

# Architettura dell'Informazione

## 8. Realizzare Siti Dinamici e Applicazioni Web

Paolo Milazzo

Dipartimento di Informatica, Università di Pisa

<http://pages.di.unipi.it/milazzo>

[milazzo@di.unipi.it](mailto:milazzo@di.unipi.it)

Master in Turismo e ICT

A.A. 2015/2016

# Introduzione (1)

- Un sito web costituito da un certo numero di pagine HTML è un sito **statico**
  - ▶ Il contenuto di un documento HTML viene determinato dall'autore del documento stesso e non cambia mai
  - ▶ HTML non consente di descrivere pagine che si modificano o si aggiornano automaticamente in risposta ad eventi “esterni” (ad esempio: inserimento di dati nei form da parte degli utenti)
- HTML infatti è un linguaggio di **mark-up**, non un linguaggio di **programmazione**
  - ▶ Consente di descrivere la struttura di una pagina web
  - ▶ Non consente di scrivere programmi che ricevano dei dati e li elaborino
- Per poter realizzare siti web più evoluti servono quindi linguaggi più evoluti che incorporino aspetti di programmazione

## Introduzione (2)

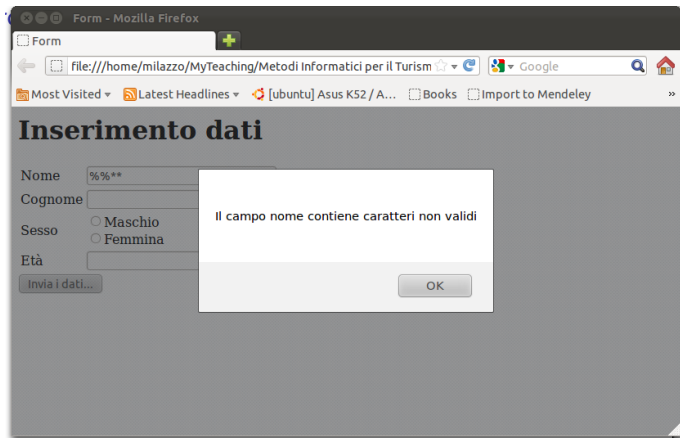
- Esistono numerosi linguaggi e numerose tecnologie per realizzare siti web evoluti
- Si differenziano principalmente in due categorie:
  - ▶ Linguaggi (e tecnologie) **client side**
  - ▶ Linguaggi (e tecnologie) **server side**
- I linguaggi client side vengono utilizzati all'interno dei documenti HTML e vengono eseguiti nel computer dell'utente che visita il sito
- I linguaggi server side si utilizzano per scrivere programmi che verranno eseguiti sul server web e che si occupano di generare "al volo" documenti HTML

# I linguaggi (e le tecnologie) client side (1)

- Vantaggi dei linguaggi client side:
  - ▶ Consentono una elevata interattività con l'utente (apertura di finestre "pop-up", risposta a eventi legati ai movimenti del mouse, ecc...)
  - ▶ Responsività: essendo eseguiti sul computer dell'utente (e non sul server) forniscono tempi di risposta molto rapidi in quanto durante le interazioni con l'utente non ci sono scambi di dati tra client e server
  - ▶ Non sovraccaricano di lavoro il server
- Svantaggi dei linguaggi client side:
  - ▶ Sono un po' limitati nelle funzionalità: ad esempio, non possono accedere a eventuali database presenti sul server

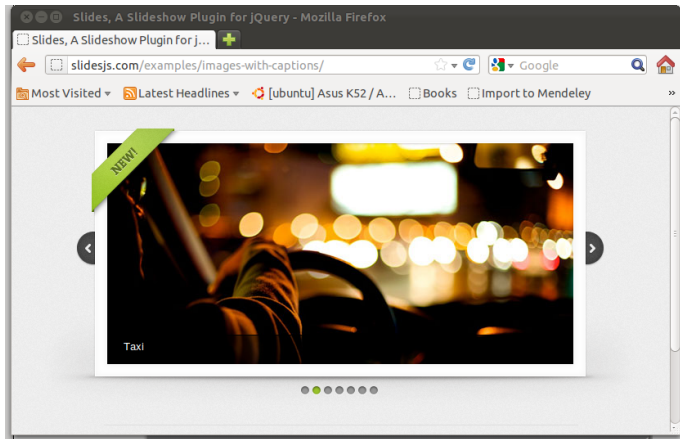
# I linguaggi (e le tecnologie) client side (2)

- Esempi tipici di applicazione dei linguaggi client side:
  - ▶ Controllo dei dati immessi in un form prima dell'invio al server



# I linguaggi (e le tecnologie) client side (3)

- Esempi tipici di applicazione dei linguaggi client side:
  - ▶ Visualizzazione di immagini a rotazione (image slideshow)



# I linguaggi (e le tecnologie) client side (4)

- Esempi tipici di applicazione dei linguaggi client side:
  - ▶ Calendario



July 2007						
Mo	Tu	We	Th	Fr	Sa	Su
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

# I linguaggi (e le tecnologie) client side (5)

- Tra i linguaggi **client side** più utilizzati abbiamo:
  - ▶ **Javascript**: è un linguaggio di programmazione che consente di eseguire semplici elaborazioni di dati all'interno di un browser. Funziona da tempo con tutti i principali browser senza bisogno di applicazioni aggiuntive (plug-in)
  - ▶ **Flash**: è una tecnologia per la realizzazione di siti web animati che richiede l'installazione di uno specifico esecutore (plug-in) nel computer dell'utente
  - ▶ **Applet Java**: è una tecnologia basata sul linguaggio di programmazione Java per la realizzazione di programmi anche molto complessi da eseguire nel computer dell'utente. Richiede l'installazione di una Java Virtual Machine nel computer dell'utente



# Javascript (1)

- Tra il linguaggi client side vediamo in particolare Javascript
- Javascript è un linguaggio di **scripting**, ossia si usa per realizzare degli **script**
- Uno script è un programma solitamente piuttosto piccolo e semplice che non richiede di eseguire calcoli sofisticati
- Javascript è **IL** linguaggio di scripting per il web
  - ▶ Al giorno d'oggi la stragrande maggioranza delle pagine web contiene codice Javascript
- Il codice Javascript viene inserito all'interno del documento HTML utilizzando il tag `<script>`
- Il browser riconosce il codice Javascript e lo esegue
- Il tag `<script>` consente anche di specificare il nome di un file separato che contiene il codice Javascript da eseguire

# JavaScript (2)

- Un semplice esempio di documento HTML+JavaScript:

```
<html>
  <head>
    <title>La mia prima applicazione JavaScript</title>
  </head>
  <body>
    <script>
      alert("Hello, world!");
    </script>
  </body>
</html>
```

- Questo script apre un alertbox (una finestrella) con scritto Hello, world!
  - ▶ JavaScript consente a un documento HTML di interagire con l'utente!

## Javascript (3)

- E' anche possibile associare l'esecuzione di una porzione di codice JavaScript all'avvenimento di un particolare evento
  - ▶ Pressione di un bottone
  - ▶ Movimento del mouse su un elemento del documento HTML
  - ▶ Modifica dei campi di un form
  - ▶ ....
- Esiste un set di eventi che possono essere associati a qualunque elemento del documento HTML tramite l'uso di attributi.
  - ▶ Ad esempio: `<input type="button" onclick="....codice JavaScript....">`
  - ▶ Esegue il codice JavaScript quando il bottone viene cliccato (ossia premuto)

## Javascript (4)

Il set di eventi (di solito) include:

- `onclick` Click sull'elemento
- `ondblclick` Doppio click sull'elemento
- `onmousedown` Il tasto (sinistro) del mouse viene premuto
- `onmouseup` Il tasto (sinistro) del mouse viene rilasciato
- `onmouseover` Il mouse si trova sopra l'elemento
- `onmouseout` Il mouse si sposta da sopra l'elemento
- `onkeypress` Pressione di un tasto della tastiera mentre il focus è sull'elemento
- `onkeydown` Un tasto della tastiera è premuto mentre il focus è sull'elemento
- `onkeyup` Un tasto della tastiera è rilasciato mentre il focus è sull'elemento
- `onfocus` Si pone il focus sull'elemento
- `onblur` Si sposta il focus su un altro elemento
- `onchange` L'elemento viene modificato
- `onsubmit` Si preme il bottone "submit" (per i form)
- `onreset` Si preme il bottone "reset" (per i form)
- `onselect` L'utente seleziona il testo contenuto nell'elemento
- `onload` Al caricamento della pagina/dell'elemento (e.g. body/immagini)
- `onunload` Alla chiusura della pagina
- `onresize` La finestra del browser viene ridimensionata

## Javascript (5)

Un altro Hello World che usa gli eventi:

```
<html>
  <head>
    <title>La mia prima applicazione JavaScript</title>
  </head>
  <body>
    <input type="button" value="Clicca qui"
      onclick="alert('Hello, world!')">
  </body>
</html>
```

- Quando si clicca sul bottone “Clicca qui” si apre un alertbox con scritto Hello, world!

## Javascript (6)

Con l'approccio a eventi si può facilmente realizzare un *rollover* (un'immagine che cambia quando le si passa sopra il mouse):

```

```

- L'immagine visualizzata inizialmente è logo.png
- Quando il mouse si posiziona sull'immagine logo.png viene sostituita da logo2.png
- Quando il mouse si sposta da sopra l'immagine viene di nuovo visualizzata l'immagine iniziale

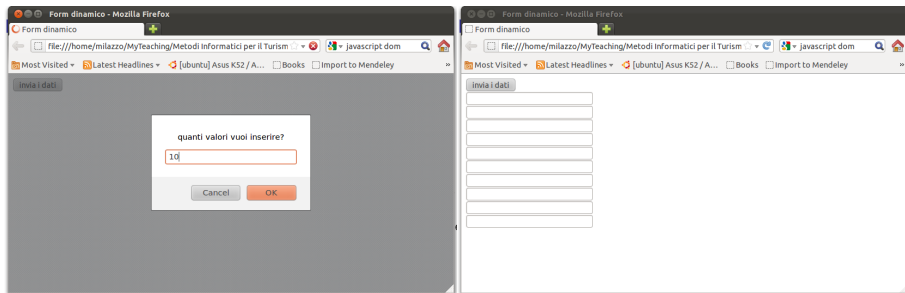
## Javascript (7)

- Javascript può essere usato anche per creare dinamicamente il contenuto della pagina web
- Ad esempio, il seguente documento HTML usa Javascript per creare un form con un numero di caselle di testo scelto dall'utente

```
<html>
  <head><title>Form dinamico</title></head>
  <body>
    <form id="myform">
      <input type="submit" value="invia i dati"><br>
    </form>
    <script>
      var n = parseInt(prompt("quanti valori vuoi inserire?"));
      var f = document.getElementById("myform");
      for ( var i=0; i<n ; i++) {
        var nuovo = document.createElement("input");
        nuovo.setAttribute("type","text");
        nuovo.setAttribute("name","v"+i);
        f.appendChild(nuovo);
        f.appendChild(document.createElement("br"));
      }
    </script>
  </body>
</html>
```

# Javascript (8)

- Ecco l'esempio eseguito nel browser:





## Javascript (9)

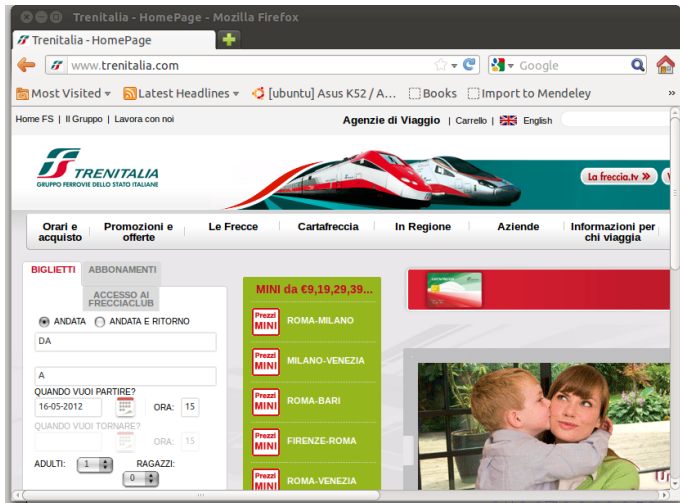
- Javascript può essere usato quindi come linguaggio per la creazione pagine web animate
- In questo contesto Javascript può essere usato anche per modificare dinamicamente aspetti di presentazione trattati con CSS
- Questo modo di utilizzare il “pacchetto” HTML+Javascript+CSS viene spesso denominato **Dynamic HTML**

# I linguaggi (e le tecnologie) server side (1)

- Vantaggi dei linguaggi server side:
  - ▶ Consentono di realizzare **applicazioni web** anche molto sofisticate che accedono a database, interagiscono con altri programmi e mettono in comunicazione utenti diversi del sito
  - ▶ Sono più sicuri: i programmi non “viaggiano” sulla rete ma rimangono all'interno del server
- Svantaggi dei linguaggi server side:
  - ▶ Minore responsività: ad ogni interazione con l'utente (click su un bottone, ecc...) segue una comunicazione tra client e server
  - ▶ I programmi possono sovraccaricare il server

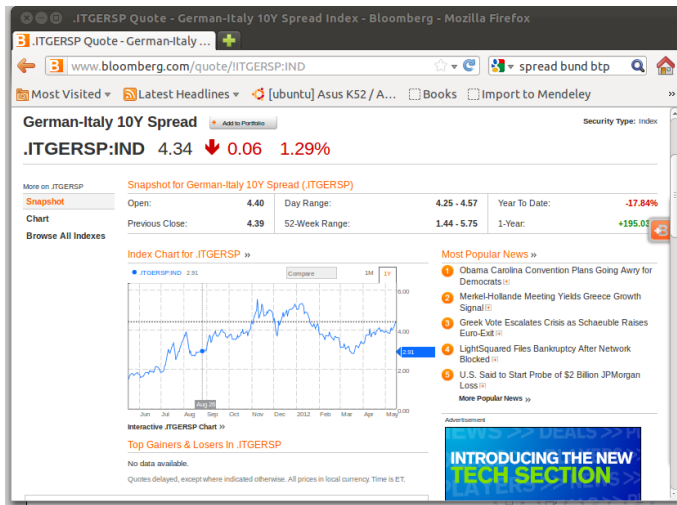
# I linguaggi (e le tecnologie) server side (2)

- Esempi tipici di applicazione dei linguaggi server side
  - ▶ Servizi di prenotazione online



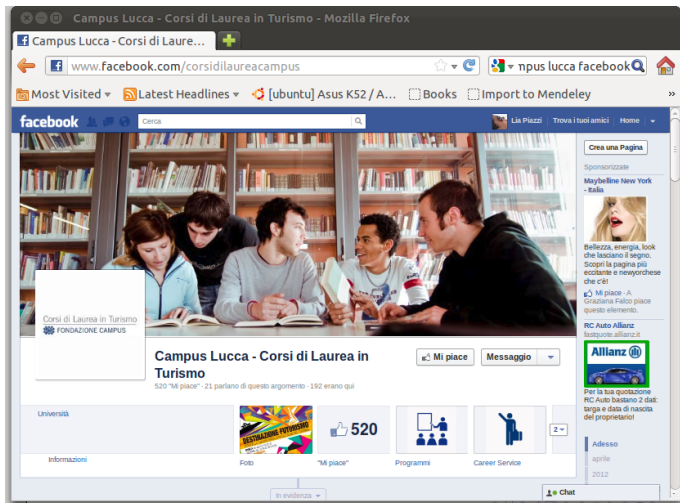
# I linguaggi (e le tecnologie) server side (3)

- Esempi tipici di applicazione dei linguaggi server side
  - ▶ Siti su finanza aggiornati in tempo reale, home banking e trading on line



# I linguaggi (e le tecnologie) server side (4)

- Esempi tipici di applicazione dei linguaggi server side
  - ▶ Social network



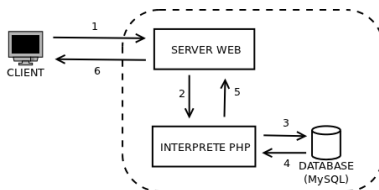
# I linguaggi (e le tecnologie) server side (5)

- Tra i linguaggi **server side** più utilizzati abbiamo:
  - ▶ PHP
  - ▶ ASP
  - ▶ Java Server Pages
  - ▶ Ruby on Rails
  - ▶ ....
- sono linguaggi di programmazione che consentono di scrivere programmi che generano documenti HTML elaborando dati ricevuti dall'utente, estraendo dati da un database e molto altro

# PHP (1)

- Vediamo in particolare il linguaggio PHP
- PHP viene eseguito (tramite l'utilizzo di un programma detto “interprete”) da un server web
- Il codice PHP viene usato per generare dinamicamente i documenti HTML che il client deve ricevere e visualizzare nel browser
  - ▶ In un sito web statico (non-PHP) il documento HTML viene preso dal file system e inviato direttamente al client
  - ▶ In un sito web dinamico (tipo PHP) i file del web server sono passati all'interprete PHP che genera il documento HTML da inviare al client
- Quindi: l'output di un “programma” PHP è un documento HTML
- Il codice PHP può fare quello che gli pare per creare il documento HTML
  - ▶ Ad esempio: accedere a un database SQL, consultare altri siti web, ecc....

## PHP (2)



- Come viene eseguito una applicazione web PHP:

- ➊ Il client richiede al server web una pagina PHP, ad es:  
`http://www.prova.it/index.php`
- ➋ Il server web vede che si tratta di una pagina PHP e inoltra la richiesta all'interprete PHP (spesso eseguito sullo stesso computer)
- ➌ L'interprete PHP esegue il programma contenuto nella pagina PHP possibilmente facendo delle richieste a un database
- ➍ Il database risponde fornendo i dati richiesti
- ➎ L'interprete crea un nuovo documento HTML tenendo conto anche dei dati estratti dal database e lo restituisce al server web
- ➏ Il server web gira il documento HTML al client in risposta alla richiesta iniziale



# PHP (3)

- I file PHP sono strutturati come un documento HTML
  - ▶ ma contengono sezioni di codice PHP delimitate da `<?php` e `?>`
- L'interprete PHP effettua il parsing del file (lo scandisce) e sostituisce le sezioni di codice PHP con il codice HTML risultante dalla sua esecuzione
- Tutto ciò che è al di fuori del tag `<?php . . . . ?>` viene ignorato dall'interprete e dato in output così com'è
- Solitamente l'estensione usata per i file PHP è `.php`
- Per “eseguire” lo script bisogna necessariamente passare tramite un web-server:
  - ▶ non è sufficiente aprire il file con il browser
  - ▶ per utilizzare PHP sul proprio computer bisogna avere installato un server HTTP e un interprete PHP

## PHP (4)

Un semplice esempio di file PHP (il solito Hello World):

```
<html>
<head><title>Hello World in PHP</title></head>
<body>

<?php
    print "<p>Hello, World!</p>";
    print "<p>Questo e' un frammento di codice PHP.
        Nota che fa parte di un documento HTML standard!</p>";
?>

</body>
</html>
```

Il corrispondente HTML generato dall'interprete PHP e inviato al client:

```
<html>
<head><title>Hello World in PHP</title></head>
<body>

<p>Hello, World!</p>
<p>Questo e' un frammento di codice PHP.
Nota che fa parte di un documento HTML standard!</p>

</body>
</html>
```

# PHP (5)

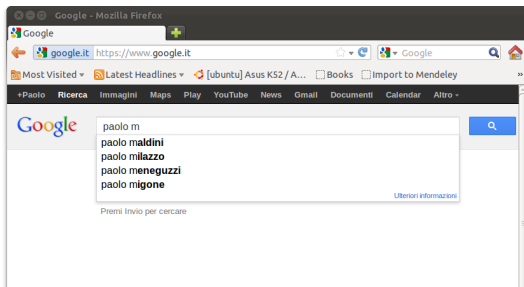
- E' importante sottolineare che l'output di PHP è un documento HTML
- Il client riceverà tale output senza vedere il codice PHP che lo ha originato
- In generale il client non percepisce la differenza tra un sito web statico e un sito web dinamico
  - ▶ in entrambi i casi il client riceve un documento HTML e lo visualizza nel browser

# Combinazioni client side/server side (1)

- Spesso si fa un uso combinato dei linguaggi e delle tecnologie client side e server side
- Ad esempio, è molto comune che un documento HTML generato da una applicazione PHP contenga del codice Javascript da eseguire nel browser dell'utente
  - ▶ Questo consente di sfruttare i vantaggi di entrambi gli approcci
  - ▶ Esempio tipico sono i form: spesso le pagine web che contengono dei form vengono generate da applicazioni server side che includono nella pagina il codice Javascript necessario per effettuare controlli sulla correttezza dei dati immessi dall'utente nel form

## Combinazioni client side/server side (2)

- Un modo sofisticato di far dialogare il codice Javascript con le applicazioni server side è tramite la tecnologia **AJAX**
- AJAX è una tecnologia che consente al codice Javascript di effettuare delle richieste a una applicazione server side
  - ▶ Consente di realizzare pagine che si aggiornano o si modificano in piccole parti **senza dover ricaricare tutta la pagina**
- L'applicazione che ha portato al successo la tecnologia AJAX sono i suggerimenti per le ricerche su google
  - ▶ Man mano che si inseriscono parole nella casella di ricerca di google vengono suggerite altre parole per rendere la ricerca più specifica



# I cookies (1)

- I cookies sono uno strumento molto usato nell'ambito della realizzazione di siti web dinamici e di applicazioni web
- Servono per memorizzare informazioni da usare in pagine di uno stesso sito visitate in momenti successivi
  - ▶ Un cookie è un piccolo file di testo che il sito lascia sul computer del client
- Possono essere creati, gestiti e consultati sia utilizzando linguaggi client side che server side
- Tipico esempio di utilizzo:
  - ▶ La funzione “ricordati di me” che vi consente di non inserire la password ogni volta che vi collegate a un sito che la prevede
  - ▶ In questo caso il sito vi lascia un particolare cookie quando inserite la password e la volta successiva che vi collegate non ve la richiede perché desume la vostra identità dal cookie che trova sul vostro computer

# I cookies (2)

- E' bene sapere che:

- ▶ I cookies sono file **temporanei**: devono avere una scadenza (che puo' essere anche di qualche mese) dopo la quale il browser li cancella dal computer automaticamente
- ▶ I cookies possono essere cancellati in qualunque momento dall'utente usando una specifica opzione del browser
- ▶ Il browser garantisce che ogni sito possa avere accesso **solo ai propri cookies**
  - ★ Questo è un aspetto di sicurezza molto importante in quanto i cookies spesso contengono **informazioni riservate** (chiavi di accesso, dati sensibili, ecc..)
  - ★ Un sito può però consentire ad altri siti di visualizzare i propri cookies, tipicamente sulla base di accordi commerciali...