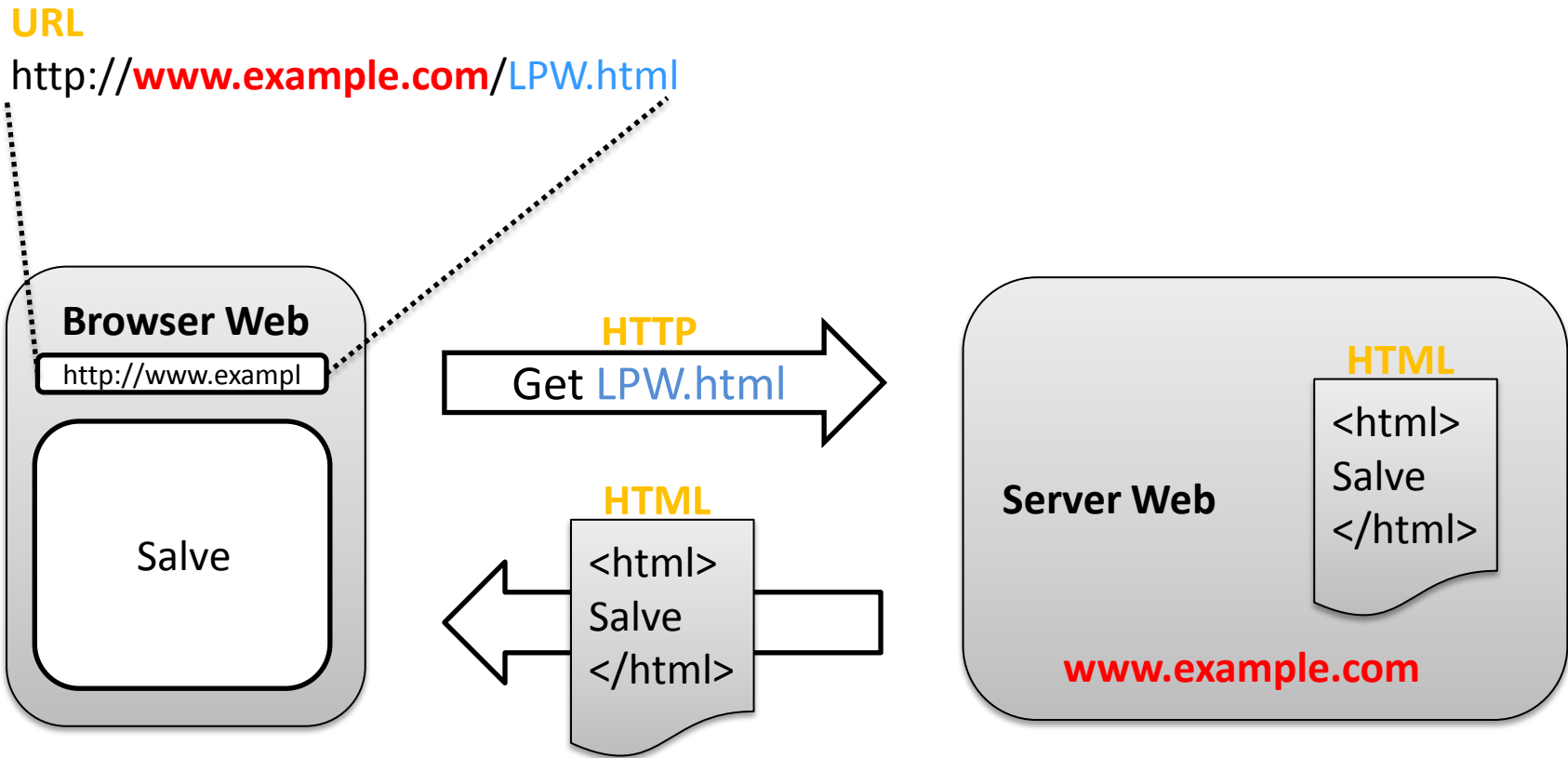


Laboratorio Progettazione Web

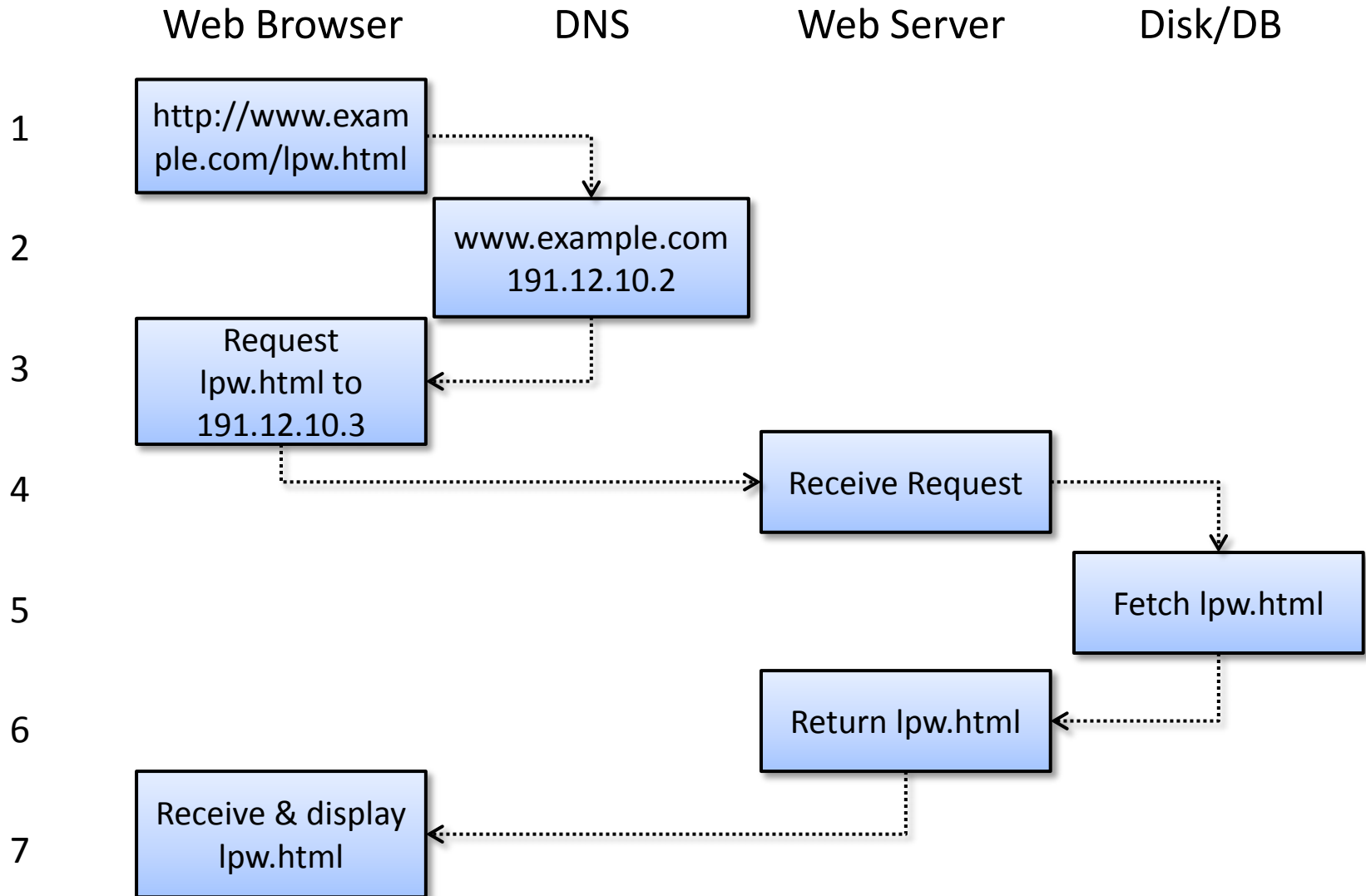
Applicazioni Web – Lezione 2

Andrea Marchetti – IIT-CNR
andrea.marchetti@iit.cnr.it
2011/2012

Architettura Web



Timeline



I Livelli in un'applicazione web

- Un'applicazione Web si sviluppa su tre livelli secondo il modello Model-View-Control (MVC)
 - Dati (Model)
 - Logico (Control)
 - Presentazione (View)

Livello Dati (Model)

- Il livello dati consente di modellare e gestire il contenuto informativo dell'applicazione.
- Tipici esempi di applicazioni presenti a questo livello sono:
 - File System con la sua struttura ad albero
 - Server dati
 - DB relazionali (MySQL)
 - DB XML (eXist)
 - DB Json (MongoDB)
 - Altri server come SMTP

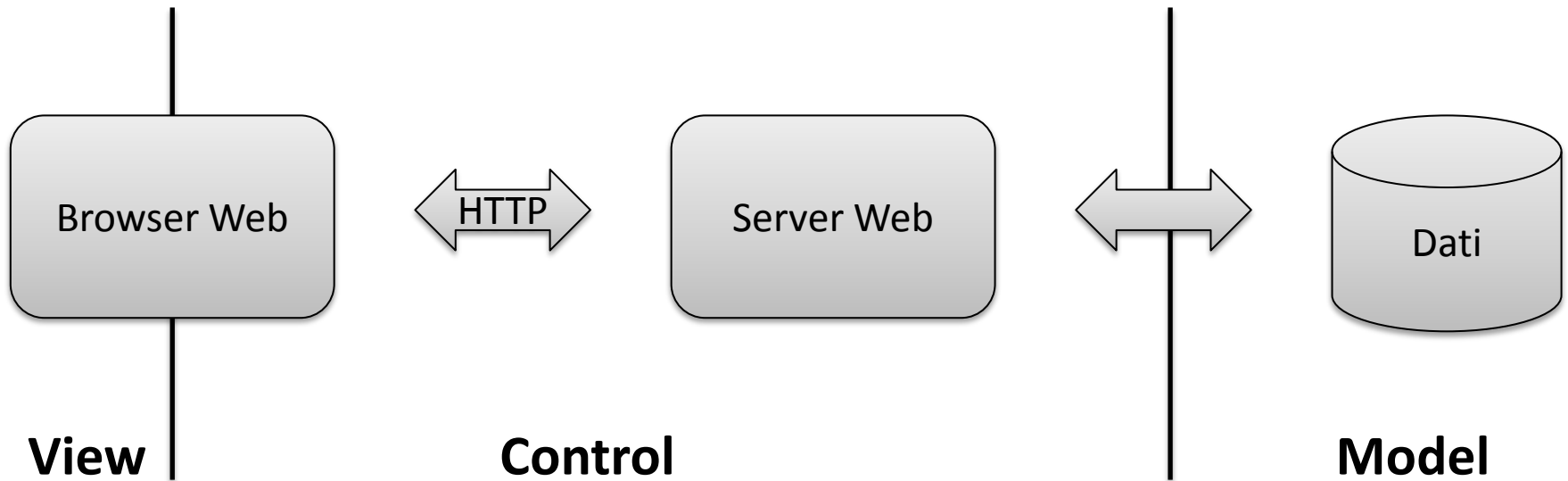
Livello Presentazione (View)

- Il livello di presentazione costituisce l'interfaccia utente dell'applicazione Web e corrisponde al browser Web.
- Le tecnologie di riferimento sono:
 - HTML – formato per la presentazione di ipertesti
 - CSS – foglio di stile per documenti HTML

Livello Logico (Control)

- Il livello logico di un'applicazione Web corrisponde alla componente elaborativa dell'applicazione
- Il livello logico può risiedere
 - Lato server: PHP, PYTHON, JSP, Javascript, Java servlet...
 - Lato browser (client): Javascript, Applet, Flash

Suddivisione Fisica dei livelli



La parte logica (Control) può risiedere sia sul Browser che sul Server

Dove mettere il controllo

- La parte di controllo deve potere accedere ai dati
- I dati possono essere:
 - Quelli provenienti dai sistemi di input/sensori del browser: mouse, tastiera, sistemi touchscreen, microfono, videocamera, GPS, accelerometro, ...
 - Quelli che risiedono su DB lato server
- Normalmente si preferisce far viaggiare meno dati possibili e quindi mettere il controllo là dove sono i dati

Control su Browser

- Dipende da cosa gestisce il browser
 - Javascript (Js) è la tecnologia standard
 - Tecnologie non standard: Applet, Adobe Flash, VBScript
- **Un programma Js può comportarsi in modo differente sui vari browser !!!**
 - IE, Firefox, Chrome, Safari, Opera, ...
 - Con le librerie javascript cross-platform questo problema si riduce
- Tipicamente la programmazione su browser riguarda l'input dell'utente –es. controllo di dati inseriti tramite FORM
- **Molti client hanno capacità di calcolo ridotte !!!**
 - Smartphone, Tablet, ...

Control su Server

- Maggiori potenzialità
 - Capacità di calcolo
 - Gestione centrale di grosse moli di dati
- Pagine Web Dinamiche
 - Il server non si limita a restituire pagine HTML che già risiedono sul disco.
- Se il server deve gestire molte richieste rischia di essere sovraccaricato !!!
 - E sempre possibile scalare la capacità computazionale

Tecnologie Server side

- L'utente può interagire con il server ad esempio sottomettendo dati che il server elabora per restituire poi una risposta sotto forma di pagina HTML.
- Un tipico caso e' l'accesso da parte dell'utente ad un database che risiede sul server. **In questo caso la pagina web funziona come una interfaccia per accedere ai dati che risiedono sul server.**
- Ci sono varie tecnologie server side
 - CGI: interfaccia per programmi compilati
 - Linguaggi di script: PHP, ASP, JSP, Perl, Python
 - Java Servlet

Tecnologia Server Side

- ◆ La computazione avviene lato server e può avvenire tramite **programmi compilati** o **script interpretati**
- ◆ Nel caso di programmi compilati il web server si limita ad invocare, su richiesta del client, un eseguibile. L'eseguibile può essere scritto in un qualsiasi linguaggio che supporti l'interazione con il web server – tipicamente Java, Visual Basic, C++.
- ◆ Storicamente il primo approccio è stato il **Common Gateway Interface (CGI)**

Tecnologia Server Side – programmi compilati

- La computazione server side con programmi compilati ha il vantaggio di un tempo di esecuzione molto veloce rispetto agli script, ma lo svantaggio di mandare in esecuzione una istanza del processo per ogni chiamata e quindi caricare molto il server in caso di molti accessi simultanei.
- Alcuni approcci tentano di rimediare a questo svantaggio – Servlet e ASP.NET – mandando un esecuzione una sola istanza di processo che gestisce chiamate multiple

Tecnologia Server Side - script

- Nel caso di esecuzione di script, il web server ha al suo interno **un motore** in grado di interpretare il linguaggio di script usato.
- Si perde in velocità di esecuzione, ma si guadagna in facilità d'uso
- In alcuni approcci rimane il problema dell'esecuzione simultanea di più processi (Perl), in altri viene eseguito un solo processo che gestisce tutte le chiamate (Cold Fusion, ASP)
- E' possibile gestire più tecnologie serverside da uno stesso server web

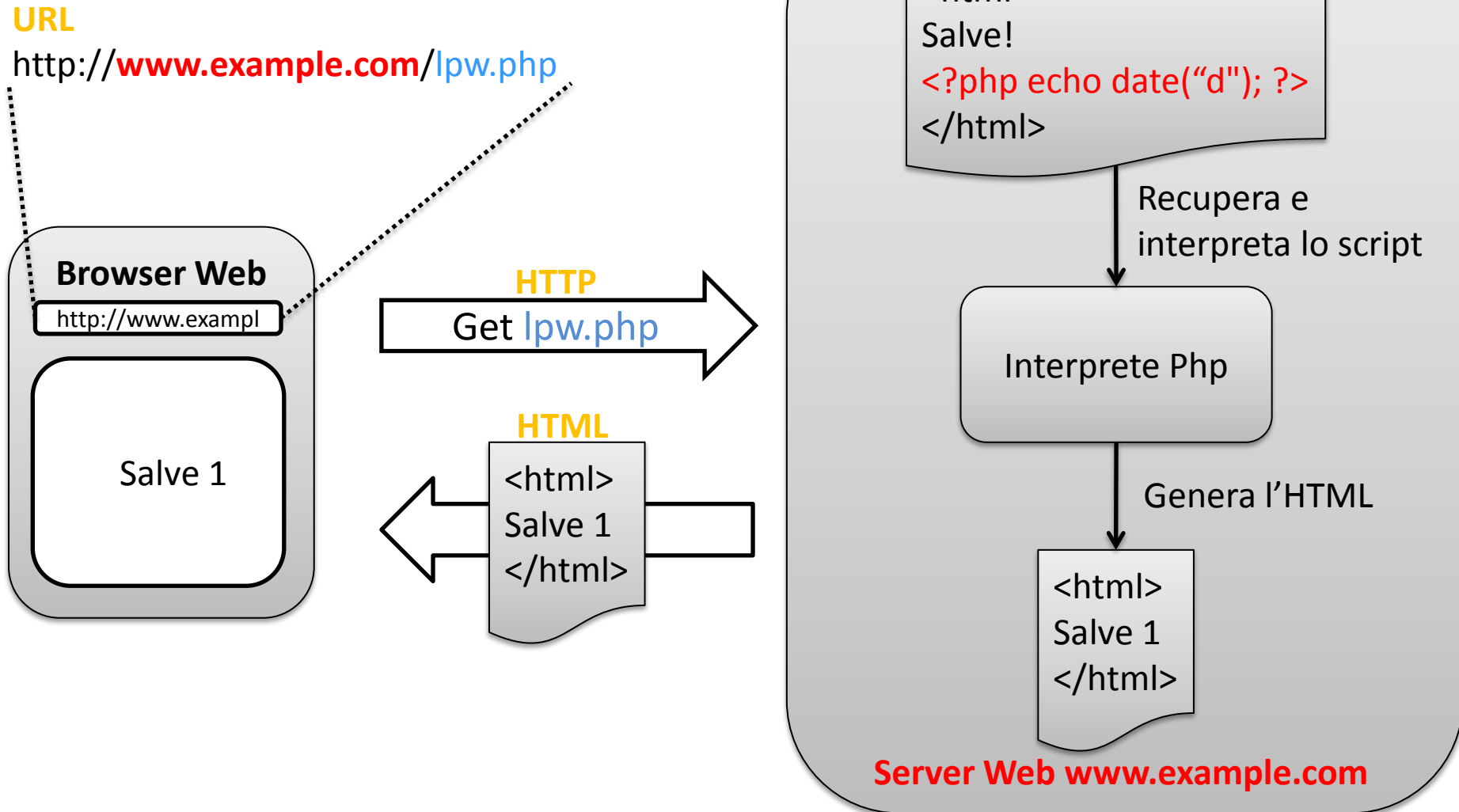
Situazione Test di autovalutazione al 26/02/2012:

	21/02 22 TEST	26/02 41 TEST
Sistemi Operativi	Sufficiente	Sufficiente++
Linguaggi di programmazione	Scarsa	Sufficiente --
HTML	Sufficiente++	Sufficiente++
JS	Scarsa ++	Scarso/Sufficiente
CSS	Sufficiente	Sufficiente

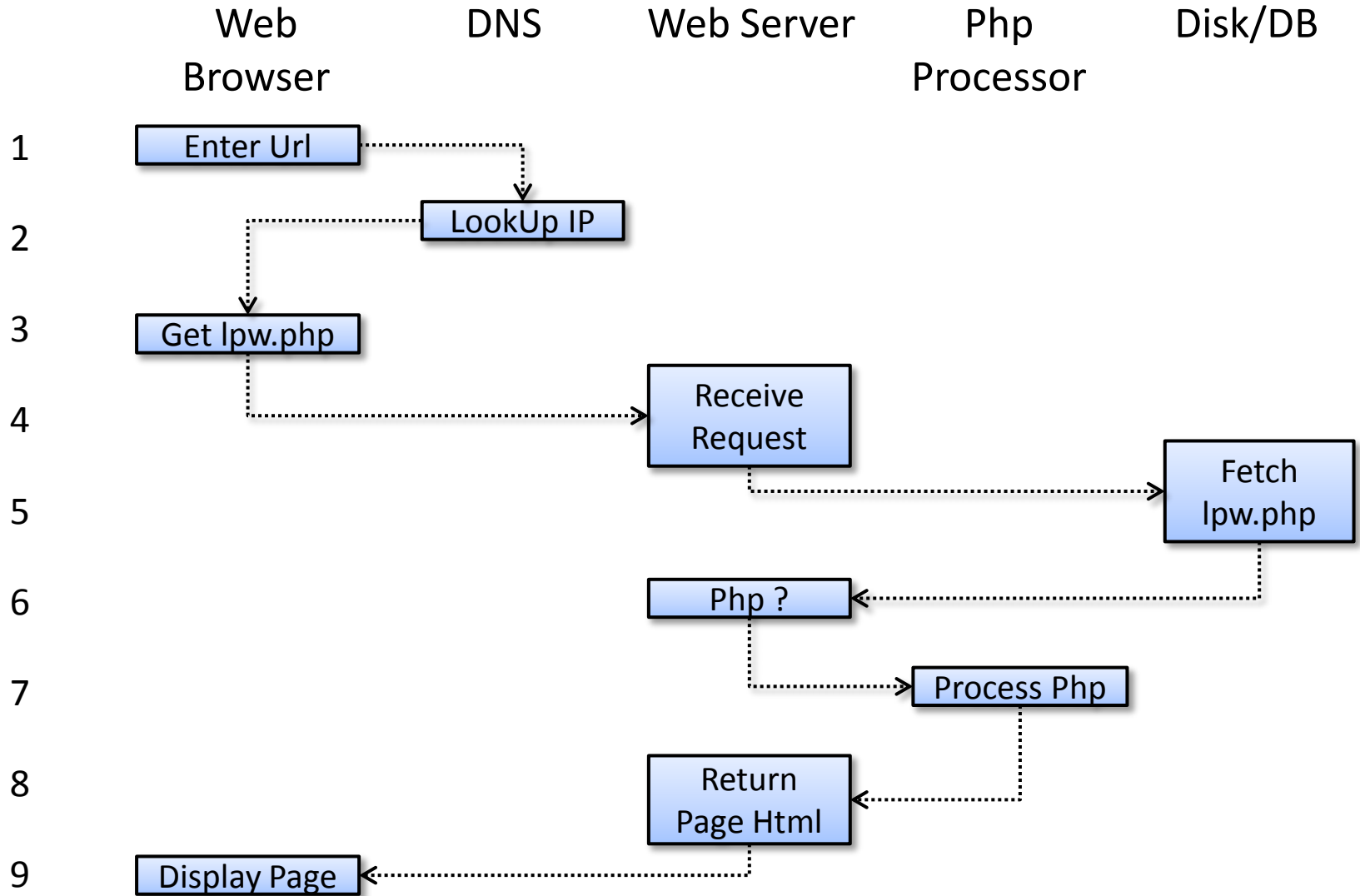
Prossime lezioni

- La prossima lezione di terrà giovedì **8 marzo 2012**
- Seminario di Oreste Signore su
 - Origini, principi e futuro del WWW**
 - 14-16 Aula Magna edificio E
- La lezione del 15 marzo sarà su HTML e sarà tenuta da Maurizio Tesconi
- La lezione del 22 marzo sarà un'introduzione a PHP

Linguaggi di scripting



Timeline



Caratteristiche di uno script PHP

- Requisito: Apache con il processore PHP
- Struttura di un programma PHP
- Termina con l'estensione .php
 - Quando il server gestisce un documento richiesto con estensione .php lo passa direttamente al Processore PHP
 - Si può configurare il server web affinché invochi sempre il processore php anche per documenti con estensione .htm o .html . Utile se si vuole nascondere che stiamo usando php.
- Il processor php restituisce alla fine dell'elaborazione un file HTML
- Se il documento .php contiene solo codice HTML il risultato sarà il documento stesso

Modi per generare HTML

```
<html>  
Salve!  
<?php echo date("h:i:s"); ?>  
</html>
```

```
<?php  
    echo "<html>";  
    echo "Salve ";  
    echo date("h:i:s");  
    echo "</html>";  
?>
```

Il Processore PHP interpreta solo il codice contenuto tra `<?php` e `?>` il resto lo fa passare

```
<html>  
Oggi è il <?php echo date("d/m/y"); ?></br>  
Sono le ore <?php echo date("h:i:s"); ?>  
</html>
```

Esercizio 0

- Installare uno dei 3 programmi precedenti sulla directory htdocs di apache
 - .../xampp/htdocs
- Se qualcosa non funziona verificare:
 - Avere usato l'estensione .php per il file contenente il programma php
 - Verificare che il server apache sia partito

I Esercizio – Gioco del Lotto

- Simulare estrazione del Lotto
- Requisiti
 - Costrutto HTML per le tabelle html.it
 - Funzione PHP che genera numeri casuali php.net
 - `rand(1,99)`

N O T I Z I A R I O
delle estrazioni del lotto del 02 Febbraio 2008

CITTA'	Primo estratto	Secondo estratto	Terzo estratto	Quarto estratto	Quinto estratto
Bari	33 <small>trigesimo</small>	9 <small>nono</small>	77 <small>settantasette</small>	78 <small>settantotto</small>	14 <small>quattordici</small>
Cagliari	44 <small>quarantafour</small>	64 <small>sessantafour</small>	71 <small>settantuno</small>	88 <small>ottantotto</small>	68 <small>sessantotto</small>
Firenze	90 <small>nonanta</small>	40 <small>quaranta</small>	85 <small>ottantacinque</small>	18 <small>diciotto</small>	38 <small>trigesimo</small>
Genova	79 <small>settantannove</small>	12 <small>due</small>	35 <small>trigesimo</small>	4 <small>quattro</small>	44 <small>quarantafour</small>
Milano	81 <small>ottantuno</small>	1 <small>uno</small>	88 <small>ottantotto</small>	71 <small>settantuno</small>	11 <small>undici</small>
Napoli	18 <small>diciotto</small>	38 <small>trigesimo</small>	80 <small>ottanta</small>	81 <small>ottantuno</small>	60 <small>sessanta</small>
Palermo	70 <small>settanta</small>	34 <small>trigesimo</small>	12 <small>due</small>	5 <small>cinque</small>	71 <small>settantuno</small>
Roma	61 <small>sessantuno</small>	75 <small>settantacinque</small>	9 <small>nono</small>	67 <small>sessantasette</small>	49 <small>quarantannove</small>
Torino	44 <small>quarantafour</small>	64 <small>sessantafour</small>	46 <small>quarantasei</small>	11 <small>undici</small>	82 <small>ottantadue</small>
Venezia	10 <small>dieci</small>	31 <small>trigesimo</small>	78 <small>settantotto</small>	8 <small>otto</small>	45 <small>quarantacinque</small>
Nazionale	80 <small>ottanta</small>	74 <small>settantafour</small>	54 <small>cinquantafour</small>	41 <small>quarantuno</small>	18 <small>diciotto</small>

Una possibile soluzione

```
<html>
<table>
  <tr><td>Bari</td>
    <td><?php echo rand(1,99); ?></td>
    <td><?php echo rand(1,99); ?></td>
    <td><?php echo rand(1,99); ?></td>
    <td><?php echo rand(1,99); ?></td>
    <td><?php echo rand(1,99); ?></td>
  </tr>
  <!-- Ripeti per tutte le ruote -->
</table>
</html>
```

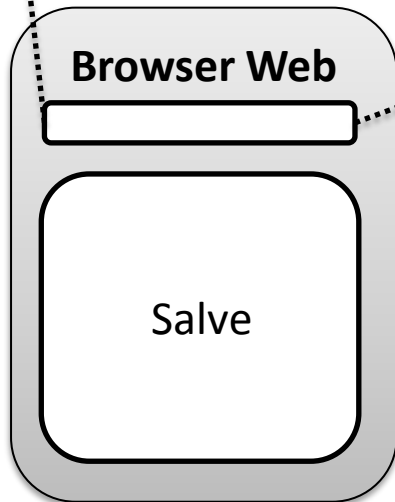

Passaggio di parametri con la richiesta

- Struttura di una URL
- protocol://hostname:port/path?query_string#fragment_id
- Ex di url con Query_string
 - **http://localhost/lpw.php?regione=Toscana&provincia=Pisa**

Passaggio di parametri con la richiesta

URL

http://localhost/lpw.php?reg=Toscana&pro=Pisa



1. lpw.php
2. reg=Toscana
3. pro=Pisa

lpw.php

```
<?php
$regione   = $_GET['reg'];
$provincia = $_GET['pro'];
echo "la regione è $regione</br>";
echo "la provincia è $provincia";
?>
```

Il Esercizio

- Scrivere una **URL con una Query String** dove l'utente possa **passare** una parola e un colore
 - <http://localhost/ex2.php?word=Mondo&color=red>
- Scrivere un **programma php** che **legge** i dati della URL e in base al colore restituisce una pagina HTML contenente la parola nel colore indicato

Il Esercizio

- <http://localhost/lpw.php?word=Welcome&color=red>
- La pagina HTML restituita dovrà essere visualizzata dal browser nel seguente modo



Welcome

Conclusioni

- La maggior parte dei siti web di oggi sono [Applicazioni web](#), ovvero abbinano alla parte statica del protocollo HTTP una componente di programmazione e, tipicamente, una base di dati.
- La componente di programmazione può essere delegata al server e/o al client (tecnologia server-side vs client-side). Server side permette applicazioni più complesse e la connessione ad un database.
- In questo corso tratteremo il lato server e in particolare vedremo un'introduzione al linguaggio di script PHP e la connessione al database MySQL

Costruiamo il nostro ambiente di lavoro

- Web Server, PHP Processor, MySql
- Editor di testo
- Ambiente di sviluppo integrato IDE

[W | M | L]AMP

- Pacchetto Sw contenente 3 componenti Free/Open Source opportunamente configurati
 - Il Server Web APACHE
 - Il DBMS MySql
 - Un Inteprete di linguaggio script (Php|Perl|Python)
- Sono la piattaforma su cui sviluppare applicazioni Web
 - W=Windows
 - M=Mac
 - L=Linux

[W | M | L]AMP

[W M L]AMP	Free/GPL	Win	Mac	Linux
XAMPP	☺	☺	☺	☺
EasyPHP	☺	☺		
AMPPS	☺	☺		
Z:WAMP Server	☺	☺		
VertrigoServ	☺	☺		
Bitnami	☺	☺	☺	☺
Lamp	☺			☺
Mamp	☺		☺	

Editor di Testo per PHP, HTML, JS, CSS

	Free	Win	Mac	Linux
Notepad++				
HTMLKit				

Integrated Development Environment

IDE	Free	Win	Mac	Linux
Eclipse PDT	☺	☺	☺	☺
Komodo IDE		☺	☺	☺
Netbeans	☺	☺	☺	☺
phpDesigner		☺		
PHPEclipse	☺	☺	☺	☺
PhpED		☺		☺
PHPEdit		☺		
Zend Studio		☺	☺	☺