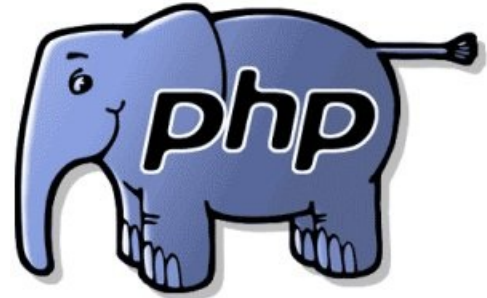




PHP programming



programmazione
lato server in PHP

Giulio Angiani
I.I.S. "Blaise Pascal" - Reggio Emilia

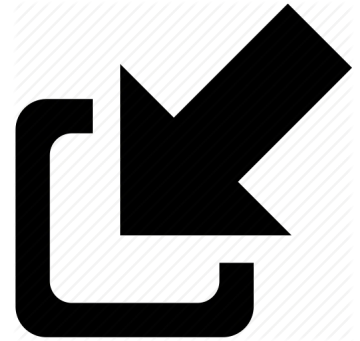


Input in PHP



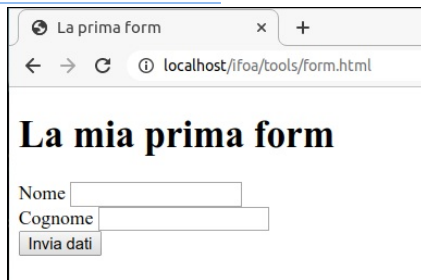
Gestire l'input verso pagine PHP

- Abbiamo visto come formattare dati presenti e già valorizzati nelle pagine scritte in PHP
- Ma come facciamo ad interagire con un utente esterno?
- Come arrivano dall'esterno i dati ad una pagina PHP?

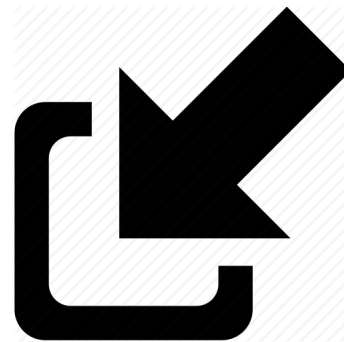


Gestire l'input verso pagine PHP

- Due modalità
 - ricezione dati via **GET**
 - ricezione dati via **POST**
- Cosa significa?
- Un utente può inviare dati ad una pagina PHP da un'altra pagina html utilizzando due metodi:
 - Cliccare su un **link**
 - Riempiendo un **form** e premendo un bottone per inviare i dati
- Un esempio di link è <http://www.ilmiosito.it>
- Un esempio di form è



The screenshot shows a web browser window with the title 'La prima form'. The address bar displays 'localhost/ifoatools/form.html'. The main content area contains the heading 'La mia prima form' in bold. Below the heading, there are two input fields: 'Nome' and 'Cognome'. At the bottom of the form, there is a button labeled 'Invia dati'.



Input via GET

- per passaggio di dati via **GET** si intende la possibilità per una pagina web di inviare ad un'altra pagina web, scritta in PHP, delle **informazioni** tramite **link**
- Ogni link è associato ad una **URL** che, come abbiamo visto precedentemente, è formato da varie parti
- Ad esempio: l'URL: **http://www.ifo.it/corsi.php?idutente=123&idcorso=ABC**
 - **http://** : protocollo
 - **www.ifo.it** : dominio
 - **/corsi.php** : percorso della pagina da cercare
 - **?** : carattere che divide la pagina dai **parametri**
 - **idutente=123&idcorso=ABC** : i dati da passare
- una **serie** di **parametro=valore** divisi da un **&** se sono più di uno



Input via GET

- Nel caso specifico: **idutente=123&idcorso=ABC**
 - due parametri
 - **idutente** che vale **123**
 - **idcorso** che vale **ABC**
- NOTA: i valori passati alla pagina sono **SEMPRE** delle stringhe anche se contengono valori numerici!
- Passando il mouse sul link vediamo in basso l'URI alla quale ci stiamo collegando
- A destra la pagina renderizzata, a sinistra il contenuto dell'html



```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Il mio primo link</title>
  </head>
  <body>
    <h1>Il mio primo link</h1>
    <a href="http://www.ifo.it/corsi.php?idutente=123&idcorso=ABC">Clicca qui per il corso</a>
  </body>
</html>
```



Input via GET

- ma dove finiscono i dati?
- Il link chiama la pagina php **corsi.php**
- I dati passati via **GET** finiscono in un **array associativo** predefinito da php chiamato **\$_GET**
- nel codice della pagina **corsi.php** ci limitiamo a stampare il contenuto dell'array **\$_GET**

```
<?php
    // corsi.php
    var_dump($_GET)
?>
```

PHP

- il comando **var_dump** ci stampa a video il contenuto di variabili. In questo caso vediamo che **\$_GET** è
 - un array associativo con **2** valori
 - uno con **chiave** idutente e **valore** 123 (stringa)
 - uno con **chiave** idcorso e **valore** ABC (stringa)



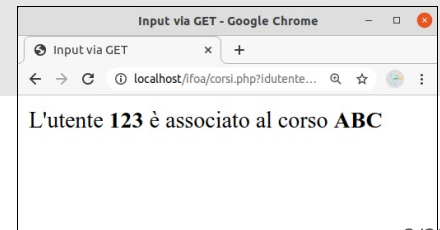
Input via GET

- proviamo a creare un OUTPUT che utilizzi questi valori
- dobbiamo:
 - creare una pagina che abbia un output in html corretto
 - recuperare i valori passati in input e utilizzarli separatamente
 - stampare l'output

```
<?php
// corsi.php
// recupero il valore dell' id utente e lo metto in una variabile di appoggio
$utente = $_GET["idutente"]; // N.B. La CHIAVE dell'array $_GET è quella usata nell'HTML
$corso = $_GET["idcorso"]; // Il nome della variabile può essere qualunque
// dopo aver recuperato i valori dall'$_GET li uso nella pagina html che sto creando
$html = "
    <!DOCTYPE html>
    <html>
        <head><title>Input via GET</title>
        </head>
        <body>
            L'utente <b>{$utente}</b> è associato al corso <b>{$corso}</b>
        </body>
    </html>
";
print($html); // stampo la pagina completa
```

?>

PHP



Input via GET - Esercizio

- Creare la pagina HTML **dati.html** che contiene un link alla pagina **dati.php**
- Nel link inserire due parametri
 - **nome** che vale **Alessandra**
 - **cognome** che vale **Bianchi**
- Nella pagina dati.php formattare un output come quello in figura prendendo i parametri dall'array **\$_GET**



- p.s. puoi provare ad inserire due link con nomi diversi che puntino alla stessa pagina di destinazione per vedere se funziona!

Input via GET - Soluzione

Il testo html deve contenere una riga come la seguente, per passare dei valori a **dati.php**

```
<a href='dati.php?nome=Alessandra&cognome=Bianchi'>Prova per Alessandra</a>
```

HTML

```
<?php
```

```
// corsi.php
// recupero il valore dell' id utente e lo metto in una variabile di appoggio
$nome = $_GET["nome"]; // N.B. La CHIAVE dell'array $_GET è quella usata nell'HTML
$cognome = $_GET["cognome"]; // Il nome della variabile può essere qualunque
// dopo aver recuperato i valori dall'$_GET li uso nella pagina html che sto creando
$html = "
    <!DOCTYPE html>
    <html>
        <head><title>Saluto</title>
        </head>
        <body>
            Ciao {$nome} {$cognome} !
        </body>
    </html>
";
print($html); // stampo la pagina completa
```

PHP

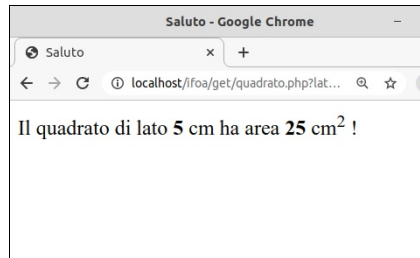
```
?>
```



Input via GET - Esercizio

- Creare una pagina HTML **quadrato.html** che contiene un link alla pagina **quadrato.php**
- Nel link inserire un parametro **lato** che rappresenta il lato di un quadrato e sia associato ad un valore numerico (per esempio 5)
- Nella pagina **quadrato.php** formattare un output indicando l'area del quadrato con il lato specificato

ad esempio: avendo nel link il parametro **lato=5** dovremmo ottenere un risultato come in figura



Input via GET - Soluzione

Il testo html deve contenere una riga come la seguente, per passare dei valori a **quadrato.php**

```
<a href='quadrato.php?lato=5'>Lato 5</a>
```

HTML

```
<?php
```

```
// recupero il valore del lato dal parametro lato e lo metto in una variabile di appoggio
// trasformo il valore in numero con l'istruzione predefinita intval
// perché l'input alle pagine PHP è sempre una stringa anche se ha un valore numerico
$lato = intval($_GET["lato"]); // N.B. La CHIAVE dell'array $_GET è quella usata nell'HTML
$area = $lato*$lato; // metto il valore dell'area in una variabile
$html = "
    <!DOCTYPE html>
    <html>
        <head><title>Area del quadrato</title>
        </head>
        <body>
            Il quadrato di lato <b>{$lato}</b> cm ha area <b>{$area}</b> cm<sup>2</sup> !
        </body>
    </html>
";
print($html); // stampo la pagina completa
?>
```

PHP



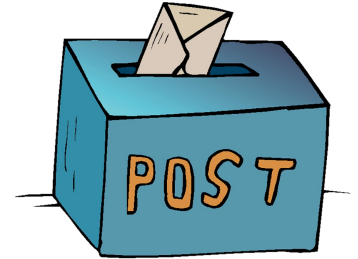
Input via GET - Esercizi per il lettore

- 1) Scrivere una pagina **numeri.html** che contiene un link alla pagina **confronto.php** Nel link inserire due parametri **a** e **b** che sono associati a valori **numerici** Nella pagina **confronto.php** indicare qual è il maggiore dei due o se sono uguali
- 2) Scrivere una pagina **rubrica.html** che contiene un link alla pagina **rubrica.php** Nel link inserire un parametro **nome** associato ad un valore **stringa** Nella pagina **rubrica.php** stampare il numero di telefono associato al nome ricevuto (se presente in un array associativo) altrimenti rispondere con "Nome non in rubrica"
- 3) Scrivere una pagina **genera.html** che contiene un link alla pagina **stampa.php** Nel link inserire un parametro **riga** associato ad un valore **intero** Nella pagina **stampa.php** stampare una tabella con tante righe quanto specificato nel parametro ricevuto



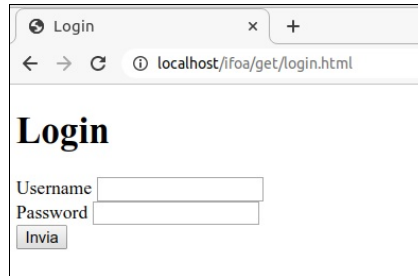
Input via POST

- per passaggio di dati via **POST** si intende la possibilità per una pagina web di inviare ad un'altra pagina web, scritta in PHP, delle **informazioni** tramite **form**
- Ogni form è associata ad una **ACTION** e contiene alcuni **campi** editabili dove l'utente inserisce i dati da inviare al programma php

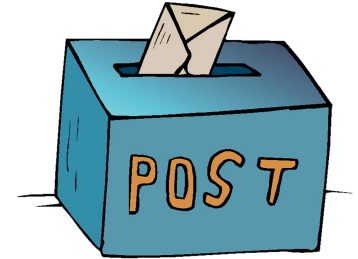


Input via POST

- Ad esempio: il form di **login** in figura contiene
 - 1 campo input con name **usr** di tipo **text** che conterrà lo **username**
 - 1 campo input con name **pwd** di tipo **password** che conterrà la sua password
 - 1 bottone (campo input di tipo **submit**) per l'invio dei dati
 - la specifica del tipo di invio **POST**
 - la **action** ovvero l'URL della pagina php che dovrà analizzare i dati



A screenshot of a web browser window. The title bar says "Login". The address bar shows "localhost/foa/get/login.html". The page content includes the heading "Login", followed by "Username" and "Password" labels next to input fields. Below the fields is a button labeled "Invia".



Input via POST

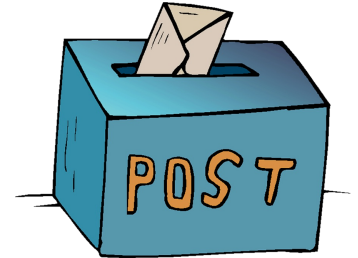
L'operazione di **submit** della **form** consiste nell'inviare alla pagina specificata nella **action** una serie di parametri associati ai valori inseriti

Nel caso specifico, inserendo

- **giulio** nel campo **usr**
- **ifoa** nel campo **pwd** (si vedono solo asterischi perché il campo è definito password)

invieremo alla pagina **login.php** i seguenti valori:

parametro	valore
usr	giulio
pwd	ifoa
submit	Invia



Input via POST

Vediamo come è scritta una form in html

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Login</title>
  </head>
  <body>
    <h1>Login</h1>
    <form action="login.php" method="post">
      Username <input type="text" name="usr"><br>
      Password <input type="password" name="pwd"><br>
      <input type="submit" name="submit" value="Invia"><br>
    </form>
  </body>
</html>
```



Input via POST - Leggere i valori passati

Vediamo come è scritta una form in html

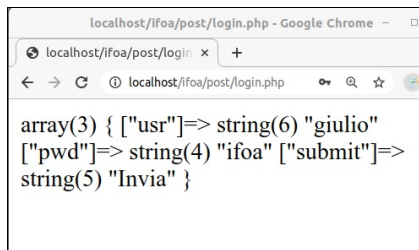
Similmente a come successo per il metodo GET:

- I dati passati via **POST** finiscono in un **array associativo** predefinito da php chiamato **\$_POST**
- nel codice della pagina **login.php** ci limitiamo a stampare il contenuto dell'array **\$_POST**

```
<?php
// login.php
var_dump($_POST)
?>
```

PHP

ottenendo:




Input via POST

- proviamo a creare un programma che faccia la **somma** di due numeri inseriti in un form e inviati ad una pagina PHP
- dobbiamo:
 - creare una pagina HTML che contenga un **form** (chiameremo questo file numeri.html)
 - inserire nel form due campi (che chiameremo **numero1** e **numero2**) di tipo text
 - inserire nel form un bottone per inviare i dati

```
<!DOCTYPE html>
<html>
  <head>
    <title>Somma</title>
  </head>
  <body>
    <h1>Login</h1>
    <form action="somma.php" method="post">
      Primo numero <input type="text" name="numero1"><br>
      Secondo numero <input type="text" name="numero2"><br>
      <input type="submit" name="submit" value="Somma"><br>
    </form>
  </body>
</html>
```

HTML



The screenshot shows a Google Chrome browser window titled "Somma - Google Chrome". The address bar displays "localhost/foa/post/numeri.html". The page content features a large heading "Login". Below the heading, there are two text input fields labeled "Primo numero" and "Secondo numero". At the bottom of the form is a submit button labeled "Somma".



Input via POST - Recupero dei dati

- I dati sono contenuti nell'array associativo \$_POST
- Le chiavi dell'array \$_POST sono i nomi dei campi indicati nel form
- Recuperiamo i valori che ci interessano da:
 - `$_POST["numero1"]`
 - `$_POST["numero2"]`

```
<?php
```

```
$n1 = intval($_POST["numero1"]); // N.B. La CHIAVE dell'array $_POST è quella usata nell'HTML  
$n2 = intval($_POST["numero2"]); // N.B. La CHIAVE dell'array $_POST è quella usata nell'HTML  
$somma = $n1+$n2; // metto il valore dell'area in una variabile  
$html = "
```

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head><title>Somma di due numeri</title>
```

```
  </head>
```

```
  <body>
```

```
    La somma dei due numeri è <b>{$somma}</b>
```

```
    <br>
```

```
    <a href='numeri.html'>torna al form</a>
```

```
  </body>
```

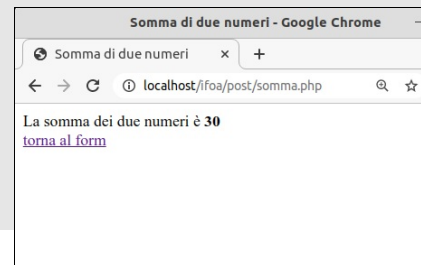
```
</html>
```

```
";
```

```
print($html); // stampo la pagina completa
```

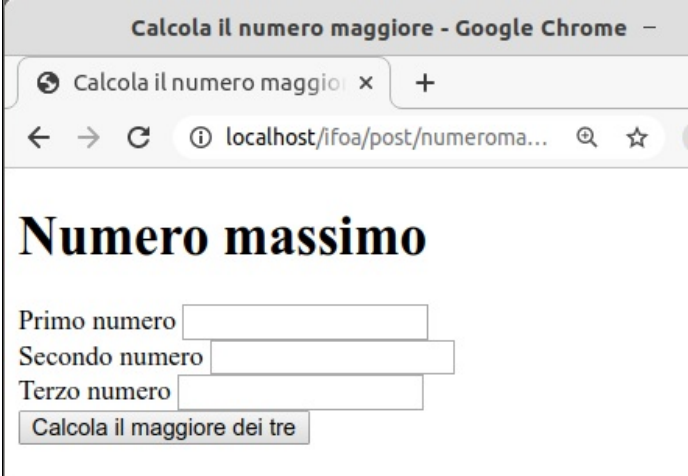
```
?>
```

PHP



Input via POST - Esercizio

Scrivere una pagina html **numeromassimo.html** dove inserire tre numeri interi Inviare la pagina con un POST a **massimo.php** che risponde indicando il numero massimo fra quelli inseriti dall'utente



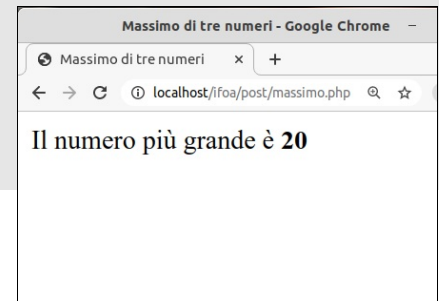
The screenshot shows a Google Chrome browser window with the title "Calcola il numero maggiore - Google Chrome". The address bar displays "localhost/foa/post/numeroma...". The main content area features the heading "Numero massimo" in a large, bold, black serif font. Below the heading, there are three input fields for numbers, each preceded by a label: "Primo numero", "Secondo numero", and "Terzo numero". At the bottom of the form is a button labeled "Calcola il maggiore dei tre".

Input via POST - Soluzione

- I dati sono contenuti nell'array associativo \$_POST
- Le chiavi dell'array \$_POST sono i nomi dei campi indicati nel form
- Recuperiamo i valori che ci interessano da:
 - \$_POST["numero1"]
 - \$_POST["numero2"]
 - \$_POST["numero3"]

```
<?php
    $n1 = intval($_POST["numero1"]);
    $n2 = intval($_POST["numero2"]);
    $n3 = intval($_POST["numero3"]);
    if (($n1 > $n2) && ($n1 > $n3)) {
        $massimo = $n1;
    }
    else if (($n2 > $n1) && ($n2 > $n3)) {
        $massimo = $n2;
    }
    else {
        $massimo = $n3;
    }
    [...]
?>
```

PHP



Input via POST - Gli altri campi della form

Una form HTML può contenere campi di vario tipo

- input
 - text
 - password
 - checkbox
 - radio
 - submit
 - date
 - file
- select
- textarea

Ref: https://www.w3schools.com/html/html_form_elements.asp



The screenshot shows a Google Chrome browser window with the title 'Campi di una form - Google Chrome'. The address bar shows 'localhost/foa/post/campi.html'. The page content includes the title 'Campi di una form' in a large, bold, serif font. Below the title, there are several form elements: a text input field labeled 'Testo libero', a password input field labeled 'Una Password', a checkbox labeled 'Una casella', a radio button labeled 'Scelta singola', and three radio buttons labeled 'Valore1', 'Valore2', and 'Valore3'. At the bottom, there is a dropdown menu labeled 'Casella a discesa' with 'Italiana' selected.

Input via POST - Recuperare i dati dalla form

- **NOTA :** Non per tutti i campi il valore passato è sempre presente in **\$_POST**
- E' presente **sempre** per i campi
 - text, password, submit, date, select, textarea
- E' presente **solo se** sono stati selezionati dall'utente per
 - checkbox, radio, file

qualche esempio:

```
Array (
    [testo] => "Giulio"
    [pwd] => "ifoa"
    [spunta] => "spunta1"
    [scelta] => "valore2"
    [discesa] => "ENG"
    [note] => "il mio primo form"
    [submit] => "Invia dati"
)
```

ARRAY \$_POST



The screenshot shows a web browser window with the title 'Campi di una form - Google Chrome'. The address bar shows 'localhost/foa/post/campi.h...'. The form itself has the title 'Campi di una form' and contains the following fields:

- Testo libero: A text input field containing 'Giulio'.
- Una Password: A password input field with masked characters '....'.
- Una casella: A checkbox that is checked.
- Scelta singola: Three radio buttons labeled 'Valore1', 'Valore2', and 'Valore3'. 'Valore2' is selected.
- Casella a discesa: A dropdown menu showing 'English'.
- A text area containing the text 'il mio primo form'.
- A submit button labeled 'Invia dati'.

Input via POST - Recuperare i dati dalla form

- **NOTA** : Non per tutti i campi il valore passato è sempre presente in **\$_POST**
- E' presente **sempre** per i campi
 - text, password, submit, date, select, textarea
- E' presente **solo se** sono stati selezionati dall'utente per
 - checkbox, radio, file

qualche esempio:

Campi di una form

Testo libero

Una Password

Una casella ☐

Scelta singola ☒ Valore1 ☐ Valore2 ☐ Valore3

Casella a discesa

inserisci qui le note

Note

```
Array (  
    [testo] => "Giulio"  
    [pwd] => "ifoa"  
    [discesa] => "ITA"  
    [note] => ""  
    [submit] => "Invia dati"  
)
```

ARRAY \$_POST



Input via POST - Esercizi per il lettore

1) Scrivere una pagina **login.html** che contiene un form con post alla pagina **login.php**

- Nella pagina inserire due campi **usr** e **pwd** che indicano nome utente e password
- Nella pagina **login.php** predisporre una array associativo **utenti** che abbia per chiave gli username e per valore le password.
- Il file login.php deve rispondere con:
 - "OK utente loggato" nel caso sia stata inserita una coppia utente/password esistente
 - "utente/password errati" nel caso sia stata inserita una coppia utente/password non esistente



Input via POST - Esercizi per il lettore

- Scrivere una pagina **registrazione.html** che contiene un form con post alla pagina **registrazione.php**
- Nella pagina inserire:
 - un campo **nome** e un campo **cognome** di tipo testo
 - un campo **annodinata** di tipo select con valori compresi fra 1989 e il 1999
 - un campo **privacy** di tipo checkbox
 - un campo **sex** di tipo radio con le due possibilità
- Il file **registrazione.php** deve rispondere con:
 - "OK utente registrato" se tutti i campi sono compilati e il campo privacy è spuntato e l'utente è maschio
 - "OK utente registrata" se tutti i campi sono compilati e il campo privacy è spuntato e l'utente è femmina
 - "Dati non completi" se l'utente non ha compilato i campi nome, cognome, sesso e annodinata
 - "Privacy obbligatoria" se l'utente ha compilato tutti i campi tranne la privacy





Giulio Angiani
I.I.S. "Blaise Pascal" - Reggio Emilia