

La pagina HTML

Una pagina HTML è in origine un file di testo non formattato, che può avere estensione .htm (o .html). Oltre al testo, al suo interno vengono utilizzati degli elementi speciali chiamati "tag" che servono a dare un significato al testo. Ad esempio, racchiudendo una parola tra i tag <a> si attribuisce il significato di collegamento ipertestuale.

```
<a href="destinazione_o_percorso"> parola </a>
```

Una pagina vuota

Essistono dei tag obbligatori che vengono usati sempre, ad esempio per indicare dove inizia e dove finisce il codice del corpo della pagina (body)

```
<html>
  <head>
    <title> </title>
  </head>

  <body>

  </body>
</html>
```

Form

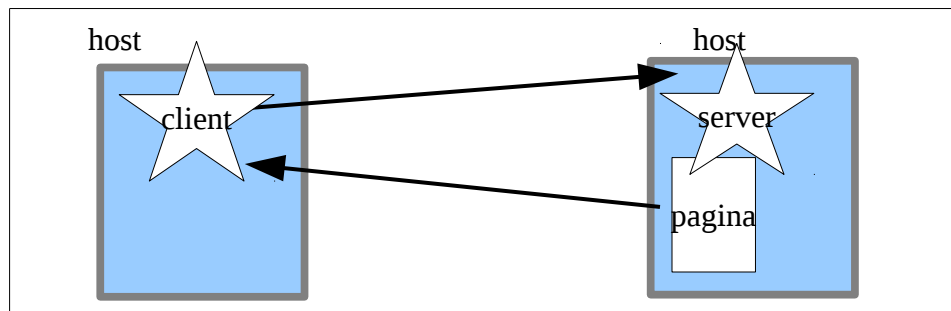
I form, chiamati anche moduli o formulari, sono degli elementi all'interno di una pagina web che consentono all'utente, ad esempio, di scrivere un testo nella pagina e di premere un pulsante. Un esempio di form è l'area dove si digita una parola desiderata, nei motori di ricerca. Un altro esempio è una pagina in cui l'utente digitati i propri dati anagrafici per registrarsi a qualche servizio.

Riassumendo, il form può essere considerato un'interfaccia tramite la quale l'utente può inserire dati di input all'interno di campi predisposti dal programmatore. Un semplice form può essere creato in una comune pagina HTML usando il tag <form>.

```
</body>
  <form>
    <input name="vartesto" type="text" />
    <input name="varpass" type="password" />
    <input value="invia" type="submit" />
  </form>
</body>
```

Funzione del browser

Il browser è un programma, ne sono un esempio Mozilla Firefox, Opera, Microsoft Internet Explorer. Di solito viene utilizzato da un utente per visualizzare le pagine web presenti su un altro computer, cioè su un altro host in una rete di computer. Il browser svolge la funzione di un client (richiede un servizio). Nell'altro host un altro programma (server) risponde fornendo la pagina richiesta. Nelle seguenti figure il simbolo della stella rappresenta un software.



Identificarsi in una rete

Quando si è collegati in una rete di computer, per poter distinguere un computer da un altro, in alcune applicazioni, viene usato un indirizzo numerico, chiamato indirizzo TCP/IP, ad esempio come il seguente: 192.168.10.4

In una rete non vi dovrebbero mai essere due host con lo stesso indirizzo IP.

L'indirizzo può essere stabilito manualmente dall'amministratore del sistema oppure automaticamente per mezzo di un servizio DHCP.

Se si possiede un computer collegato ad una rete è possibile visualizzare la configurazione del proprio indirizzo TCP/IP, digitando il seguente comando dal terminale dei comandi:

- Per windows, digitare: `ipconfig /all`
- Per Linux, digitare: `ifconfig -a`

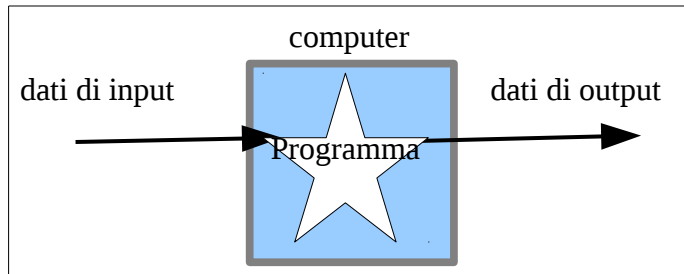
Alcune combinazioni numeriche dell'indirizzo TCP/IP sono riservate ad un uso speciale.

Ad esempio, se un indirizzo termina con .255 è considerato un indirizzo di broadcast, cioè indica tutti i computer della rete. È riservato anche l'indirizzo che termina con .0

Un altro esempio è l'indirizzo 127.0.0.1 che indica (su ogni computer) il computer attualmente in uso.

I dati e i programmi

Ricordiamo che un sistema operativo serve per far accendere il computer, e far eseguire i programmi. Un programma in esecuzione, di solito, elabora i dati in ingresso (di input) ed eventualmente produce come risultato dati in uscita (di output). I dati di input possono essere prodotti dall'utente o da un file di dati. I dati di output possono essere diretti verso l'utente o su un file di dati.



Pagine statiche e dinamiche

Un sito web può essere composto sia da pagine "statiche" che da pagine "dinamiche".

Sono considerate "**statiche**" le pagine che vengono create e modificate dal webmaster, usando il codice HTML/CSS (con estensione htm o html), e poi trasferite sul server, spesso usando un protocollo ftp. Sono considerate "**dinamiche**" le pagine che vengono generate automaticamente dal server, ogni volta che vengono richieste dal client. Questo è possibile perchè, oltre ai tag HTML, esse contengono anche istruzioni e comandi eseguibili dal computer. Le pagine dinamiche possono avere diverse estensioni, tra cui: .php, .asp, .jsp, .cgi, .py, che dipende dal linguaggio di programmazione che contengono.

Il termine "statico" e "dinamico" si riferisce quindi **al metodo con cui viene generato il contenuto** delle pagine web.

Bisogna precisare che nelle pagine web potrebbero trovarsi anche un altro linguaggio di programmazione, di cui non si è parlato: il Javascript.

Il Javascript è un linguaggio interpretato dal browser (dal client) e non permette al server di generare il contenuto delle pagine. Il linguaggio Javascript permette solo di rendere la pagina più "interattiva", cioè di farla reagire alle azioni e agli "eventi" generati con mouse e tastiera dall'utente. La programmazione in Javascript è detta **lato client**.



Per scrivere un programma si deve usare un linguaggio di programmazione. Il linguaggio HTML non è un linguaggio di programmazione, ma permette solo di creare pagine che contengono testo (informazioni racchiuse nei tag). Riassumendo:

una pagina che contiene solo HTML e CSS è considerata "statica";

una pagina a cui si aggiungono anche istruzioni in Javascript è interattiva, ma è ancora considerata "statica";

solo aggiungendo istruzioni in un linguaggio interpretato dal server la pagina può diventare "dinamica".

In quest'ultimo caso, infatti, si parla di programmazione **lato server**.

Possono essere utilizzati diversi tipi di software per lavorare sul web.

L'interprete del linguaggio PHP e il server web Apache sono software liberi e sono disponibili per diversi sistemi operativi.

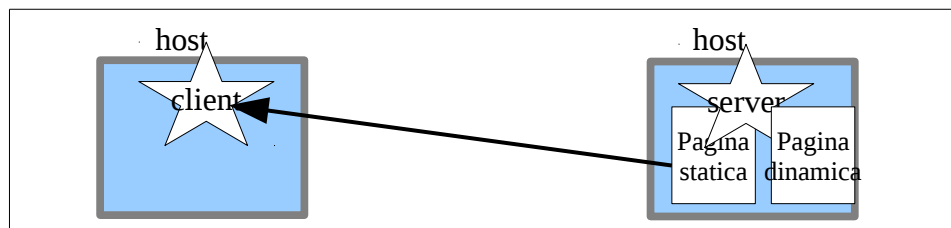
Il sistema operativo su cui sarebbe preferibile utilizzarli sarebbe GNU/Linux, anch'esso libero, ma il PHP può essere provato anche su Microsoft Windows, installando EasyPHP.

Domande e risposte

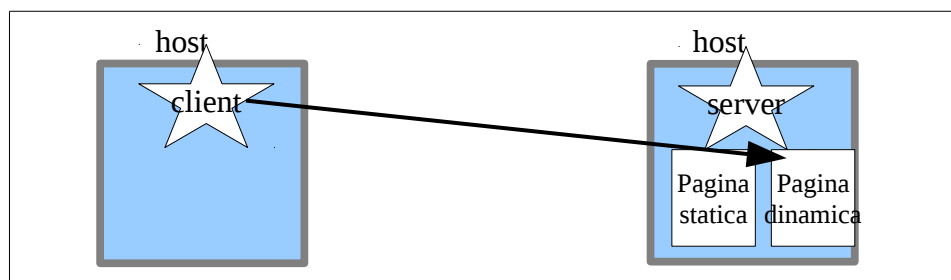
1. Chi esegue la pagina dinamica?
2. Da dove può prendere i dati di input la pagina dinamica?
3. Dove invia i suoi eventuali dati di output?

Risposte:

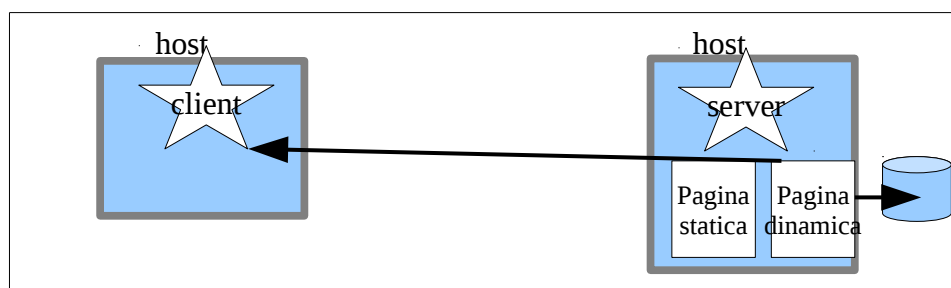
1. La pagina dinamica è eseguita su l'host che funziona come server. Per questa ragione, una pagina che contenga solo istruzioni in linguaggio Javascript non è considerata dinamica, poiché il Javascript è eseguito dal client web (il browser).
2. Un modo semplice di raccogliere dati di input è quello di usare un form, e, come già detto, il form può essere contenuto in una pagina statica. L'utente può, quindi, attraverso un browser, visualizzare il form presente in una comune pagina html (statica)



L'utente può inserire i dati di input nel form e inviarli al server che li farà elaborare dalla pagina dinamica (specificata nello stesso form).



3. Dopo l'elaborazione, la pagina dinamica può salvare i dati ricevuti in una memoria (immagine di un cilindro) e/o produrre un output visivo, restituendo una nuova pagina al browser dell'utente.



Come visualizzare il contenuto dinamico

Di solito le pagine web prima di essere visualizzate dal browser devono essere ottenute da un computer remoto. Quando il browser chiede al server web remoto una pagina, specifica un indirizzo con protocollo "http":

`http://indirizzo_del_server/pagina.php`

In questo esempio, la pagina ha estensione .php, quindi non viene inviata direttamente al browser ma viene prima elaborata ed eventualmente trasformata dal programma interprete presente nel server web in modo che poi il browser possa trovare al suo interno solo il codice html e quindi visualizzarla come una normale pagina html.

(Il browser non saprebbe interpretare le istruzioni in linguaggio php oppure asp, jsp, ecc.).

Se la pagina avesse avuto estensione .html, e fosse stata sullo stesso computer dove si trova il browser, ad esempio sul Desktop, sarebbe stato possibile aprirla anche essendo off-line, con un doppio click, senza bisogno di chiamare in esecuzione l'interprete php del server.

In questo caso, invece, non si può aprire la pagina dinamica direttamente con il browser.

Per aprire una pagina dinamica si deve essere sempre collegati con un server e quindi ogni volta si deve digitare:

`http://indirizzo_del_server/pagina.php`

Un piccolo trucco: a volte capita agli sviluppatori di avere un server web installato nello stesso computer dove si trova anche il browser. In questo caso si deve comunque digitare un indirizzo http (locale) su quest'ultimo:

`http://127.0.0.1/pagina.php`

Un esempio semplice

Un esempio di pagina php può essere trovata nell'ultima pagina di questo documento. Essa contiene sia codice HTML che codice PHP (evidenziato in nero).

Digitando il suo indirizzo, come:

`http://indirizzo_del_server/pagina.php`

questa pagina visualizzerà nel browser una data e un'ora. Si tratta della data e dell'ora del computer dove è in esecuzione il web server e **non** della data e dell'ora del computer di chi apre la pagina nel browser.

Ogni volta che l'utente caricherà questa pagina dinamica, il web server eseguirà le istruzioni PHP in essa contenute ed eventualmente potrà generare automaticamente (al volo) una nuova pagina web, con una nuova data e una nuova ora...

Un esempio reale

Quelli visti fino ad ora sono solo semplici esempi di pagine dinamiche, ma chi realizza "veri" siti internet, oggi, non deve costruire un sito scrivendo manualmente tutto il codice perché è possibile trovare diversi "siti" web dinamici, pronti per essere usati, liberamente scaricabili e modificabili.

Tanto per fare alcuni nomi, possiamo ricordare: dokuwiki.org, mediawiki.org e wordpress.org.

Ognuna delle applicazioni citate può essere scaricata come un archivio compresso che contiene tutti i file necessari al funzionamento del sito dinamico.

I file dovranno essere decompressi e caricati sull'host del sito (ad esempio, tramite servizio ftp).

Approfondimento sul web 2.0

In passato, l'attività dello sviluppatore di siti web consisteva nella costruzione di pagine statiche usando dei linguaggi (come HTML, CSS o Javascript) che potevano essere interpretati dal browser.

Il cliente consegnava i contenuti allo sviluppatore web che poi li pubblicava nel sito.

Questo rendeva il processo di pubblicazione abbastanza lento e costoso.

Questa attività portava alla realizzazione di pagine con estensione .html che appartengono al cosiddetto **web 1.0**, in contrapposizione al più recente **web 2.0**.

Usando le funzionalità del web 2.0 il costo dello sviluppo dei siti è diminuito perché esistono dei siti web dinamici pronti in cui si deve solo personalizzare solo l'aspetto e la grafica del sito.

Inoltre, utilizzando pagine web dinamiche, ovvero applicazioni eseguite sul server, è possibile ricevere dati da parte degli utenti, elaborarli e memorizzarli in un database.

In pratica, il contenuto può essere creato dagli stessi utenti del sito (ne è un esempio la wikipedia).

Nei servizi offerti dagli host che ospitavano siti secondo il web 1.0, c'erano solo il server web e il server ftp. Nei servizi offerti dal web 2.0 sono utilizzati anche un servizio di database e un interprete di un linguaggio di programmazione lato server.

Offerte di hosting

Diverse aziende offrono diversi tipi di servizi a diversi prezzi.

Di solito è compreso sempre l'acquisto del nome di dominio e il pagamento del canone deve essere ripetuto ogni anno.

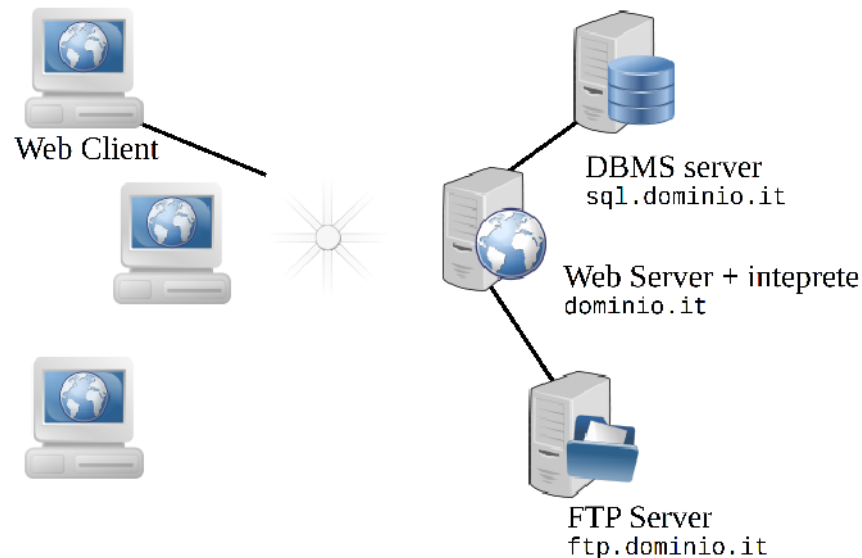
caratteristiche	primo esempio	secondo esempio
sistema operativo	GNU/Linux	Microsoft Windows Server
server web	Apache	Microsoft IIS
linguaggi supportati	PHP 5.3, PERL 5.5	ASP.net 4.0
DBMS	MySQL	Microsoft SQL Server
spazio database	5 GB	10 GB
spazio web	100 GB	10 GB
traffico mensile	10 GB	5 GB

La scelta dipende soprattutto dal tipo di linguaggio che si vuole utilizzare (PHP, ASP oppure JSP).

Il costo dipende anche dal numero di utenti che si prevede di avere.

La spesa aumenta di dieci volte se invece di affittare un host condiviso da più siti (hosting) si chiede un host dedicato per un solo sito (housing o server dedicato).

Ogni servizio potrebbe essere disponibile su diversi host (su diversi indirizzi), con diversi username e password.



Lo schema composto da tre elementi (web client, web server con pagine dinamiche e DBMS server) viene detto anche **modello Three Tier** e le applicazioni eseguite sul server web vengono chiamate **applicazioni web**.

Il modello Three Tier ha il vantaggio di suddividere tre compiti su tre macchine specializzate per questi compiti:

- il compito di memorizzare i dati è affidato al DBMS
- il compito di eseguire le istruzioni per l'elaborazione dei dati (la parte logica) è lasciato all'interprete delle pagine dinamiche
- il compito di rappresentare l'interfaccia grafica all'utente è lasciato al browser

Questa divisione dei compiti offre il vantaggio di poter sostituire uno dei tre elementi senza dover modificare gli altri due. Ad esempio, posso cambiare il tipo di DBMS senza apportare modifiche all'interfaccia della pagina html.

Esempio d'uso del tag <form>

```
<body>
<form method="get" action="programma.php" id="primo_modulo">

    <label for="u">Username</label>
    <input id="u" name="varutente" type="text" autofocus required
placeholder="Scrivi qualcosa..." />

    <label for="p">Password</label>
    <input id="p" name="varpw" type="password" required />
    <input type="submit" value="invia!" />
</form>
</body>
```

<form> serve a racchiudere il modulo

attributi

- id = identificatore del form
- method = obbligatorio, può valere solo "get" oppure "post". Il metodo "get" invia i dati scrivendoli in chiaro sulla barra dell'indirizzo, anche se si tratta di password. Il metodo "post" non li scrive sulla barra dell'indirizzo, ma li invia comunque in testo in chiaro, quindi visibile. Per nascondere qualcosa, sul serio, si deve usare la crittografia ed è necessario usare il protocollo "https".
- action = obbligatorio, contiene il percorso della pagina dinamica (o della funzione) che riceve i dati inviati premendo il pulsante "submit" del form

<input> elemento per inserire o inviare i dati

attributi

- id = identificatore dell'elemento di input
- name = obbligatorio, nome dell'elemento o della variabile di input
- value = valore di input predefinito (oppure il testo su un pulsante)
- autofocus = evita all'utente di dover fare click nel campo di input
- required = obbliga l'utente a scrivere qualcosa in un campo
- placeholder = visualizza un testo che scomparirà...
- type = obbligatorio, specifica il tipo di input (vedi tabella successiva)

<label> serve per aggiungere un'etichetta ad un campo

La posizione dell'etichetta nella pagina, rispetto al campo di input a cui si riferisce, è importante solo per la comprensione da parte dell'utente che deve inserire i dati.

attributi

- for = riferimento ad un tag <input>... se si specifica l'attributo for, l'utente può cliccare sull'etichetta oltre che sul campo.

Tabella riassuntiva dei tipi di input

valore di type	descrizione	attributi
"checkbox"	casella quadrata (vero o falso)	vedere esempio
"date"	casella di testo per una data	name
"email"	casella di testo	name
"file"	casella di testo con pulsante sfoglia(browse)	name
"hidden"	casella invisibile all'utente	name
"number"	casella di testo numerico	name
"password"	casella di testo nascosto	name
"radio"	casella di scelta opzionale	vedere esempio
"reset"	pulsante per pulire il modulo	name
"submit"	pulsante per inviare i dati	name, value,
"text"	casella di testo	name
"url"	casella di testo che inizia con "http://"	name

I più usati sono solo: text, password, checkbox, radio, file, url, reset e submit.

Questo perché ancora non tutti i browser supportano gli altri tipi di campi di input (come "date") e quindi, di solito, si usa ancora il semplice input di tipo testuale. Usando l'input 'text', invece di 'date', l'utente può scrivere qualsiasi cosa, anche usando un formato sbagliato. Ad esempio, può scrivere una data in un formato sbagliato: 31/12/10 invece di 31/12/2010.

Per questo motivo, per convalidare la correttezza formale del contenuto di un campo, spesso si usano ancora delle funzioni in Javascript. Le funzioni in Javascript possono essere usate anche per realizzare tutto quello che non è supportato dal browser dell'utente.

Ad esempio, spesso si usa il Javascript per fare in modo che il cursore sia lampeggiante dentro la prima casella di input senza che l'utente debba farci click dentro. Nei nuovi browser questa funzione si può ottenere semplicemente usando l'attributo "autofocus".

Lo stesso si può fare perché un campo sia obbligatorio (attributo "required").

Esempio radio button

In quale nazione sei nato?

Scegli una voce

- ☐ Italia
- ☐ Spagna
- ☐ Altro

invia!

Questo gruppo di input a scelta (singola) può essere ottenuto usando il seguente codice:

```
<body>

<form method="get" action="programma.php" id="secondo_modulo">
In quale nazione sei nato?
  <fieldset>
    <legend>Scegli una voce</legend>
    <input id="uno" type="radio" name="var" value="it" />
    <label for="uno">Italia </label> <br />

    <input id="due" type="radio" name="var" value="es" />
    <label for="due">Spagna </label> <br />

    <input id="tre" type="radio" name="var" value="altro"/>
    <label for="tre">Altro </label>
  </fieldset>

  <input type="submit" value="invia!" />

</form>
</body>
```

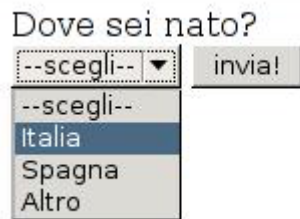
Note sul codice:

<fieldset> e <label> sono usati solo per l'estetica.

I campi di input "radio" corrispondono a pulsanti circolari che permettono di selezionare SOLO uno dei tre valori (Italia, Spagna o Altro) per poi poterlo memorizzare dentro una singola variabile (var) che poi viene inviata.

Supporre che l'utente debba inserire la sua provincia di residenza tra 38 possibilità: in questi casi, quando le scelte sono numerose, e l'elenco dei termini diventa lungo, in alternativa all'elemento `<input>` di tipo "radio", si può usare il tag `<select>` che permette di nascondere le scelte in un menù a tendina (menù a discesa) usando al suo interno il tag `<option>`. `<select>` ha il vantaggio di occupare meno spazio nella pagina.

Esempio di menù a tendina



```
<form method="get" action="programma.php" >
Dove sei nato?
<select name="var" >
  <option value="">--scegli--</option>
  <option value="it">Italia</option>
  <option value="es">Spagna</option>
  <option value="altro">Altro</option>
</select>

  <input type="submit" value="invia!" />
</form>
```

Anche in questo esempio, si può fare una sola scelta, che verrà memorizzata in una sola variabile.

Altre volte l'utente deve avere la possibilità di fare più di una scelta.

Ad esempio, nel caso in cui debba rispondere alla domanda: quali lingue conosci?

Esempio di checkbox

```
<form method="get" action="programma.php" id="secondo_modulo">
<p>Quali lingue conosci?</p>

  <input id="uno" type="checkbox" name="var_it" value="it" />
  <label for="uno">Italiano</label> <br />

  <input id="due" type="checkbox" name="var_es" value="es" />
  <label for="due">Spagnolo</label> <br />

  <input id="tre" type="checkbox" name="var_en" value="en"/>
  <label for="tre">Inglese</label>

  <input type="submit" value="invia!" />
</form>
```

Esercizio

Si vuole realizzare un form per chiedere all'utente quali sono i paesi esteri che ha visitato, a scelta tra: Svizzera, Australia, Francia e altro.

Esercizio

Si vuole realizzare un form per chiedere all'utente di selezionare la sua fascia di età, a scelta tra: 0-13, 14-17, 18-35 e maggiore di 35.

Soluzioni

Nel primo esercizio l'utente potrebbe scegliere anche tutte e tre le opzioni contemporaneamente (tutte e tre vere), quindi si possono usare tre caselle "checkbox".

Nel secondo esercizio l'utente deve scegliere solo una casella, quindi si possono usare tre caselle "radio", oppure una casella "select".

Esempio di pagina dinamica "programma.php"

```
<html>
  <head>
    <title> Pagina di output </title>
  </head>

  <body>

    <h1> Output </h1>
    <h2> Data del server </h2>
    <p>
      <?php
        echo date('Y-m-d') . "\n";
      ?>
    </p>
    <h2> Ora del server </h2>
    <p>
      <?php
        echo date('h:i:s A') . "\n";
      ?>
    </p>
    <p> Il server aspettava le seguenti variabili, nelle quali sono stati
inseriti i seguenti valori:
    </p>

    <?php
      $varutente = $_POST['varutente'];
      $varpw = $_POST['varpw'];
      $var = $_POST['var'];
      $vartempo = $_POST['vartempo'];
      $varfelice = $_POST['varfelice'];
    ?>
    <table border=1>
      <tr>
        <th>$varutente</th>
        <th>$varpw</th>
        <th>$var</th>
        <th>$vartempo</th>
        <th>$varfelice</th>
      </tr>
      <tr>
        <?php
          print("<td>".$varutente."</td>\n");
          print("<td>".$varpw."</td>\n");
          print("<td>".$var."</td>\n");
          print("<td>".$vartempo."</td>\n");
          print("<td>".$varfelice."</td>\n");
        ?>
      </tr>
    </table>
  </body>
</html>
```