

Laboratorio di Basi di Dati

Applicazioni di rete

Anno accademico 2014-2015

Marco Mesiti

**Parte di questi lucidi è tratta da una versione precedente
di Marco Mesiti. Stefano Valtolina e Sergio Mascetti**

Prima parte: introduzione



Informazioni sul corso

- 24 lezioni – Aula Tau (via Comelico)
 - Lunedì 10:30 – 12:30
 - Mercoledì 12:30 – 14:30
- Iniziamo ai '45 e proseguiamo fino alla mezza

Mese	giorni
Ottobre	1,6,8,13,15,20,22,27,29
Novembre	3,5,10,12,17,19,24,26
Dicembre	1,3,10,11,15,17



Sito del corso

- Lo stesso del corso di teoria
 - <http://homes.di.unimi.it/dbandsec/DB/1415/>
- Trovate:
 - Tutte le regole del corso (comprese quelle d'esame)
 - Il calendario delle lezioni e il materiale didattico
 - Nome utente: db1415
 - Password: db5141
 - Programma del corso
 - Ultime notizie:
 - controllate le ultime notizie la sera prima di una lezione. Se ci sono modifiche, le trovate là.



Contatti con il docente

- Ricevimento studenti:
 - Lunedì 15:30-16:30 (o su appuntamento via mail)
- Prima di scrivere una mail, controllate che quello che state per chiedere non sia sul sito
- Prima di scrivere una mail, leggete qua:
 - <http://vigna.dsi.unimi.it/comescrivere.php>
 - (ci vogliono 5 minuti e vi fate anche una risata)



Argomenti del corso di laboratorio

- Basi di HTML + linguaggio di scripting PHP
- Progettazione di una base di dati
- SQL
- Costruzione, interrogazione e accesso a basi di dati relazionali (MySQL + PHP)



Progetto

- Il progetto è pensato per gruppi di 2 studenti. Si sconsiglia di fare il progetto da soli
- Il progetto verrà valutato in trentesimi
- La valutazione è individuale. Il voto del laboratorio fa media con il voto dello scritto del corso di basi di dati
- Il voto minimo per passare entrambe le prove è 18. Qualora uno dei componenti del gruppo non raggiunga la sufficienza, sarà tenuto a svolgere un nuovo progetto da concordare nelle modalità e nei tempi con il docente
- Il voto è valido per un anno. Dopo tale data occorrerà sostenere un nuovo progetto
- Se durante la discussione del progetto si riscontra che uno dei membri del gruppo non ha partecipato alla realizzazione del progetto, l'intero gruppo sarà tenuto a fare un nuovo progetto



Studenti frequentanti

- Seguire almeno 18 lezioni di laboratorio
- Ottenere parere positivo su due questionari che saranno proposti durante le lezioni di laboratorio
- Gli studenti frequentanti che superano entrambe le prove
 - entro fine Marzo 2015 riceveranno un bonus di **3 punti** sul voto finale
 - entro la fine di Luglio 2015 riceveranno un bonus di **1 punto** sul voto finale



FAQ

- **Per la realizzazione del progetto possiamo rivolgerci ai docenti?**

Assolutamente SI. Gli studenti possono presentare le loro idee ai docenti durante le ore di ricevimento.

- Durante il corso, verranno organizzate delle lezioni per discutere i progressi nello svolgimento del progetto



FAQ

- **Quant'è importante la parte di progettazione rispetto all'implementazione?**

Fondamentale. Un qualunque applicativo deve fondarsi su una buona progettazione rigorosa e metodologica. Una qualunque casa costruita su buone fondamenta cresce bene. Se si sbagliano le fondamenta, la casa crolla. Per cui non stupitevi se "vi si manda a casa" senza aver visto il vostro programma "girare". Se non ha alla base una buona progettazione è inutile vederlo.



FAQ

- **Nel caso in cui partecipi solo ad una parte di lezione di laboratorio (arrivo tardi / esco prima della fine) conta comunque la mia presenza?**

No. Lo studente deve essere presente durante tutta la lezione. Nel caso in cui arrivi in ritardo o esca prima, non deve firmare il modulo di presenza. Verranno effettuati controlli per verificare l'effettiva presenza. Se uno studente risulta assente nonostante la sua firma sia presente verrà considerato come studente non frequentante.



FAQ

- **Se consegno una documentazione tecnica, con meno informazioni di quelle richieste nel template posso comunque discutere il progetto?**
- **No.**



FAQ

- **Posso utilizzare un altro DBMS o un altro linguaggio di script?**
- **No.** Durante il corso vengono insegnati questi applicativi e si ritiene utile che voi li impariate per la vostra crescita professionale



Seconda parte: programmare per il web

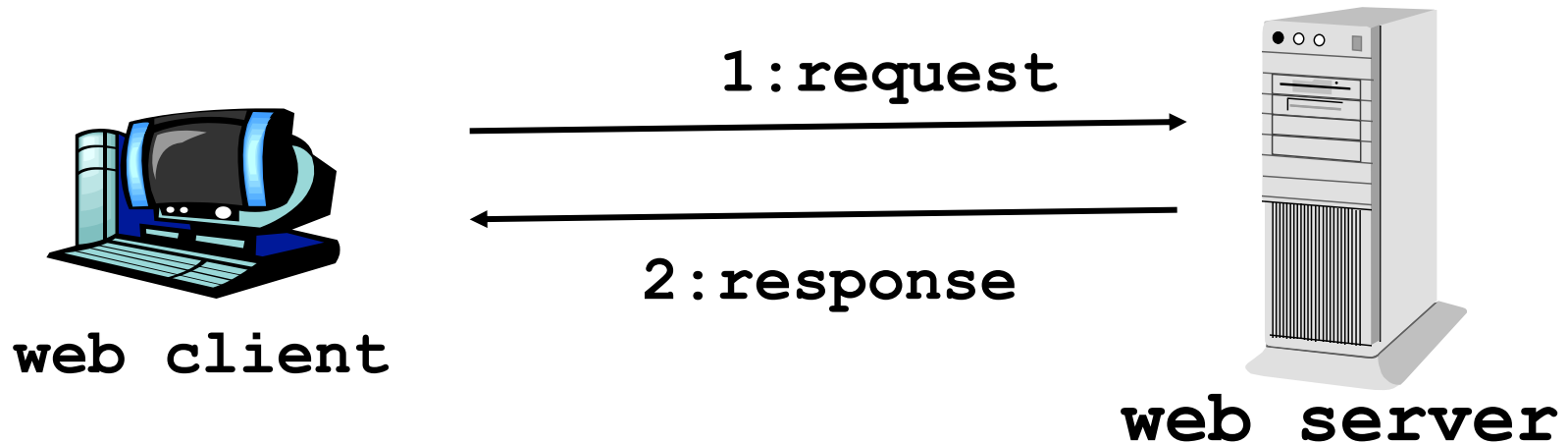


Pagine web statiche: l'idea

- Pagine che mostrano sempre le stesse informazioni
 - Esempio: pagina in cui vengono utilizzati i tag fondamentali di HTML
 - Per modificare il contenuto della pagina è necessario modificarla "a mano"



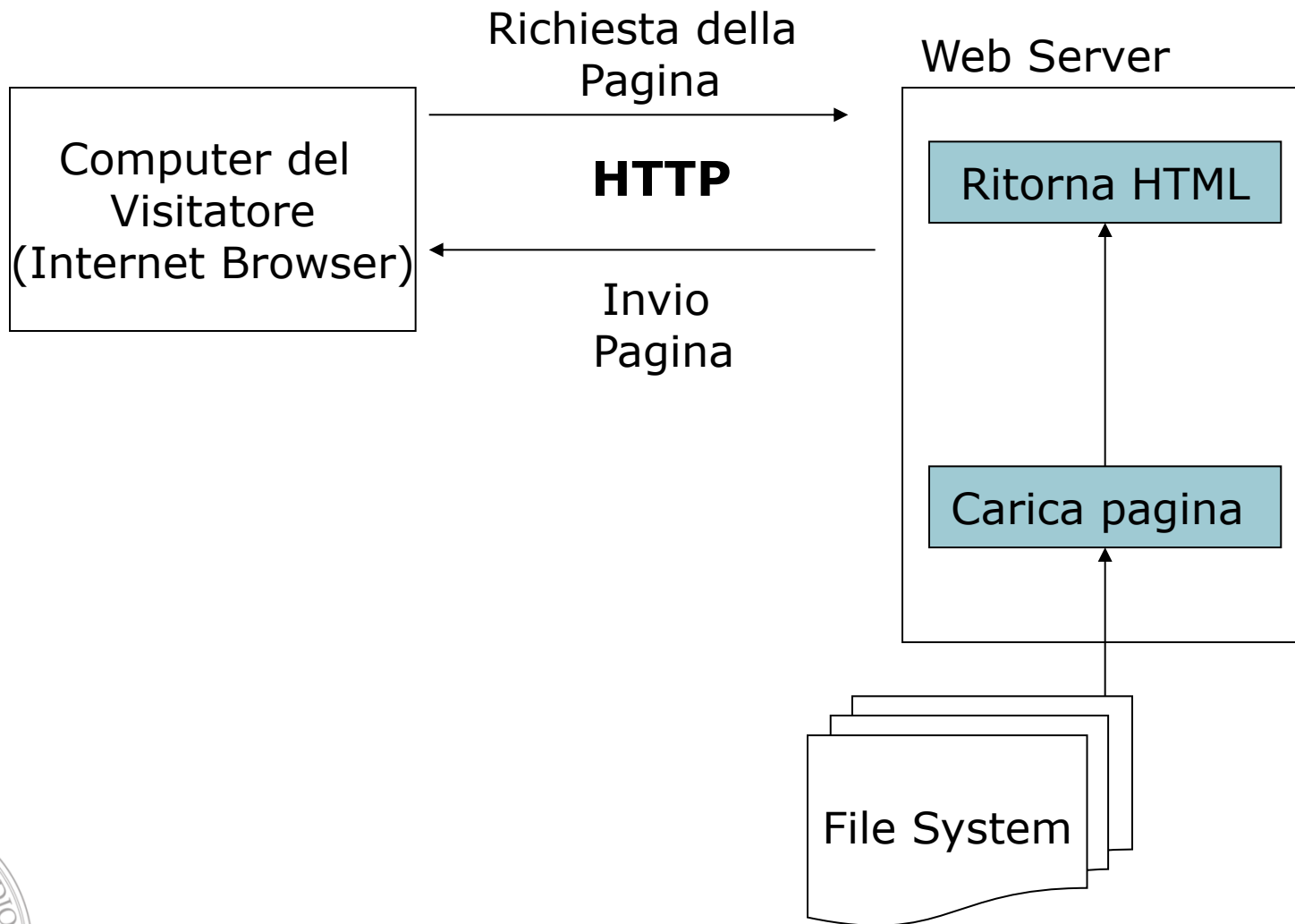
Pagine web statiche



- Per ogni pagina, il web server memorizza un file (es: HTML)
- Quando richiesto, il server invia il file al client
- Il browser visualizza la pagina web



Caricamento di una pagina Statica



Programmare per il web

- La quasi totalità dei siti ha comportamenti molto più complessi rispetto alle pagine in HTML semplice
 - es: gmail, facebook...
- Differenza tecnica principale:
 - Oltre al semplice trasferimento di informazioni, viene eseguito del codice
 - Sul server e/o sul client
 - Questo codice viene detto "script"

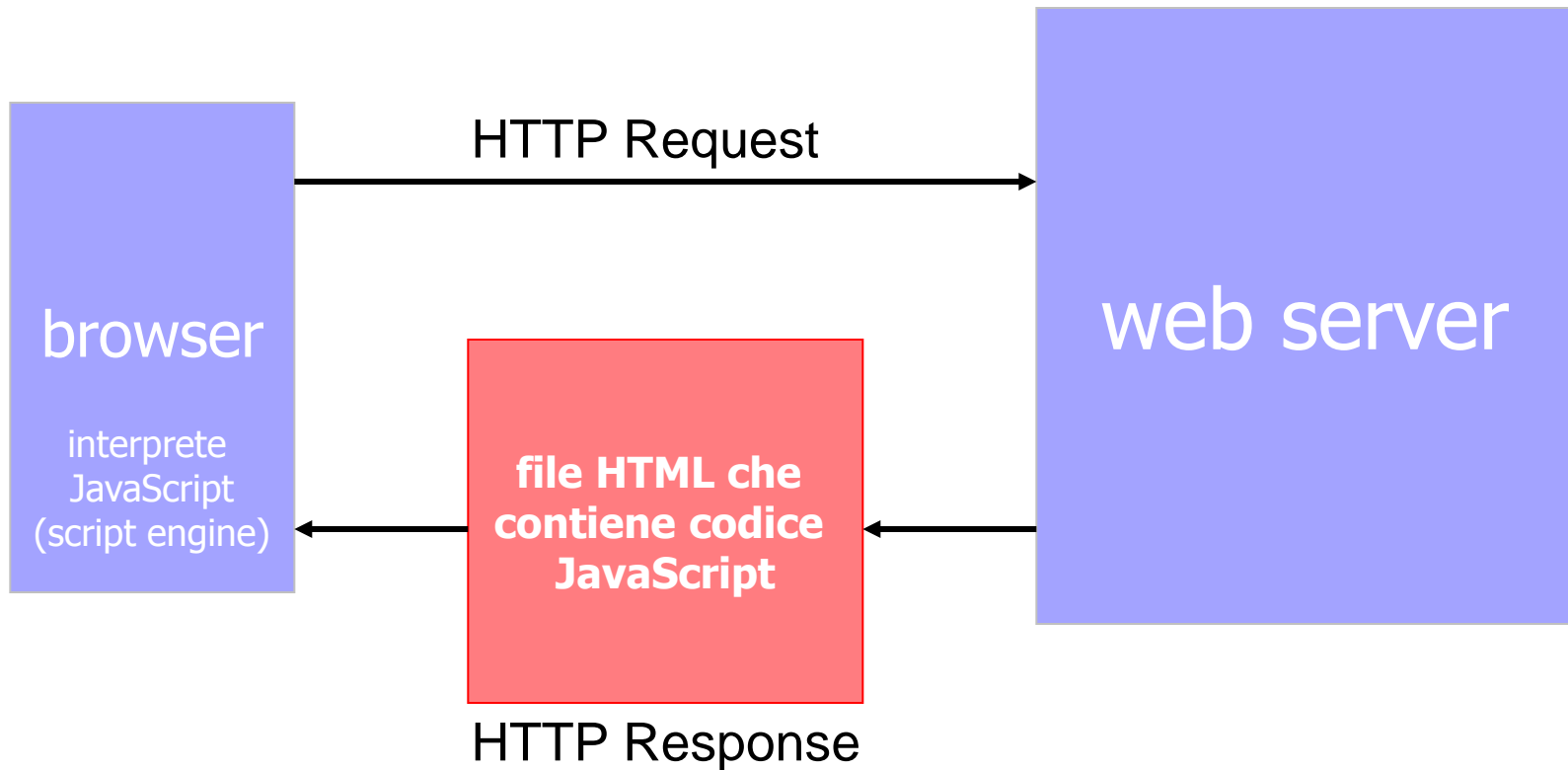


Script lato client

- Alcune pagine possono contenere del codice che viene eseguito sul client
- Quando il browser riceve la pagina
 - esegue il codice (mediante lo "script engine")
 - visualizza la pagina
- Esempio di linguaggio di script lato client:
 - Javascript



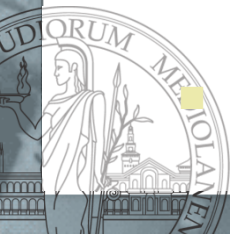
JavaScript



Script lato server e pagine web dinamiche

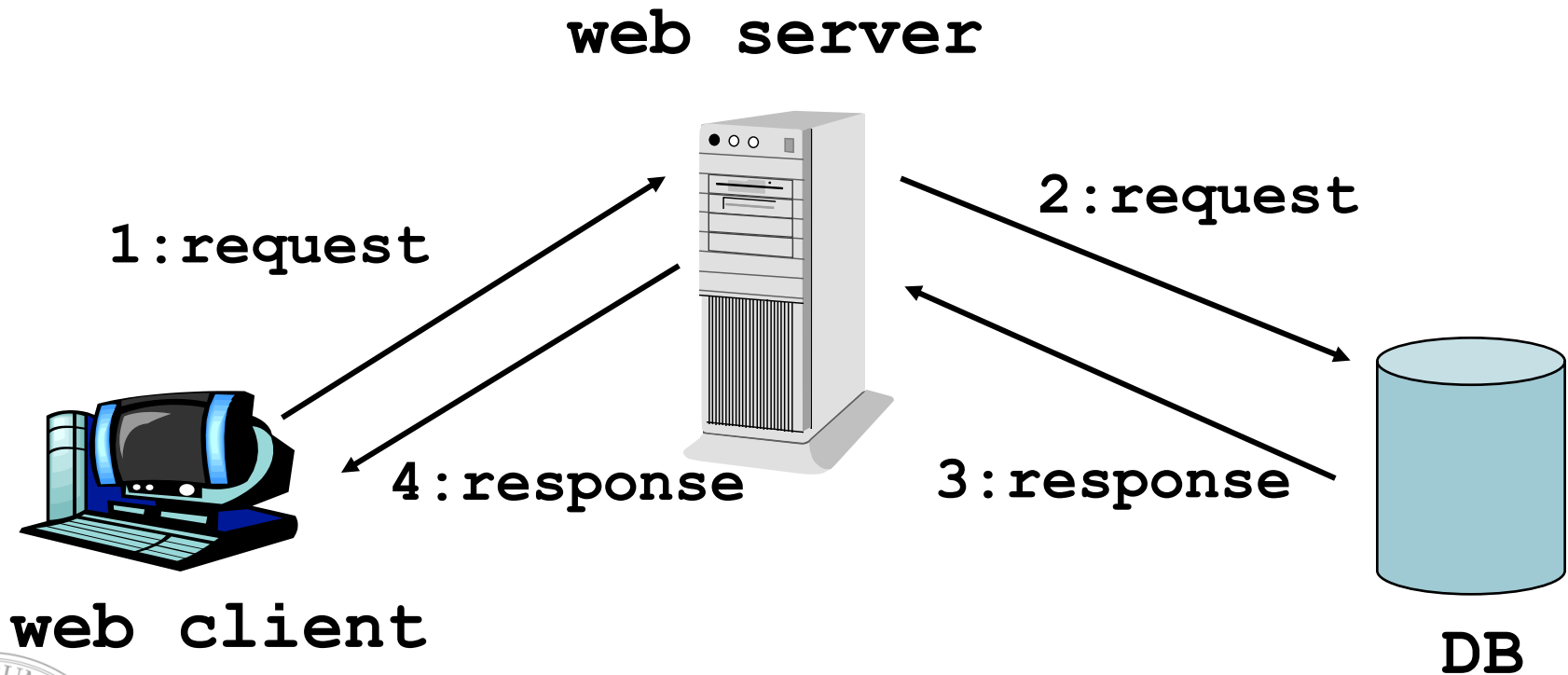
- Per ogni pagina, il web server memorizza un file che contiene:
 - informazioni di marcatura (es: HTML) e
 - del codice di script
- Quando la pagina viene richiesta il server:
 - esegue il codice che genera altre informazioni di marcatura (es: HTML)
 - invia il risultato al client

Il browser mostra la pagina

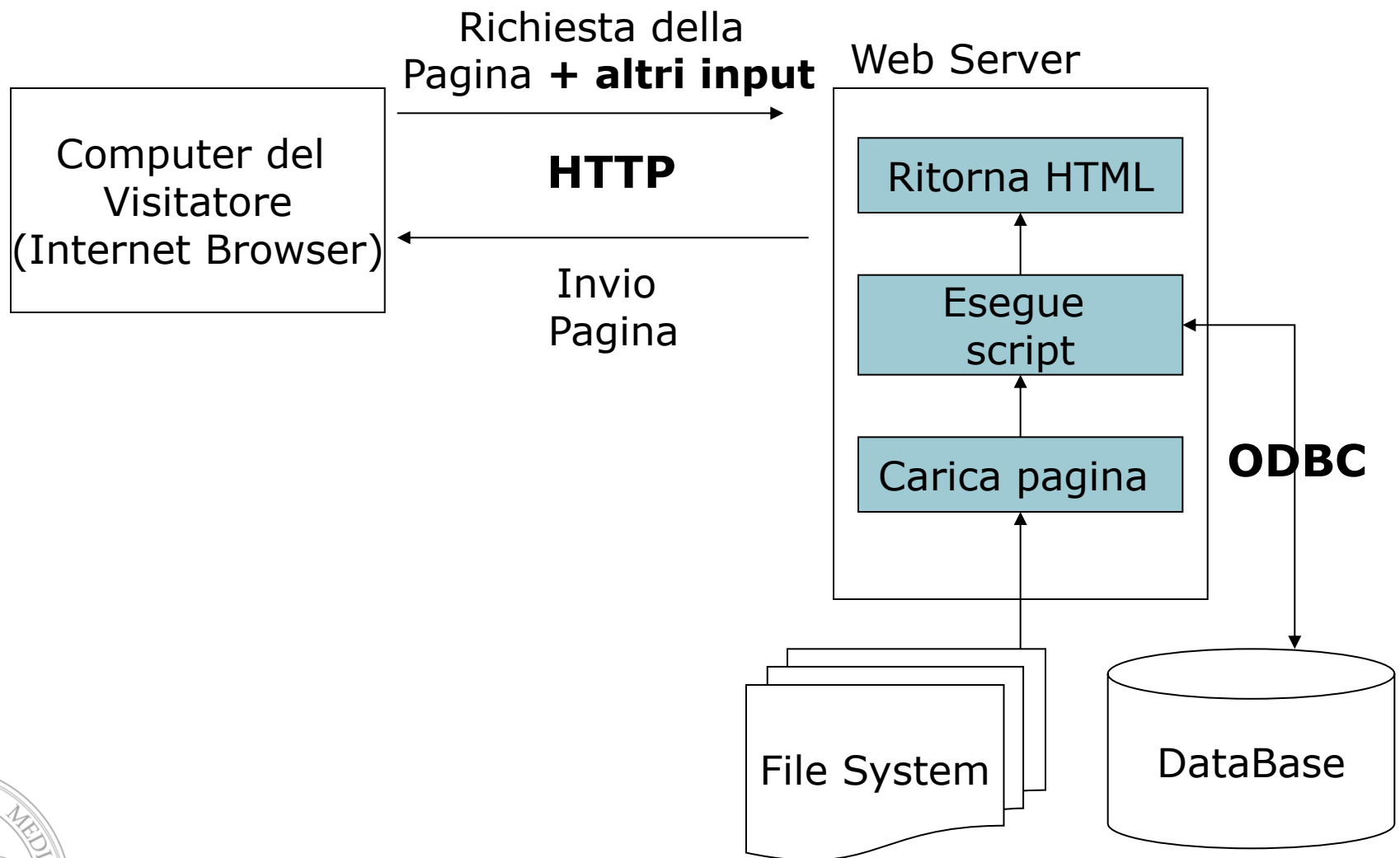


Architettura "3 -tier"

- Uno script lato server può interagire con una base di dati per leggere e/o scrivere dati



Caricamento di una pagina dinamica



Riassumiamo: Input e Output del web server

		Scripting lato client	
		NO	SI'
Scripting lato server	NO	Input: indirizzo pagina Output: HTML	Input: indirizzo pagina Output: HTML + codice di scripting (eseguito dal client)
	SI'	Input: indirizzo pagina + altre info Output: HTML	Input: indirizzo pagina + altre info Output: HTML + codice di scripting (eseguito dal client)



Quali tecnologie utilizzeremo durante questo corso?

- Sul client:
 - HTML come linguaggio di marcatura
 - CSS per definire i fogli di stile
 - JavaScript (opzionale) per scripting lato client
- Sul server:
 - Apache come web server
 - PHP scripting lato server
 - MySQL DBMS



Introduzione a HTML



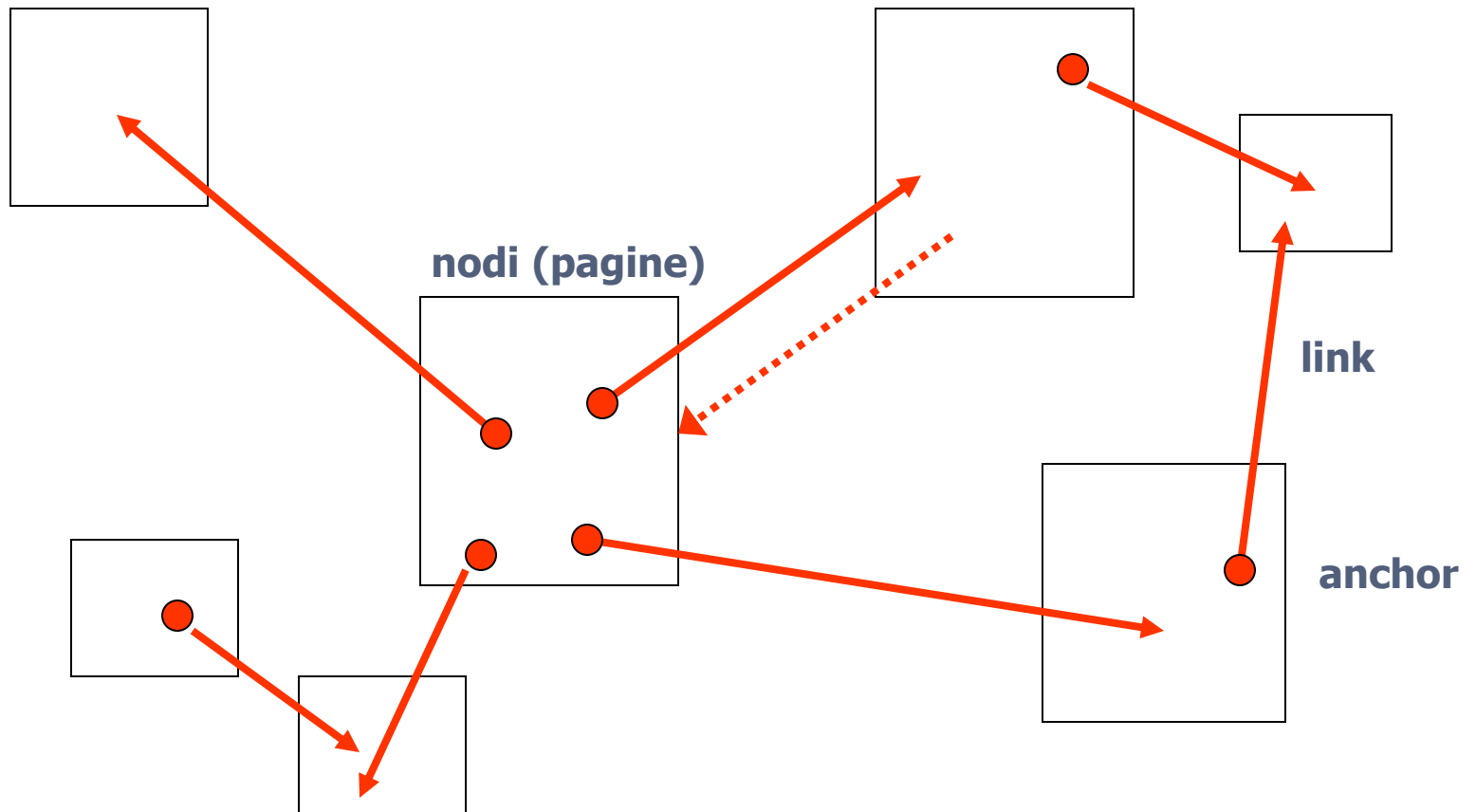
HTML: un rapido ripasso

- E' l'acronimo di **H**yper **T**ext **M**arkup **L**anguage
- Linguaggio di marcatura
 - NON di programmazione!
 - Serve per definire un ipertesto
- I TAG servono per marcare il testo
 - HTML: `ciao`
 - Risultato: **ciao**



Organizzazione di un ipertesto

lettura secondo **legami associativi**



HTML: struttura di una pagina

<html>

<head>

caratteristiche del documento

</head>

Parte opzionale

<body>

documento

</body>

</html>



HTML: è facile, prova tu!

- Crea un file (sul desktop) con estensione .html
- Apri il file con blocco note
- Inserisci all'interno del file
ciao `come` stai?
- Apri il file con il browser
 - Hai capito cosa vuol dire ``?
 - Hai capito cosa vuol dire ``?



HTML: gli attributi

- In alcuni casi un TAG ha bisogno di informazioni aggiuntive
 - Esempio: se vogliamo creare un link, bisogna indicare a quale pagina punta il link
- La sintassi per gli attributi è molto semplice
 - Esempio:
`ciao`
 - HREF è il nome dell'attributo
 - "www.google.com" è il valore impostato per l'attributo



HTML: quali tag posso usare?

- Esiste un insieme di tag che si possono usare
 - Insieme relativamente piccolo, sono qualche decina
 - Quindi li potete guardare tutti in poco tempo!
- Alcune risorse (ce ne sono tantissime)
 - <http://www.w3schools.com/>



Esercizio in aula

- Modifica la pagina che hai creato in precedenza in modo da utilizzare i seguenti tag:
 - A, P
 - H1, H3, H6
 - UL, LI
 - table, th, tr, td
- “Come si usano?”
 - Cerca sul web



HTML cosa ci interessa in particolare

- Nella gestione di pagine web dinamiche serve creare diversi strumenti di interfaccia che permettono l'interazione con l'utente:
 - Pulsanti, caselle di testo, caselle di scelta, etc...
 - Vedremo nelle prossime lezioni come si definisce il comportamento
 - Vediamo ora come si possono inserire in una pagina HTML



Le form

- Le form sono dei moduli che possono essere compilati dai visitatori di un sito permettendo così la creazione di pagine HTML interattive e non solo di consultazione
- Si deve usare il comando
`<form> descrizione </form>`



I pulsanti

- E' possibile inserire un pulsante
- Cosa succede quando viene premuto un pulsante?
 - Viene eseguito del codice lato client (con uno script) e/o
 - I dati della form vengono inviati al server
 - Nelle prossime lezioni vedremo come il server può utilizzare questi dati.
- Esempio:
`<input type="submit" value="Invia">`



Altri elementi di interfaccia

- Altri elementi possono essere creati con il tag `<input>` e valori diversi dell'attributo `type`

`type="text"`

`type="checkbox"`

`type="radio"`

`type="submit"`

`type="hidden"`

`type="password"`

`type="file"`

`type="reset"`

- **NB:** ogni elemento di un modulo deve avere un nome (`name="...."`) che viene usato al momento dell'invio dei dati

- Esempio: `<input type="password" name="pwd">`



Prova tu: Crea la pagina HTML per ottenere il seguente risultato

Prova di documento - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History

Address C:\Documents and Settings\marina\Desktop\ Go Links

Servizi riservati, per accedere devi digitare user name e password e poi premere il pulsante OK

user name marina

password *****

OK Cancella

Questa è una tabella che si utilizza per allineare i campi

TAG <INPUT> di tipo text

TAG <INPUT> di tipo password

Bottone di tipo input

Bottone di tipo reset

- Per ora NON devi definire il comportamento (cioè non deve succedere nulla quando si preme "OK") ma devi solo visualizzare questa form

Prova tu: continua

- A partire dal form che hai appena sviluppato, costruisci un altro form che una volta autenticato nel sistema ti chieda alcune informazioni per ottenere un servizio
- L'utente deve:
 - Inserire nome e cognome
 - Specificare se è maschio o femmina
 - Scegliere (anche più di una) delle attività che gli piacciono in una lista formata da sci, tennis, golf, canoa, altro
 - Confermare la lettura delle condizioni di utilizzo



Un errore comune in HTML

- Usare il tag `` per formattare del testo.
Problemi:
 - necessario usare il tag `` tante volte (con parametri diversi)
 - per cambiare la formattazione è necessario modificare il tag `` che si ripete
 - nella stessa pagina ma anche in pagine diverse
- Il tag `FONT` è stato deprecato
 - “E allora come si fa?” → fogli di stile



CSS – CASCADE STYLE SHEETS



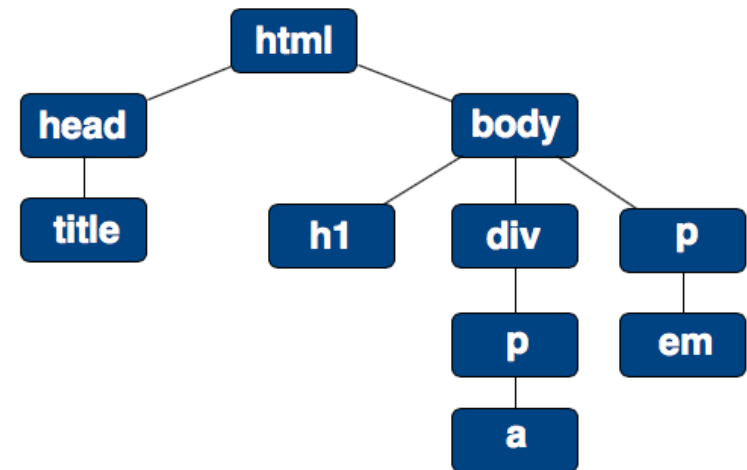
Che cos'è CSS

- **CSS** è l'acronimo di **Cascading Style Sheets**
- Permette di definire gli stili di presentazione di degli elementi contenuti in documenti HTML
- Rendono modulare e uniforme la presentazione delle pagine dello stesso sito
- Separare il **contenuto** di un documento da aspetti legati alla sua **presentazione**



Struttura ad albero di una pagina HTML

```
<html>
  <head>
    <title>Struttura del documento</title>
  </head>
  <body>
    <h1>Titolo</h1>
    <div>
      <p>Primo <a href="pagina.htm">paragrafo</a>.</p>
    </div>
    <p>Secondo <em>paragrafo</em>.</p>
  </body>
</html>
```



- Da ogni documento HTML ben formato, si può estrarre una struttura ad albero
- E' possibile associare una proprietà ad un nodo della struttura e farla **ereditare** in **cascata** a tutto il sottoalbero

Foglio CSS: collezione di regole



- Ogni regola è composta da:
 - **Selettore:** identifica la parte del documento a cui verrà applicata la regola
 - **Blocco delle dichiarazioni:** Contiene la dichiarazione di coppie **proprietà:valore**
 - La **proprietà** definisce un aspetto del selettore da modificare (margini, colore di sfondo, larghezza, etc) secondo il valore espresso

body {color: white; background: black;}

p {font: 12px Verdana, arial;}



Tipi dei valori

tipi	descrizione	esempio
numeri	interi (1, 23, ...) in virgola mobile (1.2)	p {line-height: 1.2;}
unità di misura	in,cm,mm,...	div {width: 300px;}
percentuali	Rispetto a elemento padre	div {width: 60%;}
codici colore	Rappresentazione del colore (parola chiave, rappresentazione esadecimale,...)	body {background-color: #2795b6;}
URI	URL di un documento esterno	body {background-image: url(sfondo.jpg);}
stringhe di testo		content: "Viva i CSS";

I valori non sono mai racchiusi tra parentesi a meno che non siano formati da più parole separati tra spazi o siano stringhe di testo



I selettori

selettore	descrizione	esempio
universale	serve a selezionare tutti gli elementi di un documento . Si esprime con il carattere *	* {color: red;}
Di tipo	È costituito dal nome di uno specifico elemento HTML. Serve a selezionare tutti gli elementi di quel tipo presenti in un documento.	h1 {color: green;} p {background-color: yellow;}
id e class	Sono attributi che possono essere specificati per ogni elemento HTML (il valore di id deve essere unico nel documento) Ad essi è possibile associare una rappresentazione nel foglio di stile	#titolo {color: blue;} <h1 id="titolo">...</h1> ----- .testob {color: white;} <p class="testob"> testo1 bianco.</p> <p class="testob"> testo2 bianco.</p>

Dove si definisce lo stile

- Lo stile da usare in una pagina si può definire
 - All'interno del tag <HEAD>
 - usando il tag <STYLE>
 - In un file esterno, a cui si fa riferimento all'interno del tag <HEAD>
 - usando il tag <LINK>
- Quale delle due tecniche è più comoda secondo te?
 - Sai spiegare il perché?



Esercizio 1

- Prendi il file EsercizioCSS.html sul sito del corso
 - Il file contiene un'intervista. Usando i fogli di stile, fai in modo che le domande siano in rosso e le risposte in verde.
- "Ehi, ma non mi hai spiegato come si fa, non so cosa devo fare."
 - Hai capito l'idea? Bene, cerca sul web un esempio e adattalo al tuo problema



Esercizio 1 (cont)

- Ora fai una copia del file EsempioCSS.html
 - Facciamo finta di avere due interviste
 - Le due pagine hanno la stessa formattazione
- Ora cambia la formattazione delle due pagine: metti in verde le domande e in rosso le risposte.
 - Quanti file hai dovuto modificare per apportare questa modifica?



CENNI SU JAVASCRIPT



Breve introduzione a JavaScript

- Il browser **visualizza** il documento **HTML** e interpreta (**esegue**) le istruzioni scritte in **JavaScript**

```
<script language="JavaScript">
```

```
<!--
```

```
codice JavaScript
```

```
//-->
```

```
</script>
```



Gli eventi in JavaScript

- **L'interazione** con l'utente è guidata dagli **eventi**
- **L'utente genera un evento**
- Se nel file HTML esiste del codice JavaScript associato a quell'evento (**event handler**), questo viene **eseguito**



JavaScript

- Molto spesso il codice JavaScript è scritto **all'interno dei tag HTML** come valore di nuovi attributi che sono stati introdotti per gestire gli eventi generati dall'utente. Esempio:

```
<a href="www.google.com"  
  onmouseover="alert('hai passato il  
  mouse')">passa il mouse</a>
```

Inserisci il codice in una pagina HTML che hai creato in precedenza e verifica il risultato



JavaScript: un esempio completo

- Prendi il file EsempioJS.html sul sito del corso.
- Riesci a capire a cosa serve lo script?
- Aggiungi una seconda casella di testo e fai in modo che lo script esegua gli stessi controlli anche per quella.



Javascript: Abilitazione

- Si noti che gli script lato client possono essere disabilitati
- Questo vuol dire che i controlli previsti nella visualizzazione della pagina possono mancare
- Nello sviluppo di applicazioni 3-tier questo deve essere tenuto in considerazione e molti dei controlli fatti lato client devono essere effettuati anche dal lato server



“Non capisco”

- “Se comunque devo fare dei controlli lato server (perché gli script lato client potrebbero essere disabilitati) perché devo farli anche lato client? Tanto vale farli solo lato server!”
 - Come rispondereste a questa obiezione?



Quarta parte: Conclusioni ed esercizi



Abbiamo visto

- Differenza tra pagine web
 - Statiche: HTML semplice o scripting lato client
 - Dinamiche: scripting lato server
- Architettura 3-tier
- Alcune tecnologie che useremo:
 - HTML e CSS → obbligatori
 - Javascript → opzionale



Esercizi per casa 1

- Scarica, dal sito del corso, il file `EsercizioHTML.JPG`
 - Il file contiene un'immagine di una pagina HTML
- Sfrutta quanto hai appreso in questa lezione relativamente a HTML e ai fogli di stile per realizzare una pagina analoga.
- Consiglio: ti serve il tag `<SELECT>`.



Esercizi per casa 2

- Scarica, dal sito del corso, il file PaginaContatti.JPG
- Il file contiene un'immagine di una pagina HTML
- Sfrutta quanto hai appreso in questa lezione relativamente a HTML per realizzare una pagina analoga
- Identifica quali controlli occorre fare affinché i dati che vengono inseriti con questa form possano essere considerati «buoni»



Esercizi per casa 3

- Creare una directory fileMusicali
- Inserire in tale directory:
 - Tre video musicali (con un nome significativo)
 - form HTML che richiede uno dei video contenuti nella directory
- Una pagina HTML che permetta di visualizzare (ed eseguire) uno dei video presenti nella directory



Esercizi per casa 4

- Considera la directory menuMultilivello
- All'interno è presente un file HTML in cui sono stati specificati dei menu a tendina utilizzando CSS
- Verifica il funzionamento della pagina
- Utilizzando i css in esso definito prova a creare un menu a tendina con le opzioni specificate nella seguente pagina
- Definisci una pagina per ogni opzione che contenga semplicemente il nome dell'opzione e associala alla corrispondente opzione del menu (quando clicco sulla opzione del menu, nella pagina viene caricata la corrisponde pagina)



Menu per esercizio 4

- Video
 - Catalogo
 - Prenotazione
 - Restituzione
 - Prestiti
- Clienti
 - Lista Clienti
 - Nuovo cliente
 - Modifica cliente
 - Cancellazione cliente

■ Esci

