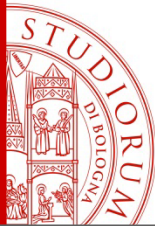


PHP





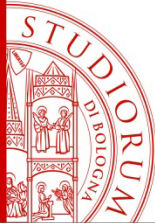
Argomenti

- PHP
 - Stack
 - Scripting Server side
 - PHP e HTTP: Sessioni, Cookie, GET e POST
 - Elementi della sintassi



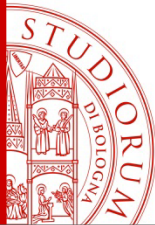


- PHP è un **linguaggio di scripting**, interpretato, originariamente concepito per la programmazione di pagine Web dinamiche **lato server**.
 - L'interprete PHP è un software libero distribuito sotto la PHP License.
 - L'acronimo è attualmente espanso come nella forma ricorsiva **PHP Hypertext Preprocessor** (originariamente era inteso come acronimo di «Personal Home Page»)
 - Citando Wikipedia «Un esempio di software scritto in PHP è MediaWiki, su cui si basano i progetti wiki della Wikimedia Foundation come Wikipedia»



Versioni di PHP

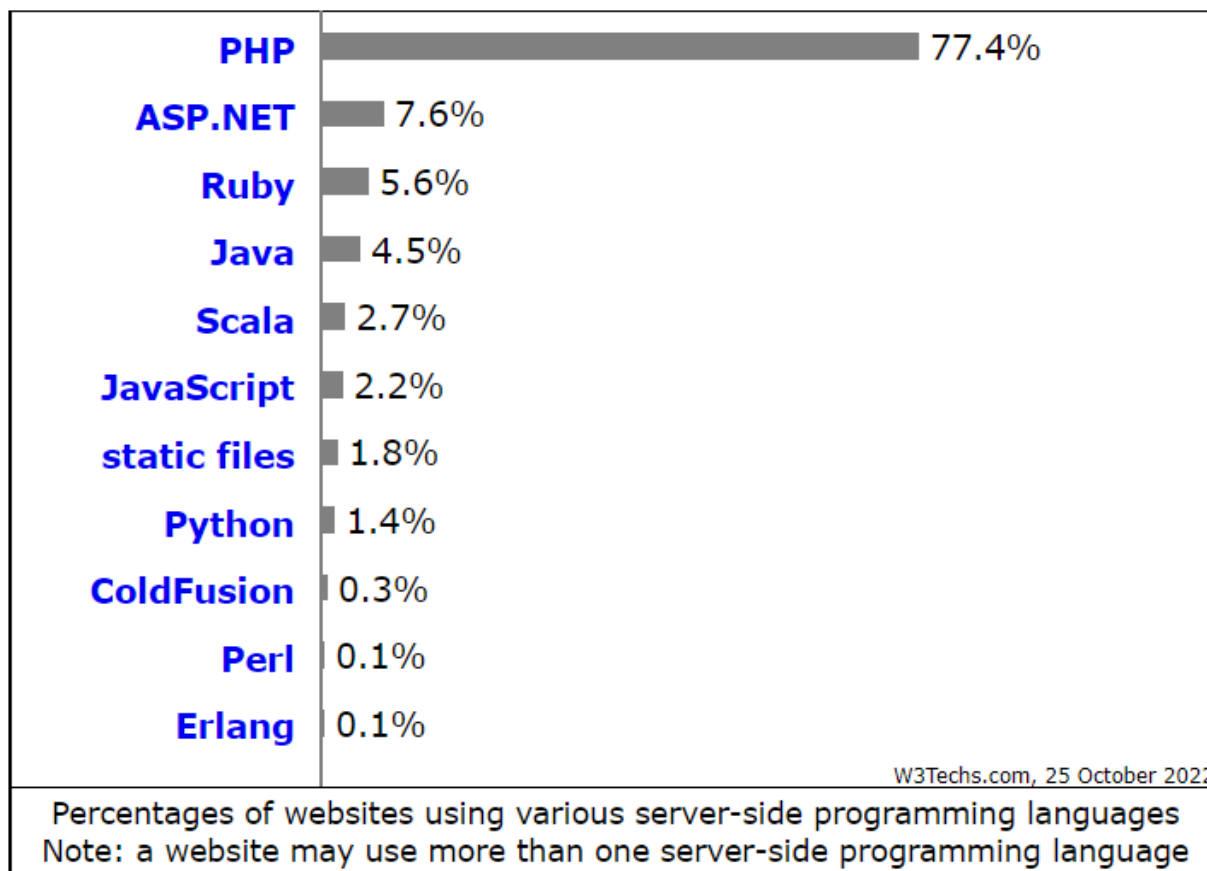
- **1994** nasce per opera di Rasmus Lerdorf che lo utilizzava nell'implementazione delle proprie pagine web. **La prima versione è rilasciata nel 1995**
- **1998** esce PHP 3.0 che rappresenta un punto di svolta, il progetto diventa espressione di un gruppo piuttosto che di una sola persona.
- **2000** esce PHP 4.0 che migliora velocità, modularità e altri aspetti.
- **2005** esce PHP 5.0 con un nuovo e solido modello ad oggetti, funzioni per la gestione di DOM XML e Web Services
- **2015** esce PHP 7.0 (la versione 6.0 non è stata mai rilasciata), basata sul nuovo motore Zend Engine III e con focus su performance e stabilità
- L'ultima versione è 8.1.11 (29 settembre 2022)

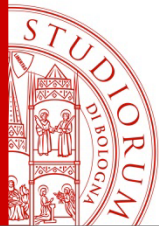


Perché PHP?

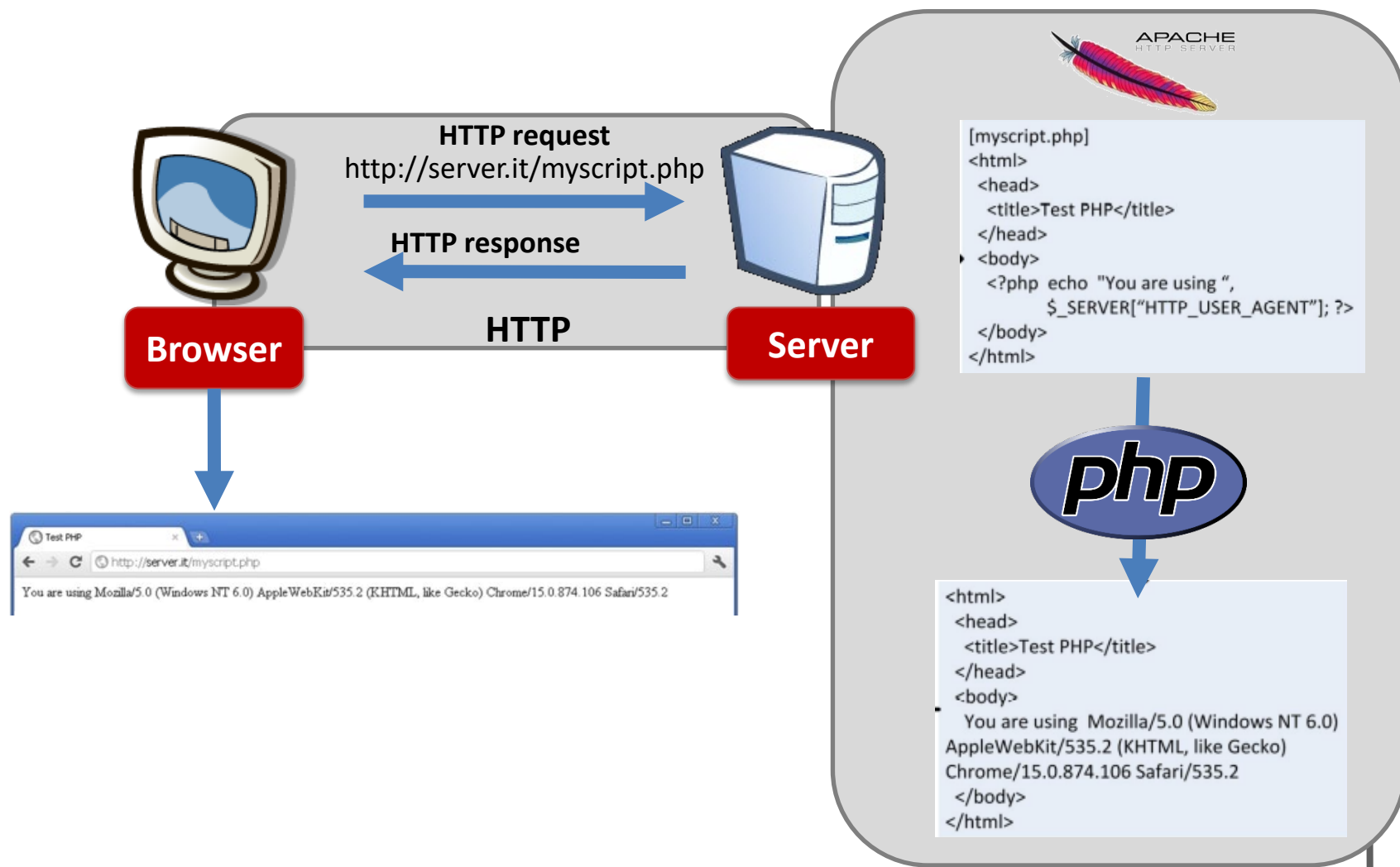
How to read the diagram:

PHP is used by 77.4% of all the websites whose server-side programming language we know.



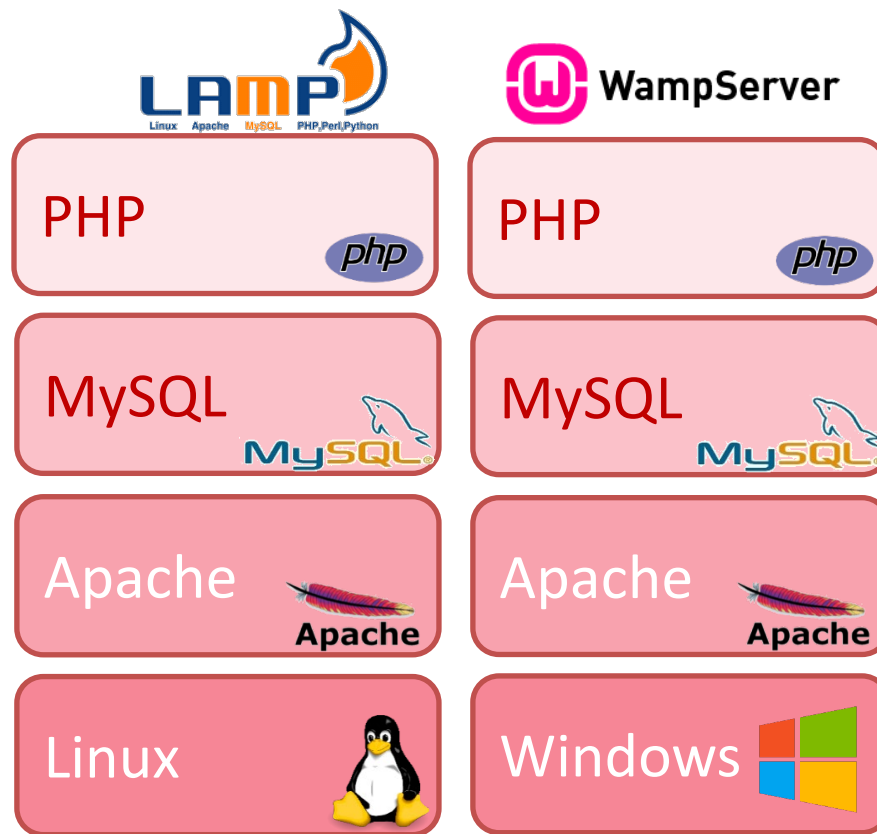


Architettura



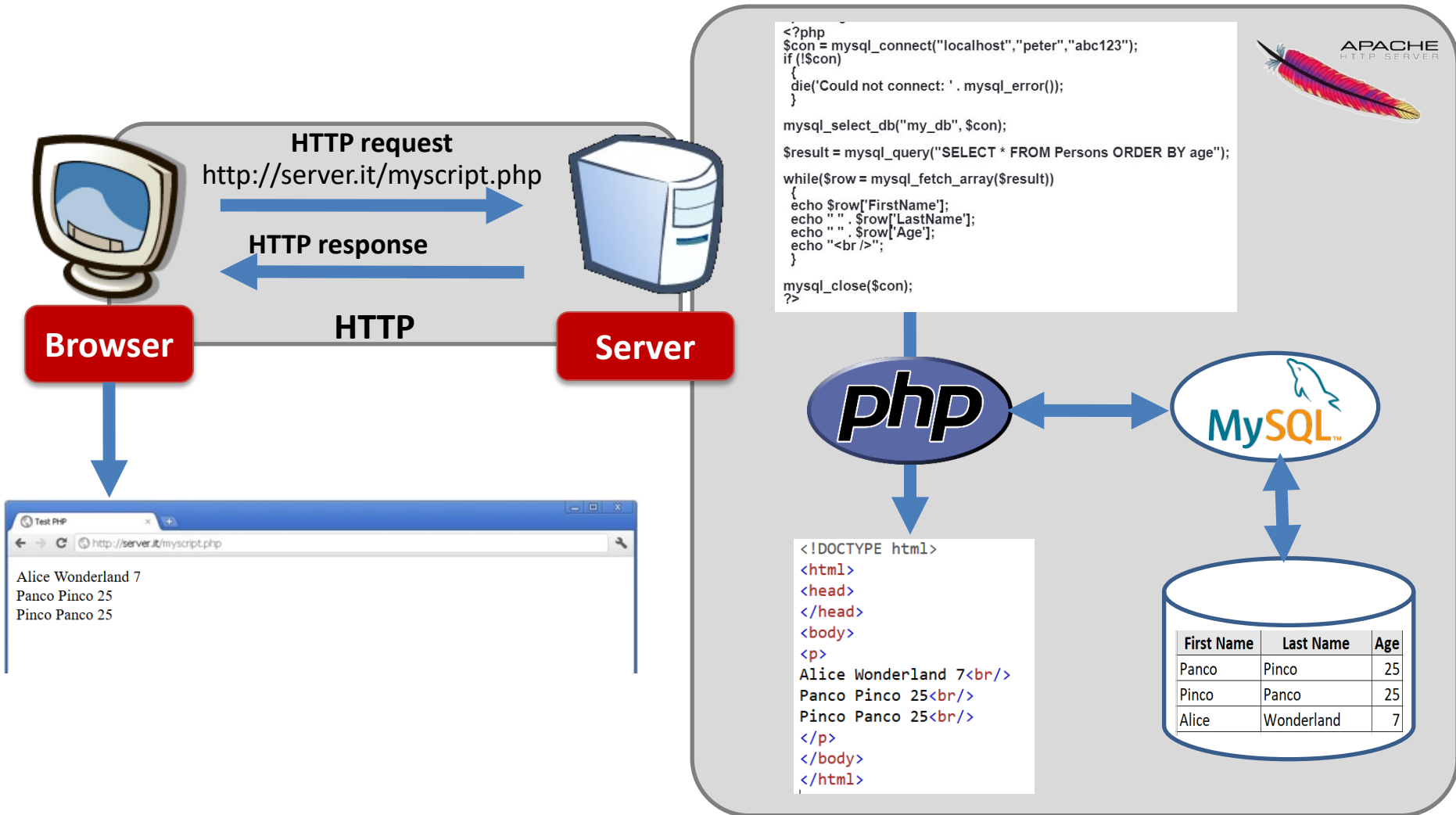
XAMP

- I solution stack che usiamo sono entrambi basati su Apache, PHP e MySQL





Architettura





Sono realizzati in PHP anche...

Etsy

YAHOO!

source
forge

WE PAY

MTV.CO.UK

Virgin
media

flickr

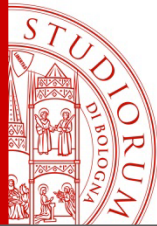


WIKIPEDIA
The Free Encyclopedia

MediaWiki

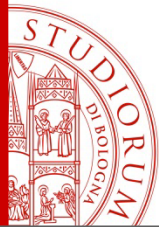
TE TechCrunch

zynga



CMS in PHP





Framework di PHP

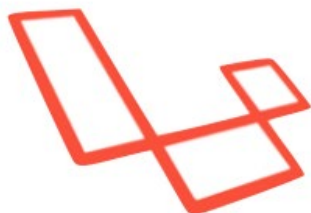


Symfony

ZF ZEND
FRAMEWORK



CodeIgniter



Laravel



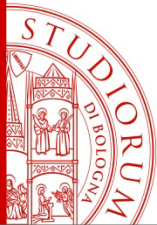
framework



SILEX

phalcon

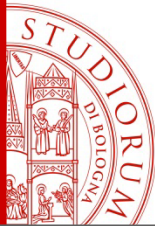




Le varie community di PHP!

PHP User Group Bologna



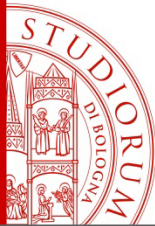


Architettura e sintassi

- L'interprete PHP preprocessa tutti i file con estensione PHP, che sono sostanzialmente file contenenti HTML, all'interno dei quali è presente del codice PHP, contenuto all'interno dei delimitatori **<?php** e **?>**
- Solo il codice contenuto tra i delimitatori viene realmente preprocessato.
- Hello world:

```
<?php // inizio codice php  
        echo "Hello, world!";  
?> // fine codice php
```

```
/* commenti multi-linea*/  
// commenti su una linea  
# commenti in stile bash
```

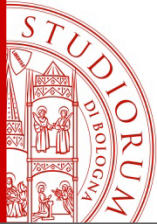


Hello world

Interprete PHP

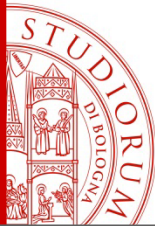
```
<html>
<head>
  <title>PHP Test</title>
</head>
<body>
  <?php echo '<p>Hello World</p>'; ?>
</body>
</html>
```

```
<html>
<head>
  <title>PHP Test</title>
</head>
<body>
  <p>Hello World</p>
</body>
</html>
```

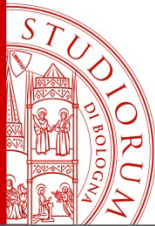
Sintassi PHP in generale

- La sintassi PHP è di tipo C-like, con differenze legate allo scopo specifico del PHP (sviluppo per il Web):
 - Tipizzazione più debole
 - Meno tipi di dati di base
 - Vettori di dimensione variabile
- Ovviamente ci sono funzioni aggiuntive, di supporto alla programmazione Web (esempio: gestione delle sessioni).



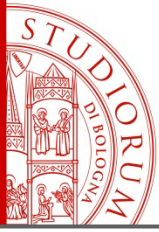
Differenze con il C

C	PHP
Tipizzazioni molto forte	Tipizzazione debole
Supporta i puntatori	Non supporta i puntatori
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
array di dimensioni fisse	Array di dimensione variabile
Non supporta array associativi	Supporta array associativi



Esempio

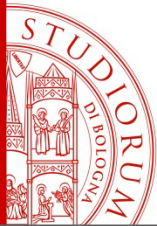
```
<!DOCTYPE html>
<html lang="it">
  <head>
    <title>Esempio</title>
  </head>
  <body>
    <p>
      <?php
        echo "Ciao, sono un micro PHP script!";
      ?>
    </p>
  </body>
</html>
```



Esempio

```
<!DOCTYPE html>
<html lang="it">
<head>
  <title>Titolo della pagina</title>
</head>

<body>
  Sono le ore <?php echo date('H:i'); ?>
  del giorno <?php echo date('d/m/Y'); ?>.
</body>
</html>
```



Esempio

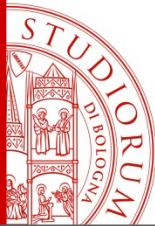
```
<!DOCTYPE html>
<html lang='it'>
<head>
  <title>Titolo
della pagina"</title>
</head>

<body> <p>
  Sono le ore <?php
echo date('H:i'); ?>
del giorno <?php echo
date('d/m/Y'); ?>.
</p></body>
</html>
```

Interprete
PHP

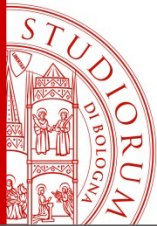
```
<!DOCTYPE html>
<html lang='it'>
<head>
  <title>Titolo della
pagina"</title>
</head>

<body><p>
  Sono le ore 15:22
del giorno 25/10/2022.
</p></body>
</html>
```



Uso di più file PHP

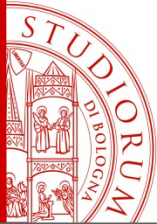
- Quando il codice è ripetuto in più pagine e/o è molto articolato, l'inserimento in interlinea con l'HTML può risultare disfunzionale.
- In questo caso si può:
 - Realizzare un file separato con estensione **.php** che contiene il codice più articolato e di uso più frequente (per esempio la gestione del/dei DB).
 - Includere il codice nel file PHP principale utilizzando **require** e **include**



Esempio

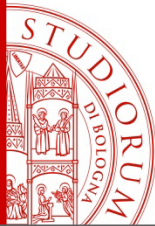
```
<?php
$host = "localhost";
$user = "silvia";
$pass = "techweb";
$database = "TW";

// connessione DBMS
$myconn = mysql_connect($host, $user, $pass) or
die('Errore...');
//connessione DB database
mysql_select_db($database, $myconn) or
die('Errore...');
?>
```



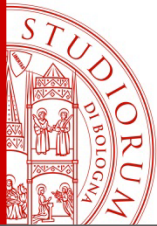
Esempio

- All'interno di tutte le pagine o i file che operano sul database TW possiamo includere il file dell'esempio (chiamiamolo **dbtw-mysql.php**), usando uno dei due costrutti a disposizione:
 - Include: **include "dbtw-mysql.php";**
 - Require: **require "dbtw-mysql.php";**
- **include()** e **require()** sono equivalenti eccetto per come esse trattano gli errori: **include()** produce un **Warning** mentre **require()** restituisce un **Fatal Error**.



Include e require

- Le primitive **include** e **require** possono essere usate anche per includere:
 - **Librerie PHP:**
in questo caso occorre ragionare con maggior attenzione sullo **scope delle variabili** (il codice che esso contiene eredita lo scope delle variabili della riga in cui si verifica l'inclusione) e del **ritorno** (ogni return termina l'esecuzione del file incluso/richiesto e torna il controllo al file php che lo ha incluso/richiesto).
 - **HTML**
- Ovviamente **non si può includere il file .php in un file .html** perché non verrebbe interpretato

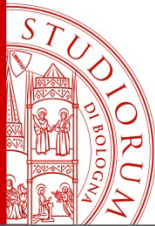


Alcuni elementi di HTTP

- Prima di passare ai dettagli della sintassi PHP, riprendiamo due elementi di HTTP che impattano sull'organizzazione delle applicazioni lato server:
 - La gestione della sessione (coi cookies)
 - I metodi GET e POST

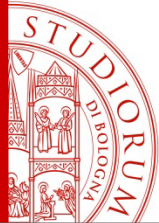
http://





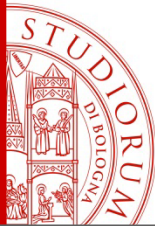
GET e POST

- **GET**: richiede di processare i dati a una specifica applicazione server.
 - Invia dati testuali, aggiungendo alla URL richiesta una query string. I dati, tutti **testuali**, sono **visibili** nella URL.
 - La query string è separata dalla URL da ? ed è composta da coppie **nome=valore** separate tra loro da & (+ sostituisce gli spazi).



GET e POST

- **POST**: invia i dati da processare una specifica applicazione server.
 - Non ha restrizioni su dimensione e tipo di dati che vengono inviati al server. In particolare può inviare anche dati di tipo **multipart/form-data** incapsulando formati binari;
 - i dati sono inviati nel body del messaggio HTTP e non sono visibili, quindi è adatto al passaggio di dati riservati, come per esempio una password.



Comunicazione HTTP

- HTTP è:
 - **Client-server**: In HTTP esistono due ruoli specifici: il client attiva la connessione e richiede dei servizi. Il server accetta la connessione, nel caso identifica il richiedente, e risponde alla richiesta. Alla fine chiude la connessione.
 - **Generico**: HTTP è indipendente dal formato dati con cui vengono trasmesse le risorse. Può funzionare per documenti HTML come per binari, eseguibili, oggetti distribuiti o altre strutture dati più o meno complicate.
 - **Statelessness**: Il server non è tenuto a mantenere informazioni che persistano tra una connessione e la successiva sulla natura, identità e precedenti richieste di un client. Il client è tenuto a ricreare da zero il contesto necessario al server per rispondere.

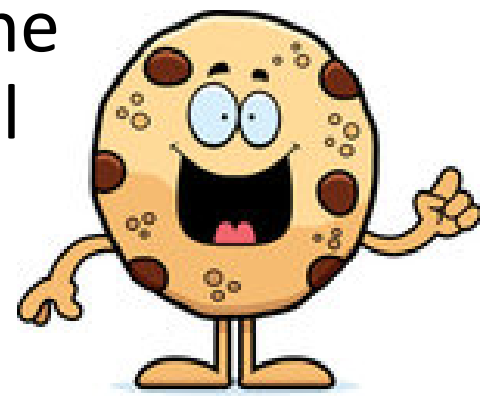
Apriamo una parentesi sui cookies

- In questo contesto vediamo i **cookies** che sono un **meccanismo di supporto alla gestione delle sessioni** basato sull'idea che sia il **client a mantenere lo stato di precedenti connessioni**, e ad inviarlo al server di pertinenza ogni volta che richiede un documento.
- **Il termine cookie (anche magic cookie) indica un blocco di dati opaco (cioè non interpretabile) lasciato dal server in consegna ad un richiedente per poter ristabilire in seguito il suo diritto alla risorsa richiesta (come il tagliando di una lavanderia).**
- Ogni server lascia in consegna i suoi cookies.



Funzionamento

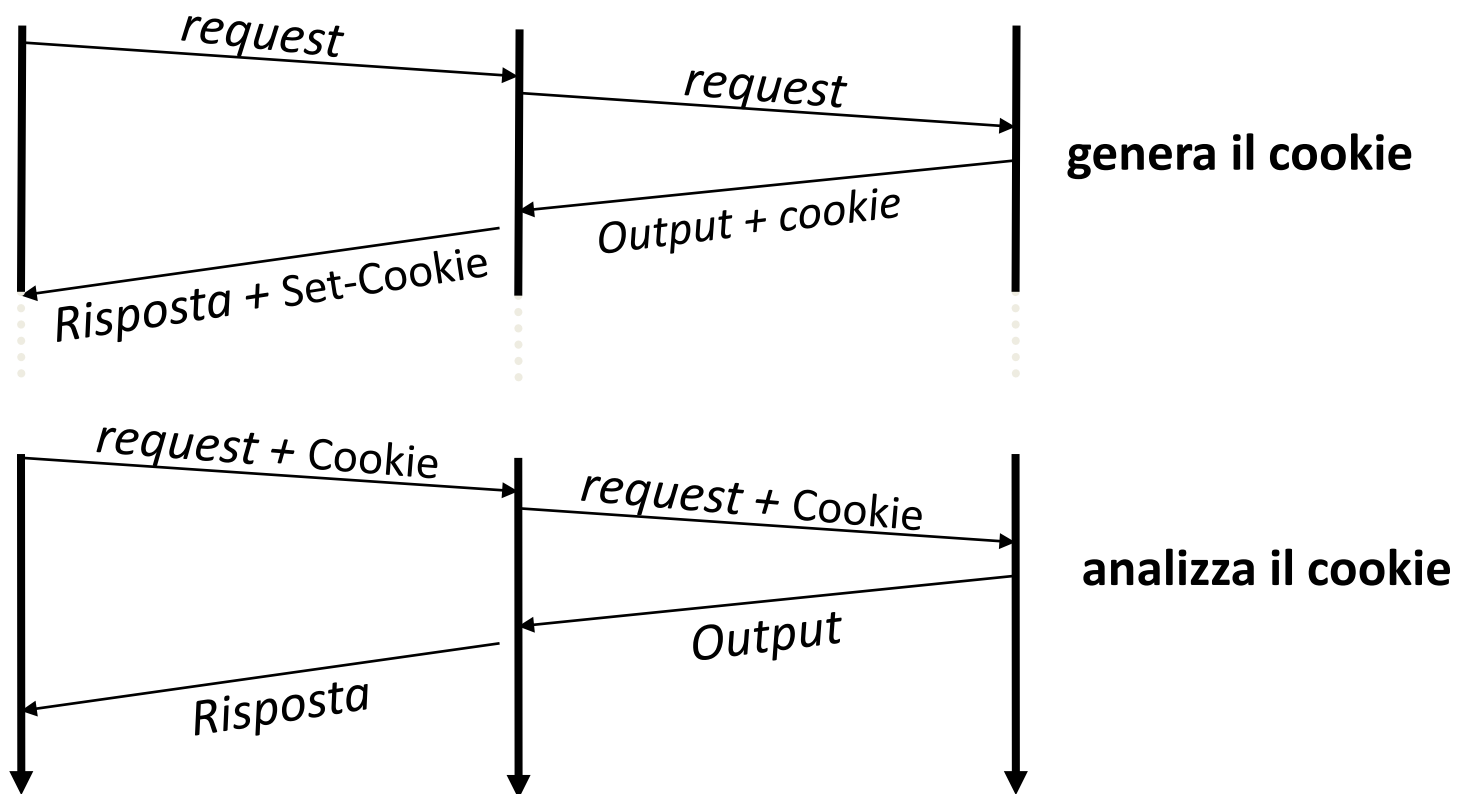
- Alla prima richiesta di uno user-agent, il server fornisce la risposta ed un header aggiuntivo, il **cookie**, con dati arbitrari e con la specifica di usarlo per ogni successiva richiesta.
- Il server associa a questi dati le informazioni sulla transazione.
- Ogni volta che lo user-agent accederà a questo sito, rifornirà i dati opachi del cookie che permettono al server di ri-identificare il richiedente, e creare così un profilo specifico.

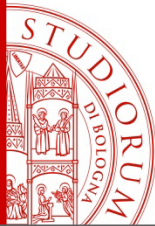


Funzionamento



applicazione





Header dei cookies

- I cookies dunque usano due **header**, uno per la risposta, ed uno per le richieste successive:
 - **Set-Cookie**: header della risposta, il client può memorizzarlo e rispeditirlo alla prossima richiesta.
 - **Cookie**: header della richiesta. Il client decide se spedirlo sulla base del nome del documento, dell'indirizzo IP del server, e dell'età del cookie.

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html
Set-Cookie: PREF=ID=5e66ffd215b4c5e6:
TM=1147099841:LM=1147099841:S=Of69MpW
Bs23xeSv0; expires=Sun, 17-Jan-2038 1
9:14:07 GMT; path=/; domain=.google.c
om
```

Cancellare i cookie

- L'utente può **cancellare** i cookie dal browser per varie ragioni:
 - **Sicurezza**, esistono diversi tipi di attacchi basati sui cookie
 - **Privacy**: cancellando i cookie si limita il controllo dei server sulle attività dell'utente.
 - **Efficienza** e consumo di spazio
- Rimandiamo alla lezione di Programmazione di Reti



Cookie

- I cookies contengono le seguenti informazioni:
 - **Comment**: stringa leggibile di descrizione del cookie.
 - **Domain**: il dominio per cui il cookie è valido
 - **Max-Age**: La durata in secondi del cookie.
 - **Path**: l'URI per il quale il cookie è valido
 - **Secure**: la richiesta che il client contatti il server usando soltanto un meccanismo sicuro (es. HTTPS) per spedirlo
 - **Version**: La versione della specifica a cui il cookie aderisce.



Cookie law

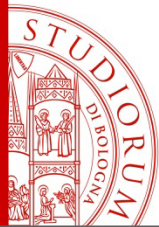
- La **cookie law** è un provvedimento a livello comunitario (direttiva 2009/136/CE) nato per rendere **consapevoli** gli utenti riguardo ai **rischi per la privacy** connessi all'uso dei cookie (in particolare quelli utili a conoscere le preferenze dell'utente).
- Come previsto dal Provvedimento dell'8/5/2014, il **2 giugno 2015** è scaduto il termine per dare attuazione alle prescrizioni del Garante della Privacy in materia di cookie.
 - **Tutti i siti web che utilizzano i cookie hanno dovuto adeguarsi alla nuova direttiva** e obbligatoriamente informare gli utenti della presenza dei cookie.
 - **L'informativa all'utente dipende dal tipo di cookie** o meglio da come il cookie viene utilizzato dal server, con quali finalità.



Cookie tecnici e analitici

- I **cookie tecnici** sono usati dal gestore del sito per mettere in opera alcune funzioni o rendere più agile la navigazione.
- Consentono ad esempio di:
 - autenticarsi per accedere ad aree riservate
 - Aggiungere o rimuovere prodotti nel carrello
 - Memorizzare alcuni criteri preselezionati come per esempio la lingua.
- I **cookie analitici** sono usati dal gestore del sito per raccogliere alcune informazioni in forma aggregata sugli utenti.
- I **cookie tecnici e analitici non richiedono consenso** preventivo degli utenti (non serve il banner con l'informativa breve) ma vanno citati nell'informativa estesa.





Molti lo chiedono comunque...

- Per non incorrere in contenziosi



Cookie di profilazione

- I cookie di **profilazione**:

- sono usati dal gestore del sito per raccogliere **dati personali** sui visitatori, sulla base dei quali il server costruisce un **profilo dell'utente/consumatore** che poi viene usato anche per proporre prodotti e servizi.
- Sono cookie che necessitano del consenso preventivo da parte dell'utente quindi in questo caso occorrerà mettere sul sito sia l'informativa estesa sia quella breve.



Informativa

Questo sito utilizza anche cookie di profilazione per inviarti pubblicità e servizi in linea con le tue preferenze. Se vuoi saperne di più o negare il consenso a tutti o ad alcuni cookie [clicca qui](#).
Proseguendo la navigazione acconsenti all'uso dei cookie.

Cookie law



**GARANTE
PER LA PROTEZIONE
DEI DATI PERSONALI**

Il tuo sito/blog installa cookie? Cosa devi fare

IMPORTANTE: per una corretta interpretazione degli adempimenti previsti, si raccomanda la consultazione del **Provvedimento del Garante dell'8 maggio 2014** e dei «**Chiarimenti in merito all'attuazione della normativa in materia di cookie**».
I documenti sono disponibili su www.garanteprivacy.it/cookie

**Segnalarli
nell'informativa**

Art. 2, par. 5, Direttiva 2009/136/CE
e art. 122, comma 1, Codice privacy

**Inserire il banner e
richiedere il consenso
ai visitatori**

Art. 2, par. 5, Direttiva 2009/136/CE
e art. 122, comma 1, Codice privacy

**Notificare
al Garante**

Art. 37, comma 1, lett. d),
Codice privacy

CHE TIPO DI COOKIE INSTALLI?

LEGENDA: ✓ adempimento previsto ✗ adempimento non previsto



Nessun cookie



**Tecnici o analitici
prima parte**



Analitici terze parti
(se sono adottati strumenti che riducono il potere identificativo dei cookie e la terza parte non incrocia le informazioni raccolte con altre di cui già dispone) – vedi punto 2 dei «Chiarimenti in merito all'attuazione della normativa in materia di cookie»



Analitici terze parti
(se **NON** sono adottati strumenti che riducono il potere identificativo dei cookie e la terza parte non incrocia le informazioni raccolte con altre di cui già dispone) – vedi punto 2 dei «Chiarimenti in merito all'attuazione della normativa in materia di cookie»



Di profilazione prima parte



Di profilazione terze parti



La notificazione è a carico del soggetto terza parte che svolge l'attività di profilazione.

Prima domanda



BONUS

- **DOMANDA 1 :**

In un sito di e-commerce che registra gli utenti esclusivamente per gestire le loro ordinazioni sono necessari:

- ☐ Cookie analitici
- ☐ Cookie tecnici
- ☐ Cookie di profilazione
- ☐ Nessun cookie

Prima domanda

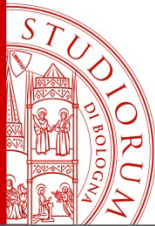


BONUS

- **DOMANDA 2 :**

Nel sito della domanda 1 (sito di e-commerce che registra gli utenti esclusivamente per gestire le loro ordinazioni) la cookie law richiede :

- ☐ Nulla, nessuna azione aggiuntiva
- ☐ Segnalare i cookie nell'informativa
- ☐ Banner con consenso dei visitatori
- ☐ Segnalazione al garante



Alcuni esempi di codice

- Vedremo alcuni elementi sintattici generali e introdurremo esempi che riguardano:
 - La gestione del DB
 - La gestione delle Sessioni
 - L'uso di get e post

Domande?

