

Scrivere pagine con tecnologia [ASP](#) (Active Server Page) per leggere FORM (schede che introducono interattività nelle pagine Web):

- **Prerequisito:** possedere accesso a sito per memorizzare ed eseguire pagine ASP
- **Scopo:** realizzare moduli HTML per l'invio dei dati attraverso metodo GET e pagine ASP in grado di *riceverli* e *interpretarli* (per maneggiarli useremo linguaggi di scripting: [JavaScript](#) o Jscript cioè la versione microsoft o ancora VBScript) al fine di produrre un output dimostrativo delle potenzialità degli **oggetti Request** e **Response**.

Primo esercizio:

- Creare un [form](#) (scheda) di inserimento dati salvando il documento con nome **insdat.htm** che si vuole con il seguente layout:

Inserimento dati

Inserisci il tuo cognome:

Inserisci il tuo nome:

Scrivere nel file con nome **reg.asp** il programma che elabora i dati della scheda visualizzando sulla finestra del browser un messaggio di benvenuto personalizzando nome e cognome

[Soluzione](#)

Secondo esercizio:

- Creare un [form](#) (scheda) di inserimento dati salvando il documento con nome **insdat1.htm** che si vuole con il seguente layout:

Inserimento dati

Inserisci il tuo cognome:

Inserisci il tuo nome:

Inserisci la tua età:

Inserisci il tuo INDIRIZZO:

- Scrivere nel file con nome **reg1.asp** il programma che elabora i dati della scheda : un messaggio di benvenuto personalizzando nome e cognome aggiungendo informazioni sull'età e l'indirizzo

[Soluzione](#)

Soluzione **primo esercizio** :

Il file **insdat.htm** che, *per test*, si posiziona nella medesima cartella della pagina ASP lato Server:

```
<HTML>
<HEAD><TITLE>Scheda</TITLE></HEAD>
<BODY>
  <H2>Inserimento dati</H2>
  <FORM METHOD="GET" ACTION ="reg.asp">
  <P>Inserisci il tuo cognome: <INPUT TYPE="TEXT" NAME = "Cognome"></P>
  <P>Inserisci il tuo nome: <INPUT TYPE="TEXT" NAME = "Nome"></P>
  <P><INPUT TYPE = "SUBMIT" VALUE = "Invio"></P>
  <P><INPUT TYPE = "RESET"></P>
  </FORM>
</BODY>
</HTML>
```

Si modificherà tale file **insdat.htm** (quando memorizzato lato Client), impostando l'URL completo come valore dell'attributo ACTION; es. ACTION = "http://UserID.somee.com/reg.asp"

Il file **reg.asp** cioè la pagina ASP in grado di ricevere i dati è memorizzato lato Server:

```
<HTML>
<HEAD><TITLE>Tecnologia ActiveServerPage</TITLE></HEAD>
<BODY>
  <H2>Benvenuto</H2>
  <BR>
  <% rem uso di oggetto Request e metodo Querystring
    rem per estrarre il valore che il metodo GET memorizza in variabile di ambiente
    rem appendendola all'URL (Uniform Resource Locators)

    RESPONSE.WRITE ("<B>")
    RESPONSE.WRITE (REQUEST.QUERYSTRING ("Cognome"))
    RESPONSE.WRITE (REQUEST.QUERYSTRING ("Nome"))
    RESPONSE.WRITE ("</B>")
  %>
</BODY>
</HTML>
```

nb:

Il metodo Write dell'oggetto [Response](#) richiede una stringa di testo tra virgolette o una funzione che restituisca una stringa.

I commenti sono preceduti su ogni riga da *rem* oppure dal *carattere apice*

tag <%= visualizza il contenuto della variabile ad esempio <%= variabile%> su unica riga
è infatti la forma contratta di <% RESPONSE.WRITE variabile %>

“ ”
_ per inserire su più righe un'istruzione
ad es: <% RESPONSE.WRITE strigaInizio_
strigaProseguo %>

Soluzione **secondo esercizio**:

Il file **insdat1.htm** memorizzato **lato client**

```
<HTML>
<HEAD>
<TITLE>Scheda</TITLE>
</HEAD>
<BODY>
  <H2>Inserimento dati</H2>

  <FORM METHOD="GET" ACTION = " http://UserID.somee.com/reg1.asp">

  <P>Inserisci il tuo cognome: <INPUT TYPE="TEXT" NAME = "Cognome"></P>
  <P>Inserisci il tuo nome: <INPUT TYPE="TEXT" NAME = "Nome"></P>
  <P>Inserisci la tua eta': <INPUT TYPE="TEXT" NAME = "Anni" SIZE = "3"
                          MAXLENGTH = "3"></P>
  <P>Inserisci il tuo INDIRIZZO: <INPUT TYPE="TEXT" NAME = "Indirizzo"
                          SIZE = "80"></P>

  <P><INPUT TYPE = "SUBMIT" VALUE = "Invio"></P>
  <P><INPUT TYPE = "RESET"></P>
  </FORM>
</BODY>
</HTML>
```

Il file **reg1.asp** memorizzato **lato Server**

```
<HTML>
<HEAD>
<TITLE>PROGRAMMA ActiveServerPage: formato di pagina su Web Server microSoft</TITLE>
</HEAD>
<BODY>
<H2>Benvenuto/a</H2>
<BR>
<% ' lettura da form          (uso di apice per commento)
  rem che acquisisce dati  (uso di rem per commento)
  RESPONSE.WRITE ("<B>")
  RESPONSE.WRITE (REQUEST.QUERYSTRING ("Cognome") & " " &_
                  REQUEST.QUERYSTRING ("Nome"))

  RESPONSE.WRITE ("<BR>")
  RESPONSE.WRITE ("La tua eta' e' " & REQUEST.QUERYSTRING ("Anni") & " anni ")
  RESPONSE.WRITE ("<BR>")
  RESPONSE.WRITE ("Il tuo indirizzo e' " & REQUEST.QUERYSTRING ("Indirizzo"))
  RESPONSE.WRITE ("</B>")
%>

  <!-- il simbolo & serve per concatenare stringhe-->

</BODY>
</HTML>
```

Riepilogo **tecnologia ASP** (ambiente microsoft):

commenti: uso di apice oppure uso di rem

inizio/fine: <%%>

per **concatenare:** &

per scrivere in **risposta:** RESPONSE.WRITE ("stringa") dove response è un oggetto e write un metodo per scrivere a video

per scrivere la data corrente: RESPONSE.WRITE Date() con uso di funzione che restituisce una stringa

per **estrarre il valore** in risposta ad un metodo **GET:**

```
REQUEST.QUERYSTRING ("NAMEvalore")
```

per **estrarre il valore** in risposta ad un metodo **POST:**

```
REQUEST.FORM ("NAMEvalore")
```

per **assegnare** il valore NAME estratto con metodo POST, ad una variabile:

```
variabile = request.form("NAMEvalore")
```

Scegliendo il **metodo post**, i dati contenuti nel modulo non sono appesi all'URL ma impacchettati a creare un blocco di informazioni e spediti (con uno o più messaggi e-mail) al programma riferito nell'attributo ACTION che li elaborerà.

Il **metodo ServerVariables** permette di richiedere al server una delle [variabili di sistema](#) come ad esempio HTTP_USER_AGENT che identifica il nome del navigatore che il client sta usando in quel momento.

Es: <% var = Request.ServerVariables ("HTTP_USER_AGENT")%>

Per includere anche script nel codice asp e utilizzare così funzioni create, ad esempio, in javascript o vbscript, si richiamano tramite il **comando nativo** <% **Call** nomefunzione %>

[Esempio](#)

Non è possibile definire una funzione senza ricorrere a linguaggi di scripting, dato che **non è possibile assegnare nomi a blocchi di codice asp**: l'[ASP](#), non è in grado di definire funzioni o procedure.

Per **importare** alcune funzioni si può usare il **comando** <!-- **#include** -->

Infatti, per inserire in un file asp, il contenuto di un file esterno che può essere di testo, html, asp, grafica o qualsiasi altro file presente sul server la sintassi è la seguente:

```
<!-- #include file="nomefile.est" -->
```

[Esempio](#)

e va usata sempre al di fuori dei tag <%,%>, che delimitano gli script ASP.

Pagine ASP con linguaggio JavaScript

- **Scopo:** realizzare moduli HTML per l'invio dei dati attraverso metodo GET e pagine ASP in grado di riceverli, interpretarli e **maneggiarli** con uso di linguaggi di scripting: JavaScript o Jscript cioè la versione microsoft al fine di produrre un output dimostrativo delle potenzialità dell'oggetto Request.
- Scelta del linguaggio **a livello di file**

```
<%@ LANGUAGE = JavaScript %> <!-- necessario blank tra @ e LANGUAGE -->
```

Esercizio:

In risposta ad un **modulo** che invia come dati il nome (**nome**), il cognome (**cognome**) ed il sesso (**sesso**), si visualizza sulla finestra del browser un messaggio di benvenuto personalizzato in modo diverso a seconda del sesso

```
<%@ LANGUAGE = JavaScript %> <!-- Scelta del linguaggio a livello di file -->
<%
    // Recupero dei dati del modulo ..... questo è un commento in javascript

var nome = Request.QueryString("nome");      // var per definire variabili senza indicarne il tipo

    // una variabile può contenere diversi tipi di dati (numeri, stringhe, ecc.)
var cognome = Request.QueryString("cognome");
var sesso = Request.QueryString("sesso");
var dizione;

if (sesso="F")
    dizione = "una femmina"
else
    dizione = "un maschio"
%>

<html><head><title>Esercitazione sull'oggetto Request</title></head>
<body>

    <b>Grazie per aver spedito il modulo!</b><br ><br >
    Ecco i dati che mi hai inviato:<br><br>
    Nome: <b><%=nome%></b><br >
    Cognome: <b><%=cognome%></b><br><br>
    Stando a quanto dici dovresti essere <%=dizione%><br><br>

</body>
</html>
```

- Invece di scegliere il linguaggio a livello di file si può usare il **tag** :

`<SCRIPT LANGUAGE = RUNAT => CODICE di script </SCRIPT>`

E' *preferibile inserire nell'intestazione* (HEAD) il codice di script come commento se si usa tale tag.

Ad esempio: `<HEAD><SCRIPT> <!--CODICE di script1 --></SCRIPT> </HEAD>`

Se si omette la scritta RUNAT=Server, il motore delle ASP compila le stesse sulla macchina client. Si possono anche specificare sezioni con diversi tipi di linguaggio

Language=[VBScript|JavaScript].

Se non si specifica il linguaggio, quello di **default** è **quello che viene usato sul Web Server**. Si veda un [esempio](#).

Esercizio:

In risposta ad un **modulo** che invia dati, si visualizza sulla finestra del browser la data corrente nella forma **gg/mm/aa** (ad esempio 6/12/2009) facendo uso dei **metodi getDate(), getMonth() e getYear()**

```
<html>
```

```
<head><title> Pagina ASP con linguaggio JavaScript </title>
```

```
<SCRIPT LANGUAGE=Javascript RUNAT=Server> // anche LANGUAGE = Jscript
<!-- nasconde lo script per i vecchi browser (attenzione: può nascondere il codice)
```

```
function PrintDate() // definizione della funzione utente di nome PrintDate
```

```
{
```

```
    var x
```

```
    x = new Date() // crea una nuova istanza dell'oggetto nativo Date2
```

```
    Response.Write(x.getDate() + "/" + (x.getMonth()+1) + "/" + x.getYear())
```

```
}
```

```
// il simbolo + serve per concatenare
```

```
//-->
```

```
</SCRIPT>
```

```
</head>
```

```
<body>
```

```
    <% Call PrintDate %> <!-- richiamo di funzione tramite comando nativo -->
```

```
</body>
```

```
</html>
```

NB: si poteva usare in alternativa il **metodo toString()** per scrivere la data sul navigatore:

```
Response.Write(x.toString()) // scrive la data sul navigatore nella forma gg/mm/aa
```

¹ In alcuni web hosting free non inserire il codice a commento

² Anche tra le **funzioni inerenti** alla **data** in **VBScript** esiste **Date()** che è una funzione del linguaggio che restituisce la data corrente; altre funzioni sono:

Time() Permette di ottenere l'ora corrente

DateSerial(anno, mese, giorno) Compone e restituisce la data in base al passaggio dei tre parametri.

Per esempio: `<%= DateSerial(99, 4, 23) %>` restituirà 23/4/99. Al posto dei numeri possiamo mettere anche della variabili, oppure delle differenze numeriche.

TimeSerial(ora, minuti, secondi) Funziona in modo identico alla funzione DateSerial, comprese le differenze numeriche, solo che dà come risultato un orario.

Timer() Restituisce i secondi trascorsi dalla mezzanotte.

Esercizio: Scriptj1b.asp identico al precedente ma senza i tag di struttura del documento

```
// scrive la data corrente ad esempio 6/12/2009 con uso tag script  
// con metodi getDate(), getMonth() e getYear()
```

```
<% Call PrintDate %>  
<SCRIPT LANGUAGE=Javascript RUNAT=Server>  
function PrintDate()  
{  
    var x = new Date()           // x è un'istanza della classe Date: un'oggetto che occupa memoria  
    Response.Write(x.getDate()+ "/" + (x.getMonth()+1) + "/" + x.getYear())  
                                // il simbolo + per concatenare  
}  
</SCRIPT><!-- notare la possibile assenza dei tag di struttura del documento -->
```

file di prova **lato client**:

```
<HTML>  
<HEAD><TITLE>Uso di JavaScript</TITLE></HEAD>  
<BODY>  
    <FORM METHOD = "GET" ACTION="http://UserID.somee.com/Scriptj1b.asp" >  
        <INPUT TYPE="SUBMIT">  
    </FORM>  
</BODY>  
</HTML>
```

Esercizio: Scriptj2b.asp (con scelta del linguaggio a livello di file e uso dell'istruzione **for**)

Con scelta del linguaggio **a livello di file**, si realizzi un ciclo che visualizza il numero di riga con range tra 1 e 10.

```
<%@ LANGUAGE = JavaScript %>  
<html>  
<head><title>Esempio di pagina ASP</title></head>  
<body>  
<%  
  
for (i=1;i<=max;i++) { %>           // sintassi C like  
  
<b>Riga numero <%=i%></b>  
<br />  
  
<% } %>  
</body>  
</html>
```


Esempio con script sia in linguaggio JavaScript sia VBScript

```
<html>
<head>
<title>Tabella e data</title>
```

```
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
```

```
<!-- per browser che non interpretano gli script
```

```
Sub Echo
```

```
Response.Write "<tr><td> Name </td><td>Value </td></tr> "
```

```
Set Params = Request.QueryString ' L'istruzione Set imposta la variabile su server
```

```
' il For Each itera per ogni stringa inserita nell'URL dopo il punto ?
```

```
' es. http://UserID.somee.com/es2.asp?pippo=3
```

```
' p assume solo il valore pippo e Params(p) prende il parametro 3
```

Name	Value
------	-------

pippo	3
-------	---

pluto	4
-------	---

paperino	5
----------	---

```
For Each p in Params
```

```
Response.Write " <tr><td>" & p & "</td><td> " & Params(p) & "</TD></TR> "
```

```
Next
```

```
End Sub
```

```
-->
```

```
</SCRIPT>
```

```
<SCRIPT LANGUAGE=JavaScript RUNAT=Server>
```

```
<!-- per browser che non interpretano gli script
```

```
function PrintDate()
```

```
{ var x
```

```
  x = new Date()
```

```
  Response.Write(x.toString())
```

```
}
```

```
//-->
```

```
</SCRIPT>
```

```
</head>
```

```
<body>
```

```
<table>
```

```
<!-- la table non ha i tag td e tr ecc. ecc. essi saranno definiti dalla subroutine Echo definita dallo
script in vbscript cioè il comando Call Echo chiamerà la subroutine e restituirà i valori che
compileranno la tabella -->
```

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

```
</table>
```

<!-- stesso discorso soltanto che PrintDate restituisce soltanto il valore della data corrente perchè chiama la funzione definita nello script JavaScript -->

<% Call PrintDate %>

</body>

</html>

<!-- effetto su browser :

se <http://UserID.somee.com/es2.asp>

Name Value

Mon Dec 2 11:39:58 EST 2002

se <http://UserID.somee.com/es2.asp?paola=2>

Name Value

paola 2

Mon Dec 2 11:44:25 EST 2002

<http://UserID.somee.com/es2.asp?paola=2 & marco=3>

Name Value

paola 2

marco 3

Mon Dec 2 11:48:58 EST 2002

-->

Un esempio di censimento

censimento

benvenuti. vi preghiamo di compilare la scheda seguente

per inoltrare la scheda usare il pulsante **inoltre**

nome:

sex: maschile femminile

interessi (selezionare le voci interessate):

musica

sport

tecnologie

giochi

possibile [soluzione](#) sia con JavaScript sia con VBScript

Esercizio:

Con scelta del linguaggio JavaScript nell'**intestazione** (usando il tag SCRIPT), si proponga all'utente la possibilità di impostare una homepage predefinita come pagina di apertura (da caricare all'aprirsi) del browser all'atto della connessione o di aggiungerla tra i preferiti.

Soluzione: vedi **Imposta il sito come Home Page**

Tutorial scritto il 30/04/2003 da **Luca Ruggiero**

Sito Web: <http://www.lukeonweb.net>

Soluzione più elegante:

http://www.webmasterpoint.org/html/trucco_02.asp?id=1313&voto=3&cat=guida&dat=5.43.20

Trucchi di HTML: Preferiti e home page predefinita

Guida al linguaggio Html:

la base della scrittura di un sito web a cura di [Lorenzo Pascucci](#)

Esercizio:

Con scelta del linguaggio JavaScript nell'**intestazione** (usando il tag SCRIPT), si proponga all'utente la possibilità di definire il colore del testo di una pagina o diversificarne lo stile.

Si crei, pertanto, da JavaScript un foglio di stile CSS usando la **proprietà style dell'oggetto document** (compatibilità IE)

Soluzione: http://www.mrwebmaster.it/fareweb/javascript/guida_javascript/lezione18.htm

Esercizio:

Con scelta del linguaggio JavaScript, si proponga all'utente la possibilità di utilizzare uno tra diversi motori di ricerca.

[Soluzione](#)

Esercizio:

Con scelta del linguaggio JavaScript, si proponga all'utente la possibilità di inviare un messaggio di posta ad uno tra diversi destinatari.

[Soluzione](#)

Esercizio:

Con tecnologia ASP e scelta del linguaggio JavaScript spedire al browser dell'utente un foglio elettronico (applicazione EXCEL) elaborato al volo.

[Soluzione](#)

Esercizio:

Con tecnologia ASP e scelta del linguaggio JavaScript spedire al browser dell'utente un documento (applicazione WORD) elaborato al volo.

[Soluzione](#)

Esercizio:

Con tecnologia ASP e scelta del linguaggio JavaScript si progetti un'applicazione che presenti un'interfaccia amichevole per graficare funzioni.

Soluzione: vedi la possibilità di graficare **funzioni reali** con JavaScript

da home di Carlo Ciaramella <http://www.ciaramella.net>

Vedi anche, per [grafici a barre](#)

Soluzione con tratto spesso (IE non riesce a gestire porzioni molto piccole nell'uso di **span**):

```
<html>
<head><title> Grafico: funzione AM</title>

<SCRIPT language="JavaScript">
<!--

function grafico()
{
  scala= 10;          // zoom ascisse

  dx= 0.01;
  nT= 10;             // max 10 se dx =0,01

  min = 0;           // ascisse
  max= 2*Math.PI*nT;

  y0= 250;           // offset per visualizzazione centro schermo

  A0 = 100;          // ampiezza

  for (x=min; x<max; x= x+dx)
  {
    y= A0*(1 + Math.sin(x/20))* Math.sin(x) +y0;

    document.write("<span style='background-color:black;position:absolute; left:"+x*scala +"
    top:"+y+"; width:1px; height:1px;'>&nbsp;</span>");

  }
}

//-->
</SCRIPT>
</HEAD>
<BODY>

<a href="#" onClick="grafico();return true;">Grafica</a> segnale AM
</BODY>
</HTML>
```

Soluzione con uso di immagine (punto nero):

```
<html>
<head><title> Grafico: funzione AM</title>

<SCRIPT language="JavaScript">
<!--

function grafico()
{
  scala= 10;      // zoom ascisse
  dx= 0.01;
  nT= 10;        // max 10 se dx =0,01
  min = 0;       // ascisse
  max= 2*Math.PI*nT;

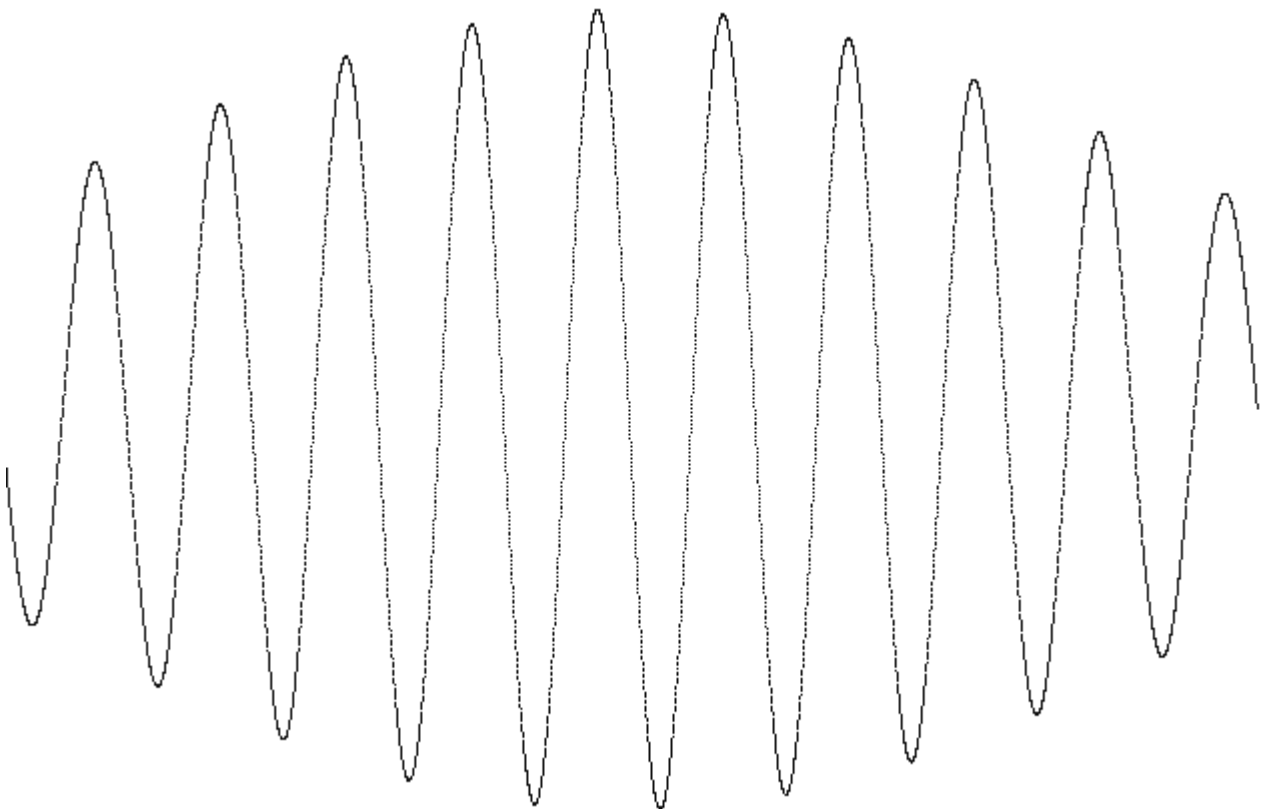
  y0= 250;      // offset per visualizzazione centro schermo
  A0 = 100;     // ampiezza

  for (x=min; x<max; x= x+dx)
  {
    y= A0*(1 + Math.sin(x/20))* Math.sin(x)+y0;

    document.write("<DIV style='background-color:white;
                    position:absolute; left:"+x*scala+"; top:"+y+";
                    width:1px; height:1px;
                    <img SRC='black.gif' border=0 width=1 height=1>
                    </DIV>");

  }
}

//-->
</SCRIPT>
</HEAD>
<BODY>
<a href="#" onClick="grafico();return true;">Grafica</a> segnale AM
</BODY>
</HTML>
```



Interazione ASP – Excel

```
<%@ LANGUAGE = JavaScript%>
<%
    Response.Expires=0 // la pagina scadrà dopo 0 minuti
                        // serve a non memorizzare la pagina all'interno della cache del Browser

    Response.buffer = true // bufferizzare il contenuto della pagina
                          // (cioè spedire al browser dell'utente la pagina solo dopo che è
                          // terminata l'elaborazione da parte del server)

    Response.ContentType = "application/vnd.ms-excel"
                          // Dichiarazione del tipo di documento scelto (excel)
    Response.AddHeader ("content-disposition", "inline; filename=esempioJ.xls")

    Response.Write("<table width=240>")
    Response.Write("<tr>")
    for (i = 1; i<= 5; i++)
    {
        Response.Write("<td width=40>")
        Response.Write(i)
        Response.Write("</td>")
    }
    Response.Write("<td width=40><b>=Somma(A1:E1)</b></td>")
    Response.Write("</tr>")
    Response.Write("</table>")

    Response.flush // metodo Flush per inviare all'utente il contenuto del buffer
    Response.end // metodo End per interrompere l'esecuzione della pagina.
%>
```

Con possibile test realizzato eseguendo sul client il file.htm seguente:

```
<HTML>
<HEAD><TITLE>Apri foglio Excel</TITLE></HEAD>
<BODY>
<H2>Visualizza foglio Excel</H2>
<FORM METHOD="GET" ACTION ="http://UserID.somee.com/excel_JS.asp">
<P><INPUT TYPE = "SUBMIT" VALUE = "Invio"></P>

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

Interazione ASP – Word

```
<%@LANGUAGE = JavaScript%>
<%
    Response.Buffer = true; // Attivazione del buffer

                                // Dichiarazione del tipo di documento scelto
    Response.ContentType = "application/msword";

                                // Scelta del file su cui scrivere
    Response.AddHeader("content-disposition", "inline; filename=stampa.doc");

    // Scrittura di un testo

    // Genero il file stampa.rtf oppure stampa.doc non fisicamente sul Server
    // ma conservato in una locazione di memoria virtuale e muore appena l'utente lo chiude,
    // ad esempio dopo aver eseguito la stampa.
    // Questo particolare processo evita problemi di concorrenzialità negli accessi.

    Response.Write("Hello Word :-");

    Response.Flush; // Invio dell'output bufferizzato
    Response.End;   // Chiusura del documento
%>
```

Con possibile test realizzato eseguendo sul client il file.htm seguente:

```
<HTML>
<HEAD><TITLE>Apri documento</TITLE></HEAD>
<BODY>
<H2>Visualizza documento</H2>
<FORM METHOD="GET" ACTION ="http://UserID.somee.com/stampa\_word.asp">

<P><INPUT TYPE = "SUBMIT" VALUE = "Invio"></P>

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

Per **inserire** nella propria **home page** un **motore di ricerca**

Il metodo più semplice è copiare ed usare la QueryString di ricerca tipica di un motore

```
<html>
<head>
<title>Esempi di ricerca su Google</title>

<script language="JavaScript">
<!--
function Ricerca() {

    var termine = document.ricerca.cerca.value;    // document.nome_form.nome_input.value

    location.href = "http://www.google.it/search?q=" + termine + "&ie=ISO-8859-1&hl=it&lr=";

    // la proprietà href serve per impostare una stringa che specifica l'URL interamente.
}
-->
</script>
</head>

<body>
<div align="center">                                // solo al click su bottone attiva la ricerca

    <form name="ricerca">
        <b>Cerca su Google</b>
        <input type="text" name="cerca" size="30">
        <input type="button" value="Cerca" onClick="Ricerca()">
    </form>

<!-- se si preme il tasto INVIO file:///DRIVER:/percorso/motoreI.htm?cerca=termine_digitato -->
</div>

<div align="center">    // attiva la ricerca anche su pressione di INVIO in casella di testo

    <form action="http://www.google.com/search" method="get">
        <b>Ricerca con Google</b>
        <input type="text" value="" name="q" size=40 maxlength=256>&nbsp;
        <input type="submit" value="Google">&nbsp;&nbsp;
    </form>

</div>
</body>
</html>
```


Per **inserire** nella propria **home page** la ricerca di termini con **più motori**

```
<html>
<head>
<title>Esempi di ricerca su motori</title>

<script language="JavaScript">
<!--
function Ricerca() {

var termine = document.ricerca.cerca.value;
var su = document.ricerca.su.value;

//Effettua la ricerca su Google

if (su == "google") {
location.href = "http://www.google.it/search?q=" + termine + "&ie=ISO-8859-1&hl=it&lr=";
}

//Effettua la ricerca su Yahoo
if (su == "yahoo") {
location.href = "http://it.search.yahoo.com/search/it?p=" + termine;
}

//Effettua la ricerca su MSN
if (su == "msn") {
location.href = "http://search.msn.com/results.asp?RS=CHECKED&FORM=MSNH&v=1&q=" +
termine;

/* oppure
"http://search.msn.it/results.aspx?cp=1252&PI=9547&DI=120&FORM=MSNH&q=" + termine;
*/
}

//Effettua la ricerca su Altavista
if (su == "altavista") {
location.href = "http://it.altavista.com/q?pg=q&q=" + termine + "&kl=XX&what=it";
}

//Effettua la ricerca su Arianna
if (su == "arianna") {
location.href = "http://arianna.libero.it/search/abin/search?query=" + termine;

/* oppure "http://arianna.libero.it/search/abin/search?query="+ termine + "&radiobutton=arianna";
*/
}

//Effettua la ricerca su Virgilio
if (su == "virgilio") {
location.href = "http://search.virgilio.it/search/cgi/search.cgi?vrs=0&db=v&op=and&q=" +
termine + "&l=XX";
}

//Effettua la ricerca su Il Trovatore
if (su == "trovatore") {
location.href = "http://search.iltrovatore.it/cgi-bin/index.cgi?q=" + termine + "&cs=iso-8859-1";
}
}
```

```

//Effettua la ricerca su All the web
if (su == "web") {
    location.href = "http://www.alltheweb.com/search?cat=web&cs=utf8&q=" + termine +
        "&_sb_lang=pref";
}

//Effettua la ricerca su Hot Bot
if (su == "hot") {
    location.href = "http://www.hotbot.com/default.asp?query=" + termine +
        "&ps=&loc=searchbox&tab=web&provKey=Inktomi";
}

//Effettua la ricerca su EXite
if (su == "ex") {
    location.href = "http://www.excite.it/search/web/results?offset=0&q=" + termine + "&l=";
}

//Effettua la ricerca su Lycos
if (su == "lycos") {
    location.href = "http://cerca.lycos.it/cgi-bin/pursuit?query=" + termine +
        "&NDB=asp_lycos&id_window=120&tld=all&family=off&cat=it";
}

//Effettua la ricerca su looksmart
if (su == "look") {
    location.href = "http://www.looksmart.com/r_search?key=" + termine;
}

//Effettua la ricerca su WiseNut
if (su == "wise") {
    location.href = "http://www.wisenut.com/search/query.dll?q=" + termine + "&p=0";
}

//Effettua la ricerca su About.com
if (su == "about") {
    location.href = "http://search.about.com/fullsearch.htm?terms=" + termine;
}

//Effettua la ricerca su Godado
if (su == "godado") {
    location.href = "http://www.godado.it/godado.pperl/?Keywords=" + termine;
}

//Effettua la ricerca su HTML.it
if (su == "html") {
    location.href = "http://cerca.html.it/cgi-bin/cerca.cgi?gr=off&q=" + termine;
}

//Effettua la ricerca su dizionario informatico
if (su == "diz") {
    location.href = "http://www.dizionarioinformatico.com/cgi-lib/diz.cgi?name=" + termine;
}
}
//-->
</script>
</head>

```

```
<body>
<div align="center">
  <form name="ricerca">
    <b>Cerca su </b>
    <select name="su">

      <option value="google">Google</option>
      <option value="yahoo">Yahoo</option>
      <option value="msn">MSN</option>
      <option value="altavista">Altavista</option>
      <option value="arianna">Arianna</option>
      <option value="virgilio">Virgilio</option>
      <option value="trovatore">Il Trovatore</option>
      <option value="web">All the Web</option>
      <option value="hot">Hot Bot</option>
      <option value="ex">eXite</option>
      <option value="lycos">Lycos</option>
      <option value="look">Looksmart</option>
      <option value="wise">WiseNut</option>
      <option value="about">About.com</option>
      <option value="godado">Godado</option>
      <option value="html">HTML.it</option>
      <option value="diz">dizionario informatico</option>
    </select>
    <b>il termine</b>
    <input type="text" name="cerca" size="30">
    <input type="button" value="Cerca" onClick="Ricerca()">
  </form>
</div>

</body>
</html>
```

Email multipla

Script per e-mail multipla

Scegli il destinatario (Invia la mail a tutti i destinatari)

Per altri esercizi:
consulta il [sito della docente](#)

- pbiasotti@libero.it
- stclassi@yahoo.it
- mavirapetto@libero.it

```
<html>
<head> <title>email multipla</title>
<style>
<!--
  body{color: #192939; font-family:Verdana}
  div {text-align:center; font-weight:bold}
  p {font-style:italic}
  a:link{color:#3366CC}
  a:active{color:#FF0000}
  a:visited {color:#3366CC}
  a:hover{font-weight:bold}
-->
</style>
<script language="javascript">
<!--
function EmailMultipla() {

  var selezione = document.formEmail.opzione.checked;
  var destinatario = document.formEmail.destinatari.selectedIndex;
  if (selezione == true) {
    location.href =
      "mailto:pbiasotti@libero.it?Subject=nome_oggetto&Cc=stclassi@yahoo.it&Bcc=paolabiasotti@libero.it";
  }
  else {
    switch(destinatario) {
      case 0: location.href = "mailto:pbiasotti@libero.it"; break;
      case 1: location.href = "mailto:stclassi@yahoo.it"; break;
      case 2: location.href = "mailto: mavirapetto@libero.it"; break;
    }
  }
}
//-->
</script>
</head>
<body>
<div>Script per e-mail multipla</div>

<form name="formEmail">
  Scegli il destinatario
  <select name="destinatari">
    <option>pbiasotti@libero.it</option>
    <option>stclassi@yahoo.it</option>
    <option>mavirapetto@libero.it</option>
  </select>
  (Invia la mail a tutti i destinatari <input type="checkbox" name="opzione">)
  <input type="button" value="Invia" onClick="EmailMultipla()">
</form>

<p>Per altri esercizi:<br>
  consulta il <a href="http://new345.altervistra.org">sito</a> della docente
</body>
</html>
```

Tecnologia ASP : sintassi e mezzi a disposizione per le applicazioni.

Che cosa sono le ASP ?

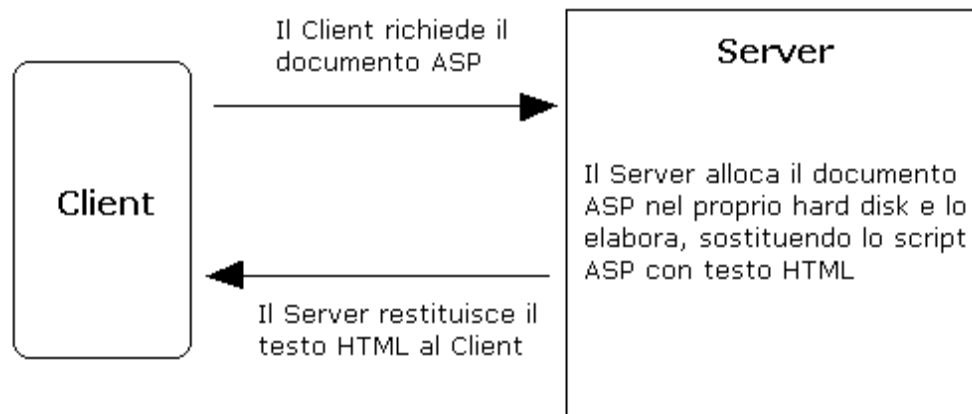
ASP significa Active Server Pages, cioè tradotto in italiano Pagine Attive sul Server. Quindi le ASP non sono altro che pagine ipertestuali interpretate dal Server.

Il funzionamento è il seguente:

- Un navigatore di Internet richiede una certa pagina al Server tramite il proprio browser
- Il Server recupera la pagina richiesta, interpreta il codice html normalmente, in presenza di codice ASP, lo elabora e restituisce in formato html il risultato;

Quando il Server termina l'elaborazione della pagina ASP, restituisce l'intero contenuto al browser del richiedente, che lo visualizza a video.

Interazione Client/Server per documenti ASP



Esaminandone "l'anatomia", possiamo dividere una pagina ASP in tre parti:

- 1) Testo
- 2) Marcatori HTML
- 3) Comandi script

In un documento con estensione **.asp** è consentito utilizzare variabili, cicli e istruzioni di controllo, grazie alla possibilità di richiamare la sintassi di un linguaggio di scripting, come ad esempio il VBscript e il JavaScript³, ma anche perl e rexx e altri.

La scelta del linguaggio dipende in primo luogo dalle necessità del programmatore e dal tipo di esecuzione che si vuole avere: se si vogliono eseguire gli script dal *lato server* in *ambiente microsoft* è preferibile utilizzare il VBscript, mentre se ci si vuole affidare alla potenza degli "scripting engine" (motore che interpreta i comandi dei linguaggi di scripting e li esegue) dei singoli navigatori è sicuramente meglio utilizzare il JavaScript, semplice ed efficace.

Il codice ASP è sempre delimitato da i due marcatori `<%` e `%>`. Ad esempio la seguente riga:

```
<% x="ciao" %>
```

assegna alla variabile x la stringa "ciao".

³ Nato come LiveScript (Netscape anni '95, contemporaneo al linguaggio Java della Sun) poi JavaScript (Netscape e Sun) proposto come Jscript in ambiente microsoft.

Una pagina può essere costituita solamente da codice ASP, oppure avere ASP immerso nel codice HTML: in entrambi i casi l'estensione deve essere .asp.

All'interno della pagina è possibile sempre **inserire un codice in un linguaggio diverso** da ASP: in questo caso deve però essere dichiarato o a livello di file o dove tale codice va eseguito.

Supponendo allora di voler aggiungere uno script in JavaScript, la giusta sintassi è la seguente:

```
<SCRIPT LANGUAGE = JavaScript RUNAT= [server oppure client]>
```

Ciò si rende necessario ogni qualvolta si ha bisogno di **funzioni o procedure**, dato che **ASP, non è in grado di definirle**.

<p>Contenuto della pagina ASP richiesta <i>nome.asp</i> con uso di JavaScript</p> <pre><%@ LANGUAGE = JavaScript %> <% for (i = 3; i<=7; i++) {%> <P style="font-size:<%=i%>em">Hello World!</P> <%}%></pre> <p>Nb: si noti l'uso di :<%= <i>variabile</i>%> forma contratta di <% RESPONSE.WRITE <i>variabile</i> %></p>	<p>Cosa appare nel sorgente della pagina restituita dal Server:</p> <pre><P style="font-size:3em">Hello World!</P> <P style="font-size:4em">Hello World!</P> <P style="font-size:5em">Hello World!</P> <P style="font-size:6em">Hello World!</P> <P style="font-size:7em">Hello World!</P></pre>
--	--

Come si può notare neanche una riga dello script è restituita al browser, ma solo la sua elaborazione.

Per test <http://5ai.somee.com/es3j.asp>

oppure con uso di **VbScript** di default lato Server - hosting **somee** <http://5ai.somee.com/es3.asp>

Una volta definita una funzione in un qualsiasi linguaggio di script, per richiamarla basta usare l'istruzione **call**. La sintassi di tale **comando nativo** è:

```
<% Call nomefunzione %>
```

Oppure per **importare** alcune funzioni si può usare il **comando #include**. Per inserire in un file asp, il contenuto di un file esterno che può essere di testo, html, asp, grafica o qualsiasi altro file presente sul server la sintassi è la seguente:

```
<!-- #include file="nomefile.est" - -->
```

e va usata sempre al di fuori dei tag <%,%>, che delimitano gli script ASP.