

# Laboratorio di Progettazione Web



AA 2012/2013

*Andrea Marchetti*  
*Maurizio Tesconi*

*IIT- CNR*

[andrea.marchetti@iit.cnr.it](mailto:andrea.marchetti@iit.cnr.it)

[maurizio.tesconi@iit.cnr.it](mailto:maurizio.tesconi@iit.cnr.it)

# Programma del Corso

- Introduzione alle applicazioni web
- Introduzione a PHP – i costrutti di base
- Il database MySQL
- Accesso a MySQL in PHP
- Progetto Finale
- Seminari
  - **Il WWW** – Oreste Signore - responsabile W3C Office Italiano
  - **Principi di Grafica orientata al Web** – Beatrice Rapisarda – IIT/CNR
  - **Principi di Usabilità e Accessibilità per il Web** – IIT/CNR
- Laboratorio

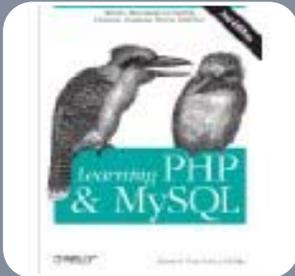
# Riferimenti e Materiale

- I lucidi, l'orario aggiornato e comunicazioni varie saranno disponibili sulla pagina web del corso:

<http://didawiki.cli.di.unipi.it/doku.php/bdd-infuma/>

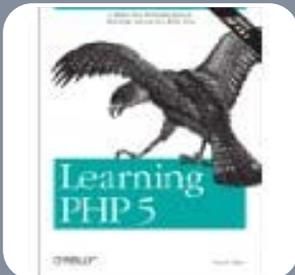
- <http://www.php.net>
- <http://php.html.it>
- Numerosi altri siti reperibili sul web cercando PHP e MySQL
- Numerosi libri sono disponibili: cercare quelli che trattano PHP e MySQL

# Riferimenti e Materiale



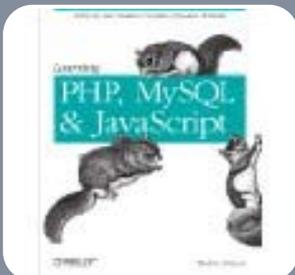
## Learning PHP & MySQL: Step-by-Step Guide to Creating Database-Driven Web Sites

- Michele E. Davis e Jon A. Phillips
- 9 feb. 2009



## Learning PHP 5

- David Sklar
- 17 dic. 2008



## Learning PHP, MySQL, and JavaScript

- Robin Nixon
- 3 lug. 2009

# Applicazioni Web

- Applicazioni
  - Architettura Standalone
  - Architettura Client-Server
- Architettura Web
  - Browser Web, Server Web
  - URL, HTTP, HTML
  - Esempi
- Applicazioni Web
- Tecnologie Client-side e Server Side

# Applicazioni Web – Def.

- + Una applicazione web fornisce **un servizio all'utente finale** per mezzo dell'infrastruttura web.
- + In una applicazione web l'utente **interagisce** con il sito
- + Si parla anche di **siti dinamici**, dove le pagine non sono statiche ma costruite dinamicamente in base all'interazione con l'utente

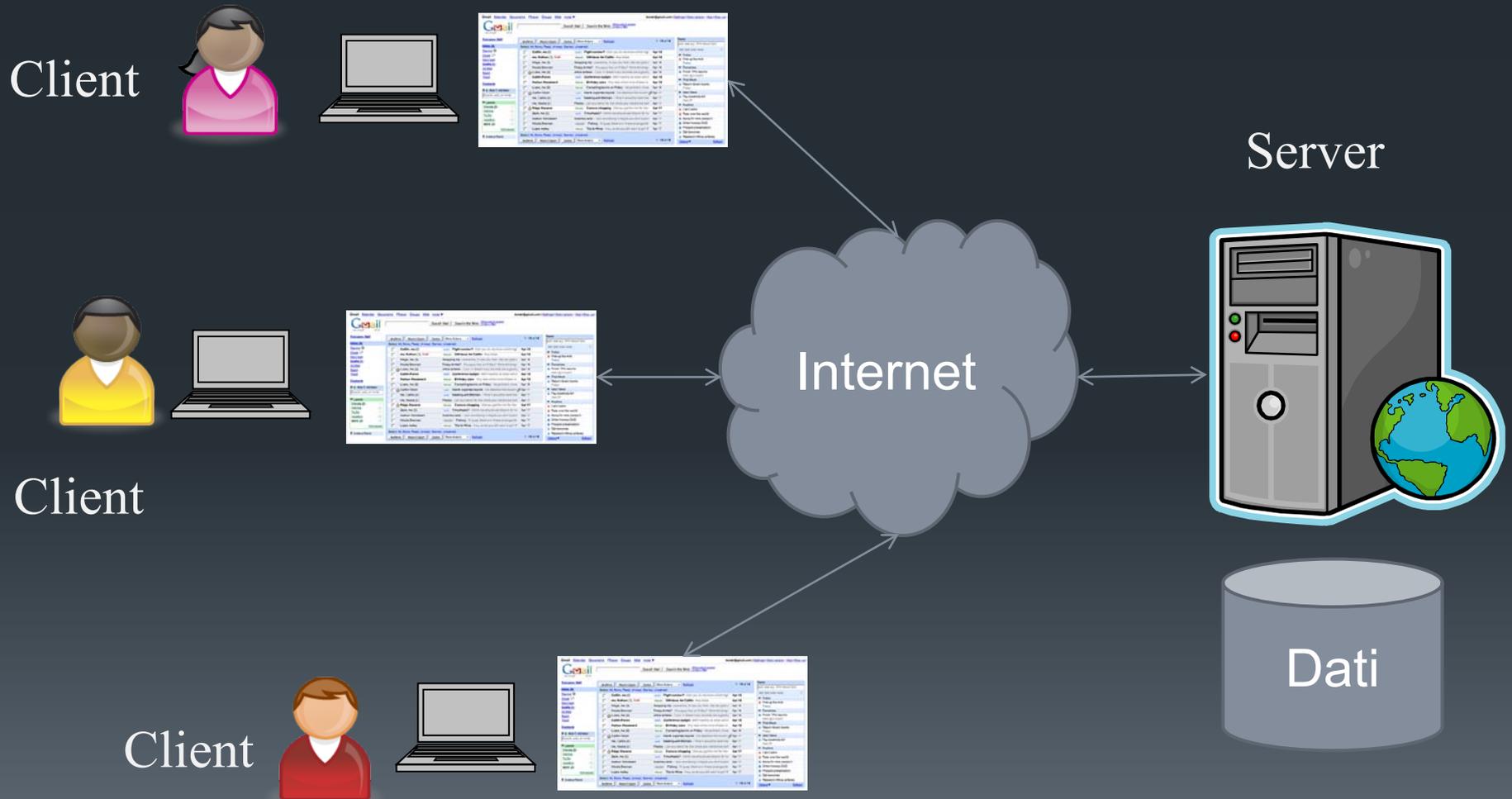
# Architetture Applicazioni

- **Standalone**
- **Client-Server**
- Peer to Peer
- Cloud Computing
-

# Architettura standalone



# Architettura client-server

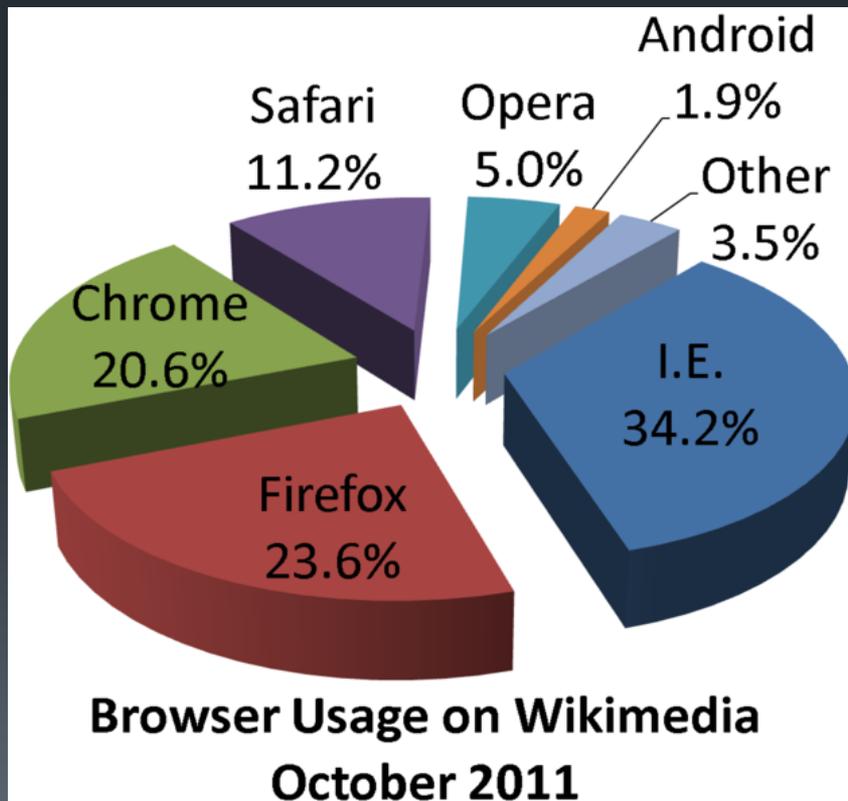


# Standalone Vs Client-Server

- Ubiquità
  - Con il modello Client-Server posso accedere ai miei dati da qualsiasi client connesso
- Cooperazione
  - Con il modello Client-Server posso pensare di interagire con altri utenti
- Connessione
  - Senza una connessione un applicazione Client-Server è praticamente inutilizzabile
- Protezione dei dati
  - In un applicazione Client-Server la protezione dei dati è demandata a terzi
- Esistono altri modelli/architetture di applicazioni (peer-to-peer, cloud computing, )

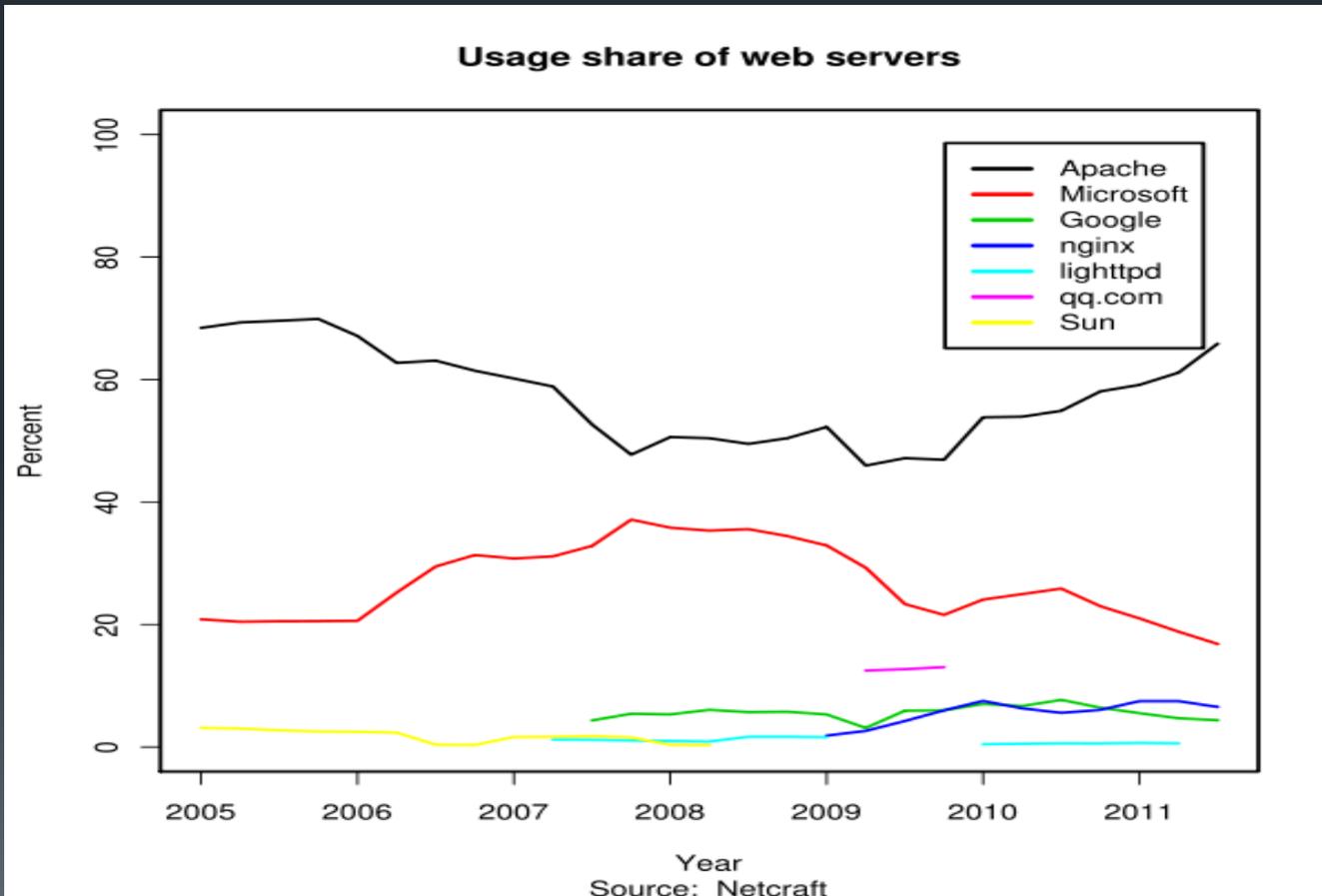
# Architettura web

- Architettura Client – Server
- I client sono detti Browser Web



# Architettura web

- I server sono detti Server Web o anche Http Server



# Architettura Web

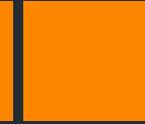


<http://www.example.com/LPW.html>

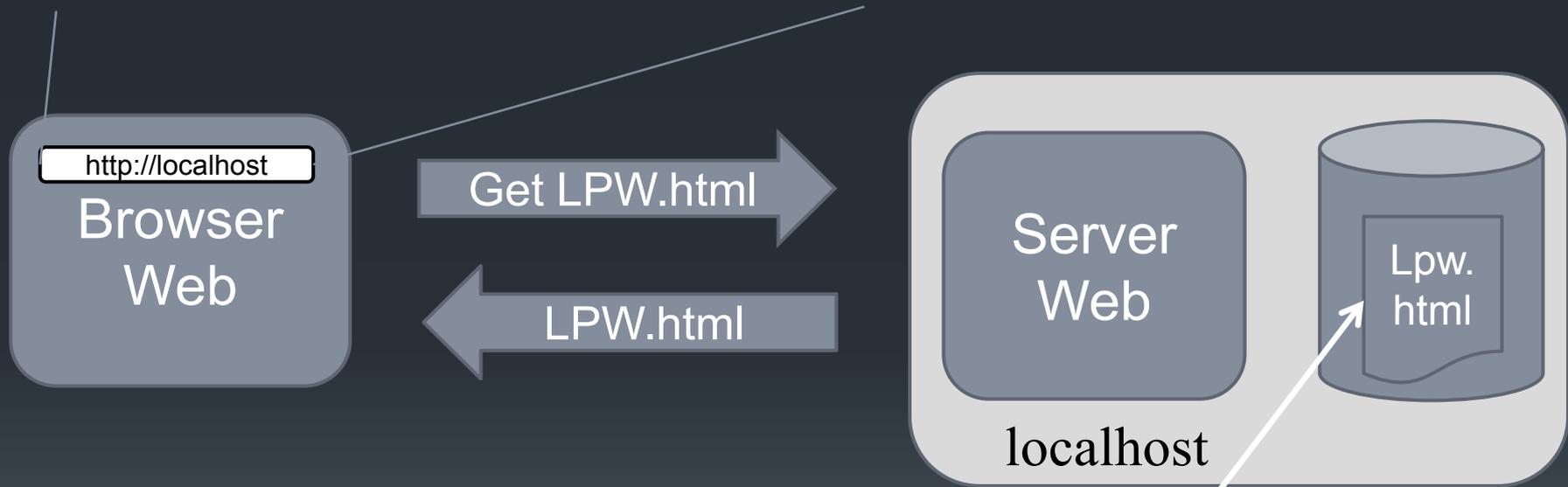


`C:/www/xampp/htdocs/LPW.html`

# Architettura Web



http://localhost/LPW.html



Browser e Server  
sulla stessa macchina

C:/www/xampp/htdocs/LPW.html

# Esempio

- LPW.html

- `<html>Salve! </html>`

# Esempio + dinamico

- LPW.php
  
- <html>
- Salve!
- <?php echo date("d/m/y h:i:s"); ?>
- </html>

# Architettura Web

- Fin dalle origini il WWW si basa sulle seguenti tecnologie
  - URL – Uniform Resource Locator
  - HTTP – Hyper Text Transfer Protocol
  - HTML – Hyper Text Markup Language

# Architettura Web - URL

- Uniform Resource Locator
- protocol://hostname:port/path?query\_string#fragment\_id
  - Protocollo
  - Hostname
  - Porta di ascolto del server web (opzionale se 80)
  - Path
    - Assoluto -> a partire dalla root configurata del server web di solito htdocs
    - Relativo -> a partire dall'ultima pagina caricata dal browser
  - Query\_string: Coppie attributo=valore
    - Query=Comuni&Regione=valle%20%20d'aosta
  - Fragment\_id
- Esempi di url validi
  - <http://www.google.it/search?q=www&num=10&hl=en>
  - [Localhost/lpw.html](http://localhost/lpw.html)
- Per approfondire <http://en.wikipedia.org/wiki/URL>

# Architettura web - HTTP

- In un modello Client-Server i client possono essere differenti, ovvero creati da soggetti diversi.
- Tutti i client si devono uniformare a come il server espone i suoi servizi.
- Protocolli di comunicazione (Application Layer Protocol)
  - FTP
  - HTTP
  - SMTP
  - IRC
  -
- TP = Transfer Protocol

# Architettura web - HTTP

- L'infrastruttura web si basa sul protocollo di comunicazione **HTTP**
  - **HyperText** Transfer Protocol
  - Trasferimento di IperTesti = documenti HTML (HyperText Markup Language)
- HTTP functions come un protocollo **richiesta-risposta**
- Il browser invia una richiesta HTTP al server web contenente:
  - La richiesta "Get lpw.html HTTP1.1"
  - Una lista di Headers
    - Accept: text/plain
    - Host: localhost
    - User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)

# Architettura web - HTTP



- Il server elabora la richiesta e restituisce un messaggio di risposta contenente oltre che la richiesta anche delle meta-informazioni come l'esito dell'operazione.
- Per approfondire <http://en.wikipedia.org/wiki/HTTP>

# Ambiente di lavoro

- Mac OS X:
  - MAMP: <http://www.mamp.info/en/index.html>
  - XAMPP: <http://www.apachefriends.org/en/xampp.html>
- Windows:
  - XAMPP: <http://www.apachefriends.org/en/xampp.html>
  - EasyPhp: <http://www.easyphp.org/>
  - Lighty2Go: <http://www.lighty2go.com/get-lighty2go/>
  - Wamp Server: <http://www.wampserver.com/en/>
- Linux
  - XAMPP: <http://www.apachefriends.org/en/xampp.html>
  - Bitnamp: <http://bitnami.org/stack/lampstack>
  
- Un editor di testo come Notepad++



XAMPP

