

Una breve introduzione al PHP

Guida al PHP

Corso pratico

Andrea Aralla

Sommario

1	INTRODUZIONE.....	3
1.1	Prerequisiti.....	3
1.2	Argomenti.....	3
1.3	Strumenti di sviluppo.....	3
2	Server Web.....	4
2.1	Configurare Apache tramite XAMMP.....	4
3	Esecuzione Lato Server e Lato Client.....	5
3.1	Lato Client JAVASCRIPT.....	5
3.2	Lato Server PHP.....	5
4	Storia e Configurazione del PHP.....	6
4.1	Storia del PHP.....	6
4.2	Configurazione.....	6
4.3	Come funziona un web server?.....	6
5	Prima Pagina PHP.....	7
5.1	Creazione di una pagina statica.....	7
5.2	Creazione di una pagina dinamica.....	8
5.3	Prime conclusioni.....	10
5.4	Sintassi PHP di base.....	11
5.5	Le variabili.....	12
5.6	Tipi di dati in PHP.....	15
5.6.1	STRING.....	15
5.6.2	INTEGER.....	15
5.6.3	FLOAT.....	16
5.6.4	BOOLEAN.....	16
5.6.5	ARRAY.....	16
5.6.6	NULL.....	17
5.7	Form HTML e codice PHP.....	18
6	Connessione con Database in locale (Xampp).....	21

Guida al PHP

1 INTRODUZIONE

1.1 Prerequisiti

- HTML

1.2 Argomenti

- *PHP Core: controllo, stringhe, array*
- *OOP*
- **DBMS**
- *XML (web services)*
- *Sicurezza*
- *Generazione file PDF*

1.3 Strumenti di sviluppo

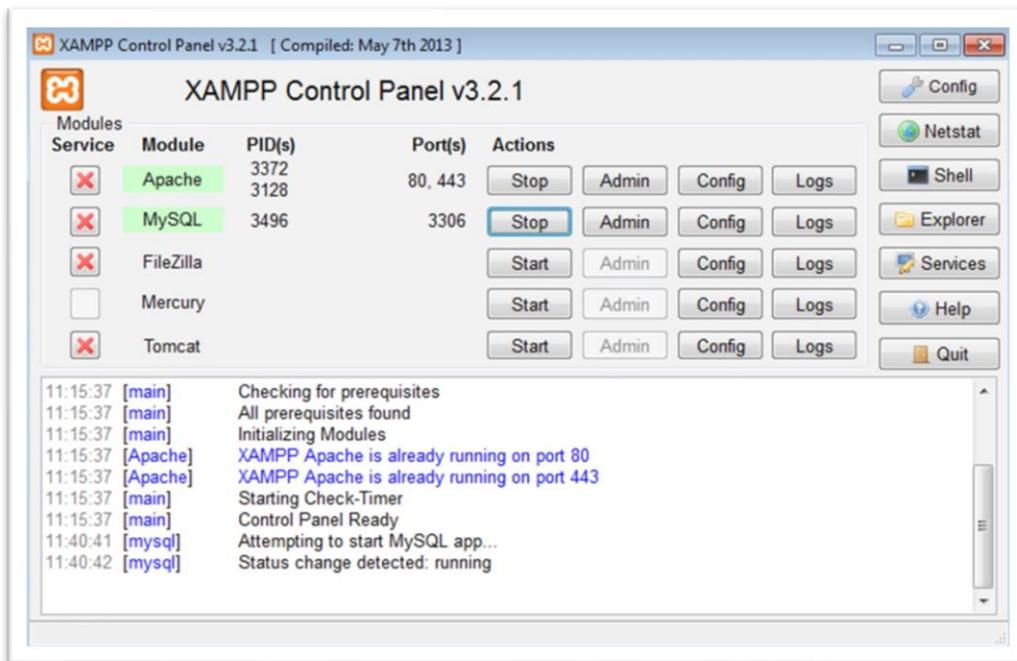
- Web Server: XAMPP
- Browser WEB: Firefox, Chrome
- IDE: Notepad++

2 Server Web

2.1 Configurare Apache tramite XAMPP

XAMPP è una piattaforma software multiplatforma e libera costituita da Apache HTTP Server, il database MariaDB e tutti gli strumenti necessari per utilizzare i linguaggi di programmazione PHP e Perl. Il nome è un acronimo dei software sopra citati (la X sta per x-platform, l'abbreviazione di cross-platform in lingua inglese ovvero multiplatforma).

Su queste piattaforme si appoggiano spesso siti web e altre piattaforme di sviluppo web dinamico e Content management system come WordPress, Drupal e Joomla!.



1 - L'ambiente XAMPP

Il web server XAMPP è scaricabile da <https://www.apachefriends.org/it/index.html>

La sua configurazione è molto semplice.

3 Esecuzione Lato Server e Lato Client

3.1 Lato Client JAVASCRIPT

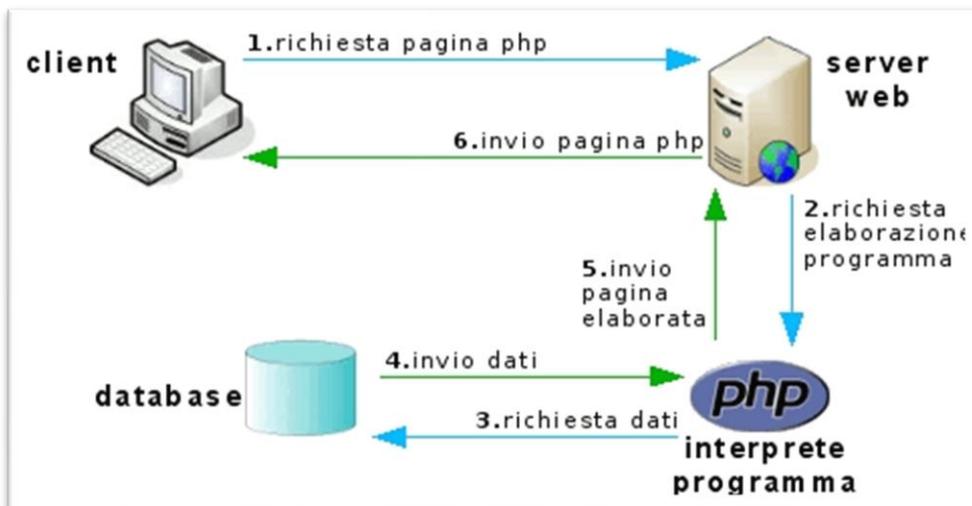
La trattazione di questo argomento verrà svolta in una guida a sé stante.

3.2 Lato Server PHP

Per far “girare” un’applicazione PHP abbiamo bisogno essenzialmente di 3 cose:

- Il **client** (o browser ad es. Chrome o Firefox) che effettua le richieste ad un Web server.
- Un **Web server** che fornisce al browser le pagine richieste restituendogli codice HTML. XAMMP è ad esempio un server web
- Un **programma** (in comunicazione con il web server) **che produce codice HTML** a partire da un file con estensione PHP (ossia una pagina in codice PHP)

- 1) Il Client elabora una richiesta di pagina dinamica in PHP.
- 2) Il Web Server, su cui è installato l’interprete PHP elabora la richiesta
- 3) I dati vengono elaborati e richiesti opportunamente. In genere si tratta di una *query* in SQL che permette di “interrogare” un Database
- 4) Il Database invia i dati richiesti all’interprete PHP
- 5) L’interprete PHP elabora i dati e “crea” una pagina di risposta
- 6) La pagina di risposta viene inviata al Client che la visualizza.



2 - Generazione di una pagina dinamica

4 Storia e Configurazione del PHP

4.1 Storia del PHP

Si rimanda a <https://it.wikipedia.org/wiki/PHP>

4.2 Configurazione

Per simulare la pubblicazione delle pagine PHP, che di fatto è “una richiesta remota fatta in locale dal nostro PC”, non possiamo mettere i file sorgente dove ci pare.

- In XAMMP la cartella è C:/XAMMP/htdocs

4.3 Come funziona un web server?

Tutto inizia quando digitiamo un indirizzo nella barra del browser.

Supponiamo di digitare: `http://www.PCremoto/percorsoFile/`

Alla pressione del tasto invio, il browser si connette al PC di nome “www.PCremoto” che risiede da qualche parte nel web e richiede al suo Web Server di fornirgli il codice HTML della pagina specificata in percorso “percorsoFile”, se questa pagina esiste.

Avviando il Web Server Apache sul nostro PC tramite l’interfaccia di XAMPP il server che viene contattato è quello che sta girando sul nostro PC.

- Apache va a cercare le pagine da noi richieste all’interno di una **directory predefinita** C:/XAMMP/htdocs

Nella **barra indirizzi** del **browser** dobbiamo invece scrivere:

- `http://127.0.0.1/Miepagine/file.htm`

o se abbiamo configurato in XAMMP come porta la 8080:

- `http://127.0.0.1:8080/Miepagine/file.htm`

Apache cercherà il *file.htm* in:

- C:/XAMMP/htdocs/Miepagine/file.htm

e restituirà al browser il codice HTML contenuto nel File.

5 Prima Pagina PHP

5.1 Creazione di una pagina statica

Copiate ed incollate il seguente codice nel Notepad++ **selezionando come linguaggio il PHP**

```
<html>
<head>
<title>Pagina Statica</title>
</head>
<body>
    Ciao, io sono una semplice pagina HTML!
</body>
</html>
```

Salvate poi il file con il nome **Statica.php** in:

- C:/XAMMP/htdocs/MyPages/

Digitate nella barra indirizzi del BROWSER:

- <http://127.0.0.1:8080/MyPages/statica.php>

Premete invio. Il risultato che otterrete sarà sempre lo stesso ad ogni pressione del tasto aggiorna.

5.2 Creazione di una pagina dinamica

Copiate ed incollate il seguente codice nel blocco note.

```
<html>
<head>
<title>Pagina Dinamica</title>
</head>
<body>
    Ciao, io sono una semplice pagina dinamica!
<br>

<?php
echo "Dal 1 gen 1970 sono passati i seguenti secondi ".time();
?>

<p>

La data di oggi e'
<?php echo " " . date("d-m-Y")
?>

<p>

<?php
echo "L'ora e' " . date("h:i:sa");
?>

</body>
</html>
```

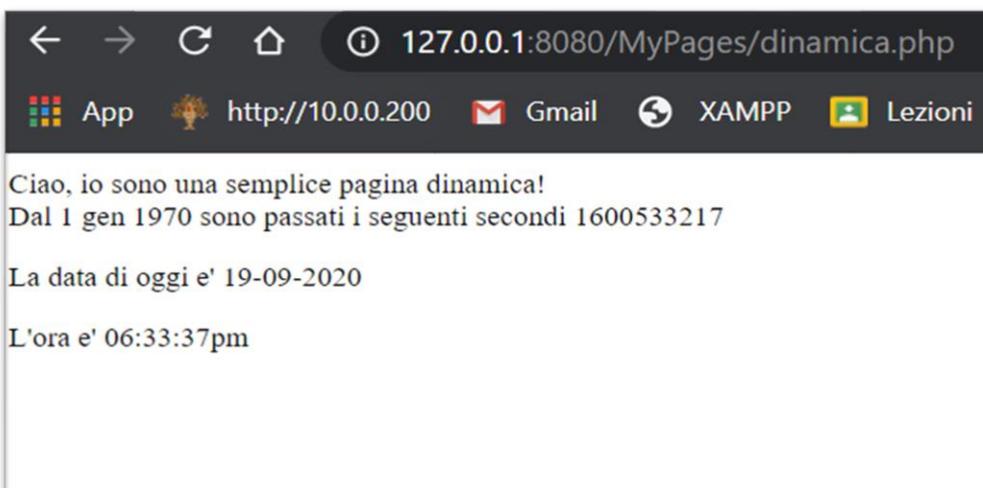
Salvate poi il file con il nome **dinamica.php** in:

- C:/XAMMP/htdocs/MyPages/

Digitate nella barra indirizzi del BROWSER:

- <http://127.0.0.1:8080/MyPages/dinamica.php>

Premete invio. Il risultato che otterrete cambierà ad ogni pressione del tasto aggiorna.



Un altro esempio è quello che permette di inserire la data di oggi in maniera dinamica. Il file è **data.php**

```
<html>

<head>
  <title>La data di oggi! </title>
</head>

<body>
  <p>
  Pagina dinamica in PHP che stampa la data di oggi.
  <br>

  <?php
  echo "Today is <br>";
  print(Date("l F d, Y"));
  echo "<br>";
  ?>
</body>

</html>
```



4 - Inserimento della data

5.3 Prime conclusioni

L'esempio differisce dal precedente per il codice aggiuntivo contenuto tra i tag `<?php.....?>`

Quando l'interprete PHP trova questi delimitatori esegue le istruzioni PHP contenute al loro interno e produce l'opportuno codice html. In questo caso ad esempio:

l'istruzione `echo` stampa la frase *"Dal 1 gen 1970 sono passati i seguenti secondi"* e la funzione `time()` restituisce i secondi a partire dalla data prima citata.

Se, una volta caricata la pagina, si prova a visualizzare il codice HTML prodotto (selezionando **HTML** da **visualizza** nella barra dei menu) il risultato che vedremmo sarebbe di questo tipo:

```
<html>
<head>
<title>Pagina Statica</title>
</head>
<body>
  Ciao , io sono una semplice pagina dinamica !
  Dal 1 gen 1970 sono passati i seguenti secondi 110022300
</body>
</html>
```

1 codice HTML prodotto

Nessuna traccia del codice PHP! Esso è stato sostituito dalla riga in rosso generata al momento della chiamata.

5.4 Sintassi PHP di base.

Uno script PHP può essere posizionato ovunque nel documento HTML.

Uno script PHP inizia con **<?php** e finisce con **?>**.

```
<?php  
// PHP code goes here  
?>
```

Un file PHP può contenere sia codice HTML che codice PHP.

5.5 Le variabili

L'essenza di un qualsiasi linguaggio di programmazione sono le variabili e anche il PHP non fa eccezione.

Le variabili sono individuate dal carattere "\$" e sono *Case-Sensitive* cioè, nell'esempio, \$color è diversa da \$COLOR o da \$coLOR.

```
<!DOCTYPE html>
<html>
<body>

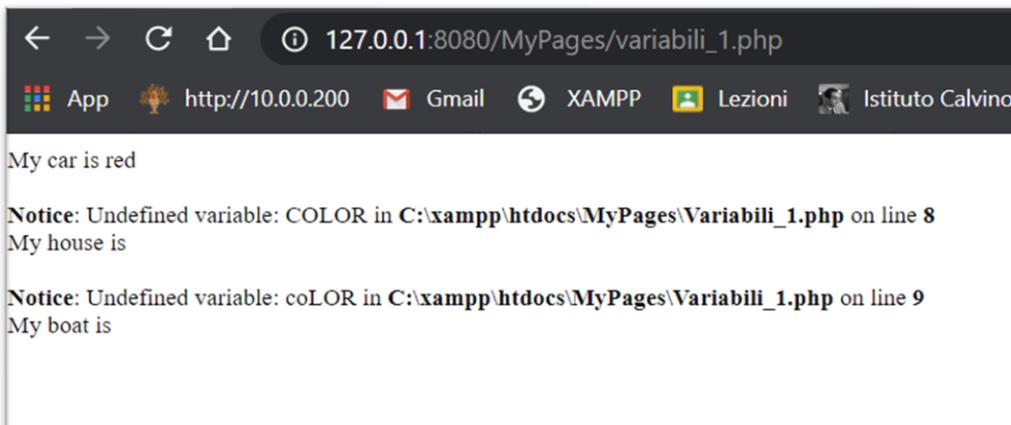
<?php
$color = "red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>

</body>
</html>
```

\$color è una variabile, in questo caso inizializzata col valore "red"

ATTENZIONE!
Le variabili in PHP sono **Case-Sensitive**

Il risultato nel Browser sarà:



Le variabili sono individuate dal carattere "\$".

Le variabili sono Case-Sensitive cioè, nell'esempio, \$color è diversa da \$COLOR o da \$coLOR

Una variabile può essere una lettera (come x o y) o avere una descrizione più estesa (età, prezzo, IVA_totale).

Regole per le variabili in PHP:

- Una variabile inizia sempre col carattere \$
- Una variabile inizia sempre con una lettera
- Una variabile non può iniziare con un numero
- Una variabile può contenere soltanto lettere, numeri e underscores (A-z, 0-9, e _)
- In nomi delle variabili sono case-sensitive (\$age e \$AGE sono due variabili distinte)

Output

Il comando PHP echo è in genere utilizzato per visualizzare i dati sullo schermo

Esempio:

<pre><?php \$txt = "PHP"; echo "I love \$txt!"; ?></pre>	I love PHP!
--	-------------

Vediamo la somma di due variabili:

<pre><!DOCTYPE html> <html> <body> Somma di due variabili <p> <?php \$x = 5; \$y = 4; echo "la somma è"; echo \$x + \$y; ?> </body> </html></pre>	Somma di due variabili la somma è 9
---	--

Il comando **print** può essere usato con o senza parentesi print o print()

<pre><?php print "<h2>PHP is Fun!</h2>"; print "Hello world!
"; print "I'm about to learn PHP!"; ?></pre>	PHP is Fun! Hello world! I'm about to learn PHP!
--	--

<pre><?php \$txt1 = "Learn PHP"; \$txt2 = "W3Schools.com"; \$x = 5; \$y = 4; print "<h2>" . \$txt1 . "</h2>"; print "Study PHP at " . \$txt2 . "
"; print \$x + \$y; ?></pre>	<p>Learn PHP Study PHP at W3Schools.com 9</p>
---	--

5.6 Tipi di dati in PHP

Le variabili possono memorizzare dati di diversi tipi e i diversi tipi di dati possono fare cose diverse. PHP supporta alcuni tipi di dati fra cui:

- String
- Integer
- Float
- Boolean
- Array
- Object (per la programmazione ad oggetti che verrà vista più avanti)
- NULL

5.6.1 STRING

Una stringa è una sequenza di caratteri, come "Hello world!".

Una stringa può essere qualsiasi testo racchiuso tra virgolette. Puoi utilizzare virgolette singole o doppie:

<pre><?php \$x = "Hello world!"; \$y = 'Hello world!'; echo \$x; echo "
"; echo \$y; ?></pre>	<pre>Hello world! Ciao Mondo!</pre>
---	-------------------------------------

5.6.2 INTEGER

Un tipo di dati intero è un numero non decimale compreso tra -2.147.483.648 e 2.147.483.647.

Regole per interi:

- Un numero intero deve contenere almeno una cifra
- Un numero intero non deve avere un punto decimale
- Un numero intero può essere positivo o negativo
- I numeri interi possono essere specificati in: notazione decimale (base 10), esadecimale (base 16), ottale (base 8) o binaria (base 2)

Nell'esempio seguente \$ x è un numero intero.

La funzione var_dump() di PHP restituisce il tipo di dati e il valore:

<pre><?php \$x = 5985; var_dump(\$x); ?></pre>	<pre>int(5985)</pre>
--	----------------------

5.6.3 FLOAT

Un float (numero in virgola mobile) è un numero con un punto decimale o un numero in forma esponenziale.

Nell'esempio seguente \$ x è un float. La funzione var_dump () di PHP restituisce il tipo di dati e il valore:

<pre><?php \$x = 10.365; var_dump(\$x); ?></pre>	float(10.365)
--	---------------

5.6.4 BOOLEAN

Un booleano rappresenta due possibili stati: TRUE o FALSE.

\$ x = vero;

\$ y = falso;

I booleani vengono spesso utilizzati nei test condizionali.

<pre><?php \$x = true; \$y = false; echo "valore di x: "; var_dump(\$x); echo "
"; echo "valore di y: "; var_dump(\$y); ?></pre>	valore di x: bool(true) valore di y: bool(false)
--	---

5.6.5 ARRAY

Un array memorizza più valori in una singola variabile.

Nell'esempio seguente \$ cars è un array.

La funzione var_dump () di PHP restituisce il tipo di dati e il valore:

<pre><?php \$cars = array("Volvo", "BMW", "Toyota"); var_dump(\$cars); ?></pre>	array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" }
---	---

5.6.6 NULL

Null è un tipo di dati speciale che può avere un solo valore: NULL.

Una variabile di tipo di dati NULL è una variabile a cui non è assegnato alcun valore.

Suggerimento: se una variabile viene creata senza un valore, viene automaticamente assegnato un valore NULL.

Le variabili possono essere svuotate anche impostando il valore su NULL:

<pre><?php \$x = "Hello world!"; \$x = null; var_dump(\$x); ?></pre>	NULL
--	------

5.7 Form HTML e codice PHP

Di seguito due piccoli listati. Il primo è un form HTML, **form1.html**, che lancia il codice PHP **esegui_form_1.php** che vanno, ovviamente, **salvati all'interno della stessa directory**:

- C:/XAMMP/htdocs/Miepagine/form1.html
- C:/XAMMP/htdocs/Miepagine/esegui_form_1.php

Il form HTML, una pagina statica...

```
<html>
  <head>
    <title>Un semplice form HTML</title>
  </head>
  <body>
    <form action="esegui_form_1.php" method="post">
      <table>
        <tr><td>Nome:</td><td><input type="text"
          name="VarNome"></td></tr>
        <tr><td>Cognome:</td><td><input type="text"
          name="VarCognome"></td></tr>
        <tr><td>Giorni Lavorati:</td><td><input type="text"
          name="VarGiorniLav"></td></tr>
        <tr><td>Costo Giornaliero:</td><td><input type="text"
          name="VarCostoG"></td></tr>
        <tr><td colspan="2"><input type="submit"
          value="Calcola"></td></tr>
      </table>
    </form>
  </body>
</html>
```

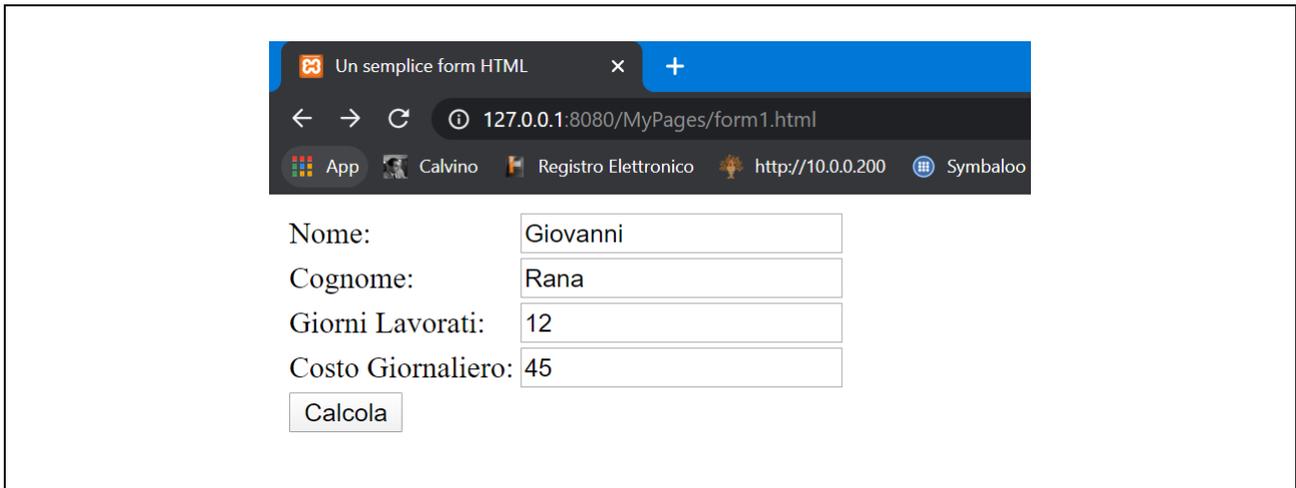
2 form1.html

... e il codice php che genera dinamicamente il contenuto sul browser.

```
<html>
  <head>
    <title>Esegue il form 1</title>
  </head>
  <body>
    <?php
      print("<h2>Il compenso di $_REQUEST[VarNome]
        $_REQUEST[VarCognome]</h3>\n");
      print("<h3>Giorni Lavorati:
        $_REQUEST[VarGiorniLav]</h3>\n");
      print("<h3>Costo Giornaliero: $_REQUEST[VarCostoG]
        euro</h3>\n\n");
    ?>
  </body>
</html>
```

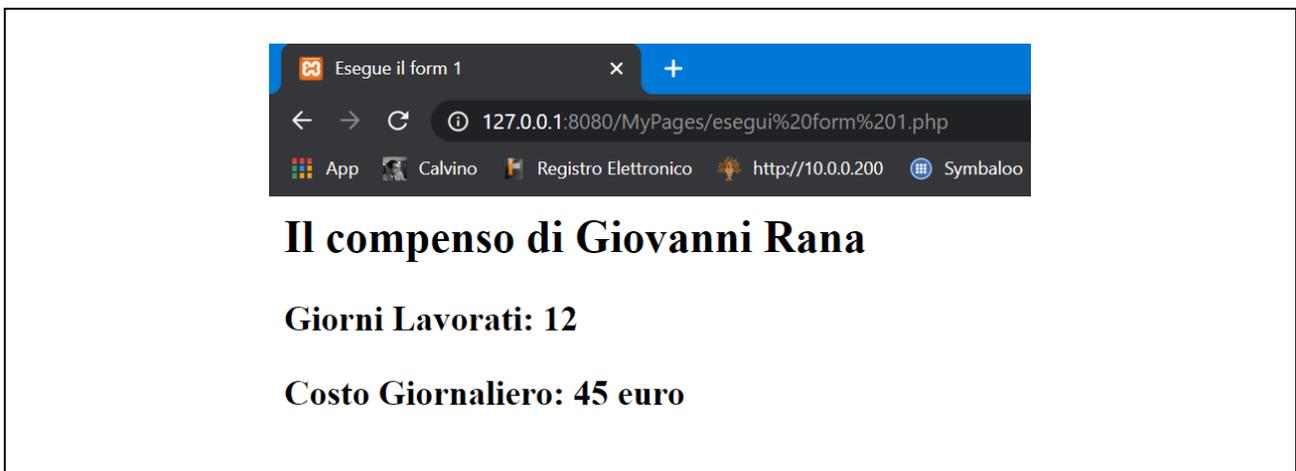
3 esegui_form_1.php

Risultato: il form...



4 form html

... la pagina generata in php...



5 esecuzione php

... e il codice sorgente generato dinamicamente.



6 codice sorgente generato dinamicamente

6 Connessione con Database in locale (Xampp)

Utilizzeremo come Data Base di esempio

INDICE DELLE FIGURE

1 codice HTML prodotto	10
2 form1.html	18
3 esegui_form_1.php	18
4 form html	19
5 esecuzione php	19
6 codice sorgente generato dinamicamente	19