



LABORATORIO DI INFORMATICA

Prof.ssa Patrizia Tarantino



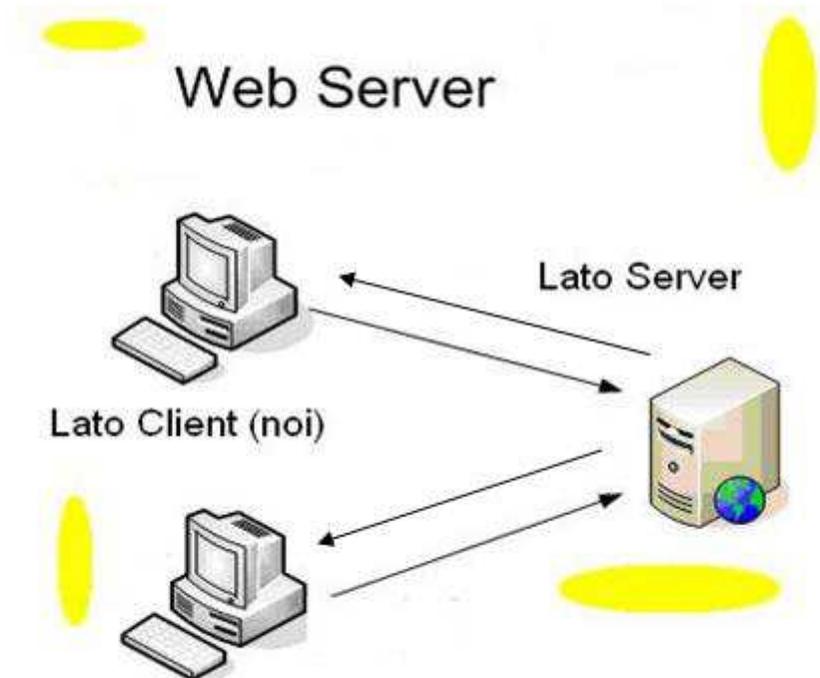


MODULO 1 - PROGRAMMAZIONE SERVER-SIDE



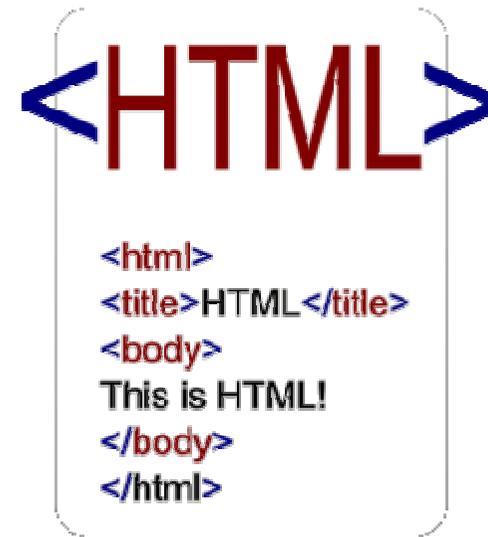
Programmazione Lato Client e Programmazione Lato Server

Quando parliamo di navigazione internet e programmazione Web, dobbiamo distinguere tra lato Client (noi) e lato Server (Host). Quando dal nostro PC digitiamo l'indirizzo del sito a cui siamo interessati, come <http://www.sito.htm>, il **browser** (FireFox ad esempio) invia la nostra richiesta al server WEB il software, che dopo averla interpretata, individua sul computer Host,(server di rete),il file richiesto e lo rinvia al browser dopo aver letto e interpretato il codice della pagina web.



CLIENT-SIDE

Il **WEB SERVER** è dunque un software che permette di gestire il server di rete (HOST). Quest'ultimo è sostanzialmente un computer avente grande capacità di memoria, che ospita le pagine web dei siti, che dopo essere state interpretate dal **web server** verranno visualizzate sul nostro Pc senza alcuna modifica. Per questo motivo questo tipo di pagina web si dice **statica (pagina HTML)**, quindi realizzata secondo i canoni della programmazione **Client Side (lato client)**.

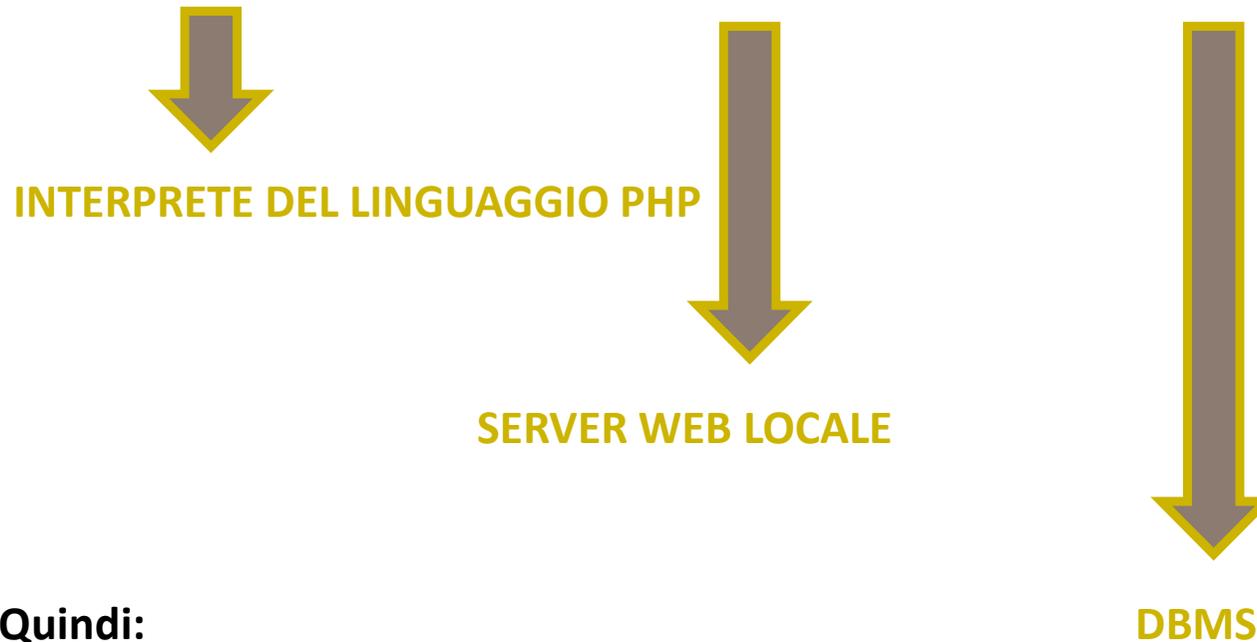


SERVER - SIDE

Se però si ha la necessità di accedere ad un database che risiede nel computer Host (cioè che risiede nel web), si dovrà procedere con la programmazione **Lato Server**, che genera pagine web di tipo **dinamico**. Questo perché accedere ad un data base, non significa solo leggere il contenuto delle sue tabelle, ma significa anche modificare tali contenuti sulla base delle azioni dell'utente navigatore. Per far fronte a questo tipo di problematica ci si riferisce alla tecnologia Server-side, (asp – php)



Di cosa abbiamo bisogno per programmare lato server?



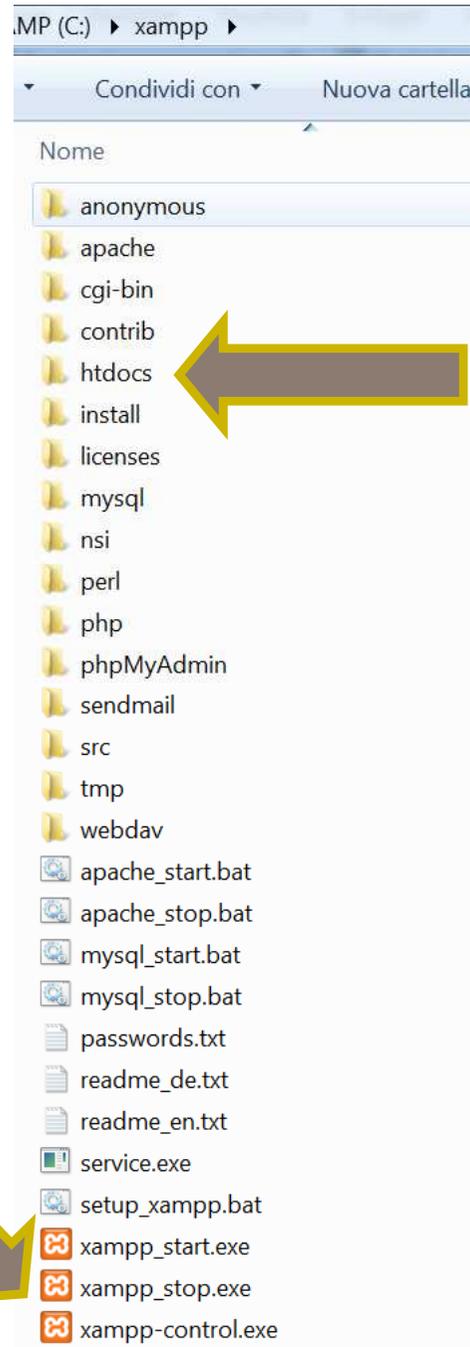
Quindi:

PER RISOLVERE QUESTO PROBLEMA POSSIAMO UTILIZZARE UN AMBIENTE INTEGRATO (ANCHE FREE) CHE FORNISCE UN WEB SERVER PER TESTARE LE PAGINE IN LOCALE, COMPLETO DI INTERPRETE PHP E E SERVER MYSQL PER GESTIRE IL DATABASE:



XAMPP

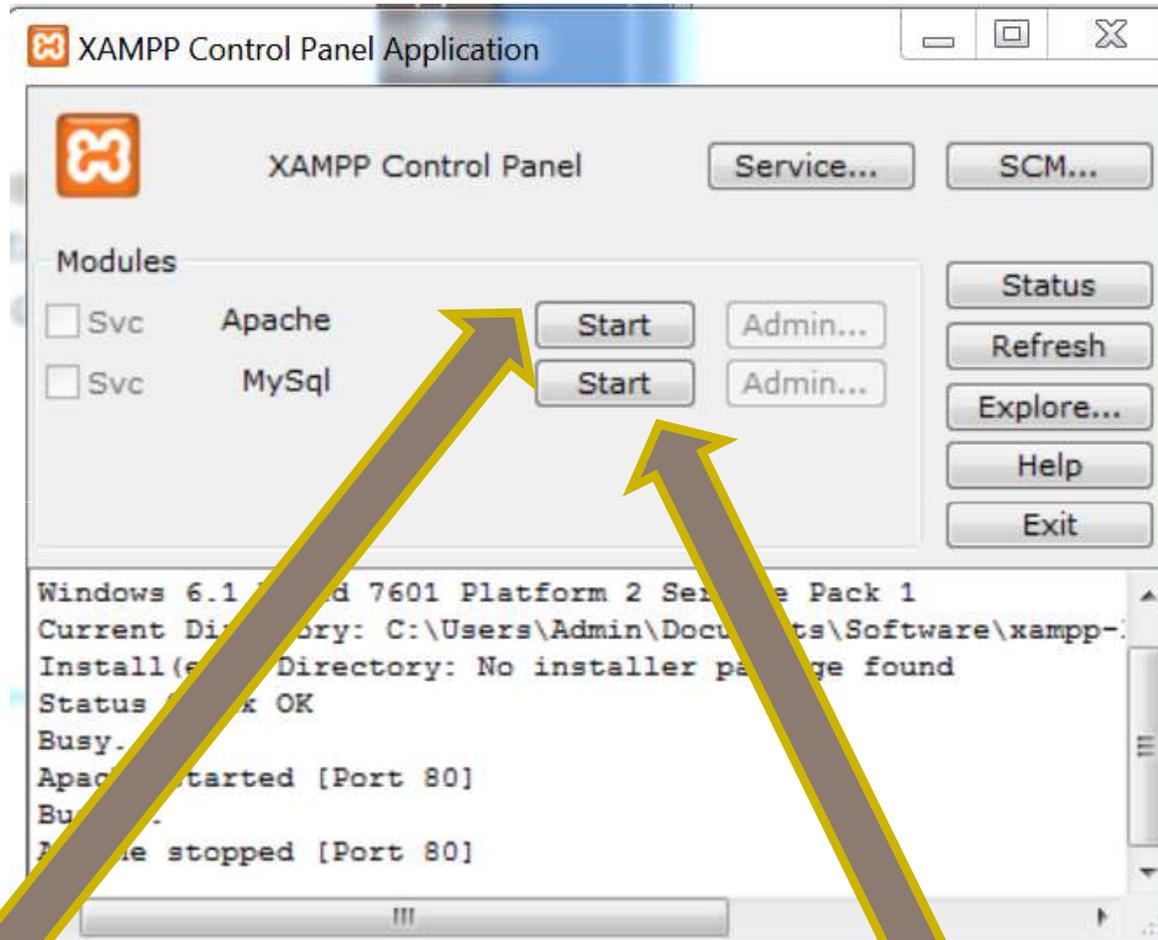
Di questo software, esiste una versione lite portable, che può essere attivata anche da chiavetta usb. Dopo aver estratto il contenuto compresso della cartella in C: nella cartella Xampp troveremo i tre file eseguibili che rispettivamente serviranno per attivare xampp, per disattivare e per accedere al pannello di controllo.



Le pagine web che dovremo testare, dovranno essere salvate nella cartella htdocs



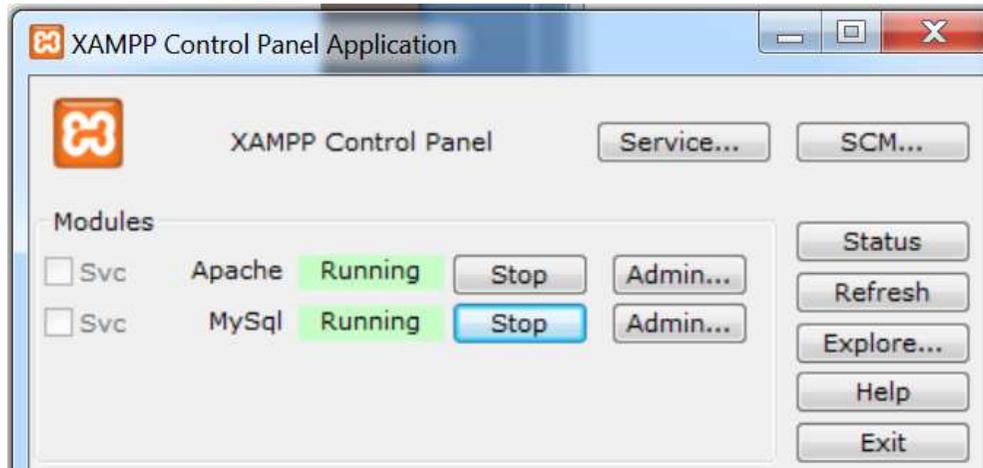
Pannello di controllo di XAMPP



Per attivare il server web e poter quindi testare le pagine php

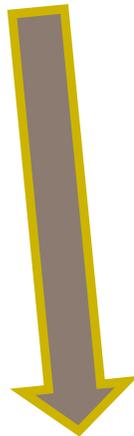
Per attivare il DBMS





Per testare se il nostro pc è pronto per iniziare a testare pagine php, verifichiamo collegandoci all'indirizzo ip della nostra postazione (dal prompt dei comandi digitiamo ipconfig)

Nella barra dell'indirizzo del nostro browser, scriviamo l'indirizzo IP ://index.html e se tutto è andato a buon fine si aprirà la pagina index di default caricata nella cartella htdocs



```

C:\Users\Admin>ipconfig

Configurazione IP di Windows

Scheda LAN wireless Wireless Network Connection 3:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

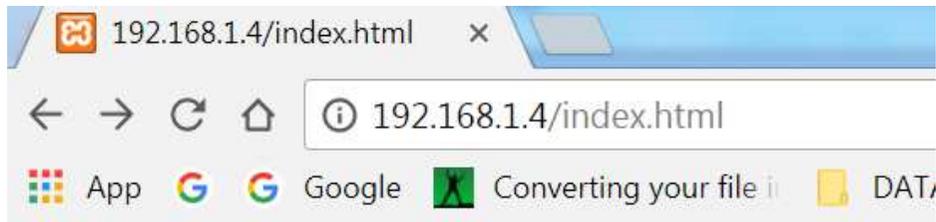
Scheda LAN wireless Wireless Network Connection 2:

    Suffisso DNS specifico per connessione: station
    Indirizzo IPv6 locale rispetto al collegamento . : fe80::e0b7:5a
15
    Indirizzo IPv4. . . . . : 192.168.1.4
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.1.1

Scheda Ethernet Bluetooth Network Connection:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

C:\Users\Admin>
    
```

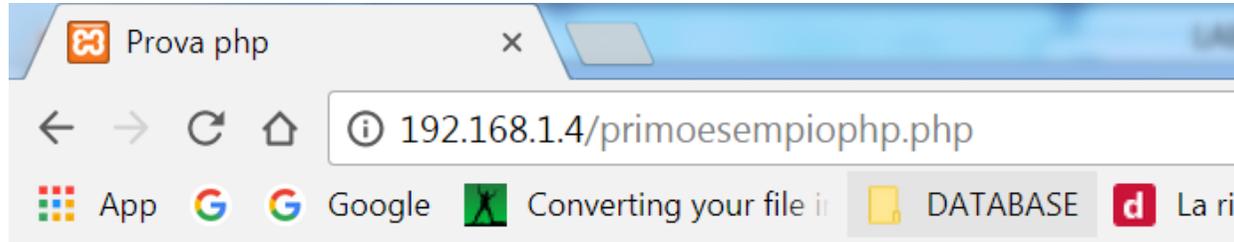


It works!

Ora con un editor, scriviamo questo codice e salviamo con estensione .php

```
<html>
<head>
<Title>Prova php</title>
</head>
<body>
<h1>PAGINA DI PROVA PHP</h1>
<?php
    echo "questo testo è scritto in <B>PHP</B> ";
?>
</body>
</html>
```





PAGINA DI PROVA PHP

questo testo è scritto in **PHP**

```
<html>
<head>
<Title>Prova php</title>
</head>
<body>
<h1>PAGINA DI PROVA PHP</h1>
<?php
    echo "questo testo è scritto in <B>PHP</B> ";
?>
</body>
</html>
```

Il codice PHP (script) viene racchiuso tra i tag `<?php ?>`, e può contenere i tag html.

La funzione **echo** invia una stringa al browser





Esistono 4 set di tag che possono essere utilizzati per delimitare blocchi di codice PHP. Soltanto due di questi

```
(<?php. . .?>  
<script language="php">. . .</script> )  
Funzionano sicuramente.
```

questi TAG – Linguaggio PHP NON SONO SEMPRE SUPPORTATI

```
<% echo ("Opzionalmente puoi utilizzare tag nello stile ASP"); %>
```

```
<script language="php"> echo ("alcuni editor non amano le istruzioni di elaborazione");</script>
```



Gli **operatori aritmetici** eseguono operazioni di puro calcolo matematico

Esempio	Nome	Risultato
$\$a + \b	Addizione	Somma del valore di $\$a$ e quello di $\$b$
$\$a - \b	Sottrazione	Rimanente di $\$b$ sottratto ad $\$a$
$\$a * \b	Moltiplicazione	Prodotto di $\$a$ e $\$b$
$\$a / \b	Divisione	Diviso di $\$a$ e $\$b$
$\$a \% \b	Modulo	Rimanente di $\$a$ diviso per $\$b$

Gli **operatori logici** sono simboli che effettuano delle scelte

Esempio	Nome	Risultato
$\$a \text{ and } \b	And	Vero se $\$a$ e $\$b$ sono veri entrambi
$\$a \text{ or } \b	Or	Vero se o $\$a$ o $\$b$ è vero
$\$a \text{ xor } \b	Or	Vero se o $\$a$ o $\$b$ è vero, ma non entrambi
$! \$a$	Not	Vero se $\$a$ non è vero
$\$a \&\& \b	And	Vero se $\$a$ e $\$b$ sono veri entrambi
$\$a \ \ \b	Or	Vero se o $\$a$ o $\$b$ è vero



Gli **operatori Bitwise** "guardano" all'interno delle variabile e scoprono se sono definite. Poi si regolano di conseguenza.

Esempio	Nome	Risultato
$\$a \& \b	And	Valido se $\$a$ e $\$b$ sono entrambi definiti
$\$a \b	Or	Valido se o $\$a$ o $\$b$ è definito
$\$a \wedge \b	Xor	Valido se o $\$a$ o $\$b$ è definito, ma non entrambi
$\sim \$a$	Not	Valido se $\$a$ non è definito e viceversa

Gli **operatori di comparazione** effettuano delle comparazioni tra variabili:

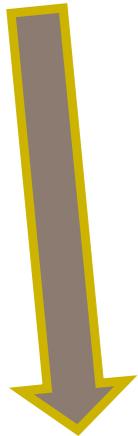
Esempio	Nome	Risultato
$\$a == \b	Uguale	Vero se il valore di $\$a$ e $\$b$ coincidono
$\$a != \b	Non uguale	Vero se il valore di $\$a$ e $\$b$ non coincidono
$\$a < \b	Più piccolo di...	Vero solo se $\$a$ è più piccolo di $\$b$
$\$a > \b	Più grande di...	Vero solo se $\$a$ è più grande di $\$b$
$\$a <= \b	Minore o uguale	Vero se $\$a$ è minore o uguale a $\$b$
$\$a >= \b	Maggiore o uguale	Vero se $\$a$ è maggiore o uguale a $\$b$



Gli operatori speciali

Esistono poi alcuni operatori speciali.

Il primo operatore è **=** e la sua funzione è quella di **assegnare** il valore ad un variabile (es. `$a = 1`).



L'altro operatore, chiamato di concatenazione, svolge appunto la funzione **di concatenare** più variabili (es. `$a = "Hello "; $b = $a . "World!"; // now $b = "Hello World!"`). Il suo simbolo è il punto (**.**).

Le variabili sono identificate da un nome preceduto dal segno **\$** ed i nomi delle variabili sono case-sensitive quindi distinguono il minuscolo dal maiuscolo



Separazione delle istruzioni

ogni istruzione termina con un punto e virgola.
Il tag di chiusura (?>) implica anche la fine di un'istruzione, perciò in tal caso si può omettere il ;

I Commenti

Il PHP supporta i commenti dei linguaggi 'C', 'C++' e della shell Unix. Per esempio:

```
<?php  
echo "Questo è un test"; // Questo è un commento su una linea C++  
/* Questo è un commento su più;  
linee ancora un'altra linea di commento */  
echo "Questo è un altro test";  
echo "Un ultimo test"; # Questo è un commento stile shell Unix
```



LE VARIABILI

Uno degli aspetti base d'ogni linguaggio di programmazione sta nelle variabili. Le variabili del PHP, in modo simile al Perl, ma **diversamente dal C, sono molto flessibili e vengono riconosciute automaticamente**. La variabile PHP viene definita nel suo contenuto, a differenza del C dove invece le variabili devono essere definite. **Le variabili possono nel PHP contenere di tutto**. Imparerete con il tempo che la variabile del PHP può essere usata come un notes dove inserire una funzione che deve essere richiamata più volte

L'inizializzazione di una variabile può essere fatta semplicemente assegnandole un valore (utilizzando l'operatore =). Non deve essere definita all'inizio, né tanto meno si deve definire di che tipo è. A volte però può essere utile e necessario farlo! Per assegnare anche il tipo alla variabile questa è la sintassi:

```
$variabile = (int) 1;
```



Esercizio 1.

Realizzare la pagina PHP che permetta di calcolare la somma di due numeri

```
1 <html>
2 <head>
3 <Title>SOMMA DI DUE NUMERI in Linguaggio php</title>
4 </head>
5 <body>
6 <h1>SOMMA DI DUE NUMERI in Linguaggio php</h1>
7 <?php
8     $A=5;
9     $B=10;
10    $C=$A+$B;
11    echo "$A + $B = $C";
12 ?>
13 </body>
14 </html>
```

Con Echo mandiamo a video la stringa che darà questo risultato se utilizziamo i doppi apici " che **mostrano il valore della variabile**

SOMMA DI DUE NUMERI in Linguaggio php

5 + 10 = 15

Utilizzando gli apici singoli, viene visualizzata la stringa così come è scritta

SOMMA DI DUE NUMER

\$A + \$B = \$C

Esercizio 2. (pagina chiamante)

Realizzare la pagina PHP che permetta di calcolare la somma dei due numeri trasmessi come parametri attraverso l'indirizzo URL



```
1 <html>
2 <body>
3 per testare IL METODO GET richiamiamo
4 la pagina php passando i valori di A e B attraverso l'URL
5 <a href="http://localhost/ESERCIZIPHP/testGETchiamato.php?A=20000&B=3000">CLICCA QUI</a>
6 </body>
7 </html>
```

La stringa che si trova dopo il punto interrogativo, contenente nomi e valori dei parametri, viene detta **query string**

L'esempio mostra un metodo non sicuro per passare i parametri a una pagina PHP che consiste nell'**aggiungere i parametri all'indirizzo URL della pagina Web che si sta richiamando**. L'indirizzo assume la seguente struttura generale:
http://localhost/pagina.php?nome1=valore&nome2=valore&...nomeN=valore
Dove sia il nome del parametro che il suo valore non deve contenere spazi. In caso il valore fosse "DI MATTEO", si dovrà indicare DI+MATTEO

Esercizio 2. (pagina chiamata)

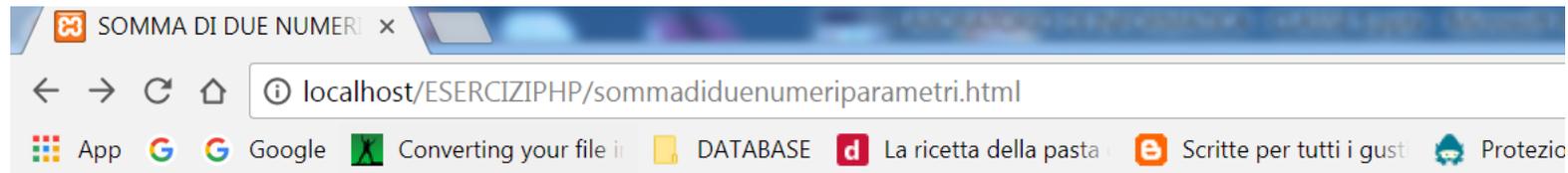
The image shows a web browser window displaying the output of a PHP script. The browser's address bar shows the URL: localhost/ESERCIZIPHP/testGETchiamato.php?A=20000&B=3000. The page content includes the text: "TESTIAMO GET: 20000+3000=23000" and "OSSERVATE NELLA BARRA DELL'INDIRIZZO I VALORI DI A E DI B". A large red arrow points from the browser window to a code editor window below it. The code editor shows the PHP script for testGETchiamato.php, which uses \$_GET to retrieve values for A and B and echoes their sum. The output text in the browser is enclosed in a red box, and a red arrow points from this box to the corresponding HTML output code in the editor.

```
1 <html>
2 <body>
3     TESTIAMO GET:
4     <?php
5     $A=$_GET["A"];
6     $B=$_GET["B"];
7     $C=$A+$B;
8     echo "$A+$B=$C"
9
10    <br>OSSERVATE NELLA BARRA DELL'INDIRIZZO I VALORI DI A E DI B SONO VISIBILI<br>
11    PROVATE A SOSTITUIRE I VALORI DI A E DI B DIRETTAMENTE NELLA BARRA <br>
12    DELL'INDIRIZZO, E RIAGGIORNATE LA PAGINA WEB.
13
14 </body>
</html>
```



Esercizio 3. METODO POST

Realizzare la pagina PHP che permetta di calcolare la somma dei due numeri indicati dall'utente.



SOMMA DI DUE NUMERI indicati dall'utente in Lingu

DIGITA IL PRIMO NUMERO

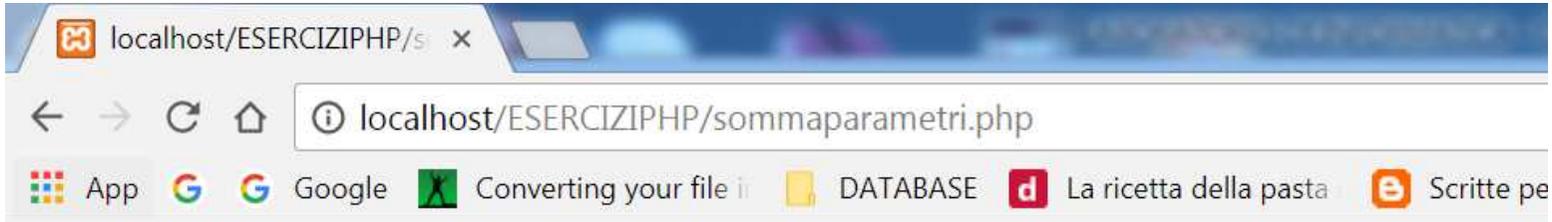
DIGITA IL SECONDO NUMERO

Il Metodo POST permette di inviare in modo riservato i dati digitati dall'utente nelle caselle di testo ad una pagina PHP che provvederà all'elaborazione

I dati che vengono passati attraverso il metodo POST sono memorizzati nell'array **\$_POST**.

Per inserire caselle di testo si utilizza `<input type="text" name="nome variabile">`

Per inserire un pulsante si utilizza `<input type="submit" value ="testo pulsante">`



I valori contenuti nella query string vengono memorizzati da PHP nell'array `$_GET`

```

C:\xampp\htdocs\ESERCIZIPHP\sommaparametri.php -
File Modifica Cerca Visualizza Formato Linguaggio
testGETchiamato.php testGET.php sommap
1 <html>
2 <body>
3 ECCO IL RISULTATO:
4 <?php
5 $A=$_POST["A"];
6 $B=$_POST["B"];
7 $C=$A+$B;
8 echo "$A+$B=$C"
9 ?>
10 </body>
11 </html>
    
```



Esercizio 4. METODO GET

Modifichiamo l'esempio precedente, sostituendo POST con GET:

The image shows a browser window at the top with the address bar containing `192.168.1.5/ESERCIZIPHP/sommaparametriGET.php?A=5&B=2`. Below the browser, the text reads: "ECCO IL RISULTATO: 5+2=7" and "Guarda cosa è cambiato nella barra dell'ind".

Two code editors are overlaid. The left editor shows the PHP code for `sommadiduenumeriparametriGET.html`:

```
1 <html>
2 <body>
3   ECCO IL RISULTATO:
4   <?php
5     $A=$_GET["A"];
6     $B=$_GET["B"];
7     $C=$A+$B;
8     echo "$A+$B=$C"
9   ?>
10  <B>Guarda cosa è cambiato nella barra dell'indirizzo!!!
11  </body>
12  </html>
```

The right editor shows the HTML code for `sommaparametriGET.php`:

```
1 <html>
2 <head>
3   <Title>SOMMA DI DUE NUMERI in Linguaggio php</title>
4 </head>
5 <body>
6   <h1>SOMMA DI DUE NUMERI Utilizzando GET</h1>
7   <form action = "sommaparametriGET.php" method="GET">
8     DIGITA IL PRIMO NUMERO<input type="text" name="A"/><BR>
9     DIGITA IL SECONDO NUMERO<input type="text" name="B"/><BR>
10    <input type="submit" value = "INVIA"/>
11  </form>
12 </body>
13 </html>
```

Differenza tra metodo POST e metodo GET

Con il metodo **POST** i dati inviati da una pagina all'altra restano riservati, pertanto questo metodo è utilizzato nella realizzazione di pagine web che contengono dati che vanno protetti, come i tipici FORM utilizzati per LOGIN E PASSWORD

Con il metodo **GET**, i dati vengono inviati direttamente tramite la bara dell'indirizzo pertanto i dati non sono protetti

Differenze tra...

GET

Indicato per pochi parametri

Solo parametri testuali

Possibilità di inserire un URL con parametri opportuni all'interno di una pagina HTML

es.

```
...<A HREF="http://www.altavista.com/cgi-bin/search?query=complexity">Ricerca Altavista sulla complessità</A>...
```

POST

Indicato per grandi quantità di dati

Dati in qualsiasi formato (testi, immagini, video)

Solo in risposta ad una form

LE FUNZIONI

PHP non richiede la dichiarazione del tipo di dato restituito dalla funzione ne il tipo dei parametri passati dalla stessa.

Questa funzione non restituisce alcun valore

La sintassi :

```
function nomeFunzione (parametri)  
    {  
    .....  
    }
```

Questa funzione restituisce il valore di somma, dunque si utilizza **return**

```
function somma($a, $b)  
    {  
    $somma = $a + $b;  
    return $somma;  
    }  
$somma = somma(10,5); // $somma sarà uguale ad 15
```

La funzione viene richiamata indicandone il nome seguito dalle () che eventualmente racchiuderanno i parametri



LE COSTANTI

Una costante può essere definita in PHP attraverso l'impiego del costrutto **define** che accetta due parametri in ingresso, questi ultimi sono: il nome della costante e il valore associato ad essa in fase di digitazione del codice

```
define('NOMECOSTANTE', valore numerico);  
Define('NOMECOSTANTE2', 'stringa');
```

A livello sintattico le costanti in PHP vengono dichiarate attraverso dei **nomi che iniziano con una lettera o con l'underscore**, per esse non è quindi previsto l'impiego del carattere iniziale \$ come avviene invece per le variabili. Per convenzione i nomi delle costanti vengono scritti interamente in **maiuscolo**.

Le costanti possono essere di tipo:
interi, float, stringhe e booleani o null

```
define('PREZZO', 100);  
define('NOME', 'Pippo');
```

