

Lezione III: Oggetti ASP e interazione tramite form HTML

La terza lezione, come le precedenti, ha avuto una durata di due ore, di cui una in aula e l'altra in laboratorio, si è tenuta alla presenza della D.O.

Tramite lezione frontale ho introdotto gli Oggetti ASP. Le pagine ASP utilizzano i concetti di base della OOP, offrendo al programmatore la possibilità di definire le classi e gli oggetti.

Ad ogni oggetto sono associati *metodi* (procedure o funzioni)¹ e *proprietà* (attributi o caratteristiche dell'oggetto)².

A un livello elementare di implementazione di pagine ASP, si utilizzano gli *oggetti predefiniti* che risolvono le operazioni di uso più comune nelle applicazioni per il Web.

L'ambiente ASP ha 5 oggetti di uso comune: *REQUEST*, *RESPONSE*, *SESSION*, *APPLICATION*, *SERVER*. Esistono anche altri due oggetti, ma hanno un uso molto specifico.

A questo punto, ho ritenuto importante spiegare la differenza concettuale tra metodo POST³ e metodo GET⁴, nell'ora successiva di laboratorio i ragazzi implementeranno i due metodi e loro stessi noteranno la differenza.

L'oggetto *REQUEST* consente la lettura dei dati passati dal browser. Esso utilizza 5 *collection*, ognuna delle quali è un insieme di elementi della pagina inviata dal browser.

La sintassi è la seguente:

Request.[Collection/Proprietà/Metodo] (Variabile)

¹ L'oggetto *Form* ad esempio dispone dei metodi *Show* e *Hide* che rispettivamente visualizzano o nascondono un form all'utente.

² Ad esempio la proprietà *Name* indica il nome che identifica un oggetto all'interno del codice; le proprietà *Height* e *Width* specificano altezza e larghezza del controllo. Alcune proprietà possono essere di sola lettura, non possono cioè essere impostate dal programmatore nel corso di una routine.

³ Con il metodo POST i dati vengono inviati allo script lato server come flusso informativo separato, e ciò consente allo script del server di leggere una quantità indefinita di informazioni: i dati vengono inviati dal client al server Web nel corpo del messaggio HTTP.

⁴ Con il metodo GET i dati inviati sono aggiunti all'URL dello script (seguono il punto interrogativo). Una volta giunti al server, i dati vengono separati e assegnati a due variabili: l'URL di script (*SCRIPT_NAME*) e la stringa di dati (*QUERY_STRING*).

Le collezioni principali sono:

- *Form*: contiene i valori di un <FORM>:
per esempio, l'istruzione

```
<% NomeStudente = Request.Form ("studente")%>
```

asigna alla variabile *NomeStudente* il valore proveniente da una casella di testo, avente la proprietà NAME uguale a *studente*, contenuta nel form di una pagina HTML.

- *QueryString*: contiene i valori delle variabili in una interrogazione via HTTP.

L'oggetto *RESPONSE* gestisce l'output di ASP verso il browser.

Per esempio:

```
<% Response.Write "Ciao" %>  
<% Response.Write "<TABLE BORDER = 1>" %>
```

Entrambe le istruzioni inviano stringhe di caratteri che devono essere interpretate dal browser dell'utente: la prima istruzione visualizza un messaggio, la seconda contiene codice HTML per creare una tabella.

È possibile impostare la proprietà *BUFFER* per decidere se inviare direttamente l'output o memorizzarlo in un *buffer* temporaneo. Il metodo *FLUSH* scarica il *buffer*, mentre il metodo *END* termina l'output verso il browser.

L'oggetto *RESPONSE* ha una sua collection *Cookies*, tramite la quale si possono mandare cookie al browser.

L'oggetto *SESSION* consente di gestire lo stato di un utente che utilizza l'applicazione ASP. Ogni volta che un utente inizia una sessione (apre per la prima volta con il browser la pagina di un sito), viene creato un oggetto *SESSION* che rimane in vita per tutta la sessione (fino a quando si chiude il browser) e comunque per un tempo massimo predefinito (timeout), in genere 30 minuti. Il timeout e l'identificatore unico di sessione sono le proprietà *TIMEOUT* e *SessionId*.

L'oggetto *APPLICATION* condivide, memorizza e ritrova informazioni in risposta a richieste di utenti all'applicazione.

L'oggetto *SERVER* consente l'accesso alle funzionalità del server, per esempio consente di accedere alla libreria degli oggetti *COM*.

Il metodo più frequentemente utilizzato è *CreateObject*, che istanzia un altro oggetto rendendolo disponibile per l'uso con tutte le sue collection, eventi, metodi, proprietà in diverse applicazioni.

Esempio:

```
<% Set miaconnessione = Server.CreateObject ("ADODB.Connection") %>
```

In laboratorio tramite una lezione interattiva ho mostrato una delle caratteristiche più importanti dei linguaggi di scripting: la possibilità che ha l'utente di interagire con una pagina Web. Con le pagine ASP l'interazione avviene nel momento in cui l'utente invia una richiesta al server Web: quest'ultimo genera come risposta una pagina Web creata dinamicamente in quel momento.

L'interazione con l'utente prevede anche la possibilità di inviare alcuni dati come *parametri* della chiamata alla pagina ASP.

Il *passaggio di parametri* a una pagina ASP viene gestito attraverso i *form*, cioè i moduli del linguaggio HTML, che permettono la costruzione di un'interfaccia grafica, formata da caselle di testo e da pulsanti. Tramite questa interfaccia, l'utente può inserire i valori e inviarli al server Web come parametri dello script.

La consegna data alla classe è stata quella di implementare i seguenti form in HTML:

Invio di dati senza un Form

[Invia i dati nascosti](#)

Invio di dati attraverso un Form

Nome

Età Meno di 18 ▼

Sesso:
 Maschio
 Femmina

Questa attività si è svolta senza particolari difficoltà.

A questo punto, i ragazzi sono stati invitati a realizzare codice ASP, per visualizzare le risposte del server conseguenti all'invio dei form.

In particolare veniva loro richiesto la realizzazione di una pagina HTML e di una pagina ASP con i seguenti layout:



È interessante notare in questo caso che i dati inviati senza form sono aggiunti all'URL dopo il punto interrogativo. Altra cosa importante è che [Invia i dati nascosti](#) ha il collegamento a

http://localhost/senzaform_get.asp?name=Nicolas&tel=0521/5426356

La classe ha eseguito il compito correttamente.

Il codice relativo alla pagina HTML è il seguente:

```
<HTML>
<HEAD>
  <TITLE>Invio di dati senza un Form</TITLE>
</HEAD>
<HR>
<H1><CENTER>Invio di dati senza un Form</Center></H1>
<HR>
<BODY>
  <CENTER>
    <A HREF="senzaform_get.asp?name=Nicolas&tel=0521/5426356">Invia i
dati nascosti</A>
  </CENTER>

</BODY>
</HTML>
```

Il codice relativo alla pagina ASP (invio di dati senza form) è il seguente:

```
<%@ LANGUAGE=VBSCRIPT %>
<% Option Explicit %>
<HTML>
<HEAD>
<TITLE>Invio di dati senza un Form</TITLE>
</HEAD>

<BODY>
<HR>
<H1><CENTER>Invio di dati senza un Form</Center></H1>
<HR>

<%
  Dim strNome, strTel

  strNome = Request.QueryString("Name")
  strTel = Request.QueryString("Tel")

  Response.Write("Ciao ..... : " & strNome & "<BR>" & _
    "Numero di telefono: " & strTel & "<BR>" )
%>

</BODY>
```

Il codice relativo alla pagina HTML (invio di dati attraverso un form) è il seguente:

```
<HTML>
<HEAD>
  <TITLE>Invio di dati attraverso un Form</TITLE>
</HEAD>
<HR>
<H1><CENTER>Invio di dati attraverso un Form</Center></H1>
<HR>
<BODY>
  <FORM METHOD=GET ACTION="form_get.asp">

    <P>Nome <INPUT TYPE=TEXT NAME=Name>

    <P>Età <SELECT NAME=Age>
      <OPTION VALUE="meno di 18"> Meno di 18
      <OPTION VALUE="18 - 30" > 18 - 30
      <OPTION VALUE="Oltre 30" > Oltre 30
    </SELECT>

    <P>Sesso:<BR>
      <INPUT TYPE=RADIO NAME=Sex VALUE=Maschio> Maschio <BR>
      <INPUT TYPE=RADIO NAME=Sex VALUE=Femmina> Femmina <BR>

    <P><INPUT TYPE=SUBMIT VALUE="Invia i tuoi dati">
      <INPUT TYPE=RESET VALUE="Reimposta">
  </FORM>
</BODY>
</HTML>
```

Il codice relativo alla pagina ASP (invio di dati attraverso un form) è il seguente:

```
<%@ LANGUAGE=VBSCRIPT %>
<% Option Explicit %>
<HTML>
<HEAD>
<TITLE>Invio di dati attraverso un Form</TITLE>
</HEAD>

<BODY>
<HR>
<H1><CENTER>Invio di dati attraverso un Form</Center></H1>
<HR>

<%
  Dim strNome, strAnni, strSex

  strNome = Request.QueryString("Name")
  strAnni = Request.QueryString("Age")
  strSex = Request.QueryString("Sex")

  Response.Write("Ciao ..... : " & strNome & "<BR>" & _
    "I tuoi anni : " & strAnni & "<BR>" & _
    "Il tuo genere: " & strSex & "<BR>" )
%>

</BODY>
</HTML>
```

A questo punto mi sono soffermato su un concetto importante che sarà usato anche in seguito.

La riga di codice nel file HTML

```
<FORM METHOD=GET ACTION="form_get.asp">
```

specifica che il metodo da utilizzare, quando i dati vengono inviati dal Client al Server, è il metodo GET.

Inoltre il pulsante

è associato al file ASP.

Si noti che il metodo GET utilizza *Request.QueryString*.

Il risultato dell'esercizio "Invia dati attraverso un Form" è il seguente:



Come esercizio ho fatto fare una attività analoga a quella precedentemente descritta utilizzando il metodo POST.