

16.6 Il passaggio di parametri

Ora che abbiamo acquisito una certa confidenza con la programmazione CGI affrontiamo un altro tema di interesse: come si effettua il passaggio di parametri a un CGI. Per prima cosa occorre dire che il modo canonico di invocare un CGI passando dei parametri è quello di impiegare un FORM HTML. Il FORM è un tag del linguaggio HTML che consente di costruire moduli per l'inserimento di informazioni. Vediamo subito un esempio in Figura 16.6. I campi vuoti di questo FORM possono essere riempiti come in Figura 16.7.

The image shows a screenshot of a Microsoft Internet Explorer browser window. The title bar reads "Sono Graditi i Vostri Commenti Microsoft Internet Explorer". The menu bar includes "File", "Edit", "View", "Go", "Favorites", and "Help". The toolbar contains icons for "Back", "Forward", "Stop", "Refresh", "Home", "Search", "Favorites", "History", "Channels", and "Find". The address bar shows "C:\httpcomments.html".

The main content area has a title "Sono Graditi i Vostri Commenti" in a large, bold, black serif font. Below the title, there are two input fields: "Nome:" followed by a text box, and "e-mail" followed by a text box. Below these fields, there is a paragraph of text: "Inserite di seguito i Vostri commenti." followed by "Verificate anche di aver inserito correttamente il Vostro nome e la Vostra e-mail." Below this text is a large, empty text area with a vertical scrollbar on the right side. At the bottom of the form, there are two buttons: "Invia Commenti" and "Cancella Tutto".

The status bar at the bottom of the window shows "Done" and "My Computer".

Figura 16.6 Esempio di FORM

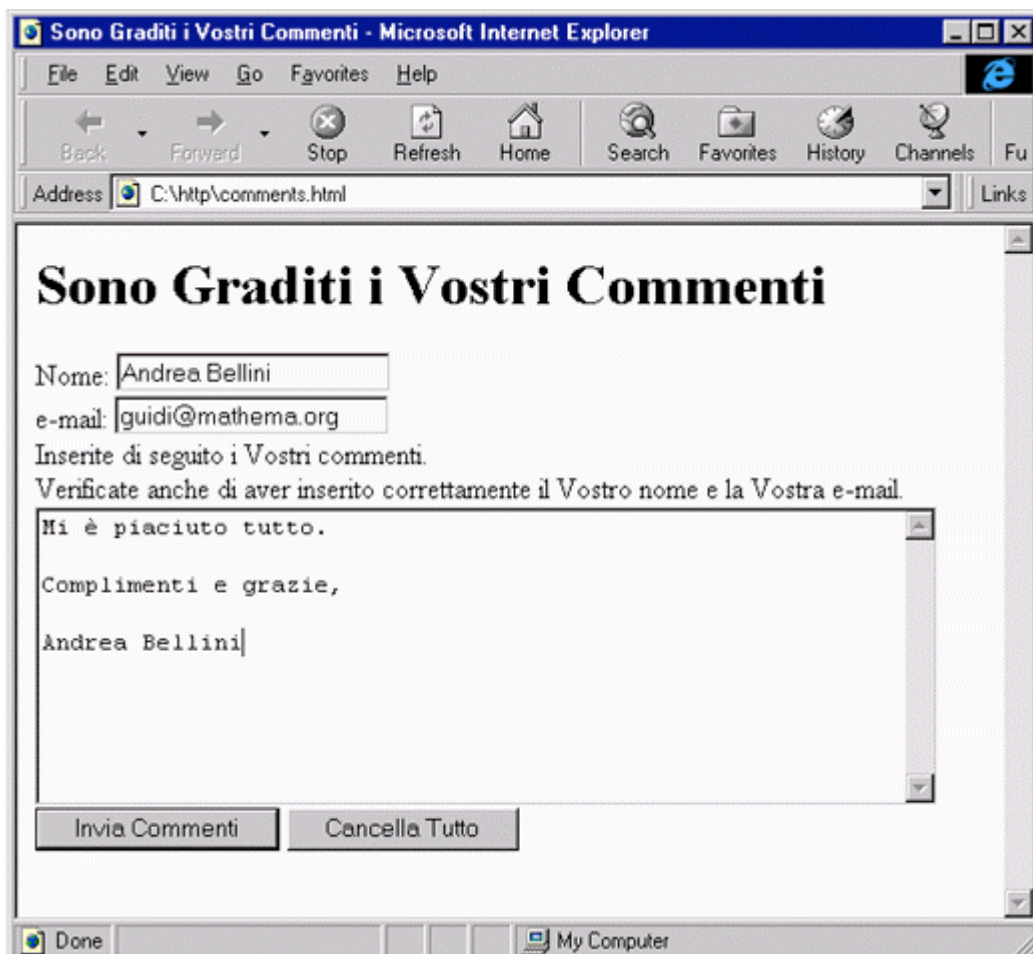


Figura 16.7 Esempio di FORM con i campi riempiti

Premendo il pulsante “Invia Commenti” si attiva il CGI che acquisisce i dati immessi, effettua delle elaborazioni e ritorna una pagina HTML per confermare l’avvenuta elaborazione. Il codice HTML che ha permesso di generare questa pagina è quello del Listato 16.3.

```
<html>
<head>
<title>Sono Graditi i Vostri Commenti</title>
</head>
<body>
<h1>Sono Graditi i Vostri Commenti</h1>
<p>
<form action="http://localhost/comments.exe" method="POST">
Nome: <input type="text" name="name"><br>
e-mail: <input type="text" name="email"><br>
Inserite di seguito i Vostri commenti.<br>
Verificate anche di aver inserito correttamente il Vostro nome e
la Vostra e-mail.<br>
<textarea name="comments" rows="10" cols="60">
</textarea>
<br>
<input type="submit" value="Invia Commenti">
<input type="reset" value="Cancella Tutto">
```

```
</form>
</body>
</html>
```

Listato 16.3 Codice HTML per la gestione della pagina delle Figure 16.6 e 16.7

La parte che ci interessa commentare è quella relativa al tag `<form>`:

```
<form action="http://localhost/comments.exe" method="POST">
Nome: <input type="text" name="name"><br>
e-mail: <input type="text" name="email"><br>
Inserite di seguito i Vostri commenti.<br>
Verificate anche di aver inserito correttamente il Vostro nome e
la Vostra e-mail.<br>
<textarea name="comments" rows="10" cols="60">
</textarea>
<br>
<input type="submit" value="Invia Commenti">
<input type="reset" value="Cancella Tutto">
</form>
```

Si osservi la prima direttiva:

```
<form action="http://localhost/comments.exe" method="POST">
```

In essa si inizia un FORM e come azione associata al FORM, ovvero come CGI da innescare in seguito alla selezione del bottone “Invia Commenti”, lanceremo il CGI identificato da

```
http://localhost/comments.exe
```

facendo uso del metodo di passaggio di parametri detto “POST”. Esistono due metodi per inviare i parametri a un CGI.

GET i parametri sono passati al CGI per mezzo della variabile di ambiente `QUERY_STRING`, e in questa variabile di solito non si possono memorizzare più di un centinaio di byte di dati;

POST i parametri vengono trasmessi al CGI per mezzo dello standard input del CGI senza alcuna limitazione di dimensione. In questo caso il CGI può conoscere quanti dati deve leggere da `STDIN` andando a leggere il contenuto di un'altra variabile di memoria `CONTENT_LENGTH`.

Il programma CGI per sapere quale metodo è stato scelto per passargli i parametri è sufficiente che legga il contenuto della variabile di ambiente `REQUEST_METHOD`. Da questa semplice introduzione si arguisce come il metodo POST sia tra i due preferibile poiché non introduce alcun limite al passaggio dei parametri. Ma in che modo vengono passati i parametri?

Ritorniamo alla definizione in HTML del FORM definito nell'esempio, a partire dalla definizione del primo campo relativo al “Nome”:

```
Nome: <input type="text" name="name">
```

La prima parte

Nome :

non ha nessun significato speciale se non quello di essere una etichetta che esplicita il significato del campo; il tag

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

invece definisce il campo da riempire. Il significato di questo tag potrebbe essere così parafrasato: “Definisci un campo di input, ovvero un campo in cui effettuare un inserimento. Il valore che sarà inserito nel campo sarà trattato come un testo (`type = "text"`) e il nome simbolico che daremo a questo campo sarà `"name"` (`name="name"`)”. Il nome simbolico del campo è molto importante. Infatti quando digiteremo qualcosa in questo campo, per esempio `"Andrea Bellini"`, nel momento in cui il FORM con i suoi contenuti inseriti dall’utente sarà inviato al CGI darà luogo a una coppia del tipo:

```
name=Andrea+Bellini
```

Si noti come lo spazio venga simbolicamente rappresentato con il simbolo `'+'`. Il successivo campo relativo all’indirizzo e-mail:

```
e-mail: <input type="text" name="email">
```

ha il medesimo significato del precedente, ma con un diverso nome simbolico, `"email"`. A questo punto la stringa di parametri che sarà passata al CGI si allunga:

```
name=Andrea+Bellini&email=guidi@mathema.com
```

Si osservi come le coppie

```
<nome=valore>
```

vengano separate per mezzo del simbolo `"&"`. Il terzo campo:

```
<textarea name="comments" rows="10" cols="60">
</textarea>
```

non è concettualmente diverso dai primi due. Anche in questo caso sarà generata una coppia:

```
comments= <testo relativo ai commenti>
```

Gli attributi `rows` e `cols` regolano la dimensione della finestra di editing. Infine si definiscono i due bottoni di invio e di reset

```
<input type="submit" value="Invia Commenti">
<input type="reset" value="Cancella Tutto">
```

Si osserva che, a eccezione degli spazi, cui è riservato il simbolo speciale `"+"`, in virtù del fatto che gli spazi sono usati molto di frequente, i caratteri speciali sono rappresentati con la seguente sequenza, detta sequenza di escape:

```
%xx
```

dove `xx` è un numero di due cifre esadecimali che indica il codice ASCII del carattere speciale che si vuole rappresentare. Per esempio, il simbolo `%` stesso può essere inviato con la sequenza di escape:

&25

dove 25 nella codifica ASCII esadecimale corrisponde al simbolo “%”.

Alla fine ciò significa che il CGI che si vede recapitare i parametri con questo formato dovrà farsi carico di decifrare queste sequenze di escape.