

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** (da memorizzare **lato SERVER**) il programma che elabora i dati della scheda visualizzando sulla finestra del browser un messaggio di benvenuto, personalizzando nome e cognome.

Soluzione

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*"

```
<input type="text" name = "NAMEvalore">
```

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>
  <% ' lettura da form      (uso di apice per commento)
    rem che acquisisce dati (uso di rem per commento)

    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 Locator)

    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.

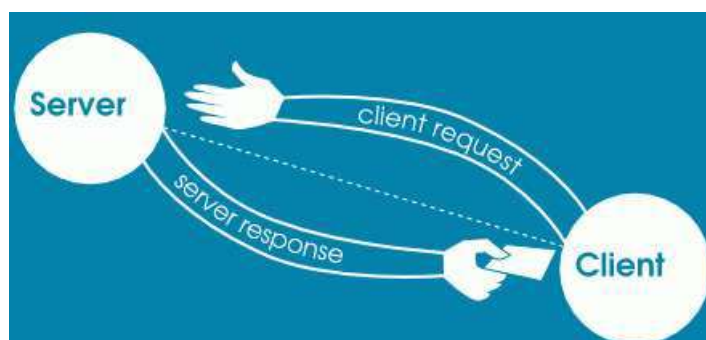
I **commenti** sono preceduti su ogni riga da **rem** oppure dal *carattere apice*

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

il simbolo **&** serve per **concatenare** stringhe

" " per inserire su più righe un'istruzione
ad es: **<% RESPONSE.WRITE strigaInizio_**
strigaProseguo %>

```
<% RESPONSE.WRITE (REQUEST.QUERYSTRING ("Cognome") & " " &_
REQUEST.QUERYSTRING ("Nome")) %>
```



Altro esercizio:

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

Compila ed invia il seguente modulo:

Introduci il tuo nome:

Introduci il tuo cognome:

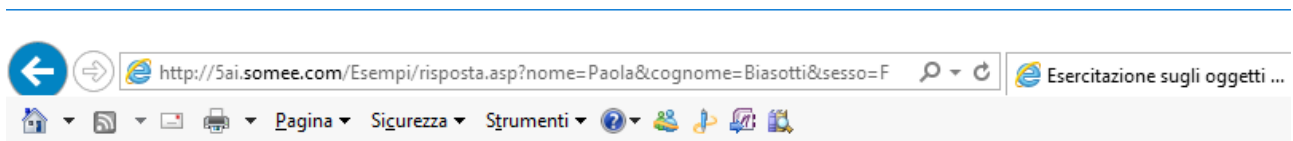
Specifica il tuo sesso:

M F

Invia il modulo

Scrivere nel file con nome **risposta.asp** (da memorizzare **lato SERVER**) il programma che elabora i dati della scheda visualizzando sulla finestra del browser dati inviati ed altre informazioni comprensive dell'IP del client.

Esempio <http://5ai.somee.com/Esempi/domanda.htm> (modulo per eseguire ASP) per ottenere la risposta in figura (caso di IP dinamico):



GRAZIE PER AVER SPEDITO IL MODULO!

Ecco i dati che mi hai inviato:

Nome: **Paola**

Cognome: **Biasotti**

Stando a quanto dici dovresti essere una femminuccia

Inoltre aggiungo:

Nome di questa pagina: **/Esempi/risposta.asp**

Nome del server: **5ai.somee.com**

Tuo indirizzo IP: **82.56.95.70**

Ecco l'intera query string che mi hai passato: **nome=Paola&cognome=Biasotti&sesso=F**

[Web hosting by Somee.com](http://5ai.somee.com)

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

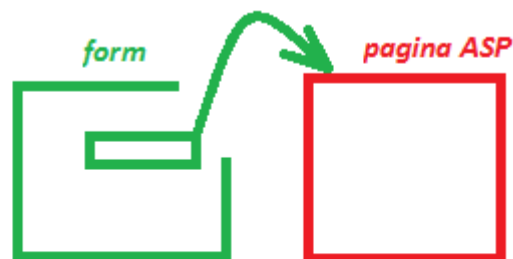
```
REQUEST.QUERYSTRING ("NAMEvalore")
```

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

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

per **assegnare** ad una variabile il valore (dell'attributo **name** nel form) estratto con metodo POST:

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



```
<input type="text" name = "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")%>

Non è possibile implementare strutture di controllo →

è necessario integrare linguaggi di script

È altrettanto impossibile 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.

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 variabli 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>
```

Esempio <http://5ai.somee.com/Esempi/lanciaJS.html> (modulo per eseguire ASP con JavaScript)
nel confronto <http://5ai.somee.com/Esempi/lancia.html> (modulo per eseguire ASP con VBScript)

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

Che cosa sono le ASP ?

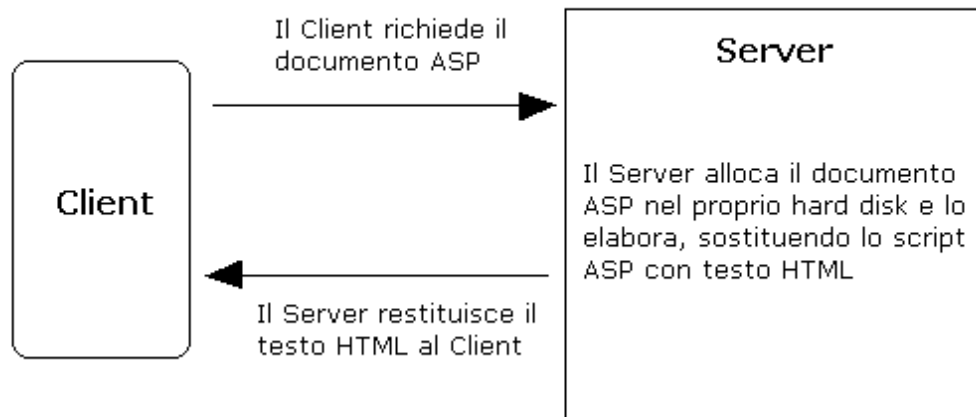
ASP significa Active Server Pages, cioè tradotto in italiano Pagine Attive sul Server. Quindi le ASP non sono altro che pagine ipertestuali interpretate dal Server.

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 JavaScript¹, 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".

¹ 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 ASP²: 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**.

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

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 %>
```

([esempi](#))

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.

([esempi](#))

² Lo scripting di default in ASP è VBScript (da https://www.w3schools.com/asp/asp_syntax.asp navigando tra [esempi](#))