

بِسْمِ اللّٰهِ الرَّحْمٰنِ الرَّحِیْمِ



javascript - SQL - ADO

آموزش مباحث پیشرفته در طراحی صفحات وب

تهیه و تنظیم:

mohammad6347@gmail.com

فصل اول

java script

مقدمه

بنام یگانه ی هستی

جاوا اسکریپت یک زبان اسکریپت می باشد . زبان اسکریپت یک زبان برنامه نویسی کم حجم است.

هر خط جاوا اسکریپت قابل اجرا می باشد. این زبان توسط اکثر **Browser** ها پشتیبانی میشود ، مانند **Explore , Netscape Internet** برنامه نویسی با زبان جاوا این قابلیت را دارد که می توان کدهای جاوا اسکریپت را در میان کدهای **HTML** نیز قرار داد .

برای مطالعه ی این دوره ی آموزشی نیاز است کاربران محترم با دوره های زیر آشنایی داشته باشند :

HTML –

– مبانی برنامه نویسی

– آشنایی با مبانی **Database** و یک برنامه مانند **Access**

جهت کد نویسی و طراحی صفحات وب نیاز به یک **Text Editor** ساده مانند **Notepad** که در تمامی ویندوزها قابل دسترسی می باشد و نیاز به نصب **Editor Text** جدید نیست .

ضمناً سیستم شما باید دارای مشخصات سخت افزاری ذیل باشد :

– حداقل سیستم مورد نیاز پنتیوم ۲۳۳

– ۱۶ مگابایت **Ram**

– حدود ۶۰۰ مگا بایت فضای آزاد

در پایان از کاربران محترم خواهشمند است در صورت داشتن هر گونه انتقاد ، پیشنهاد و ارائه سوالات احتمالی با نویسنده از طریق پست الکترونیکی زیر تماس حاصل نمایند .

<http://www.etvto.ir/ostadonline/ebrahimzadeh@etvto.ir>

چگونه کدهای جاوا اسکریپت را در HTML وارد کنیم؟

```
<html>
<head></head>
<body>
<script type=" text/ java script">
  Java Script کدهای
</script>
</body>
</html>
```

برای نوشتن توضیحات : (در HTML)

// توضیحات

/* توضیحات */

کدهای Java Script در کدام قسمت HTML داخل میشود؟

هم در قسمت HEAD :

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

اگر کدهای **Java Script** در قسمت **Head** وارد شوند ، مطمئن خواهید بود که در هنگام بارگذاری صفحه برنامه شما در حافظه **Load** میشود قبل از اینکه کسی از صفحه استفاده کند.
هم در قسمت **Body** :

```
<html>
<head>
</head>
<body>
<script type="text/java script">
    Java Script کدهای
</script>
</body>
```

اگر کدها در قسمت **Body** باشند این کد معمولا قسمتهای دیگر صفحه را ایجاد میکند.
مثال :

وقتی که کد در قسمت **Body** باشد :

```
<html>
<script type="text/java script">
function message()
{
    alert (" HELLO USER")
}
</script>
</head>
<body onload=" message()">
</body>
```

در این مثال وقتی صفحه بارگذاری میشود یک پیغام **HELLO USER** برای شما نمایش داده میشود.

چگونه یک کد **Java script** خارجی را در **HTML** اجرا میکنیم ؟

```
document.write(" This script Is External")
```

این عبارت را در یک فایل به نام **xxx.js** ذخیره میکنیم.

و حال آنرا در کد HTML خودمان فرا می خوانیم.

```
<html>
<head>
</head>
<body>
<script src=" xxx.js"></script>
</body>
</html>
```

تعریف متغیرها

۱- اسم متغیر به حروف بزرگ و کوچک حساسیت دارد.

۲- اسم متغیرها باید با حرف شروع شود.

- در تعریف متغیر استفاده از کلمه **Var** دلخواه است.

Var strname

مقدار دهی به متغیرها :

```
Var strname="name"
↳ strname="name"
Var strname=مقدار
Strname= مقدار
```

عملگرها-تعریف عملگرهای محاسباتی

جمع +

تفریق -

ضرب *

تقسیم /

باقیمانده %

افزایش یک واحدی ++

کاهش یک واحدی --
 تعریف عملگر های مقایسه ای
 == معادل بودن
 != برابر نبودن!
 > بزرگتر از
 / < کوچکتر از
 <= بزرگتر یا مساوی
 >= کوچکتر یا مساوی
 تعریف عملگر های منطقی :
 && and
 || OR
 ! NOT
 تعریف عملگر رشته ای :
 + الحاق دو رشته
 مثال:

```
Txt1="What A Very"
Txt2="Nice Day"
Txt3= txt1+txt2
مخروج= What A Very Nice Day document .write (txt3)
```

تعریف تابع و توابع در Java Script

Function اسم تابع (...arg1,arg2)

JS کدهای

{

یک تابع میتواند آرگومان هم نداشته باشد.

مثال :

```
Function result (a,b)
{
    c=a+b<br>
    return c
}
```

چگونه یک تابع را فراخوانی کنیم

- یک تابع هنگامی اجرا میشود که آن را فراخوانی کرده باشیم . به منظور فراخوانی یک تابع ملزم به استفاده از آرگومان (arg1,arg2) می باشیم ، با این توضیح که نام آرگومانها بدون () باشند . اگر یک تابع مقداری برای باز گرداندن به برنامه اصلی داشت باید آن را توسط Return برگردانیم.
مثال :

```
Function total (a,b)
{result=a + b return result}
JS کدهای sum=total(2,3)
```

فرستادن پیغام برای کاربر

اگر بخواهیم برای User یک پیغام بنویسیم :

```
Alert("This is a message")
```

جملات شرطی :

الف-IF

```
If ( شرط )
}
```

در صورت درست بودن شرط کد ها اجرا میشوند

```
{
```

ب- IF.....ELSE

```
If ( شرط )
}
```

```

در صورت درست بودن شرط
{
else
}

```

```

در صورت درست نبودن شرط
{
مثال :

```

اگر ساعت قبل از ۱۰ باشد Good Morning نشان می دهد و در غیر این صورت Good Day نشان میدهد.

```

Var d=new Date ()
Var time=d.getHours()
If (time< 10)
{
document. write (" Good Morning")
}
else
{
document .write ("Good Day")
}
}

```

Switch

```

Switch ( شرط )
}

```

```

: case Label 1

```

باشد این کدها اجرا میشود Label 1 اگر شرط برابر [Codes]

```

Break

```

```

: Case Label 2

```

باشد این کدها اجرا میشود Label 2 اگر شرط برابر [Codes]

```

Break

```

```

: Default

```


نباشد این کدها اجرا میشود Label , Label 2 اگر شرط برابر [Codes]
 {

مثال :

```

var d= new Date ()
TheDay=d.getDay()
switch (theDay)
{
  case 5 :
  document.write(" finally Friday")
  break
  case 6 :
  document.write(" Nice Saturday")
  break
  case 0 :
  document.write(" sleepy Sunday")
  break
  default
  document.write("I'm Looking Forward To This weekend")
}

```

زبانهای شیء گرا - Object Oriented

Script یک زبان برنامه نویسی Object Oriented (شیء گرا) میباشد.

در این قسمت مفهوم شیء گرای می پردازیم. یک برنامه O-O شامل مفاهیم Object ، Properties ، Methods و Events میباشد.

Objects : در دنیای حقیقی ، اشیاء مانند کامپیوتر ، دوچرخه و ... میباشدند. در زبان JS اشیاء درون Browser ها بوده و عبارتند از پنجره ها ، فرمها ، دکمه ها و Check Box ها.

Properties : اشیاء نیز دارای مشخصه هایی میباشدند. مثلاً گربه دارای پوست میباشد، کامپیوتر دارای کیبورد و دوچرخه دارای چرخ است.

Methods : کاری که اشیاء میتوانند انجام دهند Methods نامیده میشود.

Events (وقایع): عملیاتی هستند که کاربر در هنگام کار با صفحه طراحی شده ، انجام میدهد. پذیرش (Submitting) یک فرم و تغییر مکان نشانگر ماوس بر روی یک تصویر دو نمونه از حالات و وقایع اتفاق افتاده است.

این وقایع توسط دستوراتی به نام **Event Handler** برنامه، کنترل می شوند.

حلقه های تکرار

۱- حلقه تکرار **While** :

While (شرط) {

JS کدهای

}

۲- حلقه تکرار **Do...While** :

Do

}

JS کدهای

{

while (شرط)

۳- حلقه تکرار **For** :

For (مقدار اولیه ; شرط ; افزایش)

}

JS کدهای

```
{
```

مثالی از حلقه For :

```
For(i=0; i<=5; i++)
{
    document.write("The Number Is"+i)
    document.write(" ")
}
```

آبجکت String

آبجکت String : آبجکت String برای کار کردن با متن است. Length ، یک Property برای String است که تعداد کاراکترهای موجود در رشته را بر می گرداند.

:Length

مثال :

```
var str="Esfahan Information Technology Center"
document . write (" "+str+"")document.write(str.length)
```

خروجی برنامه:

Esfahan Information Technology Center 37

Fontcolor () رنگ متن را عوض میکند.

Index Of () مکان کاراکتر داده شده را بر میگرداند.

Match () یک لغت در متن را پیدا میکند.

Substr () یک لغت در متن را پیدا میکند.

To LowerCase () به حروف کوچک تبدیل میکند.

To UpperCase () به حروف بزرگ تبدیل میکند.

مثال از متد FontColor ()

```
Var txt=" Information Technology " document. write
(""+txt.fontcolor()+"" ) document. write
(""+ txt.fontcolor('red')+"
```

مثالی از متد IndexOf ()

```
Var str="Esfahan Information Technology Center"
Var pos=str.indexOf
("Information") If (pos >=0)
{
    document. write("Information Found At Position : ")
    document .write(pos+"")
}
else
{
    document . write ( "Information not<br> Found ")
}
}
```

این مثال تست میکند که آیا یک متن حاوی یک لغت خاص است یا نه. اگر آن لغت خاص در متن باشد مکان اولین کاراکتر از آن لغت را برمیگرداند. اولین مکان در متن با عدد ۰ مشخص میشود.

W3 School Is Great 0 1 2 مثال از متد Match () :

```
Var str="Active Server Page" Document.write(str.match("server"))
```

تست میکند که آیا یک رشته حاوی یک لغت خاص است یا نه.

مثال از متد substring() , substr :

```
Var str = "Information Technology") document .
write ( str.substr (2,6)) document .
write (str.substring(2,6))
```

خروجی :

Format Form

آبجکت آرایه

تعریف یک آرایه :

برای تعریف آرایه حتماً باید از کلمه **New** استفاده کرد.

```
Var My-name=new Array(5)
```

ایجاد میکند **My-name** این دستور یک آرایه پنج تایی به نام مقدار دهی به آرایه :

```
My-names [0]="Ali" My-names [1]="Hassan"
My-names [2]="Davood"
My-names [3]="Mohammad" My-names [4]="Mehdi "
```

چاپ اعضاء آرایه :

چاپ همه اعضاء آرایه

```
for ( i=0 ; i< my-names.length ; i++) document.
write(my-names[i])
Hassan
```

معرفی متد ها و خصوصیات یک آرایه

معرفی متد ها و خصوصیات یک آرایه :

Length تعداد عناصر یک آرایه را برمیگرداند.

Concat() دو آرایه را به هم الحاق میکند.

Reverse() آرایه را معکوس میکند.

Sort() آرایه را مرتب میکند.

Join() رشته ای الحاق شده از تمام اعضاء آرایه برمیگرداند.

Slice() یک تکه از آرایه ، برمیگرداند .

مثال :

```
aa=new Array(2) bb= new Array(2) cc=new Array()
aa[0]="10" aa[1]="20" bb[0]="30" bb[1]="40" cc=aa.
concat (bb) cc=aa.join(" ")
```

چیزی که بین اعضاء آرایه قرار میگیرد.

```
aa.slice(i,j)
```

آبجکت Date

آبجکت Date :

هنگامی که بخواهیم با مقادیر زمان و تاریخ کار کنیم از این **Object** استفاده میکنیم. برای تعریف یک آبجکت **Date** باید از دستور زیر استفاده کنیم.

```
()Var my-date=new Date
```

بعد از تعریف **Object** جدید با استفاده از متد ها میتوان به مقادیر آبجکت **Date** دسترسی داشته باشیم.

معرفی متدهای Date

معرفی متدهای Date :

Date() Date برمیگرداند.

آبجکت set / getDate () روزهای هفته را برمیگرداند (۱-۳۱) / تنظیم میکند.

set / getDay () روزهای هفته را برمیگرداند (۰-۶) set / getMonth () ماه را برمیگرداند (۰-۱۱) / تنظیم میکند.

set / getHour () ساعت را برمیگرداند / تنظیم میکند set / getMinutes () دقیقه را برمیگرداند / تنظیم میکند.

set / getSecond () ثانیه را برمیگرداند / تنظیم میکند.

set / getYear () سال را برمیگرداند / تنظیم میکند.

آبجکتهای Window

آبجکتهای Window :

- 1- Alert
Alert ("Hello")
- 2- Confirm

مثال:

```

var name= confirm("Press A Button") if <br>(name==true)
{
    document. write("You Press OK")
}
else
{
    document. write("You Press Cancel")
}
3- Prompt
Var name= prompt("Please enter Your Name "," ")
If (name!=null&&name!=" ")
{
    document. write ("Hello"+name)
}

```

باز کردن یک پنجره جدید - window.open

باز کردن یک پنجره جدید window.open

در این مثال وقتی روی کلید Open کلیک کنیم یک پنجره جدید باز میشود که حاوی URL سایت Yahoo میباشد.

مثال تکمیلی برای تغییر خصوصیات پنجره باز شده :

```

Function openwindow()
{
    window.open("http://www.yahoo.com" ,
        "my-new-window,toolbar=yes,location=yes,
        directories=no,status=no,menubar=yes,
        scrollbar=yes,resizeable=no,width=400,height=400")
}

```

مثالی از ترکیب متدهای alert , confirm :


```

If ( confirm( "Are you sure want to do that"))
{
  alert("You said yes")
}
else
{
  alert("You said no")
}

```

چگونه یک پیغام را در خط وضعیت (Status Bar) اینترنت اکسپلورر قرار دهیم؟

```

<head>
Function load ()
{
  window.status="Put your message here"
}
</head>
<body onload="load()">
</body>

```

گرفتن Print از صفحه

```

< head >
< script type="text/javascript " >
function printpage()
{
  window . print()
}
</script>
</head>
<body>
<form>
<input type="button" value="print this page" onclick="printpage()">
</form>
</body>

```

مثال :

```

Function newwindow (bookgif)
{
  bookwindow=window.open("bookgif" , width=140 , height=160)
  bookwindow.focus()
}
<a href="javascript:newwindow ('java-sm.gif')">welcome</a>
<a href ="javascript: newwindow('sun.gif')">Happy</a>

```

به وسیله این مثال میتوان در یک صفحه چندین پیوند را نشان داد.

آبجکت Browser

۱- تشخیص نوع Browser :

```
document .write (screen .width)
document .write ("*")
document .write(screen .height)
```

توسط Property از آبجکت Navigator می توان نوع Browser را مشخص کنیم. برای آبجکت Navigator، Property های دیگری نیز وجود دارد :

نوع Browser

Navigator . appName تشخیص ورژن

Browser . appVersion تشخیص سیستم عامل . Platform.

آدرس موجود در خط آدرس Browser را میدهد. Referrer برای اینکه آیا یک Plugin نصب است Plugins.

۲- تشخیص Resolution مانیتور :

```
document.write(window .screen .colordepth)
document.write("bit")
```

۳- تشخیص Mode رنگ کامپیوتر :

```
<a href ="sean.html"
  onmouseover="window.status='best kind in the world';
  return true "onmouseout="window.status=" " ;
  return true"> My Son</a>
```

مثال قرار دادن پیامها در Status Bar :

```
<a href ="sean.html"
  onmouseover="window.status='best kind in the world';
  return true "onmouseout="window.status=" " ;
  return true"> My Son</a>
```

هنگامی که روی لینک My Son میرویم در خط وضعیت عبارت Best Kind In The World نمایش داده میشود.

ایجاد Rollover ها

مفهوم Rollover بسیار ساده میباشد. بدین معنا که ابتدا دو تصویر وجود داشته تصویر اول در هنگام دریافت صفحه وب توسط کاربر در Browser نمایش داده میشود. بعد از اینکه کاربر ماوس را بر روی اولین تصویر برد، Browser پس از چند ثانیه تصویر دوم را با تصویر اول عوض میکند.

مثال :

```
<a href="next.html"
  onmouseover="document.arrow.src='images/redarrow.gif'
  " onmouseout="document.arrow.src='images/bluearrow.gif'">

</a>
```

Alt ، توضیحی درون مستطیل میدهد.

آبجکت Frame

بررسی تگ Frame در HTML :

مفهوم Frameset مشخص کننده چندین قاب در صفحه وب بوده و دارای تگهای Frame برای هر صفحه میباشد .

Border اندازه مرز بین قابها را بر حسب پیکسل مشخص میکند .

Cols ابعاد ستونها را بر حسب پیکسل برای هر قاب مشخص میکند .

Frameboard با مقادیر Yes , No وجود خطوط مرزی بین قابها را مشخص میکند .

Rows ابعاد سطرها را بر حسب پیکسل برای هر قاب مشخص میکند .

Frame مکان و ویژگی هر قاب را مشخص میکند.

Name برای ارجاع صفحات مجموعه قاب توسط Java Script استفاده میشود.

SRC آدرس صفحه در قاب مشخص میشود.

عملگرهای OR و AND

مثال :

```
SELECT * FROM Persons WHERE FirstName ='Ali' AND
LastName ='Hasani'
SELECT * FROM Persons WHERE (
FirstName='Ali' OR FirstName='Payam' ) AND
LastName =' Hasani'
```

فصل دوم

SQL

SQL

توضیح :

توسط SQL میتوان درون یک بانک اطلاعاتی به جستجو پرداخته (Query) و نتیجه را برگرداند. بانک اطلاعاتی شامل آبجکتی به نام جدول (Table) میباشد.

رکوردها در بانکهای اطلاعات و در جداول ذخیره میگردند.

جدول شامل سطر و ستون میباشد.

در زیر میتوان یک جدول را مشاهده کرد :

Last Name First Name Address City

مثال:

Select Last name from persons

پر کاربرد ترین دستورات SQL شامل موارد زیر است :

SELECT استخراج یک داده از بانک اطلاعاتی .

UPDATE به روز رسانی یک داده درون بانک .

DELETE پاک کردن یک داده از بانک اطلاعاتی .

INSERT وارد کردن یک داده جدید به بانک اطلاعاتی.

همچنین در SQL میتوان داده هایی نیز تعریف کرد :

CREATE TABLE ایجاد یک جدول جدید .

ALTER TABLE تغییر دادن یک جدول .

DROP TABLE پاک کردن یک جدول.

CREATE INDEX ایجاد یک اندیس .

DROP INDEX (کلید جستجو) پاک کردن یک اندیس

Active Server Pages - ASP و SQL

SQL یکی از قسمت‌های خیلی مهم ASP میباشد زیرا در ASP برای کار با بانکهای اطلاعاتی از SQL استفاده میشود. و توسط تکنولوژی ADO میتوان از SQL در ASP استفاده کرد.

دستور SELECT :

SELECT column-name(s) FROM table-name

مثال :

نام جدول Persons است

Last Name First Name Address City

Hasani Ali Esfahan

Alian Payam Esfahan

Mehdian Akbari Tehran

Javadi Hamid Tehran

SELECT LastName , FirstName FROM Persons Last

Name First Name Hasani Ali

Alian Payam

Mehdian Akbari

Javadi Hamid

مثال :

```
SELECT * FROM Persons
```

کل جدول به عنوان خروجی برمیگردد.

دستور Select همراه با شرط

SELECT column FROM table WHERE column condition value

عملگر ها در SQL:

عملگر مفهوم

= معادل بودن

<> برابر نبودن

< بزرگتر

> کوچکتر

<= بزرگتر مساوی

>= کوچکتر مساوی

between بین یک دامنه

مثال :

```
SELECT * FROM Persons WHERE city='Esfahan' first
name last name address city
hasani ali Esfahan
alian payam Esfahan
```

دستور DISTINCT (جداسازی)

SELECT DISTINCT column-name(s) FROM table-name

مثال :

```
نام جدول Order است Company Order Number
sega 3412
w3s 2312
Trio 4678
W3S 6778
SELECT DISTINCT Company FROM Order SEGA
W3S
Trio
```

دستور Order By

برای مرتب کردن سطرها

نام جدول Order است Company Order Number

sega 3412

ABS Shop 5678

w3s 3212

W3S 6778

مثال :


```
SELECT Company , OrderNumber FROM Orders ORDERBY Company
```

نتیجه:

```
Company Order Number
sega 3412
ABS shop 5678
w3s 6778
w3s 3212
```

دستور Insert

```
INSERT INTO Table-Name
(Column1 , column2) VALUES (value1,value2)
```

مثال :

```
Persons نام جدول last name First name Adress city
Hasani Javad
No 40 Esfahan
```

```
INSERTINTO Person VALUES
('Farhadi', 'Ali', 'NO 15', 'Tehran')
```

معرفی دستور Delete

```
DELETE FROM table-name WHERE column-name=some-value
```

مثال :

```
DELETE FROM Person WHERE Lastname='hasani'  
نام جدول Persons نام last name First name Adress city  
alian hasani NO 40 Esfahan  
Hasani Ali No 15 Tehran
```

```
نتیجه :  
last name First name Adress city  
alian hasani NO 40 Esfahan  
SEGA  
W3S  
Trio
```

فصل سوم

ADO

دسترسی به یک پایگاه داده از یک صفحه ASP

۱- ایجاد یک ADO Connection در یک پایگاه داده

۲- باز کردن Connection پایگاه داده

۳- ایجاد یک

4 ADO Record Set-باز کردن

5 Record set-گرفتن داده هایی که نیاز داریم از

6 Record Set-بستن

7 Record Set-بستن

Connection

۱- ایجاد یک ADO Connection به یک پایگاه داده

```
مسیر فایل نمونه C:/InetPub/wwwroot/nor.mdb DSN-LESS روش 1-1
<% var conn=Server.CreateObject
("ADODB.Connection")Conn.Provider="Microsoft.
Jet.OLEDB.4.0" Conn.Open
("C:/InetPub/wwwroot/nor.mdb")%>
```

۲- روش ODBC

در این روش ابتدا باید به Data Base ، یک ODBD Connection اضافه کنیم و سپس از طریق ADO به فایل DSN به طریق زیر Connect کنیم.

```
<% var conn=Server.CreateObject
("ADODB.Connection")conn.open("nor".DSN)%>
```

ایجاد یک ODBC Connection در پایگاه داده MS Access

۱- وارد شدن به ODBC از Control Panel

۲- انتخاب

3 -System DSN- کلیک کردن روی دکمه ADD

۴- انتخاب Microsoft Access Driver و کلیک کردن روی دکمه

5 - Finish - در مرحله بعد کلیک کردن بر روی دکمه Select و تعیین محل پایگاه داده

۶- دادن یک نام در قسمت

7 -Data Source Name- کلیک کردن روی دکمه OK برای اینکه قادر باشیم اطلاعات یک پایگاه داده را بخوانیم اطلاعات باید ابتدا در Load ,Record Set شوند. بنابر این بعد از ساختن یک Connection باید یک Record Set ایجاد کنیم.

مثال :

```
Data Base = nor.mdb <br>
نام جدول = <br>
Customer C:/Inetpub/wwwroot/nor.mdb
مسیر فایل Data Base
< % conn= Server. CreateObject("ADODB.Connection")
Connection <br>conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn . Open ("C:/webData/ nor.mdb")
Connection بلوک rs=Server. CreateObject ("ADODB.Recordset")
Record Set % بلوک rs. Open("Customer", conn) Record Set >
```

در این مثال تمام محتویات جدول Customer به rs ریخته میشود.

حال میخواهیم در انتخاب محتویاتی از Customer که میخواهند به rs انتقال داده شوند از دستور SQL استفاده کنیم.

ساختن یک Connection و Record Set و به کار بردن SQL

```
< % set conn=Server. CreateObject ("ADODB .Connection"
conn. Provider="Microsoft.jet.OLEDB.4.0" conn.
Open ( "C:/Inetpub/wwwroot/nor.mdb") (Record Set بجد)
set rs= Server. CreateObject <br>("ADODB.Recordset")
(Record Set بلزگرس) rs. Open("select * From customer ", conn)
```

گرفتن داده هایی مورد نیاز از Record Set

بعد از اینکه Record Set را باز کردیم میتوان به داده های مورد نیاز دسترسی یافت .

مثال :

دسترسی به فیلد name از جدول (rs rs) name

چاپ داده ها

مثال : چاپ فیلد name از جدول

```
rs Response.write(rs(name))
```

بستن Connection و Recordset

- برای بستن Recordset

rs.close () - برای بستن Connection

conn.close ()

منبع : سایت اینترنتی <http://www.etvto.ir/ostadonline/www.w3schools.com>