي تحدد وقت إنتهاء فعالية الصفحة.

مثال : المثال التالي سوف يحدد زمن إنتهاء فعالية الصفحة خلال ١٠ دقائق:

كود

```
<% response.expires=10%>
<html>
<body>
</body>
</html>
```

الخاصية ExpireAbsolute

الخاصية:ExpireAbsolute

تعد بالفتره الزمنيه بالوقت والتاريخ ، لإنهاء فعالية الصفحة بدقة. قبل الوقت المحدد.

الصيغة:

كود

```
<% response.ExpiresAsolute=date&time%>
```

date:تقوم بتحديد التاريخ، لتاريخ إنتهاء فعالية الصفحة.

Time:تقوم بتحديد الوقت لإنهاء فعالية الصفحة.

مثال على ذلك:

كود

```
<% response.ExpiresAbsolute=#October 11,2003 10:00:00# %>
```

```
<html>
```

```
<body>
```

```
</body>
```

```
</html>
```

ممكن أن نقوم بتحديد فعالية إنتهاء الصفحة أيضاً في نصوص الHTML ، على سبيل المثال:

كود

```
<a href="arabmoheet.net?uniqueurl=09/09/2003 2:00:00 AM"> المحيط  
</a> العربي
```

الخاصيةIsClientConnected

الخاصية: IsClientConnected

-تحدد إذا كان المستخدم متصل أو غير متصل على السيرفر.

الصيغة:

كود

```
response.isclientconnected
```

مثال على ذلك:

كود

```
<html dir="rtl">
<body>
<%
If response.IsClientConnected=true then
response.write("<center><b>أنت متصل الآن مع منتديات المحيط العربي
</b></center>")
else
response.write("الان أنت غير متصل معنا")
end if
%>
</body>
</html>
```

وهكذا نكون قد انتهينا من خصائص الكائن Response.

والدرس القادم سوف يكون شرح لمناهج الكائن Response.

الدرس الخامس

مناهج الكائن Response

هناك كائن Response Object وهي كالتالي:-

AddHeader

Clear

End

Flush

Redirect

والآن سوف أقوم بشرح كل واحدة منها على حدة

المنهج - Addheader: تضيف عنوان جديد ل. HTTP

ملاحظة: نادراً ما تضاف إلى العنوان، ولا تستطيع حذفها.

مثال على ذلك، في المثال التالي سوف تقوم بنقلك خلال 5 ثواني إلى الصفحة التي قمت أنت

بتحديده:

كود

```
<% response.addheader "refresh" ,"5;url=asprresponse.asp" %>
```

ويمكن ان نستخدم بدلاً من ذلك نصوص الHTML ، كالتالي:

كود

```
<" meta http-equiv="refresh" content="5;url=asprresponse.asp >
```

المنهج - Clear: تسمح جميع البيانات المخزنة داخل ال Buffer.

ملاحظة: هذه الخاصية تقوم بمسح فقط كل ما قبلها، وما بعده يتم طباعته.

الصيغة:

كود

```
response.clear
```

مثال على ذلك، المثال سوف يقوم بمسح جميع البيانات التي المخزنة داخل ال **buffer** فقط.

كود

```
<%  
response.Buffer=true  
%>  
<html>  
<body>  
وسهلاً بك في منتديات المحيط العربي أهلاً  
  
<%  
response.Clear  
%>  
الخير صباح  
</body>  
</html>
```

فسوف يكون الناتج فقط:

كود

```
صباح الخير
```

المنهج - End: توقف عملية المعالجة للسكريبات المخزنة داخل ال **buffer** ، وتظهر الناتج الحالي. حيث أنها تعرض السكريبات التي تم معالجتها فقط.

الصيغة:

كود

```
response.end
```

مثال على ذلك:

كود

```
<% response.buffer=true%>
<html>
<body>
<p>وسهلا بك في منتديات المحيط العربي، نتمنى لك أن تقضي أوقات ممتعة أهلا</p>
<%
Response.End
%>
في منتديات المحيط العربي مرحبا بك
</body>
</html>
```

سوف يكون الناتج:

كود

ممتعة أهلا وسهلا بك في منتديات المحيط العربي، نتمنى لك أن تقضي أوقات

المنهج - Flush: يرسل المحتوى المخزن داخل ال **buffer** إلى برنامج الاستعراض.

الصيغة:

كود

```
response.flush
```

مثال على ذلك:

كود

```
<%  
Response.Buffer=true  
%>  
<html dir="rtl">  
<body>  
<p align="center"><b>منتديات المحيط العربي مرحبا بك</b>  
</p>  
<%  
Response.Flush  
%>  
<p align="center"><b>مرحبا بك منتديات المحيط العربي</b></p>  
  
</body>  
</html>
```

النتائج:

كود

مرحبا بك منتديات المحيط العربي

العربي مرحبا بك منتديات المحيط

المنهج: redirect

-ترجع المستخدم مباشرة إلى رابط آخر.

الصيغة:

كود

`response.redirect URL`

URL: الرابط التي سوف تنتقل إليه بمجرد فتحك للصفحة.

مثال على ذلك،

كود

```
<%  
response.redirect "http://www.arabmoheet.net"  
%>
```

وبذلك نكون قد انتهينا من شرح خصائص ومناهج الكائن **Response Object**

الدرس السادس

الكائن

Request Object

الكائن: Request Object

-يستخدم للحصول على بيانات المستخدم.

User Input (مدخلات المستخدم)

عن طريق **Request Object** تستطيع إسترجاع بيانات المستخدم من الفورم.

-هناك طريقتين لإسترجاع بيانات المستخدم:

أولاً الأمر: Request.QueryString

الأمر **Request.QueryString** تستخدم لاسترجاع بيانات المستخدم التي قام بإدخالها من الفورم، وهي تكون مرئية للمستخدم (تظهر البيانات المرسلّة على عنوان المتصفح) وهي تستخدم المنهج **get** لإرسال البيانات.

مثال: كيفية إسترجاع بيانات المستخدم باستخدام المنهج: get

- أنشأ الصفحة التالية وقم بتسمية هذه الصفحة باسم: simpleform.asp.

كود

```
<html>
<body>
<%
Request.querystring="fname"
Request.querystring="lname"
%>
<form method="get" action="simpleform.asp">
الإسم الأول: <input type="text" name="fname">
</br >
اسم العائلة: <input type="text" name="lname">
</br ></br >
<input type="submit" value="إرسال البيانات">
</form>

</body>
</html>
```

شرح الكود:

أولاً- أنشأنا فورم تحتوي على التالي:

اسم الأول: وهو من نوع Text ، واسمه fname.

2- اسم العائلة: وهو من نوع Text ، واسمه lname.

3- زر الإرسال: وهو من نوع submit ، وقيمه إرسال البيانات.

4- من خصائص الفورم قيمة بتحديد المنهج الذي سوف نستخدمه في استرجاع البيانات وهو المنهج

Get ، وفي Action نحدد الصفحة التي سوف تقوم بإسترجاع البيانات المرسله إلى السيرفر.

ثانياً - استخدمنا الأمر Request.QueryString لإسترجاع بيانات الذي قام المستخدم في إدخاله.

Request.querystring "fname": -سوف تقوم باسترجاع البيانات بالنسبة لاسم الأول، حيث أن fname ترجع إلى اسم الحقل الذي قمة بتعريف الحقل اسم الأول.

Request.querystring "lname": -سوف بتقوم باسترجاع البيانات بالنسبة لإسم العائلة، حيث أن lname ترجع إلى اسم الحقل الذي قمنا بتعريف حقل اسم العائلة.

-سوف تلاحظ في نتيجة هذا الكود كيف سوف يتم الحصول على البيانات المستخدم التي قام بإدخاله عن طريق الفورم، بعد أن تقوم بإدخال البيانات سوف تلاحظ في عنوان المتصفح كيف يتم إرسال القيم إلى السيرفر، فرضاً إذا كان قيمة fname المدخلة تساوي **abdullah** وقيمة اسم العائلة تساوي **shaqaliah** فالبيانات التي تم إرسالها إلى السيرفر سوف تكون على هذا الشكل:

كود

```
http://www.arabmoheet.net/simpleform.asp?fname=abdullah&lname=shaqaliah
```

ملاحظة: تلاحظ القيم المرسله في نفس الصفحة التي قمنا بإنشاء الفورم عليها.

مثال: كيفية طباعة القيم المدخلة في الفورم?

-أنشأ الصفحة التالية وقم بتسمية هذه الصفحة باسم: **simpleform2.asp**.

كود

```
<html dir="rtl">
<body>
<%
Request.querystring="fname"
Request.querystring="lname"
%>
<form method="get" action="simpleform.asp">
```

```
<input type="text" name="fname">  
</br >  
اسم العائلة: <input type="text" name="lname">  
</br ></br >  
<input type="submit" value="إرسال البيانات">  
</form>  
<%  
response.write(request.querystring("fname"))  
response.write(request.querystring("lname"))  
%>  
</body>  
</html>
```

- باستخدام الأمر **Response.write** نقوم بطباعة قيم المدخلات. حيث نضع أمر **Request.Object** مع اسم الفورم. كما هو موضح في المثال أعلاه.

مثال : كيفية إرسال القيم وطباعتها في صفحة أخرى؟

أنشأ صفتين وليكن اسمهما **Form.asp**، **Form2.asp**

- الصفحة الأولى سوف تحتوي فقط على الفورم واسم الصفحة التي سوف نقوم بإرسال البيانات

إليها، قم بتسمية هذه الصفحة باسم: **form.asp**

كود

```
<html dir="rtl">  
<body>  
<form method="get" action="form2.asp">
```

```
<input type="text" name="fname">
</br >
اسم العائلة: <input type="text" name="lname">
</br ></br >
<input type="submit" value="إرسال البيانات">
</form>
</body>
</html>
```

2- الصفحة الثانية سوف تقوم باستقبال البيانات المرسله وطباعتها. وقم بتسميتها باسم

form2.asp.

كود

```
<html dir="rtl">
<body>
<%
dim fname,lname
fname=request("fname")
lname=request("lname")
%>
<b>اسم الأول: <%= "<B>"&fname"</b>"%></br>
العائلة اسم: <%= "<B>"&lname &"</B>"%>
</body>
</html>
```

شرح كود صفحة:form2.asp

fname:متغير يحتوي على القيمة المخزنة في ("fname Request") المدخلة من حقل اسم

form.asp. الأول في صفحة

lname:متغير يحتوي على القيمة المخزنة في ("lname Request") المدخلة من حقل اسم

form.asp. الأول في صفحة

:<%=>"&fname&""<%=> تساوي أمر الطباعة response.write وسوف تقوم

بطباعة القيمة المخزنة داخل المتغير.fname

ملاحظة: تلاحظ أنني لم أستخدم الأمر Request.querystring ، وقمت بوضع Request

فقط ، فحالا الاثنتين يساوي نفس الشيء لأن السيرفر سوف يتعرف عليها عن طريق وضع المنهج

Get.

مثال :كيفية إسترجاع قيم لها نفس المتغير ونقوم بطباعتها؟

1-الصفحة الأول، واسمها get.asp ، وتحتوي فقط على الفورم، والصفحة المرسله إليها النتائج :

كود

```
<html dir="rtl">
<body>
<form method="GET" action="names.asp">
<input type="text" name="n" size="9">الإسم الأول

<input type="text" name="n" size="9">اسم العائلة
<input type="submit" value="Submit" >
</form>
</body>
</html>
```

2-الصفحة الثانية واسمها names.asp هي التي سوف تستقبل القيمتين وتقوم بطابعتهما، في هذه الحالة سوف نستخدم loop ، لكي نقوم باسترجاع قيمتي المتغير n ونقزم بطباعة القيم :

كود

```
<html dir="rtl">
<body>
<% for i=1 to request.querystring("n").count
response.write(request.querystring("n")(i)&"</br>")
next
%>
</body>
</html>
```

شرح الكود:

1- في صفحة get.asp نلاحظ ترسل إلى صفحة names.asp ، قيمتين ولهما نفس المتغير.

2- في صفحة names.asp عندما يكون للقيم المرسله قيمتين ونريد استرجاع القيمتين لنفس

المتغير ، ونقوم بطابعتهما يجب أن نستخدم في هذه الحالة loop.

for i= 1 to request.querystring("n").count: استخدام هنا ال loop ، و i

متغير قيمته تبدأ من واحد إلى عدد القيم المرجعة من request ، و count تحسب كم قيمة توجد

داخل request ، والقيم المرجعة لدينا هي اثنتين، أي i سوف تساوي من واحد إلى اثنين.

request.querystring("n")(i: عند request.querystring("n") تحتوي على

القيم المرجعة ، لكن أي قيمه سوف تبدأ في طباعتها هل هي القيمة abd أو القيمة maher ، ولكن

عن طريق (i) عندما تعود قيمتها إلى واحد سوف تقوم عندها بطباعة القيمة الأولى المرسله،

وعندما تصبح قيمتها ٢ سوف تقوم بطباعة القيمة الثانية المرسله لنفس المتغير.

5- next: أي التالي، سوف تعمل حلقة دوران وسوف تقوم بطباعة جميع القيم إلى أن تنهي قيمة

i، فعندها سوف تخرج من loop ، وتقوم بعرض جميع النتائج.

مثال: إرسال البيانات عندما يضغط المستخدم على رابط تشعبي؟

في هذا المثال يوضح كيفية إرسال بعض البيانات إلى الصفحة عن طريق الرابط، وإظهار تلك البيانات على صفحة الهدف (وفي هذا المثال صفحة الهدف سوف تكون على نفس الصفحة).

كود

```
<html dir="rtl">
</head>
<body>
<p align="center"><a href="simple_querystring.asp?color=green"
>Example </a></p>
<%
name=request("color")
response.write("<font color='green'>"&name)%>
</body>
</html>
```

شرح الكود:

-في هذا المثال، سوف نقوم بعمل رابط يحتوي على المتغير **color** وقيمهته **green** ، حيث عندما يضغط المستخدم على هذا الرابط، سوف نقوم باستقبال هذه القيمة وبطباعتها على الصفحة الهدف.

استخدام جملة **if** الشرطية:

في هذا المثال سوف أقوم بالفحص إذا كان المستخدم قد قام بادخال جميع البيانات المطلوبة أو لا: قم بإنشاء صفتين، الصفحة الأولى وليكن اسمها **proprieteis.asp** ، والثانية وليكن اسمها **aspaccept.asp**.

1-صفحة `properteis.asp` تحتوي فقط على الفورم والصفحة التي سوف ترسل إليها بيانات الفورم، وهي كالتالي :-

كود

```
<html dir="rtl">
<body>
<form method="GET" action="aspaccept.asp">
اسم المستخدم:<input type="text" name="name" size="6">
البريد الإلكتروني:<input type="text" name="email" size="6">
الشخصي الموقع:<input type="text" name="website" size="6">
التوضيح:<textarea rows="2" name="dis" cols="12"></textarea>
</body>
</html>
```

2-صفحة `aspaccept.asp` تحتوي على الأوامر التالية:

كود

```
<html>
<head>
<%
name=request("name")
email=request("email")
website=request("website")
dis=request("dis")
%>
</head>

<body>
```

```

<%response.write("<center><B>")%>
<%
if name<>"" then
response.write(name"<BR>")
else
response.write("تدخل اسم المستخدم أنت لم")
end if
if email<>"" then
response.write(email)
else
response.write("<br>الإيميل أنت لم تقم بادخال")
end if
response.write("</br>"&website"<BR>")
response.write(dis)
%>
<%response.write("</center><B>")%>
</body>
</html>

```

شرح الكود:

if name<>"" then: جملة if سوف تفحص إذا كانت قيمة المتغير name لا تساوي فارغه، سوف تقوم بتنفيذ الشرط الذي يليها.

else: إذا كان الشرط في جملة if غير صحيح، أي إذا كانت قيمة المتغير name فارغة، سوف تقوم بتنفيذ الشرط الذي يليها.

end if: تعني الإنتهاء من جملة if الشرطية.

ثانيا الأمر Request.Form: وهي لا تختلف كثيراً عن أمر **Request.querystring**، فهي تستخدم لجمع القيم من الفورم مع استخدام المنهج **Post**، وهي لا تكون مرئية للمستخدم، ولا تقوم بتحديد كمية البيانات المرسله على عكس **Request.querystring**.

مثال: أنشأ الصفحة التالية وقم بحفظها باسم **formpost.asp**

```
<html dir="rtl">
<body>
<%
request.form("username")
request.form("pass")
%>
<form method="post" action="formpost.asp">
اسم المستخدم: <input type="text" name="fname">
</br >
كلمة المرور: <input type="text" name="lname">
</br ></br >
<input type="submit" value="إرسال البيانات">
</form>
<%= "&request.form("username")%"> اسم المستخدم"></br>
<%= "&request.form("pass")%"> كلمة المرور">
</body>
</html>
```

شرح الكود:

أولاً- أنشأت فورم تحتوي على التالي:

1- اسم المستخدم: وهو من نوع Text ، واسمه.username

2- كلمة المرور: وهو من نوع Text ، واسمها.pass

3- زر الإرسال: وهو من نوع.submit

4- من خصائص الفورم قيمة بتحديد المنهج الذي سوف نستخدمه في استرجاع البيانات وهو المنهج post ، وفي Action حددنا الصفحة التي سوف يقوم بإسترجاع البيانات فيها وهي إرسال البيانات على نفس الصفحة.

ثانياً - استخدام الأمر Request.form لإسترجاع بيانات الذي قام المستخدم في إدخاله:

Request.form "username:" -سوف تقوم باسترجاع البيانات بالنسبة لاسم المستخدم، حيث أن **username** ترجع إلى اسم الحقل الذي قمنا بتعريف الحقل اسم المستخدم.

Request.form "pass:" -سوف تقوم باسترجاع البيانات بالنسبة لكلمة المرور، حيث أن **pass** ترجع إلى اسم الحقل الذي قمنا بتعريف حقل كلمة المرور.

-سوف تلاحظ في نتيجة هذا الكود كيف سوف يتم الحصول على البيانات المستخدم التي قام بإدخاله عن طريق الفورم، بعد أن تقوم بإدخال البيانات سوف تلاحظ في عنوان المتصفح ، أنه لا يوجد بيانات مرسله. فقط سوف ترى هذا الشكل لعنوان المتصفح:

كود

```
localhost/formpost.asp
```

ملاحظة: الأمر **request.form** لا يفرق عن الأمر **request.querystring** إله بطريقة إرسال البيانات فقط، ممكن أن تكتب الأمر **request.form** لتخزين البيانات المرسله من الفورم بهذا الشكل.
("request("username.

مثال: كيفية استخدام الفورم مع radio button ؟

هذا المثال يوضح كيفية التعامل مع المستخدم من خلال **radio button** ، باستخدام **Request.form**، النموذج مستخدم بطريقة المنهج: **post:**

كود

```
<html dir="rtl">
<body>
<%
dim color
color=request.form("color")
%>
```

المفضل الرجاء قم باختيار اللون

لك:

```
<form method="POST" action="formradio.asp">
<input type="radio" value=""<font color='red'>الأحمر</font>"
name="color" <%if color="الأحمر" then response.write("checked")
%>></font> الأحمر.
<input type="radio" value=""<font color='red'>الأزرق</font>"
name="color" <%if color="الأزرق" then response.write("checked")
%>></font>
لأزرق.
<input type="radio" value=""<font color='red'>الأخضر</font>"
name="color" <%if color="الأخضر" then response.write("checked")
%>></font>
لأخضر.
<input type="submit" value="إرسال"></font><font size="2"
</form>
<%
if color<>"" then
response.write("<center><b>هو اللون المفضل لك '&color)
response.write("</b></center>")
end if
%>
</body>
</html>
```

شرح الكود:

- 1- أنشأت فورم من نوع **radio button** ، تحتوي على ثلاث اختيارات، واسم كل منهم **color** ، وقيمة الأول احمر، والثاني، أزرق، والثالث أخضر.
- 2- **color**: سوف تخزن بداخله القيمة المسترجعه من الفورم.

الأحمر، نفذ الشرط، checked سوف تقوم فقط باسترجاع القيمة التي تم اختيارها. 3-الأحمر = if color: إذا كانت قيمة color تساوي
if color<>"" then: إذا كانت color لا تساوي قيمة فارغة نفذ التالي.
5- (response.write(color: سوف تقوم بطباعة قيمة color الذي قم تم اختياره.

مثال: كيفية استخدام الفورم مع checkbox ؟

هذا المثال يوضح كيفية التعامل مع المستخدم باستخدام "checkbox" باستخدام
Request.form، النموذج مستخدم بطريقة: post

كود

```
<html dir="rtl">
<body>
<%
fruits=request("fruits")%>
لديك ما هي الفوكه المفضله
<form method="POST" action="formcheckbox.asp">

<input type="checkbox" name="fruits" value="<font color='red'>التفاح</font>"
<%if fruits="التفاح"then response.write("checked") %>> size="2" التفاح.

<input type="checkbox" name="fruits" value="<font
color='orange'>المانجا</font>"<%if fruits="المانجا"then
response.write("checked")%>>المانجا.
<input type="checkbox" name="fruits" value="<font
color='orange'>البرتقال</font>"<%if fruits="البرتقال"then
response.write("checked")%>>البرتقال.
<input type="checkbox" name="fruits" value="<font
color='red'>الخوخ</font>"<%if fruits="الخوخ"then
response.write("checked")%>> الخوخ.
<input type="checkbox" name="fruits" value="<font
color='yellow'>الموز</font>"<%if fruits="الموز"then
response.write("checked")%>> الموز.

<input type="checkbox" name="fruits" value="<font
```

```
color='green'>الكمثره</font>"<%if fruits="الكمثره"then  
response.write("checked")%>><font color="#008000" size="2">الكمثره.  
<input type="submit" value="إرسال"></font></b></p>  
</form>  
<%  
if fruits <>"" then  
response.write("<center><B>لديك الفواكه المفضلة  
هي:&fruits&"</b></center>")  
end if  
%>  
</body>  
</html>
```

نفس شرح المثال السابق، ولا يوجد اختلاف بينهما، غير أنها تظهر أكثر من قيمة.
وإن شاء الله في الدرس الخامس سوف أقوم بشرح خصائص الكائن Request

الدرس السابع

خصائص الكائن Request

1 الكوكيز: (Cookie)

ما هي cookies (الكعكات)؟

الكعكات غالباً تستخدم لتعريف المستخدم، والكعكات ملف صغير يكون موجود على السيرفر على كمبيوترك الخاص، وكل مره نفس الكمبيوتر عندما يطلب الصفحة مع المتصفح، فسيقوم الكمبيوتر بإرسال الكعكات أيضاً.

كيف يتم إنشاء الكعكات؟

الأمر `Response.cookies` هو الذي يستخدم في إنشاء الكعكات.

ملاحظة: هذا الأمر يكتب قبل الوسم `<HTML>`.

في المثال التالي، سوف أنشاء `cookie` واسمها `"firstname"` وقيمتها: `"abd"`

كود

```
<%  
response.cookie("firstname")="abd"  
%>
```

وتستطيع أن تحدد صلاحية إنتهاء هذا: `cookie:Expires=#may 10,2003#`

كود

```
<%  
response.cookies("firstname")="abd"  
response.cookies("firstname").Expires=#may 10,2003#  
%>  
<html>  
<body>  
</body>  
</html
```

>

سوف تلاحظ أنه لا يوجد شيء، وذلك لأننا قمنا فقط بإنشاء **cookie** ولم نقم باسترجاع قيمته، ولكن سوف تجد في ملف **cookies** على جهازك الخاص بك ملف صغير يحتوي على اسم **cookie** وقيمته وهذا الملف تجده على المسار التالي إذا كان الوندوز موجود على قرص الصلب C.

C📄ocuments and SettingsabdullahCookies

حيث أن **abdullah** هو اسم الجهاز الخاص بي أنا، أما أنت فضع اسم الجهاز الخاص بك الذي قمت بتعريفه عندما قمت بإزالة الوندوز، وإن لم يكن الوندوز على القرص C فقط ضع بدل C اسم القرص الصلب الذي يوجد عليه الوندوز.

كيف يتم استرجاع القيم من **cookie?**

الأمر **request.cookies** يستخدم لاسترجاع قيم **cookie** ، والمثال التالي يوضح ذلك:

كود

```
<html>
<body>
<%
request.cookies("firstname")
response.write("<center><B>"&request.cookies("firstname")&"</b></center>"
)
%>
```

</body>

</html>

طبعاً لن يكون هناك نتيجة، وذلك لأنه قمنا بتحديد تاريخ إنتهاء فعالية الكوكي، بتاريخ فانت.

الكعكات مع المفاتيح ?

-إذا كانت الكعكات تحتوي على عدد من القيم، فعندها نقول بأن الكعكات تمتلك مفاتيح.

في المثال التالي، سوف أقوم بإنشاء مجموعة من الكعكات واسمها "user" ، و "cookie user" يمتلك مفاتيح تحتوي على بيانات "user".

كود

```
<%  
response.cookies("user") ("firstname")="Abdullah"  
response.cookies("user") ("firstname")="Jaber"  
response.cookies("user") ("firstname")="mahmoud"  
response.cookies("user") ("firstname")="Shaqlaih"  
%>  
<html>  
<body>  
</body>  
</html>
```

في هذا المثال سوف يرسل السيرفر لديك كل هذه الكعكات إلى المستخدم.

قم بحفظ هذا الكود باسم `aspkeys.asp` ، وقم بقراءته على متصفحك حتى يتم إنشاء `cookie`.

لقد تم إنشاء الكعكات الآن، وفي المثال التالي سوف نقوم بقراءة القيم المخزنة داخل الكعكات:

كود

```
<html dir="rtl">
<body>
<%
dim x,y
for each x in request.cookies
response.write("<p>")
if request.cookies(x).haskeys then
for each y in request.cookies(x)
response.write("<center><b>")
response.write(x&":"&y&"]="&request.cookies(x)(y))
response.write("</br>")
response.write("</b></center>")
next
else response.write(x&"]="&request.cookies(X)&"]="</br>")
end if
response.write"</p>"
next
%>
</body>
</html>
```

والنتيجة كالتالي:

user:firstname=Abdullah

user:lastname=shaqlaih

user:contry=palestine

user:age=22

ملاحظة: عند كتابة هذا الكود سوف يقوم بطباعة جميع **cookie** الموجوده على جهازك الخاص بك سواء كانت يوجد لها مفاتيح أو لا يوجد لها، أي لو كنت انت مسجل في اي موقع، وكان يتبع نظام الكوكيز، سوف تظهر لك جميع ملفات الكوكيز التي لديك.

مثال كيف يتم تخزين اسم المستخدم، وكلمة المرور داخل الكوكيز، عند إدخالها من الفروم؟
الصفحة التالية تحتوي على الفورم، لإدخال اسم المستخدم وكلمة المرور:

كود

```
<form method="POST">  
<p>اسم المستخدم: <input type="text" name="username"  
size="20"><br>  
كلمة المرور: <input type="password" name="password"  
size="20"></p>  
<p><input type="submit" value="Submit" name="B1"></p>  
</form>
```

هنا سوف سوف يتم تخزينهم داخل الكوكيز:

كود

```
response.cookies("username")=request("username")
response.cookies("password")=request("password")
response.cookies("username").expires=date+1
response.cookies("password").expires=date+1
```

اعتقد المثال واضح ولا يحتاج للشرح، لكن عند تحديد صلاحية الإنتهاء **date+1** تعني بأنه فعال من تاريخ اليوم + يوم واحد.

مثال: كيف يتم قراءة كوكيز محدد؟

كود

```
request.cookies("username")
```



2- ServerVariables:

-تستخدم لاسترجاع قيم متغيرات السيرفر.

الصيغة:

كود

```
request.servervariables (server_variable)
```

server_variable: اسم متغير السيرفر الذي نريد استرجاع قيمته.

مثال: في بعض الأحيان نتحتاج إلى أن نعرف بعض المعلومات عن زوار الموقع أو عن السيرفر المستضيف للموقع، وهذا يتم عن طريق **ServerVariables** ، وتكون النتيجة على هيئة نص:

كود

```
<html dir="rtl">
<body>
<p>رقم IP (IP):<br>
<%Response.Write(Request.ServerVariables("remote_host"))%></p>

<p>المنهج<br>
المستخدم:<%Response.Write(Request.ServerVariables("request_method"))%></p>
```

```
p>

<p>اسم
السيرفر:<%Response.Write(Request.ServerVariables("server_name"))%></p>
البروت الذي رقم
طلبته:<%Response.Write(Request.ServerVariables("server_port"))%></p>

الوب المستخدم على برنامج<p>
السيرفر:<%Response.Write(Request.ServerVariables("server_software"))%></p>
>

</body>

</html>
```

وسوف يكون الناتج على سبيل المثال:

رقم (IP):217.21.4.46

المنهج المستخدم::Get

السيرفر اسم:arabteam2000.com

رقم البورت الذي طلبته::٨٠

Microsoft-IIS/5.0 :على السيرفر برنامج الوب المستخدم

هنا سوف اضع بعض متغيرات: **server_variable**

AUTH_TYPE: هي الطريقة الذي يستخدمه السيرفر للمصادقة على اسم المستخدم.

CONTENT_LENGTH: طول الفترة الزمنية عند الطلب

CONTENT_TYPE: نوع البيانات المطلوبة

GATEWAY_INTERFACE: نسخة السي جي أي المستخدمة في السيرفر

LOGON_USER: اسم المستخدم في حال استخدام الشبكة الداخلية للويندوز ان تي.

PATH_INFO: المسار الافتراضي للملف المطلوب على السيرفر

PATH_TRANSLATED: المسار الطبيعي للملف المطلوب على السيرفر

QUERY_STRING: نوع الاستعلام المستخدم من خلال الوصفة <A>

REMOTE_ADDR: رقم الاي بي للجهاز الزائر.

REMOTE_HOST: اسم الجهاز الزائر. Computer Name

REQUEST_METHOD: النوع المستخدم هل هو GET أو POST أو PUT.

SCRIPT_MAP: اسم النطاق Domain للصفحة.

SCRIPT_NAME: اسم الملف المطلوب مع المسار الافتراضي له

SERVER_NAME: اسم السيرفر.

SERVER_PORT: رقم البورت الذي تم الطلب من خلاله.

SERVER_PORT_SECURE: إذا كان (1) فهو مؤمن أما إذا كان (0) فهو غير مؤمن

SERVER_PROTOCOL: نوع البروتوكول المستخدم

SERVER_SOFTWARE: برنامج الويب المستخدم على السيرفر مثال Microsoft IIS 4

URL: الملف المطلوب ضمن المسار الافتراضي له

هذه إحدى المتغيرات التي نستخدمها، وهناك الكثير من المتغيرات، ومعظمها أنا أجلها، ولكن سوف أقوم بكتابة لكم الكود التالي، حيث هذا الكود سوف يقوم لك باستعراض جميع المتغيرات التي تستخدم

مع **server_variable**

```
<%  
for each x in Request.ServerVariables  
response.write(x & "<br />")  
next  
%>
```

وبذلك نكون قد انتهينا من هذه الخاصية.



-3الخاصية:TotalBytes

تستخدم لاسترجع العدد الكلي للبيانات المرسله من قبل المستخدم.

الصيغة:

```
Varbytes=Request.Totalbytes
```

مثال: إيجاد العدد الكلي للبيانات المرسلّة من قبل المستخدم؟

كود

```
<html dir="rtl">
<body>
<%
name=request.form("name")
%>
<form method="POST" action="exbytes.asp">
<p dir="rtl" align="center">:الاسم <input type="text" name="name"
size="20"><br>
<input type="submit" value="إرسال"></p>
</form>
<%
if name<>"" then
response.write("<center><B>عدد البيانات التي قمت
بإرسالها:&request.totalbytes&</b></center>")
end if
%>
</body>
</html>
```

الأمر `request.totalbytes` ، يقوم باسترجاع عدد البيانات المرسلّة من قبل المستخدم

الدرس الثامن

الكائن

Session Object

الكائن Session Object الجلسة:)

إن كائن الجلسة يستخدم لتخزين البيانات، أو تغيير الإعدادات لجلسة المستخدم (أي عندما يكون المستخدم داخل إلى الموقع). إن المتغيرات المخزنة داخل كائن الجلسة تحمل البيانات لمستخدم واحد، وهي متوفرة لكل الصفحات في تطبيق واحد.

توضيح أكثر لكائن الجلسة:

-عندما تتعامل مع أي تطبيق، فإنك تقوم بفتحه، وتعمل بعض التغييرات ومن ثم تغلقه (وهذا هو تعريف الجلسة). إن الكمبيوتر يعرف من أنت. إن يعرف متى تبدأ التطبيق ومتى تنتهي منه. ولكن على الانترنت هناك مشكلة واحدة: إن مزود خدمة الويب فإنه لا يعرفك وماذا تفعل لأن عنوان ال HTTP لا يحتوي على حالة. إن ASP يحل هذه المشكلة بواسطة إنشاء cookie خاص لكل مستخدم. إن ال cookie يرسل إلى الزبون ويحتوي على بيانات تعرف المستخدم. هذه الواجهة

تسمى Session Object.

-إن كائن الجلسة يستخدم لتخزين البيانات، أو تغيير الإعدادات لجلسة المستخدم (أي عندما يكون المستخدم داخل إلى الموقع). إن المتغيرات المخزنة داخل كائن الجلسة تحمل البيانات لمستخدم واحد، وهي متوفرة لكل الصفحات في تطبيق واحد. من البيانات التي تكون مخزنة في كائن الجلسة هي الاسم، رقم ID ، والتفضيلات. إن مزود السيرفر(الخادم) ينشئ جلسة لكل مستخدم جديد، ويدمر الجلسة عندما تنتهي صلاحيتها.

أعتقد أن مفهوم الجلسة أصبح الآن واضح لكم.

متى يبدأ الجلسة؟

تبدأ الجلسة عندما يكون:

1-المستخدم الجديد يقوم بطلب ملف ASP ، و ملف Global.asa والذي يحتوي على

`proceduer session_onstart.`

2-القيم تخزين في متغيرات الجلسة.

3-المستخدم يطلب ملف ASP ، ملف Global.asa ، يستخدم الـ `<object>` لإنشاء ال

`object` مع مدى فترة الجلسة.

متى تنتهي الجلسة؟

-تنتهي الجلسة عندما المستخدم لا يطلب أو يقوم بتحديث الصفحة في تطبيقات ال ASP لمدة محددة من الوقت، فإن المدة الافتراضية لإنهاء الجلسة هي ٢٠ دقيقة.

-إذا كنت تريد أن تضع زمن لانتهاء فعالية الصفحة لفترة أقل أو أكثر من القيمة الافتراضية، يمكنك

استخدام الخاصية Timeout.

المثال التالي يضع زمن صلاحية انتهاء الجلسة لمدة ٥ دقائق:

كود

```
<%  
session.timeout=5  
%>
```

-كما أنه يمكن أن تحدد إنتهاء أو تدمير الجلسة باستخدام المنهج Abandon.

كود

```
<%  
session.Abandon  
%>
```

ملاحظة: المشكلة الرئيسية بالنسبة للجلسات هي متى يجب أن تنتهي، فإنك لا تعرف متى يكون آخر طلب للمستخدم هل هو الطلب الأخير أم لا. ولذلك لا نعرف متى نجعل الجلسات مفتوحة، فإن الانتظار إلى فترة طويلة تستهلك موارد السيرفر. ولكن إذا تم مسح الجلسة بسرعة كبيرة فإنك سوف تخاطر في حالة ما إذا عاد المستخدم وكان السيرفر قد قام بمسح كل البيانات، وعندها سوف يظطر

المستخدم إلى البدء من جديد. إن إيجاد زمن إنتهاء الجلسة يمكن أن يكون صعباً.

إرشاد: إذا كنت تستخدم متغيرات الجلسة، فمن الأفضل أن تخزن بها كميات صغيرة من البيانات.

تخزين وإسترجاع قيم المتغيرات:

-أهم شئ بالنسبة لكائن session هو أنك تستطيع تخزين المتغيرات بداخله، كالتالي:-

كود

```
<%  
session("username")="ahmad"  
session("age")=22  
%>
```

المتغير "username" وقيمه "ahmad" ، والمتغير "age" قيمته ٢٢ .

مثال: في هذا المثال سوف اقوم بإنشاء كائن الجلسة، وعرض قيم متغيرات الجلسة:

كود

```
<html dir="rtl">  
<body>  
<%  
session("username")="ahmad"  
session("age")=22  
%>  
<%  
response.write(session("username")&session("age"))  
%>
```

```
</body>
```

```
</html>
```

- عندما تخزن القيم داخل متغيرات **session** فإنه يمكنك استدعائه في أي صفحة من تطبيقات ال

ASP، المثال التالي سوف يوضح ذلك:

سوف نقوم سوياً بإنشاء صفتين ، واسمهما **session_form.asp** والثانيه

session_result.asp:

1-الصفحة الأولى واسمها **session_form.asp:**

كود

```
<html dir="rtl">
```

```
<body>
```

```
<%
```

```
session("name")=request("name")
```

```
%>
```

```
<%
```

```
error=request("error")
```

```
if error<>"" then
```

```
if error="name_null" then
```

```
response.write("<center><b><font color='red'>عذراً لم  
تقم بإدخال قيمه داخل
```

```
</font></b></center>")
```

```
end if
```

```
end if
```

```
%>
```

```
<form method="POST" action="sesssion_result.asp">
```

```
<p>أدخل اسمك:<input type="text" name="name" size="20"><br>
```

```
<input type="submit" value="إرسال"></p>
```

```
</form>
</body>
</html>
```

شرح الكود:

-في هذه الصفحة سوف ترسل القيمة إلى نفس الصفحة.
-بعد أن أرسلت القيمة إلى نفس الصفحة، قمت بتخزين داخل متغير الجلسة.

2-الصفحة الثانية ، واسمها:session_result.asp

كود

```
<% if session("name")="" then
response.redirect("session_form.asp?error=name_null")
end if
%>
<html>
<body>
بالزائر المحيط العربي يرحب:
<%
response.write(session("name"))
%>
<p></p>
</body>
</html>
```

شرح الكود:

-في هذه الصفحة سوف يقوم بطباعة القيمة المخزنة داخل متغير الجلسة، التي خزنة قيمته داخل

المتغير في صفحة `session_form.asp`.

-أما جملة الشرطيه `if` في الأعلى الصفحة تقوم بالفحص إذا كان قيمة متغير الجلسه يساوي فارغ،

فسوف تذهب إلى صفحة `session_form.asp` وحامله معها قيمة.

-بعد أن تذهب إلى صفحة `session_form.asp` ومعها القيمة الخطأ، سوف يتم استقبال هذه

القيمة وعرض ناتج الخطأ كما هو موضح في صفحة `session_from.asp`.

-ملاحظة: في هذا المثال لم نحدد زمن إنتهاء كائن الجلسة، فعندها سوف ينتهي عند الوقت

الافتراضي ٢٠ دقيقة.

-وتستطيع أن تخزن أولوية المستخدم داخل الجلسة، ومن ثم دخول ذلك الأولوية لاختيار أي صفحة

لإرجاعها للمستخدم.



-هناك مجموعات خصائص ومنهاج وأحداث لكائن `Session Object` وهي كالتالي:-

-1 المجموعات:

تنفسك إلى:

• Contents

• StaticObjects



المجموعة

Contents

المجموعة: Contents:

-تحتوي على جميع متغيرات الجلسة.

مثال: كيفية عرض جميع متغيرات الجلسة؟

في هذا المثال سوف نستخدم ال Loop ، لعرض جميع متغيرات الجلسة الموجودة على جهازك الخاص بك:

كود

```
<%  
dim i  
for each i in session.contents  
response.write(i&"</br>")  
%>
```

مثال: في هذا المثال سوف أقوم بعرض جميع قيم متغيرت الجلسة، وإظهار عدد هذه المتغيرات.

كود

```

<html dir="rtl">
<body>
<%
dim i,j
j=session.contents.count
response.write("عدد متغيرات الجلسة = "&j&"</br>")
for i=1 to j
response.write(session.contents(i)& "</BR>")
next
%>
</body>
</html>

```

شرح الكود:

`j=session.contents.count` - المتغير `j` قيمته تساوي عدد المتغيرات الجلسة الموجوده

داخل الخاصية `contents`.

`response.write("j =&")` - عدد متغيرات الجلسة: `response.write` (" `j =&` تقوم بطباعة القيمة المخزنة داخل

المتغير `j` ، والقيمة التي بداخله عدد المتغيرات الجلسة.

`session.contents(i)` - تحتوي `i` على قيم متغيرات الجلسة.

- **ملاحظة:** بهذه الطريقة سوف تقوم باستعراض جميع الجلسات المتواجدة على جهازك الخاص

بك، سواء قمت أنت بإنشائها أو أي موقع سجلت به كان يستخدم كائن `session` عند التسجيل.



المجموعة

Staticobjects

المجموعة:Staticobjects

-تحتوي على جميع قيم متغيرات الجلسة.

مثال: إظهار القيم المخزنة داخل متغيرات الجلسة نستخدم المجموعة staticobject ، كما هو

موضح في المثال التالي :-

كود

```
<html dir="rtl">
<body>
<%
FOR each i in session.staticobjects
response.write(i)
next
%>
</body>
</html
```



-2الخصائص الكائن Session:

- CodePage
- SessionID



الخاصية

Codepage

الخاصية: Codepage

-تحتوي على كود الصفحة، أي أنها تقوم بإظهار كود محتوى الصفحة.

أمثله على محتوى كود الصفحة:

اقتباس

1256: -تعني بأن الصفحة باللغة العربية.

1252: -الصفحة باللغة الإنجليزية، والأوربية.

932: -الصفحة باللغة البيانية.

مثال: المثال التالي، سوف يظهر كود محتوى الصفحة:

كود

```
<%  
response.write(session.codepage)  
%>
```



الخاصية

SessionID

الخاصية: **SessionID**

-تضع أو تسترجع رقم ال ID لكل مستخدم، والتي تولد عن طريق السيرفر(الخادم).

الصيغة:

كود

Session.SessionID

مثال: لإسترجاع رقم ال ID لكل مستخدم:

كود

<%

Sessino.SessionID

%>



Session: المناهج الكائن -3

- Contents.Remove
- Contents.RemoveAll ()



المنهج

Content.remove

المنهج: Content.remove

- تستخدم لإزالة أو حذف متغيرات الجلسة الموجودة داخل المجموعة Contents.

مثال: حذف متغير من متغيرات: Contents

كود

```
<%  
session("test1")="first test"  
session("test2")="second test"  
session("test3")="Third test"  
session.contents.remove("test2")  
for each x in session.contents  
response.write( x & " = " & session.contents(x) & "</br>")
```

```
next
```

```
%>
```



المنهج

() Contents.RemoveAll

() Contents.RemoveAll:المنهج

-تستخدم لحذف أو لإزالة جميع متغيرات الجلسة.

مثال: المثال التالي، سوف يقوم بحذف جميع متغيرات الجلسة الموجودة داخل المجموعة

Contents:

كود

```
<%  
session("test1")="first test"  
session("test2")="second test"  
session("test3")="Third test"  
session.contents.removeAll( )  
for each x in session.contents  
response.write( x & "=" & session.contents(x) &"</br>")  
next  
%>
```

سوف تلاحظ بأن جميع متغيرات الجلسة قد حذفها من داخل المجموعة. Contents



-4 أحداث الكائن: Session

- Session_OnEnd
- Session_OnStart



الأحداث

Session_OnStart and Session_OnEnd

الحدث: Session_OnStart

-يحدث عندما السيرفر ينشأ الجلسة، وهذا الحدث يتم وضعه في ملف Global.asa.

الحدث: Session_OnEnd

-يحدث عندما تنتهي الجلسة، ويتم وضعه في ملف Globa.asa.

سوف يتم التعامل مع هذين الحدثين إنشاءً لله في درس الـ Globa.asa



وبذلك نكون قد انتهينا من الكائن Session Object.

الدرس التاسع

الكائن

Application Object

الكائن: Application Object

مجموعة من ملفات الـ ASP تعمل مع بعضها البعض لأداء مهمة معينة تسمى تطبيقات (Application).

توضيح أكثر لكائن: Application

- أي تطبيق على الشبكة هو عبارة عن مجموعة من ملفات الـ asp. إن تلك الملفات تعمل مع بعضها البعض لأداء مهمة معينة.

- إن كائن الـ Application Object في الـ ASP يستخدم لربط تلك الملفات مع بعضها البعض.

- إن كائن الـ Application Object يستخدم لتخزين ودخول المتغيرات من أي صفحة، تماماً

مثل كائن الجلسة. والفرق بينهما هو أن كل المستخدمين يتشاركون في "Application"

"Object واحد فقط."

- إن كائن الـ Application Object يجب أن يحتوي على معلومات والتي سوف تستخدم من قبل الكثير من الصفحات في التطبيق (مثل معلومات ربط قواعد البيانات). وهذا بمعنى أنه يمكنك دخول المعلومات من أي صفحة. و تعني أيضاً أنه يمكنك تغيير المعلومات من موقع واحد، ومن ثم

تنشر المعلومات الجديدة أوتوماتيكياً إلى جميع الصفحات.

وبذلك أصبح مفهوم كائن ال **Application** واضح إليكم.

-

تخزين واسترجاع قيم المتغيرات:

-متغيرات ال **Application** يمكن الوصول إليها وتغييرها بواسطة أي صفحة في التطبيق.

مثال: كيفية تعريف المتغيرات واسترجاع قيمتها؟

كود

```
<html>
<body>
<%
application("user")="Mahmoud"
response.write(application("user"))
%>
</body>
</html>
```

شرح الكود:

application("user")="Mahmoud": ال **user** هي عبارته عن متغير

Application و **Mahmoud** قيمة المتغير.

response.write(application("user": -سوف تظهر قيمة المتغير وتظهرها على برنامج الاستعراض.



المجموعات وخصائص ومنهاج الكائن **Application Object** وهي كالتالي:-

-1 المجموعات:

- **Contents**
- **StaticObjects**



المجموعة

Contents

المجموعة: Contents

-تحتوي على كل المتغيرات المضافة إلى التطبيق أو الجلسة خلال أمر ال **Script**.

مثال: المثال التالي سوف نرى كيفية استرجاع جميع متغيرات ال **Application**.

كود

```
<%  
for each x in application.contents  
response.write(x &"</br>")
```

```
next  
%>
```

مثال: المثال التالي، سوف يقوم باسترجاع قيم المتغيرات وعددها:

كود

```
<%  
application("users")="Mahmoud"  
application("age")=22  
for i=1 to application.contents.count  
response.write(i &"=" & application.contents(i)&"</br>")  
next  
%>
```

تلاحظ أنها لا تختلف عن كائن الجلسة.



المجموعة

Staticobjects

Staticobjects: المجموعة

-تحتوي على جميع قيم متغيرات الـ Application.

مثال: استخدام ال loop لإظهار جميع قيم متغيرات الجلسة.

كود

```
<%  
dim i  
for each i in application.staticobjects  
response.write(i &"</br>")  
next  
%>
```



-2 المناهج:

- Contents.Remove
- Contents.RemoveAll ()
- Lock
- Unlock



المنهج

Contents.Remove

المنهج:Contents.Remove

-تستخدم لإزالة أو حذف متغيرات Application الموجودة داخل المجموعة Contents.

مثال :كيفية حذف متغير من متغيرات ال Contents في ال:Application

كود

```
<%  
application("test1")="first test"  
application("test2")="second test"  
application("test3")="Third test"  
application.contents.remove("test2")  
for each x in application.contents  
response.write( x & "=" & application.contents(x) &"</br>")  
next  
%>
```

لا فرق بينه وبين المثال الذي تم شرحه في كائن ال session ، فلا داعي للشرح.



المنهج

() Contents.RemoveAll

المنهج:Contents.RemoveAll ()

-تستخدم لحذف أو لإزالة جميع متغيرات الجلسة.

مثال: كيفية حذف جميع متغيرات ال Contents في ال:Application

كود

```
<%  
application("test1")="first test"  
application("test2")="second test"  
application("test3")="Third test"  
application.contents.removeall( )  
for each x in application.contents  
response.write( x & "=" & application.contents(x) &"</br>")  
next  
%>
```

لا فرق بينه وبين المثال الذي تم شرحه في كائن ال session ، فلا داعي للشرح.



المنهج

Lock and Unlock

المنهج: Lock

- تمنع المستخدمين الآخرين من تغيير المتغيرات في ال (Application Object) تستخدم للتأكد من أن زبون واحد فقط يستطيع تغيير متغيرات التطبيق. (

المنهج: Unlock

- تسمح للمستخدمين الآخرين لتغيير متغيرات المخزنة في ال (Application Object) بعد أن أقفلت باستخدام المنهج. ("Lock")

الصيغة:

كود

Application.Lock

Application.Unlock

سوف يتم التطرق إليهما في درس الملف Global.asa.



-3 الأحداث:

- Application_OnEnd
- Application_OnStart



أحداث

Application_OnStart and Application_OnEnd

الحدث: Application_OnStart

-تحدث قبل إنشاء أول جلسة جديدة (عندما يتم الرجوع إلى ال "Application Object" لأول مره. وهذا الحدث يوضع في ملف.Globa.asa

الحدث: Application_OnEnd

-تحدث عندما يتم الإنشاء من التطبيق.

سوف يتم التطرق إليهما في درس الملف.Global.asa

الدرس العاشر

الكائن

Server Object

الكائن: Server Object

-تستخدم للوصول إلى خصائص والمانهج الموجوده على السيرفر(الخادم).

-هناك خصائص ومنهاج لكائن Server Object وهي كالتالي:-

1-الخصائص:

الخاصية

ScriptTimeout

الخاصية:ScriptTimeout

-تسترجع أكبر عدد من الثواني للسكربتات التي تستطيع تنفيذها قبل أن تنتهي.

الصيغة:

كود

```
server.scriptTimeout=(number of seconds)
```

number of seconds: أكبر عدد من الثواني للسكربتات تستطيع تنفيذها قبل أن ينتهي

السيرفر، وقيمتها البدائية تساوي ٩٠ ثانية.

-مثال:

كود

```
<html dir="rtl">
<body>
<%
server.scriptTimeout=200
response.write(server.scripttimeout)
%>
</body>
</html>
```

شرح الكود:

server.scriptTimeout=200: ال- ScriptTimeout سوف تنتهي بعد ٢٠٠ ثانية.

(response.write(server.scripttimeout: -سوف تظهر قيمة ال

ScriptTimeout.



2- المناهج:

Execute •

HTMLEncode •

MapPath •

URLEncode •



المنهج

Execute

المنهج: Execute

-تقوم بتنفيذ ملف ال Asp عن طريق ملف Asp آخر .

الصيغة:

كود

```
server.Execute(path)
```

path: -تحدد موقع أو مكان ملف ال ASP المراد تنفيذه.

مثال: على تنفيذ ملف ASP في صفحة أخرى:

-لنفترض بأن الصفحة الأولى تحتوي على هذا الكود وليكن اسمها: Execute.asp

كود

```
<%  
Response.write("مرحبا بكم في موقع")  
%>
```

-والصفحة الثانية وليكن اسمها Result_execute.asp تحتوي على هذا الكود التالي:-

كود

```
<html>  
<body>  
<%  
server.execute("execute.asp")  
response.write("العربي المحيط")  
%>  
</body>  
</html>
```

شرح الكود:

execute.asp -سوف تقوم باستدعاء الملف "server.execute("execute.asp: وتقوم بتنفيذ ما بداخله.



المنهج

HTMLEncode

المنهج:HTMLEncode

-تقوم بتشفير تطبيقات كود ال HTML التي قمت بتحديدھا.

مثال: على تشفير كود ال:HTML

كود

```
<html dir="rtl">
<body>
<%
Response.write("<center><b>"&server.HtmlEncode("<center><b><font
color='ffffff'>العربي المحيط</font></b></center>")&"</b></center>")
%>
</body>
</html>
```

شرح الكود:

`<center>المحيط العربي</center>`

`'server.HtmlEncode("<center><font color='ffffff:`

المنديات، أو إضافة دروس، وبها أكواد تتطلب إلى إظهارها على متصفح الانترنت، وليس ترجمة الكود وإظهار نتيجة هذا الكود.



المنهج

MapPath

المنهج: MapPath

-تقوم بتحديد المسار للمسار الحقيقي .

ملاحظة: هذا المنهج لا يستخدم في الأحداث `Session.Onend` و `Application.Onend`.

الصيغة:

كود

```
Server.MapPath(path)
```

Path: تقوم بوضع المسار داخل الملف الجذري wwwroot.

مثال: لنفترض ، الملف **Test.asp** موجود على wwwroot c:\inetpub/**/script:**

كود

```
<%  
response.write(Server.MapPath("test.asp") & "<br />")  
response.write(Server.MapPath("script/test.asp") & "<br />")  
response.write(Server.MapPath("/script/test.asp") & "<br />")  
response.write(Server.MapPath("script") & "<br />")  
response.write(Server.MapPath("/") & "<br />")  
response.write(Server.MapPath("") & "<br />")  
%>
```



المنهج

URLEncode

URLEncode: المنهج

تقوم بتشفير الرابط الذي تقوم بتحديدته أنت.

الصيغة:

كود

Server.URLEncode(string)

String: مجموعة الحروف التي تريد أن تقوم بتشفيرها.

مثال: كيفية تشفير الرابط:

كود

```
<html dir="rtl">
<body>
<%
Response.write("<center><b>"&(server.urlencode("Http://www.ArabMoheet.net"))
&"</b></center>")
%>
</body>
</html>
```

الدرس الحادي عشر

الكائن

(ASPError (ASP 3.0

الكائن: (ASPError (ASP 3.0

-تستخدم لإظهار وتوضيح البيانات لأي خطأ حدث في سكريبت صفحات ال ASP. وهذا الكائن يتم إنشائه عندما يتم استدعاء المنهج `server.Getlasterror` ، وهذه الأخطاء نستطيع الوصول إليها فقط عن طريق هذا المنهج.

في الجدول التالي يوضح خصائص الكائن: ASPError

الوصف	الخاصية
تقوم باسترجاع الكود الخطأ المتولد عن طريق الـ IIS.	ASPCode
تسترجع وصف وتوضيحات للخطأ.	ASPDescription
تسترجع مصدر الخطأ.	Category
تحدد موقع في Column الملف الذي أدى إلى الخطأ.	Column
تحدد توضيح قصير للخطأ.	Description
تحدد اسم الملف الذي تولد منه الخطأ.	File
تحدد السطر الذي حدث فيه الخطأ.	Line
تحدد مقياس كود الـ com الخطأ في الخطأ.	Number
تحدد مصدر الكود الحقيقي من السطر الذي تم حدوث الخطأ فيه.	Source

الصيغة:

كود

```
ASPError.ASPCode()
ASPError.ASPDescription()
ASPError.Category()
ASPError.Column()
ASPError.Description()
ASPError.File()
ASPError.Line()
ASPError.Number()
ASPError.Source()
```

مثال: في هذا المثال سوف أقوم بكتابة فقط الكود للتعامل مع هذه الخصائص:

كود

```
<%  
dim objErr // تقوم بتحديد متغير أنت  
set objErr=Server.GetLastError()  
response.write("ASPCode=" & objErr.ASPCode)  
response.write("<br />")  
response.write("ASPDescription=" & objErr.ASPDescription)  
response.write("<br />")  
response.write("Category=" & objErr.Category)  
response.write("<br />")  
response.write("Column=" & objErr.Column)  
response.write("<br />")  
response.write("Description=" & objErr.Description)  
response.write("<br />")  
response.write("File=" & objErr.File)  
response.write("<br />")  
response.write("Line=" & objErr.Line)  
response.write("<br />")  
response.write("Number=" & objErr.Number)  
response.write("<br />")  
response.write("Source=" & objErr.Source)  
%>
```

الدرس الثاني عشر

الكائن

Asp File System Object

الكائن: Asp File System Object

يستخدم لدخول نظام الملفات الموجوده على السيرفر، وهذا الكائن يستطيع التعامل مع الملفات والمجلدات ومسارات الدليل. أيضاً إنه من الممكن استرجاع نظام البيانات للملفات مع هذا الكائن.

هناك مناهج لكائن **File System Object** وهي كالتالي:-

ملاحظه: عند إستخدامك لهذه المناهج قد لا تؤدي عمله، وذلك يكون السبب في برامج الحماية من الفيروسات الموجوده على جهازك الخاص بك، والحل لذلك فقط عليك وقف الفحص التلقائي للفيروسات وجعله يدوياً.



المنهج **BuildPath**

المنهج: **BulidPath**

-يقوم بإضافة الاسم للمسار الموجود.

الصيغة:-

كود

```
[newpath=] FileSystemobject.BuildPath(path,name)
```

Path:-تطلب، المسار الذي يلحق بالإسم إلى.

name:-تطلب، الاسم الذي يلحق المسار

مثال:

كود

```
<%  
dim fs,path  
set fs=server.createobject("Scripting.FileSystemobject")  
path=fs.buildpath("c:mydocument","test")  
response.write(path)  
set fs=nothing  
%>
```

شرح الكود:

Fs: متغير وهو سوف يكون الخاص بالتعامل مع الملفات من خلال الجزئية **Scripting.FileSystemObject** والتي تمكننا من الوصول للملفات من خلال نظام التشغيل

Path: متغير وقيمته تساوي **test**: **path=fs.buildpath("c:mydocument","test")** تعني بأننا نريد بناء المسار **("c:mydocument","test")**.



المنهج

CopyFile

المنهج: CopyFile

-ينسخ واحد أو أكثر ملف من أي مكان إلى مكان آخر.

الصيغة:

كود

```
filesystemobject.copyfile source,destination[,overwrite]
```

Source: تحدد مسار الملف أو الملفات التي تريد أن تقوم بنسخها.

Destination: تحدد المكان الذي تريد أن تنسخ إليه الملفات.

Overwrite: إذا كانت القيمة **True** يسمح بالكتابة فوق الملف الموجود، أما إذا كانت **False**

يمنع الكتابة فوقه، والقيمة الافتراضية له هي **True**.

مثال: كيفية نسخ ملف من مكان إلى مكان آخر?

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
fs.copyfile "c:\mydocumentsweb*.htm", "c:\webpages"  
set fs=nothing  
%>
```

شرح الكود:

fs.copyfile - هذا المنهج يستخدم لنسخ ملفات من مكان لمكان.

c:\mydocumentsweb*.htm - تحدد مسار المجلد الذي يحتوي على الملفات التي تريد

نسخها.

"c:webpages:" -تحدد مسار المجلد الذي تريد أن تنسخ إليه الملفات.



المنهج

CopyFolder

المنهج: CopyFolder

-ينسخ واحد أو أكثر مجلد من أي مكان إلى مكان آخر.

الصيغة:

كود

```
filesystemobject.copyfolder source,destination[,overwrite]
```

Source: تحدد مسار المجلد أو المجلدات التي تريد أن تقوم بنسخها.

Destination: تحدد المكان الذي تريد أن تنسخ إليه المجلدات.

Overwrite: إذا كانت القيمة True يسمح بالكتابة فوق الملف الموجود، أما إذا كانت False يمنع

الكتابة فوقه، والقيمة الافتراضية له هي True.

مثال: كيفية نسخ جميع المجلدات من مكان إلى مكان آخر؟

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
fs.copyfolder "c:mydocumentsweb*", "c:webpages"  
set fs=nothing  
%>
```

شرح الكود:

fs.copyfolder: هذا المنهج يستخدم لنسخ المجلدات من مكان لمكان.
c:mydocumentsweb.: -تحدد مسار المجلد الذي يحتوي على الملفات التي تريد نسخها، و* تحدد جميع المجلدات بداخل المجلد.**Web.**
c:webpages: -تحدد مسار المجلد الذي تريد أن تنسخ إليه المجلدات.

مثال: كيفية نسخ مجلد واحد من مكان إلى مكان آخر؟

في هذا المثال سوف يتم نسخ المجلد **test** إلى المجلد **Webpages.**

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
fs.copyfolder "c:mydocumentsweb est", "c:webpages"  
set fs=nothing  
%>
```



المنهج

CreateFolder

المنهج: CreateFolder

-يقوم بعملية انشاء مجلد جديد

الصيغة:

كود

```
filesystemobject.createfolder (name)
```

name: تحدد اسم المجلد الذي تريد أن تنشاه

مثال: كيفية إنشاء مجلد؟

كود

```
<%  
dim fs,f  
set fs=server.createobject("scripting.filesystemobject")  
set f = fs.createfolder ("c:myweb")  
set fs=nothing  
set f=nothing  
%>
```

شرح الكود:

fs.createfolder: هذا المنهج يستخدم لإنشاء المجلدات.

c:myweb: تضع اسم المجلد الذي تريد انشاءه وتحدد المسار الذي سوف تضعه به.

c:webpages: *تحدد مسار المجلد الذي تريد أن تنسخ إليه المجلدات.



المنهج

CreateTextFile

المنهج: CreateTextFile

-تنشأ ملف نصي في المجلد الحالي وتسترجع الكائن **TextStream** الذي يستخدم للقراءة والكتابة من الملف.

الصيغة:

كود

```
filesystemobject.createtextfile(filename[,overwrite[,unicode]])
```

file name: اسم الملف الذي أنشأته.

overwrite: إذا كانت القيمة **True** فإن الملف تستطيع الكتابة عليه، أما إذا كانت القيمة **False**

فإنكلا تستطيع الكتابة عليها، والقيمة البدائية تكون **True**.

Unicode: إذا كانت القيمة **True** تعني بأن الملف أنشأ كرمز، أما إذا كانت القيمة **False** فإنه

تعني أن الملف أنشأ كملف **ASCII**. والقيمة البدائية تكون **False**.

مثال: كيفية إنشاء ملف والكتابة عليه؟

كود

```
<%  
dim fs,t  
set fs=server.createobject("scripting.filesystemobject")  
set t=fs.createtextfile ("c:somefile.txt",true )  
t.writeline("العربي مرحبا بك في موقع المحيط")  
t.close  
set t=nothing  
set fs=nothing  
%>
```

شرح الكود:

createtextfile: -تستخدم لإنشاء ملف.

t.writeline: -تحتوي ال t على الملف الذي تم إنشائه، **writeline**تستخدم للكتابة على الملف.

ملاحظة: ليس من الشرط ان يكون امتداد الملفات **txt** ، ممكن أن تضع أي امتداد أنت تختاره.



المنهج

DeleteFile

DeleteFile: المنهج

-تحذف واحد أو أكثر من الملفات المحددة.

الصيغة:

كود

```
filesystemobject.deletefile(filename[,force])
```

file name: اسم الملف أو الملفات التي تريد أن تقوم بحذفها.

Force: إذا كانت القيمة صحيحة فإنه سوف يتم حذف الملف، أما إذا كانت القيمة **False** أي أن

الملف ليس للحذف. والقيمة البدائية تكون **False**.

مثال: كيفية حذف ملف وفحص هل الملف موجود أم لا؟

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
if fs.fileExists("c:somefile.txt") then  
fs.deletefile("c:somefile.txt")  
response.write("<center><b>المحدد تم حذف الملف لقد</b></center>")  
else  
response.write("<center><b>عذراً الملف الذي تريد أن تحذفه غير</b></center>")  
end if  
set fs=nothing  
>%>
```

شرح الكود:

fileExists: -سوف تقوم بتحديد إذا كان الملف موجود ام لا.
if fs.fileExists("c:somefile.txt") then: -إذا كان الشرط الملف موجود نفذ الشرط
("fs.deletefile("c:somefile.txt.
deletefile: -تستخدم لحذف الملفات.



المنهج

DeleteFolder

المنهج: DeleteFolder

-تحذف واحد أو أكثر من المجلدات المحددة.

الصيغة:

كود

```
filesystemobject.deletefolder(filename[,force])
```

file name: اسم المجلد أو المجلدات التي تريد أن تقوم بحذفها.

Force: إذا كانت القيمة صحيحة فإنه سوف يتم حذف المجلد، أما إذا كانت القيمة **False** أي أن

المجلد ليس للحذف. والقيمة البدائية تكون **False**.

مثال: كيفية حذف مجلد وفحص هل الملف موجود أم لا؟

قم بإنشاء مجلد اسمه **webpages** وضعه داخل القرص الصلب **c:**

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
if fs.folderexists("c:webpages") then  
fs.deletefolder("c:webpages")  
response.write("<center><b>تم حذف المجلد</b></center>")  
else  
response.write("<center><b>عذراً المجلد غير تحذفه غير</b></center>")  
</b></center>")  
end if  
set fs=nothing  
%>
```

شرح الكود:

folderExists: - سوف تقوم بتحديد إذا كان المجلد موجود ام لا.

if fs.folderExists("c:somefile.txt") then: - إذا كان المجلد موجود نفذ الشرط

("fs.deletefolder("c:webpages.

deletefolder: - تستخدم لحذف الملفات.

بعد أن تتم عملية الحذف قم بالمحاولة مرة أخرى، وشاهد نتيجة الفحص.



المنهج

DriveExists

المنهج: DriveExists

-تفحص إذا كائن القرص الصلب موجود أو لا.

الصيغة:

كود

```
filesystemobject.driveExists(drive)
```

Drive: اسم القرص المحدد للفحص.

مثال :كيفية معرفة إذا كان القرص المحدد موجود أو غير موجود؟

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
if fs.driveExists("c:")=true then  
response.write("<center><b>The drive c:Exists</b></center>")  
else  
response.write("<center><b>The drive c:is not Exists</b></center>")  
end if  
set fs=nothing  
%>
```

شرح الكود:

DriveExists: -سوف تقوم بالفحص إذا كان القرص موجود ام لا.
if fs.driveExists("c:")==true then: -إذا كان القرص موجود نفذ الشرط.



المنهج

FileExists

المنهج:FileExists

-تفحص إذا كائن الملف موجود أو لا.

الصيغة:

كود

```
filesystemobject.FileExists(filename)
```

Filename: اسم الملف المحدد للفحص.

مثال: كيفية معرفة إذا كان الملف المحدد موجود أو غير موجود؟
قم بإنشاء الملف **somefile.txt** على القرص الصلب **c:** ثم شاهد النتيجة:

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
if fs.fileExists("c:somefile.txt")=true then  
response.write("<center><b>موجود الملف</b></center>")  
else  
response.write("<center><b>غير موجود الملف</b></center>")  
end if  
set fs=nothing  
%>
```



المنهج

FolderExists

المنهج: FolderExists

-تفحص إذا كان المجلد موجود أو لا.

الصيغة:

كود

```
filesystemobject.FolderExists(filename)
```

Filename: اسم المجلد المحدد للفحص.

مثال: كيفية معرفة إذا كان المجلد المحدد موجود أو غير موجود؟
قم بإنشاء المجلد **webpages** على القرص الصلب **c** ثم شاهد النتيجة:

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
if fs.folderExists("c:webpages")=true then  
response.write("<center><b>موجود المجلد</b></center>")  
else  
response.write("<center><b>غير موجود المجلد</b></center>")  
end if  
set fs=nothing  
%>
```



المنهج

GetAbsolutePathName

المنهج: GetAbsolutePathName

-تسترجع المسار المكمل الجذري للقرص الصلب للمسار المحدد.

الصيغة:

كود

filesystemobject.GetAbsolutePathName(path)

Path: لتحديد المسار الكامل للجذر.

مثال :

كود

```
<%  
dim fs,path  
set fs=server.createobject("scripting.filesystemobject")  
path=fs.getabsolutepathname("c:")  
response.write("<center><b>"&path"&"/b></center>")  
%>
```

القيمة الافتراضية للمسار هي ملفات. System
":getabsolutepathname("c : سوف تقوم بتحديد المسار للقرص c وسوف تكون

مثال :

كود

```
<%  
dim fs,path  
set fs=server.createobject("scripting.filesystemobject")  
if fs.fileExists("c:/somefile.txt")=true then  
path=fs.getabsolutepathname("somefile.txt")  
response.write("<center><b>"&path"&"/b></center>")  
else
```

```
response.write("<center><b>يوجد ملف بهذا الاسم تأكد من اسم لا >b></center>")
</b></center>")
end if
set fs=nothing
set path=nothing
%>
```



المنهج

GetBaseName

المنهج: GetBaseName

-تسترجع الاسم الأساسي للملف أو المجلد المحدد.

الصيغة:

كود

```
filesystemobject.GetBaseName(path)
```

Path: المسار للمجلد او الملف الذي سوف يتم استرجاع اسمه.

مثال :

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
response.write("<center><b>"&  
fs.getbasename("c:somefile.txt")&"</b></center>")  
set fs=nothing  
%>
```

شرح الكود:

تستخدم لإرجاع اسم الملف أو المجلد من المسار المحدد.



المنهج

GetDrive

المنهج: GetDrive

-تسترجع الكائن Drive تبعاً لى القرص المحدد في المسار.

الصيغة:

```
filesystemobject.GetDrive(drivespec)
```

drivespec يستطيع كتابة القرص بين قوسين (c) ، أو يكتب هكذا (c:) ، أو بهذا الشكل (c:) ، أو أي مواصفات شبكة مشتركة. (computer2share1 \)

مثال :

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
set d=fs.getdrive("c:")  
response.write("<center><b>"&d"</b></center>")  
set fs=nothing  
%>
```



المنهج

GetDrivename

المنهج: GetDrivename

-تسترجع اسم القرص من المسار المحدد.

الصيغة:

كود

```
filesystemobject.GetDrivename(path)
```

path: المسار المحدد الذي سوف يسترجع اسم القرص.

مثال :

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
d=fs.getdrivename("c:somefile.txt")  
response.write("<center><b>"&d"</b></center>")  
set fs=nothing  
%>
```

المنهج

GetExtensionName

المنهج: GetExtensionName

-تسترجع اسم امتداد الملف الموجود للمسار المحدد .

الصيغة:

كود

filesystemobject.GetExtensionName(path)

path: المسار المحدد الذي سوف يسترجع اسم امتداد الملف.

مثال :

كود

```
<%  
dim fs,d  
set fs=server.createobject("scripting.filesystemobject")  
d=fs.getextensionname("c:somefile.txt")  
response.write("<center><b>"&d"</b></center>")  
set fs=nothing  
%>
```



كود

المنهج

GetFile

المنهج: GetFile

-تسترجع الكائن File للمسار المحدد.

الصيغة:

كود

```
filesystemobject.GetFiles(path)
```

path: المسار المحدد للملف.

مثال :

كود

```
<%  
dim fs,d  
set fs=server.createobject("scripting.filesystemobject")  
if fs.fileexists ("c:somefile.txt")=true then  
set d=fs.getfile("c:somefile.txt")  
response.write("<center><b>هو تعديل تم على الملف آخر</br>")  
response.write(d.datelastmodified)  
response.write("</b></center>")  
else  
response.write("<center><b>الملف المحدد غير موجود على القرص</b></center>")  
end if  
set fs=nothing  
set d=nothing  
%>
```

شرح الكود:

dateLastModified: -تظهر آخر تاريخ ووقت تم تعديل على الملف.



المنهج

GetFileName

المنهج: GetFileName

-تسترجع اسم الملف او المجلد لآخر مركب في المسار المحدد .

الصيغة:

كود

```
filesystemobject.GetFileName(path)
```

path: المسار المحدد للملف او المجلد.

مثال :

كود

```
<%  
dim fs,d,s  
set fs=server.createobject("scripting.filesystemobject")  
d=fs.getfilename("c:somefile.txt")
```

```
s=fs.getfilename("c: est est")
response.write("<center><b>")
response.write(d "</br>"&s)
response.write("</b></center>")
set fs=nothing
%>
```



المنهج

GetFolder

المنهج: GetFolder

-تسترجع الكائن Folder للمسار المحدد .

الصيغة:

كود

```
filesystemobject.GetFolder(path)
```

path: المسار المحدد للمجلد.

مثال :

قم بإنشاء المجلد webpages على الجهاز الخاص بك على القرص C وبعدها شاهد النتيجة.

كود

```
<%  
dim fs,d  
set fs=server.createobject("scripting.filesystemobject")  
if fs.folderexists ("c:webpages")=true then  
set d=fs.getfolder("c:webpages")  
response.write("<center><b>هو تم على المجلد آخر تعديل</br>")  
response.write(d.datelastmodified)  
response.write("</b></center>")  
else  
response.write("<center><b>الملف المحدد غير موجود على القرص</b></center>")  
end if  
set fs=nothing  
set d=nothing  
%>
```



المنهج

GetParentFoldername

المنهج: GetParentFolderName

-تسترجع الاسم من الدليل الرئيسي المجلد لآخر مركب في المسار المحدد .

الصيغة:

كود

```
filesystemobject.GetParentFolderName(path)
```

path: المسار للملف أو المجلد لمعرفة الدليل الرئيسي للمجلد في المسار المحدد.

مثال :

كود

```
<%  
dim fs,d  
set fs=server.createobject("scripting.filesystemobject")  
d=fs.getparentfoldername("c:winncursrs3dagro.cur")  
response.write("<center><b>"&d)  
response.write("</b></center>")  
set fs=nothing  
%>
```



المنهج

GetSpecialFolder

المنهج: GetSpecialFolder

-تسترجع مسار لبعض مجلدات الوندوز.

الصيغة:

كود

```
filesystemobject.GetSpecialFolder(Foldername)
```

Foldername: المجلد الذي سوف يتم استرجاعه، عندما تكون القيم التالية:

0=WindowsFolder - Contains files installed by the Windows operating system

1=SystemFolder - Contains libraries, fonts, and device drivers

2=TemporaryFolder - Used to store temporary files

مثال :

كود

```
<%  
dim fs,d,s,c  
set fs=server.createobject("scripting.filesystemobject")  
d=fs.getspecialfolder(0)  
s=fs.getspecialfolder(1)
```

```
c=fs.getspecialfolder(2)
response.write("<center><b>"&d&"</br>"&s&"</br>"&c)
response.write("</b></center>")
set fs=nothing
%>
```



المنهج

GetTempName

المنهج - GetTempName: تقوم باسترجاع ملفات أو مجلدات عشوائية من قائمة ملفات الانترنت المؤقتة.

الصيغة:

كود

```
filesystemobject.GetTempName
```

مثال :

كود

```
<%  
dim fs,tfolder,tname,tfile  
set fs=server.createobject("scripting.filesystemobject")  
set tfolder=fs.getspecialfolder(2)  
tname=fs.getTempname  
set tfile=tfolder.createtextfile(tname)  
response.write(tfile)  
set fs=nothing  
set tfolder=nothing  
set tfile=nothin  
%>
```



المنهج

MoveFile

المنهج: MoveFile

-تنقل واحد أو أكثر من ملف من مكان واحد إلى آخر.

الصيغة:

كود

filesystemobject.MoveFile (source,destination)

Source: مسار الملف أو الملفات التي سوف يتم نقلها.
Destination: المكان الذي سوف يتم نقل الملفات إليه.

مثال :

قبل مشاهدة النتيجة، قم بإنشاء مجلدين المجلد الأول اسمه **Web** وضع بداخله صور امتدادها **gif** ،
والمجلد الثاني اسمه **Images** وضعهما على القرص **c**

كود

```
<%  
dim fs  
set fs=server.createobject("scripting.filesystemobject")  
fs.movefile "c:web*.gif","c:images"  
set fs=nothing  
%>
```



المنهج

MoveFolder

المنهج: **MoveFile**

-تنقل واحد أو أكثر من مجلد من مكان واحد إلى آخر.

الصيغة:

filesystemobject.MoveFolder (source,destination)

Source: مسار المجلد أو المجلدات التي سوف يتم نقلها.

Destination: المكان الذي سوف يتم نقل المجلد إليه.

مثال :

قبل مشاهدة النتيجة قم بإنشاء مجلدين المجلد الأول اسمه **Webpages** والثاني **Images** وضعهما على القرص **c.**

```
<%  
dim fs,tfolder,tname,tfile  
set fs=server.createobject("scripting.filesystemobject")  
fs.movefolder "c:webpages","c:images"  
set fs=nothing  
%>
```

أرجو من الله أن لاتنسوا من جمعة بدعاء بظهر الغيب