Lezione 3 LE PRIME ACTIVE SERVICE PAGE: 099etto REQUEST

L'oggetto **Request** può essere considerato come l'esatto contrario dell'oggetto **Response**, infatti mentre quest'ultimo invia dati al browser, **Request serve a ricevere informazioni dal browser** (quindi informazioni dal navigatore). L'utilizzo di entrambi questi oggetti con i relativi metodi, ci permettono di interagire con gli utenti del nostro sito. I metodi principali di Request sono:

• **Collection Querystring** viene utilizzato quando si raccolgono dati provenienti da un indirizzo URL

• Collection Form viene utilizzato quando i dati provengono da un form Html utilizzando i metodi

- GET (il browser effettua la codifica all'interno dell'indirizzo e visualizza i dati codificati direttamente all'interno dell'indirizzo, quindi non adatto se i dati sono dati protetti come una password)
- POST (metodo adatto all'invio di dati protetti)

Proviamo ora a capire il funzionamento della Collection Form con un esempio di utilizzo di **Request.Form:** In questo esempio vedremo come visualizzare in una pagina Asp i dati inseriti dall'utente navigatore in un Form html (tipico esempio di modulo di iscrizione). cominciamo con il preparare la pagina Html.

Codice HTML (modulo.htm)	Risultato (modulo.htm)
ntml> nead> itle>modulo /head> body> form method="POST" tion="RACCOLTA_FORM.asp"> OME: <input name="NOME" type="text"/> OGNOME: <input <br="" type="text"/> ame="COGNOME"> nput type="submit" value="INVIA" ame="INVIA"> form> body> /html>	🖉 modulo - Windows Internet Explorer
	G C:Unetpublywww.rootlynodule.htm
	File Modifica Visualizza Preferiti Strumenki ?
	Google 8 - Effettus la ricerca 🖓 🍻 -
	🚖 🅸 🍘 modulo
	NOME: INVIA

Come vedete questo è il codice di una normale pagina Html (modulo.htm), scritto in blocco note, si possono riconoscere tag html che permettono di disegnare sulla pagina una casella di testo (<input type="text") e tag per l'inserimento di un pulsante (<input type="submit"> per entrambi si è fissata la proprietà name. La riga che vi ho evidenziato, definisce il form che verrà trasmesso con il metodo POST, quindi in modo "riservato", alla pagina RACCOLTA_FORM.asp quando il navigatore (action="nomefile.asp") cliccherà sul pulsante con scritto INVIA (proprietà value="invia").

con Frontpage per inserire le caselle di testo e il pulsante, dal menu Inserisci - Modulo casella di testo/pulsante

Operano	Operando con frontpage		
COCROME Propriets polanto di comando None: Producta polanto di comando Nome: Producta Producta polanto di comando None: Producta Prod	Dopo aver inserito gli oggetti, ci clicchiamo con il tasto DX attivando il menu di scelta rapida, scegliamo <i>Proprietà campo modulo</i> per definirne le proprietà come il nome e il value. Per definire il metodo action , scegliamo proprietà modulo e indichiamo la pagina RACCOLTA_FORM.asp. Ovviamente come sempre frontpage sporca un pò il codice originale, per questo a volte conviene partire dal blocco note e poi passare a frontpage per la formattazione piuttosto che il contrario.		
II codic	e con frontpage		
<head> <meta content="it" http-equiv="Content-Language"/> <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/> <title>Nuova pagina 1</title> </head> <body></body>			
 <form action="WEBBOT-SELF" method="POST"><!--webbot bot="SaveResults"<br-->S-Label-Fields="TRUE" U-File="RACCOLTA_FORM.asp" S-Format="HTML/BR" > NOME: <input <br="" name="NOME" type="text"/>size="20"> COGNOME: <input name="COGNOME" size="20" type="text"/> <input name="INVIA" type="text"/> </form>			
 .			

Ora vediamo come realizzare la pagina asp che dovrà raccogliere le informazioni dal modulo Html, ovviamente qui vedete il risultato che apparirà a video dopo che l'utente avrà cliccato sul pulsante **Invia** (ricordate la Action = "Raccolta_Form.asp")

Codice ASP	Risultato (Raccolta_Form.asp)
------------	-------------------------------

Nel codice Asp qui sopra, ho evidenziato due righe di codice, perché su queste vorrei soffermarmi un momento per capire esattamente come funziona l'oggetto REQUEST. Il codice nome=request.form("NOME"), assegna ad una variabile denominata nome (non l'ho dichiarata ma poteva essere dichiarata), il contenuto del text box NOME del form. Il codice Response.write "il tuo nome è: " & nome manda a video il contenuto della variabile nome concatenata (&) alla stringa indicata dopo il metodo Write. Come vedete ho omesso le parentesi dopo il metodo Write, che possono essere omesse, e vi sottolineo che i nomi di variabile non vanno mai tra apici, come in tutti i linguaggi di programmazione che già conoscete.

IN SINTESI...

<FORM method="POST" action="pagina.asp">: crea un modulo html,

che sarà inviato in modo riservato a pagina.asp.

INPUT TYPE="text": Tag Html per inserire in un form Html un text box

INPUT TYPE="submit": Tag Html per inserire in un form Html un pulsante

REQUEST: Oggetto Asp che si utilizza per ricevere dati dalla pagina web, dati che vengono inseriti dall'utente navigatore in un form html e recuperati con la collection **Form**: **Request.Form("nome oggetto del form")**

&: operatore di concatenazione stringhe.

AREA PROGETTO "Simulazione E-Commerce"

Cominciamo ad operare nell'area progetto applicando quanto esposto in questa lezione e creiamo la pagina web caricamagazzino.htm che conterrà il modulo (Form), per caricare la scheda di magazzino. Vi mostrerò codici "puliti", cioè scremati da quanto frontpage aggiunge automaticamente. quindi provando questi codici vi appariranno solo i contenuti essenziali, senza formattazione ne immagini, che potremo poi aggiungere con poche mosse direttamente da frontpage. Ho colorato il codice delle pagine del progetto in viola perchè possiate riconoscerlo facilmente.

Codice HTML (caricamagazzino.htm)	Risultato (caricamagazzino.htm)
<pre></pre> (caricamagazzino.htm) <html> <html> <head> <title>CARICA MAGAZZINO</title> SCHEDA ARTICOLO <form action="registraMagazzino.ASP" method="POST"> ARTICOLO <select name="articolo" size="1"> <option></option> <option>berretto Nike</option> <option>berretto Nike</option> <option>ciondolo cuore</option> <option>ciondolo resina verde</option> <option>cionta D&G</option> <option>contra D&G</option> <option>contra pano avvitatore</option> <option>coption></option></select></form></head></html></html>	CATEGORIA INVIA ANNULLA HOME
	La casella di riepilogo

NAME="QUANTITA">	Per permettere all'utente di inserire
PREZZO <input <="" td="" type="TEXT"/> <td>movimentazioni solo a riguardo degli articoli</td>	movimentazioni solo a riguardo degli articoli
NAME="PREZZO">	disponibili, creiamo una casella di riepilogo. Il
CATEGORIA <input< td=""><td>tag Html per inserirla è: <select> e <option>.</option></select></td></input<>	tag Html per inserirla è: <select> e <option>.</option></select>
NAME="CATEGORIA">	All'interno del tag select, definiamo gli attributi
<input <="" td="" type="SUBMIT"/> <td>size 1 per rendere visibile una sola voce a</td>	size 1 per rendere visibile una sola voce a
VALUE="INVIA" NAME="B1"> 	tendina chiusa, e l'attributo name per definire il
 	nome con cui verrà richiamata nelle pagine.
<input td="" type="RESET" value<=""/> <td>Con il tag <option> è possibile definire le voci</option></td>	Con il tag <option> è possibile definire le voci</option>
="ANNULLA" NAME="B2"> 	dell'elenco.
	Oltre il pulsante INVIA (type ="submit") per
<a< td=""><td>inviare i dati in un'altra pagina, ho previsto il</td></a<>	inviare i dati in un'altra pagina, ho previsto il
href="index.htm">HOME <td>pulsante ANNULLA (type="reset"), che</td>	pulsante ANNULLA (type="reset"), che
>	automaticamente cancella quanto scritto nei
	campi del modulo per resettare appunto il form.
	Notate il tag <form action=""> che invia i dati alla</form>
	pagina registraMagazzino.asp per caricare i dati
	nella tabella del database. Vedremo come
	realizzare questa pagina nella prossima lezione.