

---

# Applicazioni Web: meccanismi per il passaggio di informazioni tramite HTTP

Corso di **Applicazioni Telematiche**

A.A. 2006-07 – Lezione n.10 – parte II

Prof. Roberto Canonico

Università degli Studi di Napoli Federico II

Facoltà di Ingegneria

---

---

## Argomenti

- Programmazione server-side
  - Uso dei form HTML per l'input di utente
  - Uso dei metodi HTTP per il passaggio di informazioni da client ad applicazione server-side
-

---

## Programmazione server-side

- Consente di generare “dinamicamente” tutto o parte del documento HTML richiesto da un browser, in seguito all’esecuzione di un programma sulla macchina su cui è in esecuzione il processo “web server”
  - Serve a costruire pagine web il cui contenuto è determinato da informazioni fornite dall’utente e/o da dati reperiti da sorgenti esterne
    - Previsioni meteo, quotazioni di borsa, news, siti di e-commerce, sistemi di e-learning, ecc. ...
  - Il web server deve essere opportunamente istruito e configurato per mandare in esecuzione un programma in modo da generare il contenuto “dinamico”
- 

---

## Programmazione server-side (2)

- Sono attualmente disponibili diverse tecniche di programmazione server side
  - Si differenziano per:
    - i linguaggi di programmazione supportati
    - i web server supportati
    - i meccanismi di “aggancio” al web server ed il loro impatto sulle prestazioni
    - il particolare ambito applicativo per il quale sono concepite
-

## Programmazione server-side (3)

- Codice *embedded* in HTML
  - Server Side Includes (SSI) – Apache
  - ASP – Microsoft
  - PHP
  - JSP
- Codice separato, associato ad una URL
  - CGI
  - Java servlet
  - NSAPI o server API

## Form HTML

- Il meccanismo dei FORM consente di inviare informazioni da un web browser ad un programma in esecuzione lato server
- Per quanto rudimentale, è lo strumento essenziale per trasformare un browser web in un'interfaccia grafica (GUI) universale
- HTML definisce degli elementi interattivi e gli eventi che determinano l'invio delle informazioni

## Elementi di una form HTML

The diagram shows a form with the following elements and labels:

- Popup menu:** A dropdown menu with the text "Dott." and a downward arrow.
- Text box:** Three text input fields labeled "Cognome: De Pippis", "Nome: Pippo", and "Email: pippozzo@gmail.com".
- Radio button:** A group of radio buttons for age selection: "Età:  0-17  18-26  27-35  >35".
- Check box:** A group of checkboxes for skills: "Conoscenze:  GNU/Linux  Windows  Office  Java".
- Buttons:** Two buttons at the bottom: "Invia query" and "Clear data".

## Form HTML: esempio

```
<FORM action="/cgi-bin/ex1.exe" method="GET">
<SELECT NAME="titolo" SIZE=1>
  <OPTION SELECTED> Sig.
  <OPTION> Sig.ra
  <OPTION> Dott.
  <OPTION> Dott.ssa
  <OPTION> Prof.
  <OPTION> Prof.ssa
</SELECT><p>
Cognome: <INPUT TYPE="text" NAME="cognome" SIZE=30><p>
Nome: <INPUT TYPE="text" NAME="nome" SIZE=30><p>
Email: <INPUT TYPE="text" NAME="email" SIZE=30><p>
Età:
<INPUT TYPE="radio" NAME="eta" VALUE="a">0-17
<INPUT TYPE="radio" NAME="eta" VALUE="b">18-26
<INPUT TYPE="radio" NAME="eta" VALUE="c">27-35
<INPUT TYPE="radio" NAME="eta" VALUE="d">&gt;35
<p>Conoscenze:
<INPUT TYPE="checkbox" NAME="skill" VALUE="Linux">GNU/Linux
<INPUT TYPE="checkbox" NAME="skill" VALUE="Windows">Windows
<INPUT TYPE="checkbox" NAME="skill" VALUE="Office">Office
<INPUT TYPE="checkbox" NAME="skill" VALUE="Java">Java
<p><INPUT TYPE="submit">
<INPUT TYPE="reset" NAME="resetbutton" VALUE="Clear data">
</FORM>
```

## Form con invio dati tramite GET

- I dati vengono inviati attraverso la URL del metodo GET mediante una query string
- Formato application/x-www-form-urlencoded:
  - space characters → "+"
  - escape (%xx) reserved characters
  - name=value pairs separated by &
- Esempio: `http://localhost/cgi-bin/ex1.exe?titolo=Dott.&cognome=De+Pippis&nome=Pippo&email=pippo@gmail.com&eta=d&skill=Linux&skill=Windows&skill=Java`
- E' possibile inviare solo una piccola quantità di dati
- I dati restano visibili nel file di log del server!

## Esempio di form con uso di GET

- ```
<form action="env.cgi" method="GET">
  Enter some text here:
  <input type="text" name="sample_text" size=30>
  <input type="text" name="other_text" size=30>
  <input type="submit"><p></form>
```
- `test.cgi?sample_text=paperino&other_text=qui+quo+qua`

Enter some text here:

## Form con invio dati tramite POST

- I dati vengono inviati dal client al server web nel body del metodo POST
- I dati non sono visibili nel file di log del server

## Esempio di form con uso di POST

- ```
<form action="env.cgi" method="POST">  
Enter some text here:  
<input type="text" name="sample_text" size=30>  
<input type="text" name="other_text" size=30>  
<input type="submit"><p></form>
```
- Nessuna apparente differenza per l'utente
- La differenza è nei messaggi HTTP che viaggiano sulla rete e nella struttura del programma server-side

Enter some text here:

Domande?

