



# PHP

# Strumenti di Lavoro

Prof. Ing. Loris Penserini  
elpense@gmail.com



# Storia

PHP nasce nel 1994 dalla tastiera di Rasmus Lerdorf.

Inizialmente è stato presentato come una serie di Common Gateway Interface (CGI), cioè binari scritti in C per tracciare il numero di visite all'interno della sua homepage personale.

Da qui il primo nome del PHP: **Personal Home Page**.

In seguito però verrà rinominato in **Hypertext Preprocessor**.

# Pronti a sviluppare

Strumenti necessari per lo sviluppo di applicazioni Web con PHP:

- La parte server, per esempio Apache web server
- La parte client, un qualsiasi browser per pagine HTML
- L'interprete PHP
- Possibilmente un buon IDE

**Soluzione tecnologica proposta per queste lezioni:**

- **XAMPP → Apache e PHP**
- **NetBeans → IDE per lo sviluppo**
- **Chrome e Edge → Browser**

# XAMPP

XAMPP Control Panel v3.2.4 [ Compiled: Jun 5th 2019 ]

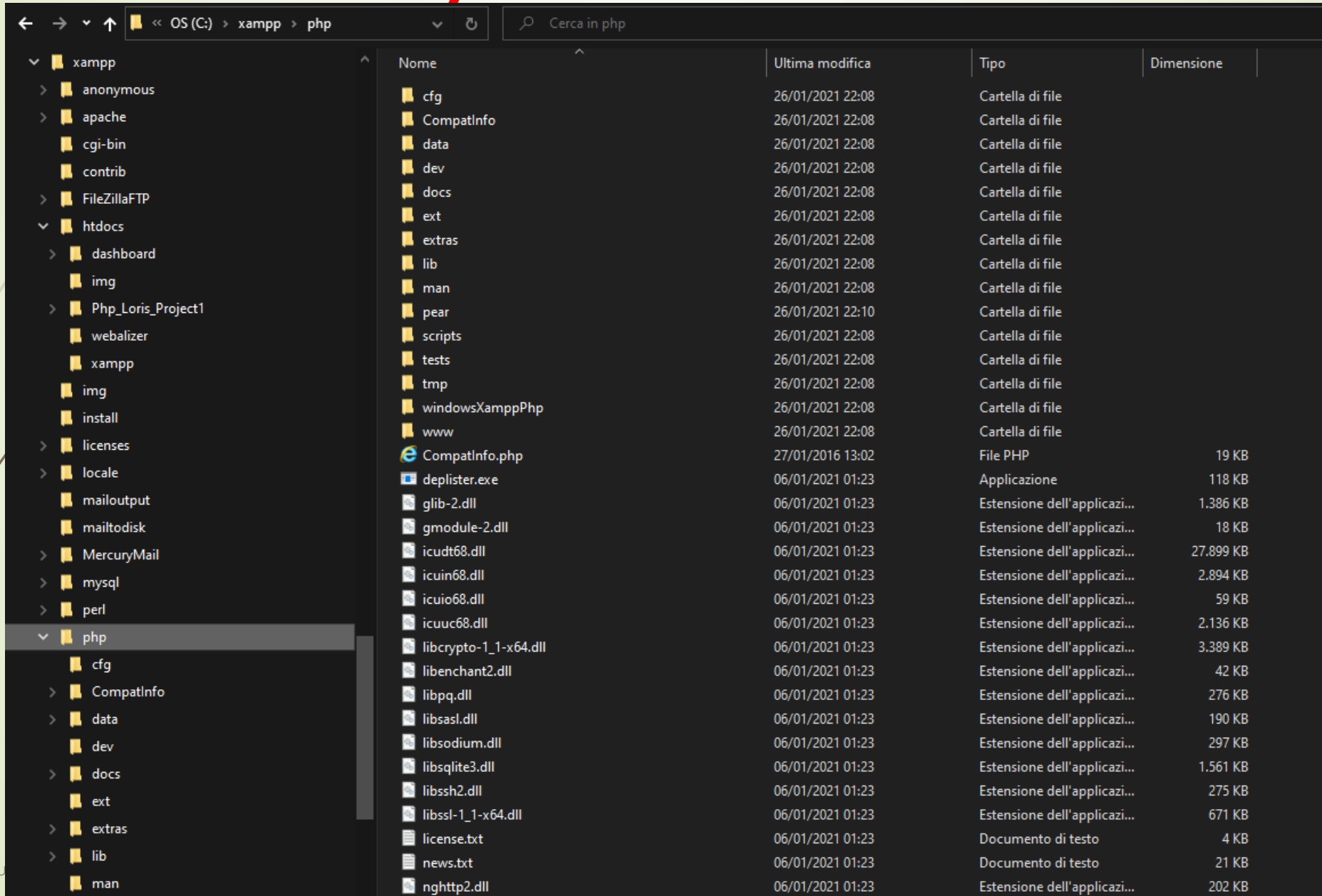
**XAMPP Control Panel v3.2.4**

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	9628 11892	80, 443	<b>Stop</b> Admin Config Logs
<input type="checkbox"/>	MySQL			Start Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

Config Netstat Shell Explorer Services Help Quit

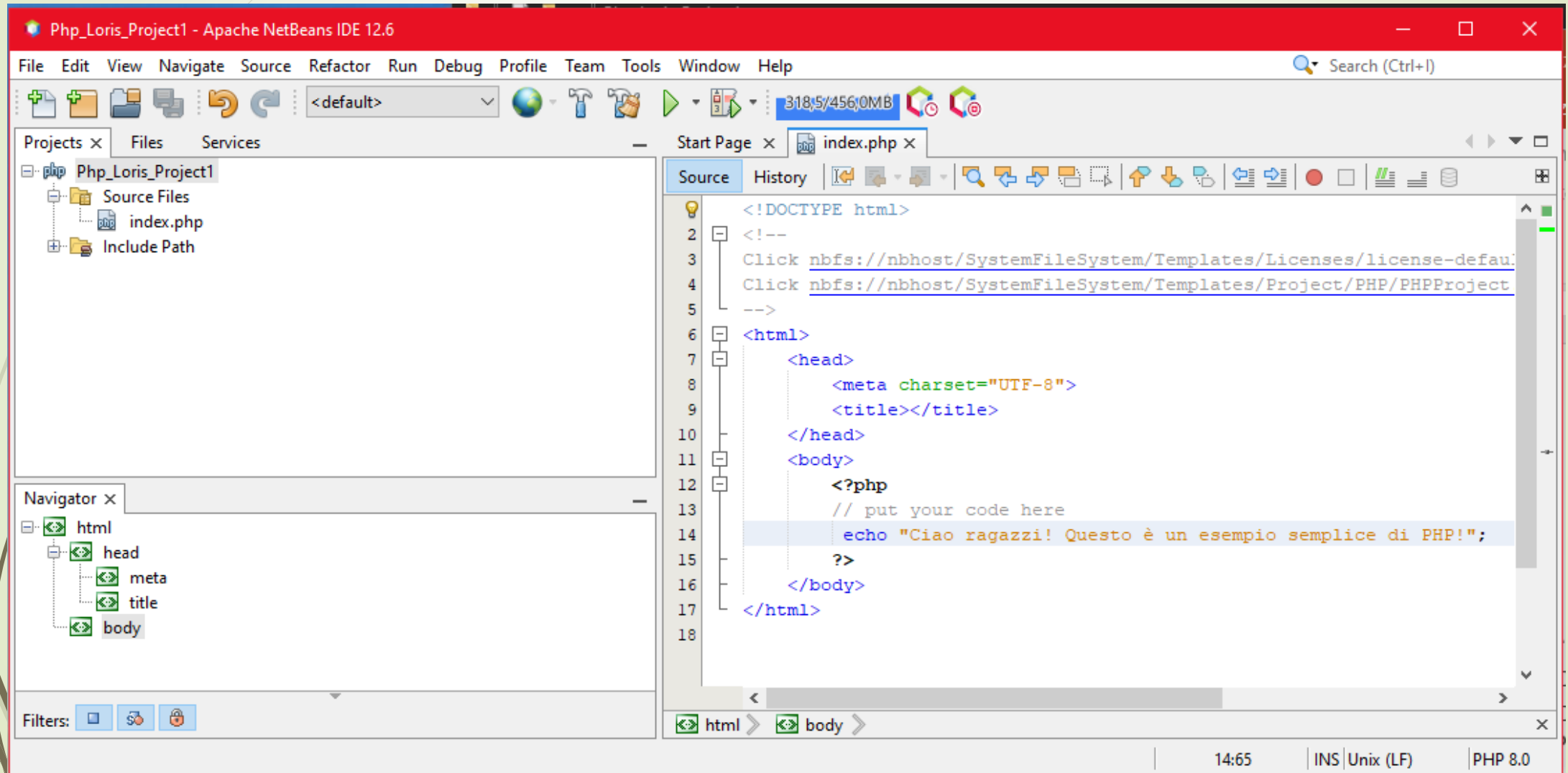
16:05:36 [main] Initializing Control Panel  
16:05:36 [main] Windows Version: Home 64-bit  
16:05:36 [main] XAMPP Version: 8.0.1  
16:05:36 [main] Control Panel Version: 3.2.4 [ Compiled: Jun 5th 2019 ]  
16:05:36 [main] You are not running with administrator rights! This will work for most application stuff but whenever you do something with services there will be a security dialogue or things will break! So think about running this application with administrator rights!  
16:05:36 [main] XAMPP Installation Directory: "c:\xampp\  
16:05:36 [main] Checking for prerequisites  
16:05:37 [main] All prerequisites found  
16:05:37 [main] Initializing Modules  
16:05:37 [main] Starting Check-Timer  
16:05:37 [main] Control Panel Ready  
16:05:41 [Apache] Attempting to start Apache app...  
16:05:41 [Apache] Status change detected: running

# XAMPP file system structure



Nome	Ultima modifica	Tipo	Dimensione
cfg	26/01/2021 22:08	Cartella di file	
CompatInfo	26/01/2021 22:08	Cartella di file	
data	26/01/2021 22:08	Cartella di file	
dev	26/01/2021 22:08	Cartella di file	
docs	26/01/2021 22:08	Cartella di file	
ext	26/01/2021 22:08	Cartella di file	
extras	26/01/2021 22:08	Cartella di file	
lib	26/01/2021 22:08	Cartella di file	
man	26/01/2021 22:08	Cartella di file	
pear	26/01/2021 22:10	Cartella di file	
scripts	26/01/2021 22:08	Cartella di file	
tests	26/01/2021 22:08	Cartella di file	
tmp	26/01/2021 22:08	Cartella di file	
windowsXamppPhp	26/01/2021 22:08	Cartella di file	
www	26/01/2021 22:08	Cartella di file	
CompatInfo.php	27/01/2016 13:02	File PHP	19 KB
deplister.exe	06/01/2021 01:23	Applicazione	118 KB
glib-2.dll	06/01/2021 01:23	Estensione dell'applicazi...	1.386 KB
gmodule-2.dll	06/01/2021 01:23	Estensione dell'applicazi...	18 KB
icudt68.dll	06/01/2021 01:23	Estensione dell'applicazi...	27.899 KB
icuin68.dll	06/01/2021 01:23	Estensione dell'applicazi...	2.894 KB
icuio68.dll	06/01/2021 01:23	Estensione dell'applicazi...	59 KB
icuiuc68.dll	06/01/2021 01:23	Estensione dell'applicazi...	2.136 KB
libcrypto-1_1-x64.dll	06/01/2021 01:23	Estensione dell'applicazi...	3.389 KB
libenchant2.dll	06/01/2021 01:23	Estensione dell'applicazi...	42 KB
libpq.dll	06/01/2021 01:23	Estensione dell'applicazi...	276 KB
libsasl.dll	06/01/2021 01:23	Estensione dell'applicazi...	190 KB
libsodium.dll	06/01/2021 01:23	Estensione dell'applicazi...	297 KB
libsqlite3.dll	06/01/2021 01:23	Estensione dell'applicazi...	1.561 KB
libssh2.dll	06/01/2021 01:23	Estensione dell'applicazi...	275 KB
libssl-1_1-x64.dll	06/01/2021 01:23	Estensione dell'applicazi...	671 KB
license.txt	06/01/2021 01:23	Documento di testo	4 KB
news.txt	06/01/2021 01:23	Documento di testo	21 KB
nghttp2.dll	06/01/2021 01:23	Estensione dell'applicazi...	202 KB

# Ambiente di Sviluppo Integrato (IDE)



# Server Web su una qualsiasi cartella

Se lo sviluppatore vuole che il browser si colleghi ad un determinato progetto PHP, dovete operare le seguenti configurazioni:

- **Variabile di ambiente per eseguire PHP da riga di comando**
- **Impostare un localhost virtuale su una directory corrente**



# Eseguire PHP da riga di comando

Configurare il sistema operativo (Windows, Mac o Linux) per poter eseguire le applicazioni php da riga di comando. Configurare le variabili di ambiente (di sistema o di utente) in modo da specificare dove il S.O. possa ritrovare il file eseguibile dell'interprete PHP.

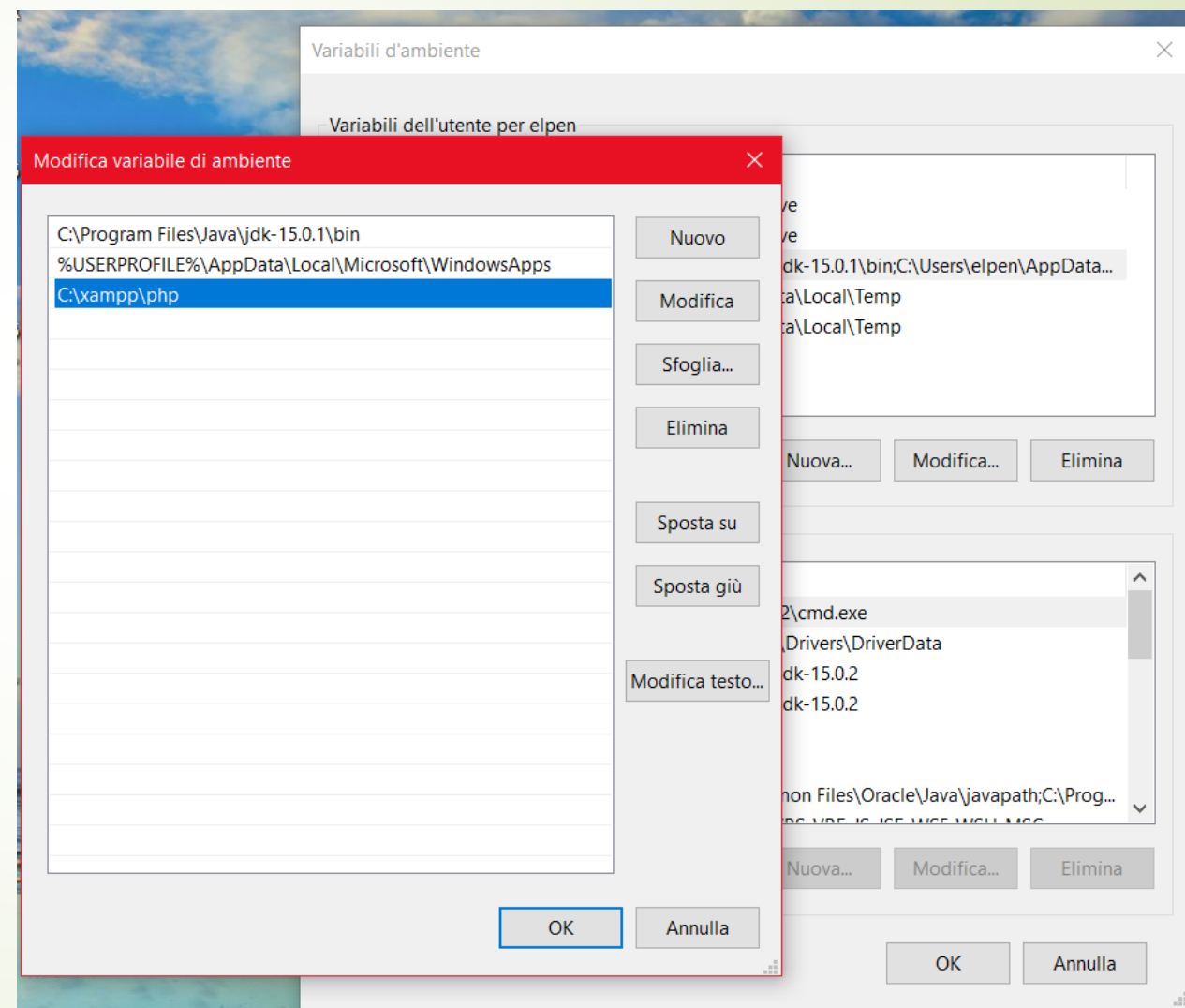
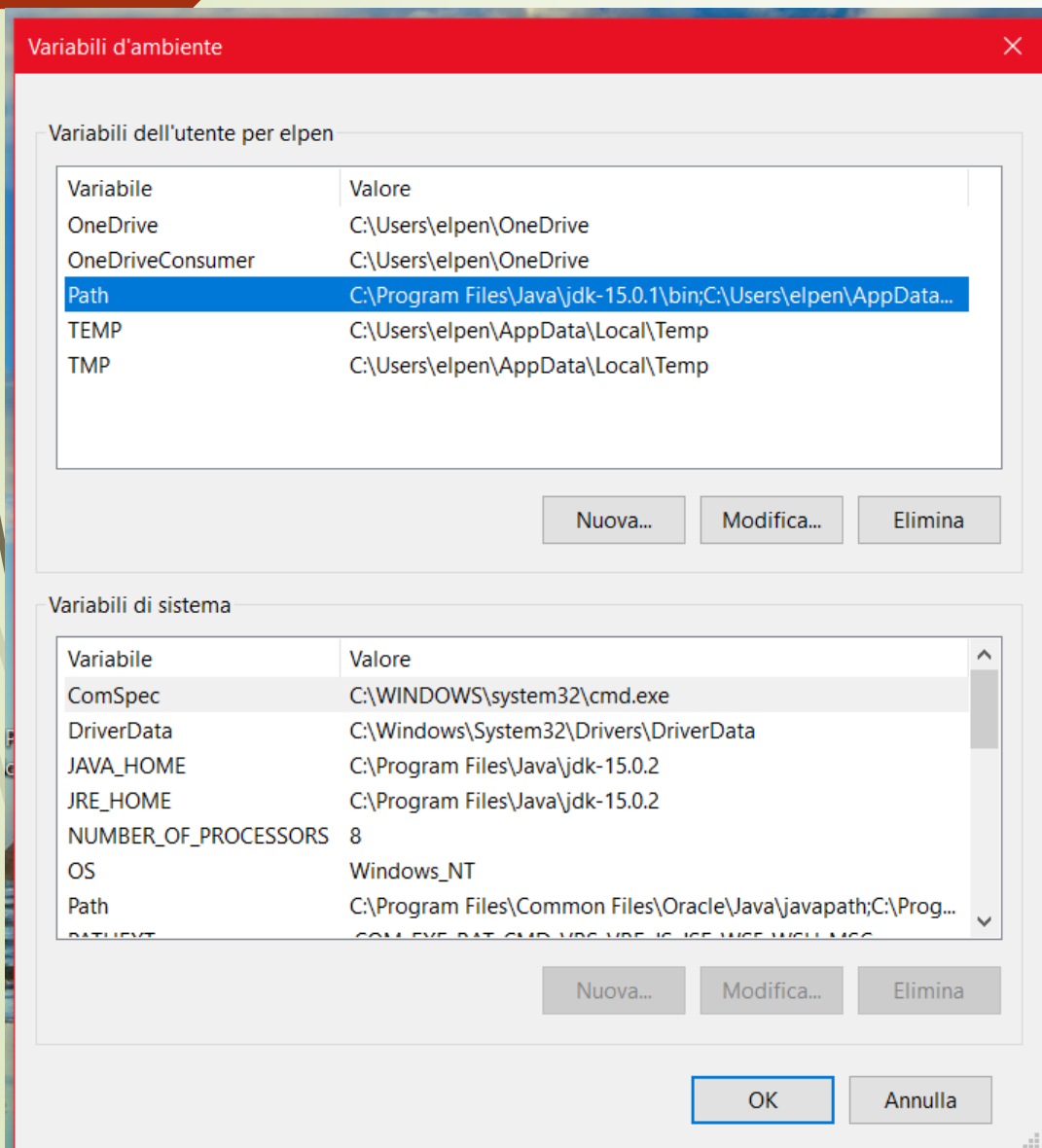
Nel caso di Windows, e utilizzando XAMPP, il file eseguibile (*php.exe*) si trova:

**...\xampp\php**

la variabile d'ambiente da modificare è: **PATH**



# Variabili di ambiente



# Impostare un Server Web

Da Windows, aprire una shell ms-dos con CMD e posizionarsi nella cartella del progetto creato con NetBeans.

Eseguire il comando:

**php -S localhost:8080** (Non includere http://...)

**Questo comando avvierà un server Web in ascolto sulla porta 8080 utilizzando la directory corrente in cui ci si trova come root del documento.**

Inserendo nella barra degli indirizzi del browser

**http://localhost:8080**

Dovresti vedere eseguire il tuo progetto PHP.

# Indirizzare una nuova directory corrente

Per sovrascrivere la root del documento predefinita (cioè la directory corrente), usa il flag -t :

```
php -S <host/ip>:<port> -t <directory>
```

Ad esempio se hai una directory public/ nel tuo progetto puoi servire il tuo progetto da quella directory usando

```
php -S localhost:8080 -t public/
```

# Caratteristiche

Linguaggio C	Linguaggio PHP
Tipizzazioni molto forte	Tipizzazione debole
Paradigmi supportati: imperativo, procedurale, strutturato	Paradigmi supportati: imperativo, funzionale, object-oriented, procedurale, riflessivo
Supporta i puntatori	Non supporta i puntatori
Utilizzato principalmente per software Desktop	Utilizzato principalmente per applicativi Web
Operazioni con le stringhe particolarmente complicate	Operazioni con le stringhe molto semplici da effettuare
Non supporta I/O filesafe	Supporta I/O filesafe
Non supporta system calls	Supporta system calls
Tipi di dati numerici supportati: int8_t, uint8_t, int16_t, uint16_t, int32_t, uint32_t, int64_t, uint64_t, int, unsigned int, signed char, unsigned char, short, unsigned short, long, unsigned long, long long, unsigned long long	Tipi di dati numerici supportati: 32-bit signed int, 64-bit signed int (long integer)
Tipi di dati float supportati: single precision float, doublelong double	Tipi di dati float supportati: double precision float
Supporta array di dimensioni fisse	Non supporta array di dimensioni fisse
Non supporta array associativi	Supporta array associativi

# Modalità di inserimento di codice PHP

Uno script PHP è semplicemente una pagina HTML all'interno della quale viene inserito il codice, cioè le istruzioni che costituiscono il programma, per cui **PHP è un linguaggio HTML-embedded**.

```
<html>
<head>
  <meta charset="UTF-8">
  <title>ITS Turismo Marche - Corso PHP</title>
</head>
<body>
  <?php
  // Esempio PHP
  echo "<h1>Ciao ragazzi! Questo è un esempio semplice di PHP!</h1>";
  ?>

  <?php
  // Esempio PHP
  phpinfo();
  ?>

</body>
</html>
```

# Output...

```
<html>
<head>
  <meta charset="UTF-8">
  <title>ITS Turismo Marche - Corso PHP</title>
</head>
<body>
  <?php
    // Esempio PHP
    echo "<h1>Ciao ragazzi! Questo è un esempio";
  >

  <?php
    // Esempio PHP
    phpinfo();
  >
</body>
</html>
```

Ciao ragazzi! Questo è un esempio semplice di PHP!

## PHP Version 8.0.1



System	Windows NT DESKTOP-TAHMGPH 10.0 build 19042 (Windows 10) AMD64
Build Date	Jan 5 2021 23:36:34
Build System	Microsoft Windows Server 2016 Standard [10.0.14393]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cscript /nologo /e:javascript configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build\dep-aux\oracle\x64\instantclient_19_9\sdk,shared" "--with-oci8-12c=c:\php-snap-build\dep-aux\oracle\x64\instantclient_12_1\sdk,shared" "--with-oci8-19=c:\php-snap-build\dep-aux\oracle\x64\instantclient_19_9\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20200930
PHP Extension	20200930
Zend Extension	420200930
Zend Extension Build	API420200930,TS,VS16
PHP Extension Build	API20200930,TS,VS16
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, zlib.*, bzip2.*

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v4.0.1, Copyright (c) Zend Technologies

zend®engine





# Dalle CGI al PHP



# Dalle CGI al PHP

In un'architettura Web di tipo client/server oltre alle interazioni consentite dal protocollo HTTP, i server Web possono consentire ad applicazioni esterne (Web client) di far eseguire, da remoto, codice sul server, tramite il protocollo di **Common Gateway Interface (CGI)**.

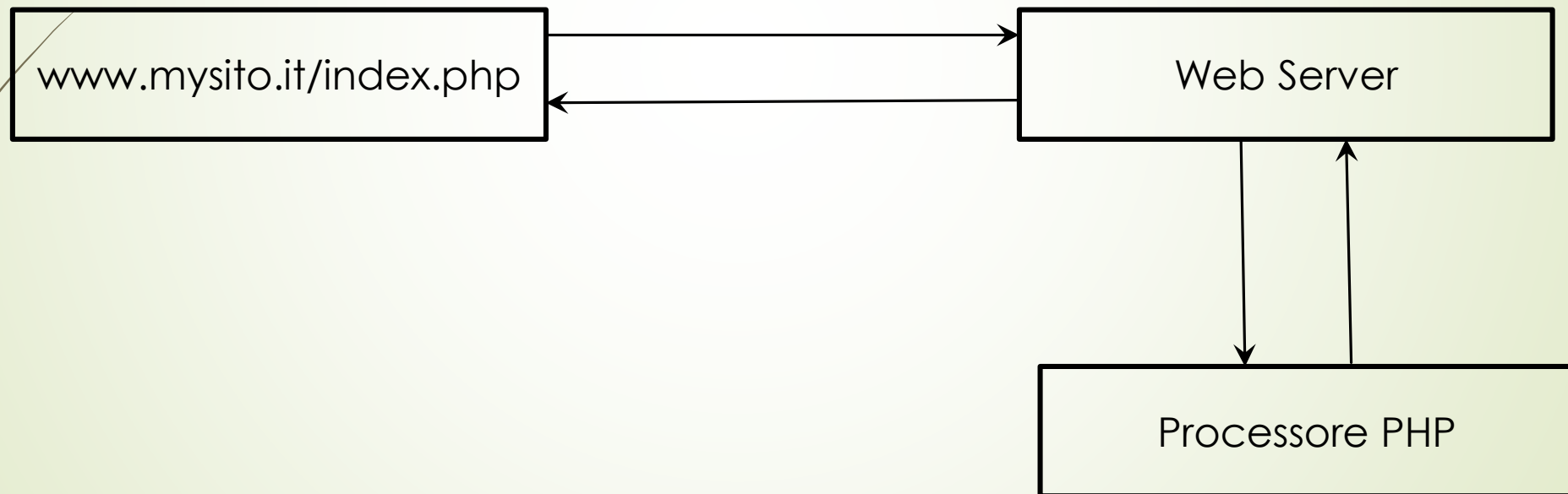
CGI introduce un nuovo tipo di risorsa, cioè il server **HTTP serve risorse dinamiche generate da un'applicazione esterna**. Il server HTTP agisce da tramite tra lo user client e l'applicazione esterna.

Lo standard CGI stabilisce come un'applicazione esterna (o script CGI) comunichi con il Web server.

**Quindi, uno script CGI è un eseguibile all'interno di un S.O. che può essere implementato in qualsiasi linguaggio di programmazione, come per esempio in PHP.**

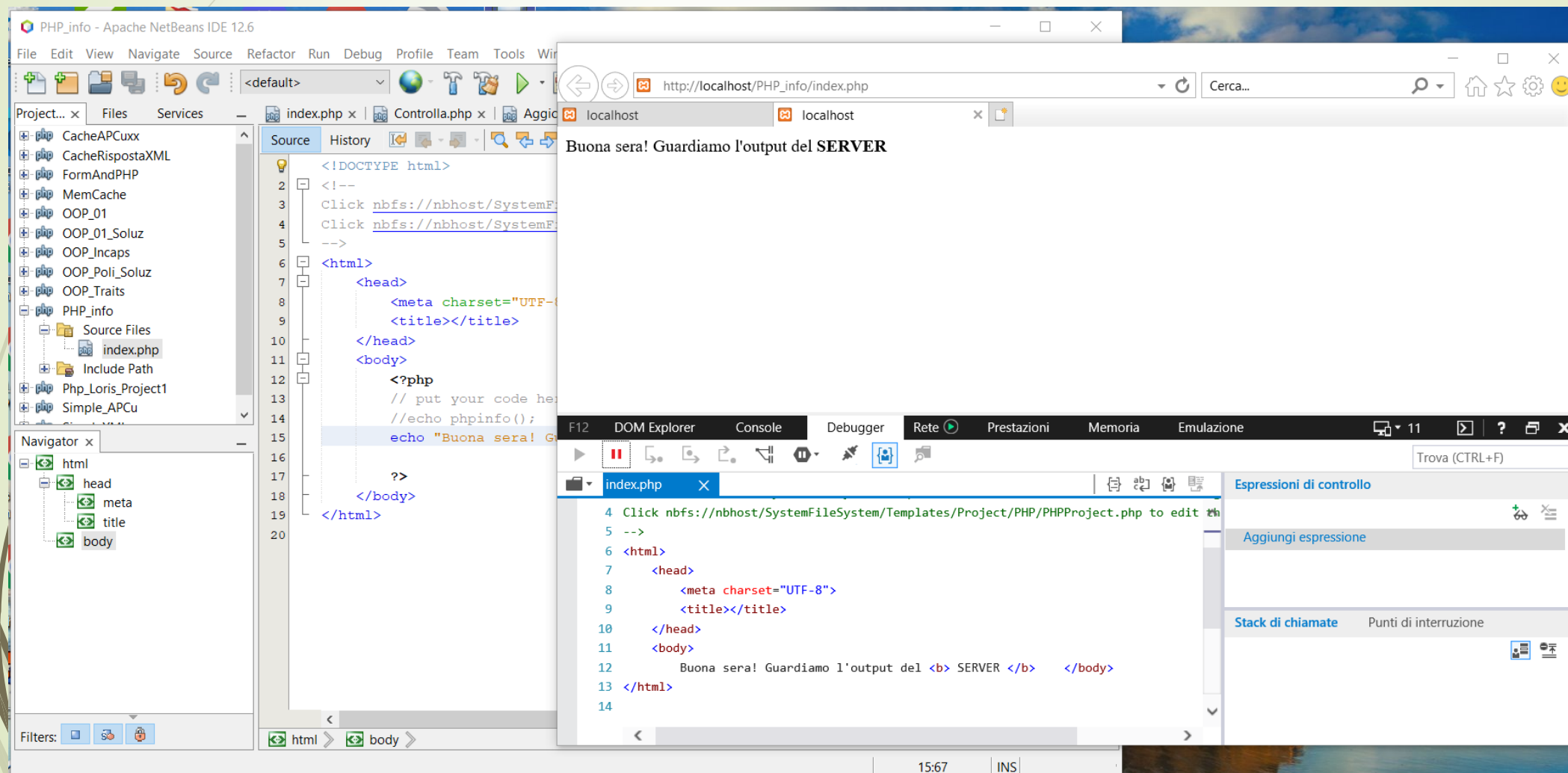
# Architettura di comunicazione in PHP

Tutto il codice contenuto nel file .php (tra i tag `<?php ... ?>` viene inviato al preprocessore PHP, per l'interpretazione e l'esecuzione. Poi i risultati vengono spediti al Server che li rigira al browser del Client...



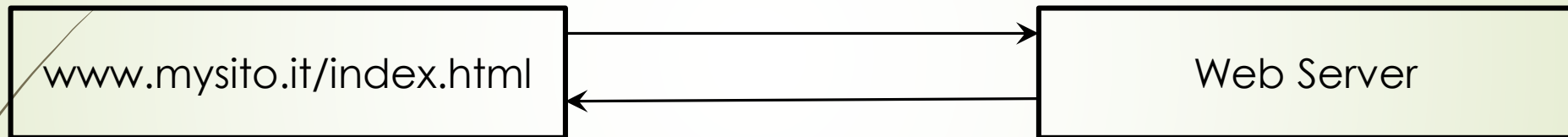
# Architettura di comunicazione in PHP

Tutto il codice contenuto nel file .php (tra i tag `<?php ... ?>`) viene inviato al preprocessore PHP, per l'interpretazione e l'esecuzione. Poi i risultati vengono spediti al Server che li rigira al browser del Client... **SOLO I RISULTATI**



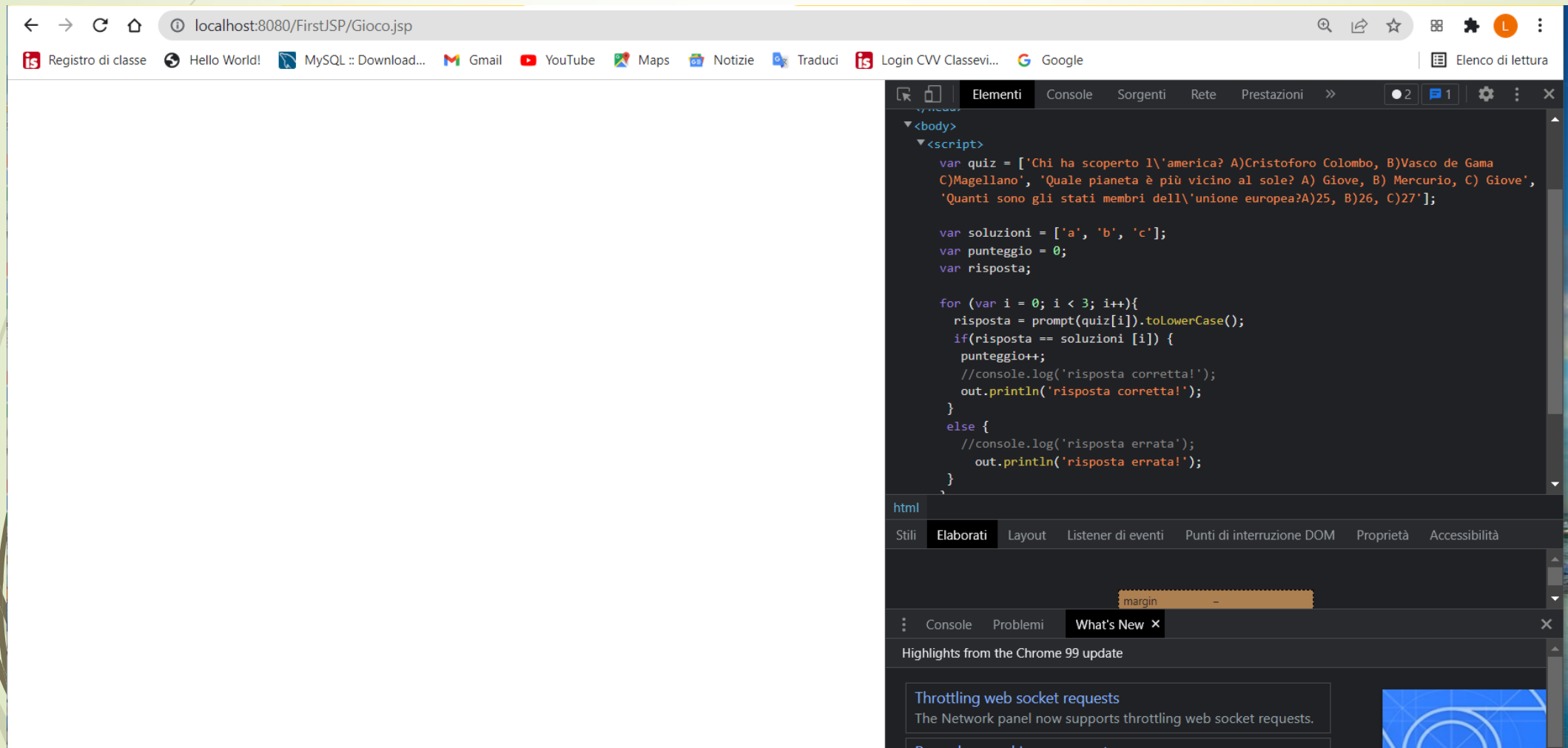
# Architettura di comunicazione in JSP

Il codice JavaScript viene scaricato sul PC dell'utente e poi viene interpretato ed eseguito all'interno del browser...



# Architettura di comunicazione in JSP

Tutto il codice contenuto nel file .html (tra i tag `<script>... </script>` viene eseguito dal browser del Client...



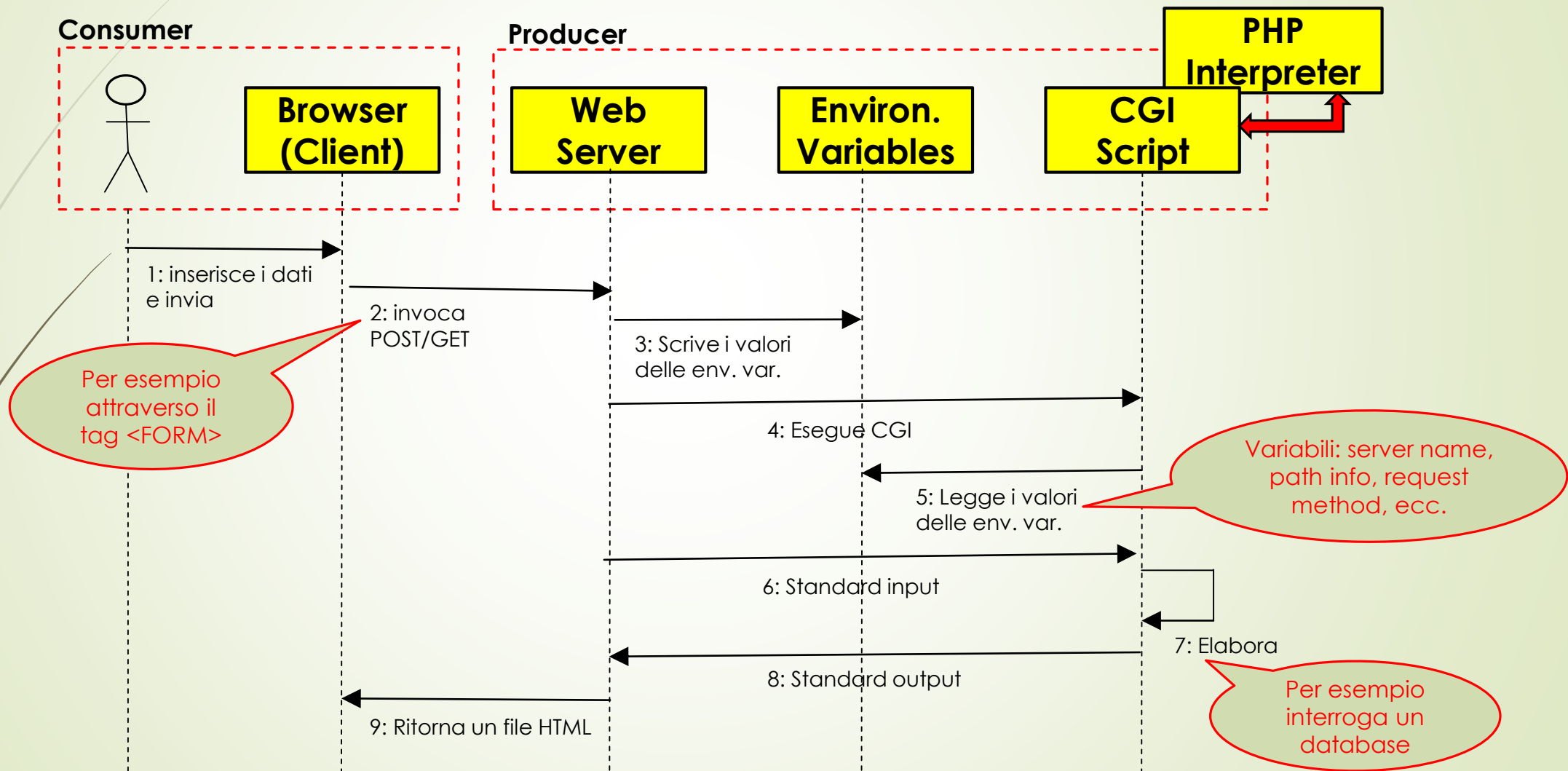
# CGI e HTTP in pratica

## Il server HTTP:

- Riconosce URL corrispondenti all'invocazione di script CGI (POST/GET)
- Esegue lo script CGI (che confeziona la risposta in formato HTML e la ritorna al server HTTP)
- Restituisce la risposta nello standard output, in formato HTML, al client



# Sequenza di messaggi tra gli attori





# Invocare un programma PHP come script CGI

```
24
25 <section>
26   <h1>Ricerca libro</h1>
27   <br>
28   <form action="http://localhost/FormAndPHP/myphp.php" method="post">
29     <table>
30       <tr>
31         <td>Autore</td>
32         <td><input type="text" name="autore"></td>
33       </tr>
34       <tr>
35         <td>Titolo_del_libro</td>
36         <td><input type="text" name="titolo"></td>
37       </tr>
38       <tr>
39         <td><input type="submit" value="CERCA"></td>
40       </tr>
41     </table>
42   </form>
43 </section>
44
```

# Passaggio di parametri tra FORM e PHP

```
12 <?php
13 //PHP contiene delle variabili, definite superglobal che sono
14 //accessibili indipendentemente dallo scope, come gli
15 //array associativi: $_GET, $_POST, $_SERVER, usati qui di seguito.
16 //L'array $_SERVER contiene diverse informazioni relative al server,
17 //al client che ha effettuato la richiesta e sulla richiesta stessa.
18 if ($_SERVER['REQUEST_METHOD'] == 'GET') {
19     //Utilizzo l'array associativo $_GET[]
20     //che contiene i parametri passati;
21     echo '<h3>ATTIVATO script PHP con uso del GET </h3><br>';
22     $autore = filter_var($_GET['autore'], FILTER_SANITIZE_STRING);
23     $titolo = filter_var($_GET['titolo'], FILTER_SANITIZE_STRING);
24     echo 'AUTORE: '.$autore.'<br>'. 'TITOLO LIBRO: '.$titolo;
25 }elseif($_SERVER['REQUEST_METHOD'] == 'POST'){
26     //Utilizzo l'array associativo $_POST[]
27     //che contiene i parametri passati;
28     echo '<h3>ATTIVATO script PHP con POST </h3><br>';
29     $autore = filter_var($_POST['autore'], FILTER_SANITIZE_STRING);
30     $titolo = filter_var($_POST['titolo'], FILTER_SANITIZE_STRING);
31     echo 'AUTORE: '.$autore.'<br>'. 'TITOLO LIBRO: '.$titolo;
32 }
33 ?>
```



# **GRAZIE!**