

# Linguaggi

*Corso di Laurea in Ingegneria delle Telecomunicazioni  
A.A. 2010-2011*

Alessandro Longheu

*<http://www.diit.unict.it/users/alongheu>*

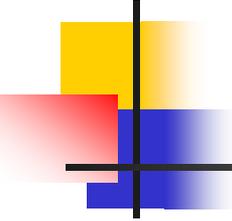
*[alessandro.longheu@diit.unict.it](mailto:alessandro.longheu@diit.unict.it)*

---

*- lezione 11 -*

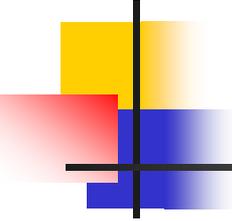
## **Web Programming**

# Web Programming



---

- Definizione di applicazione web
- Schemi architetturali:
  - Thin vs. Thick Web Client
  - Architetture N-tier per applicazioni web
  - Architettura per applicazioni basate su web services
- Tipi di documenti web: statici, dinamici, attivi
- Linguaggi per applicazioni web (lato client e server)
- Stratificazione logica di un'applicazione web

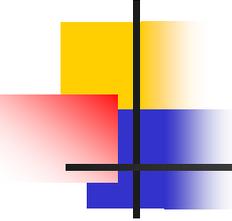


# Web Programming

---

## **Applicazione web:** due punti di vista

- **Estensione di un sito web** che permette ad un utente di invocare un'applicazione e influenzarne lo stato sul server
  - Include: commercio elettronico, webmail, discussion forum
  - Esclude: motori di ricerca, contatore di accessi
- **Estensione di un'applicazione** client-server in cui il browser sostituisce il client da installare come front-end dell'applicazione
  - Nel back-end dell'applicazione è necessario avere un web server



# Web Programming

---

- **Thin Web Client**

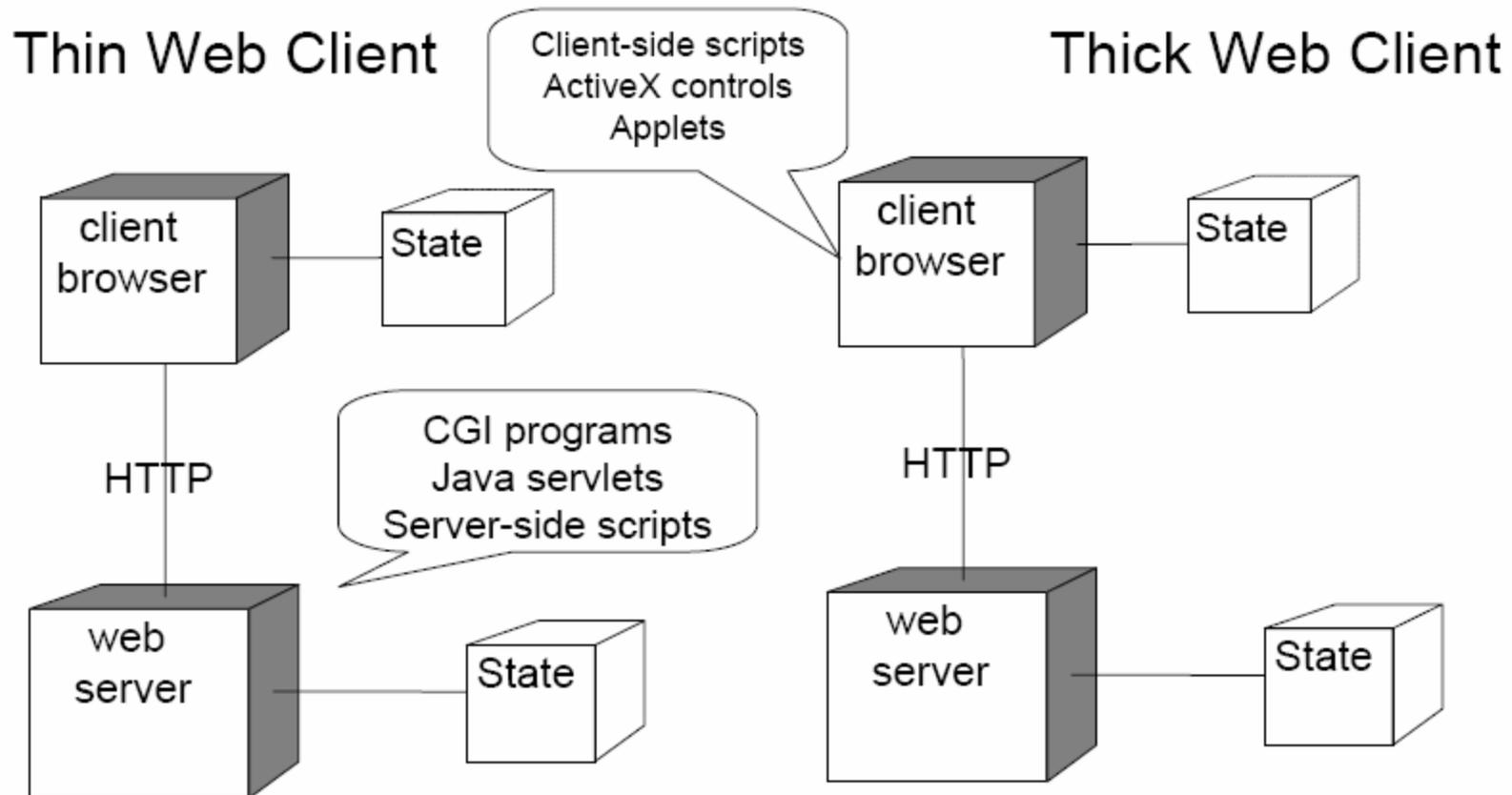
- Tutta la business logic è eseguita dal lato server; configurazione minima richiesta al client, quindi browser che supporta i forms e comunicazione tra client e server solo con HTTP

- **Thick Web Client**

- Uso di interfacce avanzate o parte della business logic eseguita sul lato client; tipicamente il client usa client-side scripting (DHTML), Java applets o controlli ActiveX; la comunicazione tra client e server è anche in questo caso solo tramite HTTP

# Web Programming

## Architetture 2-tier (2 livelli) con thin e thick client:

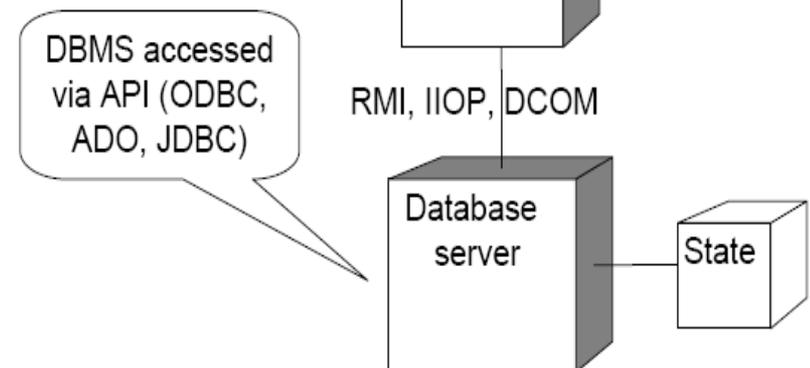


# Web Programming

- I livelli utilizzati in un'architettura client/server in realta' potrebbero non limitarsi a 2 (client e server), ma avere 3 o piu' livelli con compiti differenti
- **architettura 3-tier per applicazioni web:**

## Glossario...

- API=Application Programming Interface=librerie
- IIOP=Internet ORB Protocol (protocollo per programmazione distribuita). ORB (Object Request Broker) è un software che consente di chiamare da un programma una funzione remota (associata ad un altro software su un'altra macchina)
- RMI=Remote Method Invocation è un'API Java per l'invocazione remota di metodi
- DCOM=Distributed COM è una tecnologia microsoft per l'invocazione remota di metodi



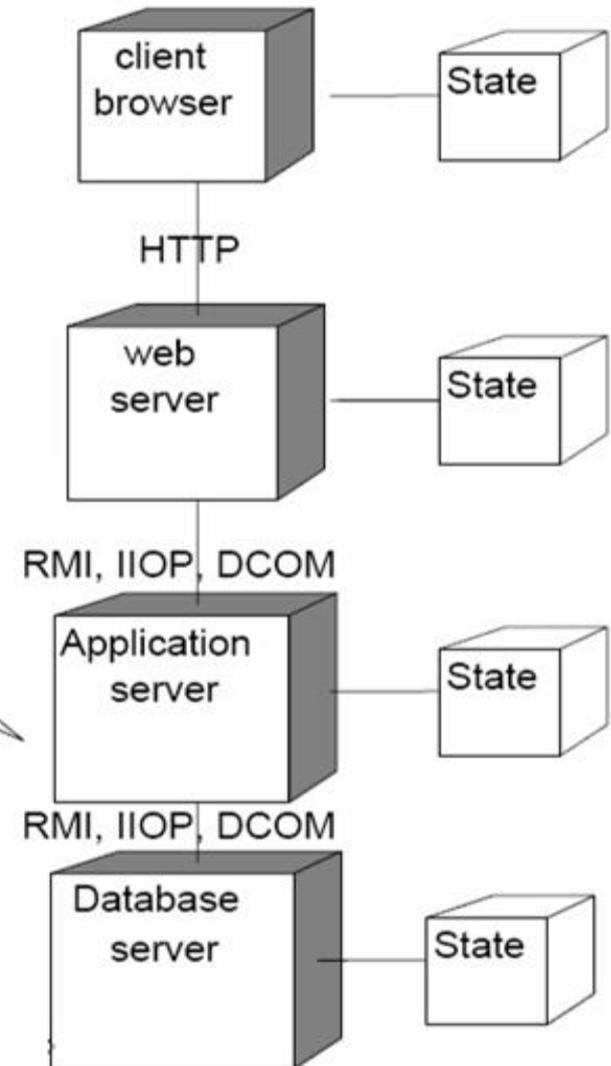
# Web Programming

## Architettura 4-tier per applicazioni web

### Glossario...

- La programmazione distribuita si basa sul fatto che programmi su macchine diverse possono cooperare fra loro per un fine comune. Per cooperare devono parlare la stessa lingua, quindi sono stati sviluppati diversi protocolli più o meno standard per realizzare questa lingua. Se i programmi sono scritti in modo da utilizzare questa lingua comune, si chiamano componenti e assumono nomi diversi:
  - COM Component, se in accordo con Microsoft
  - Enterprise Javabean, se in accordo con SUN
- Una applicazione legacy è un programma datato che continua ad essere usato poiché l'utente (tipicamente un'organizzazione) non vuole o non può rimpiazzarlo (costi). Sono molto frequenti

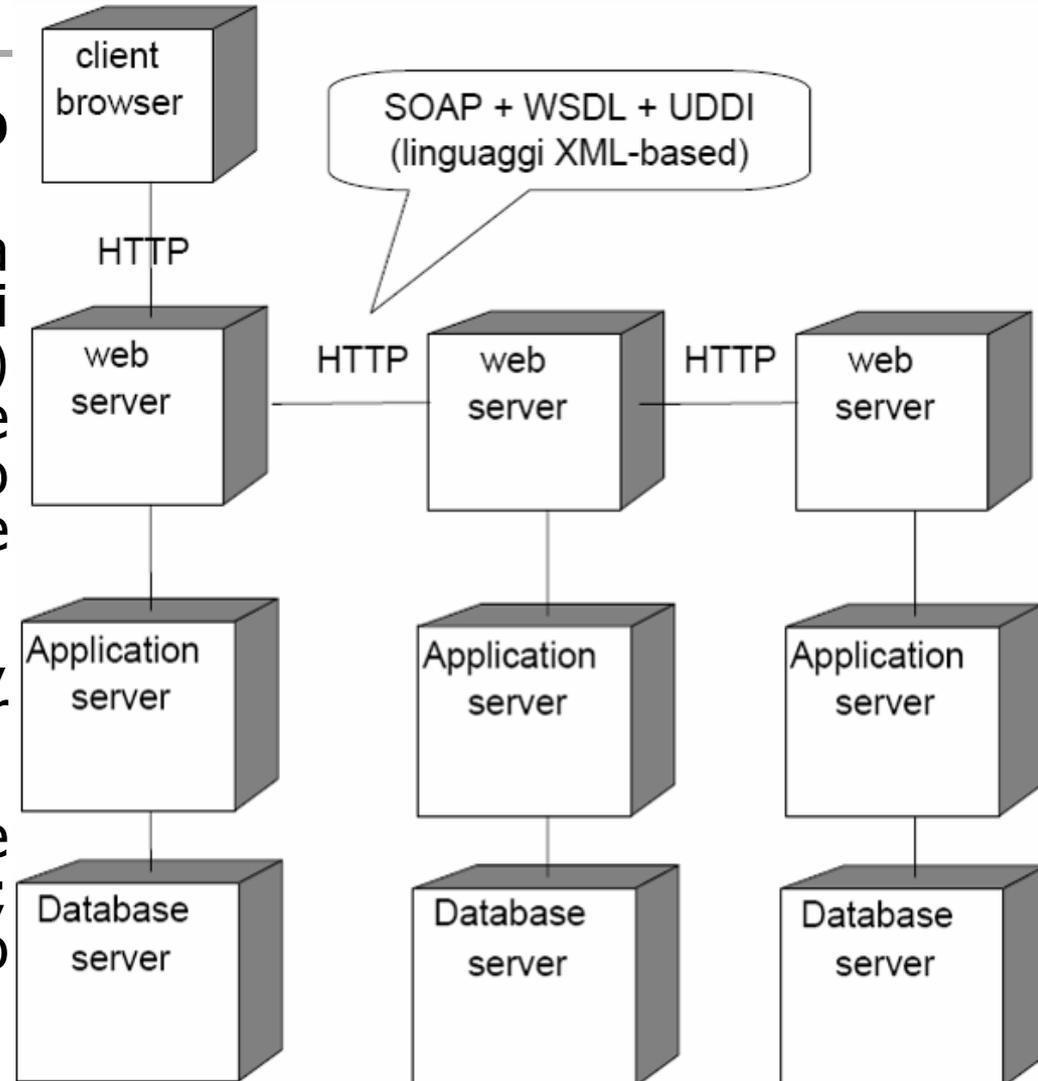
COM comp.  
Ent. JavaBeans  
Legacy apps



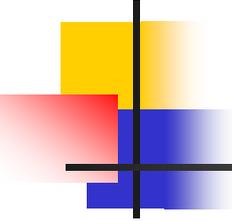
# Web Programming

## Architettura 4-tier per web services.

- Un web service è un sistema software in cui diversi programmi (servizi web) dislocati in diverse macchine (su piu' siti) possono lavorare insieme per un fine comune.
- Oltre i web-services, esistono altre tecnologie per questo scopo, come CORBA.
- Serve una lingua comune per lo scambio dei dati; SOAP, WSDL, UDDI sono esempi



# Web Programming



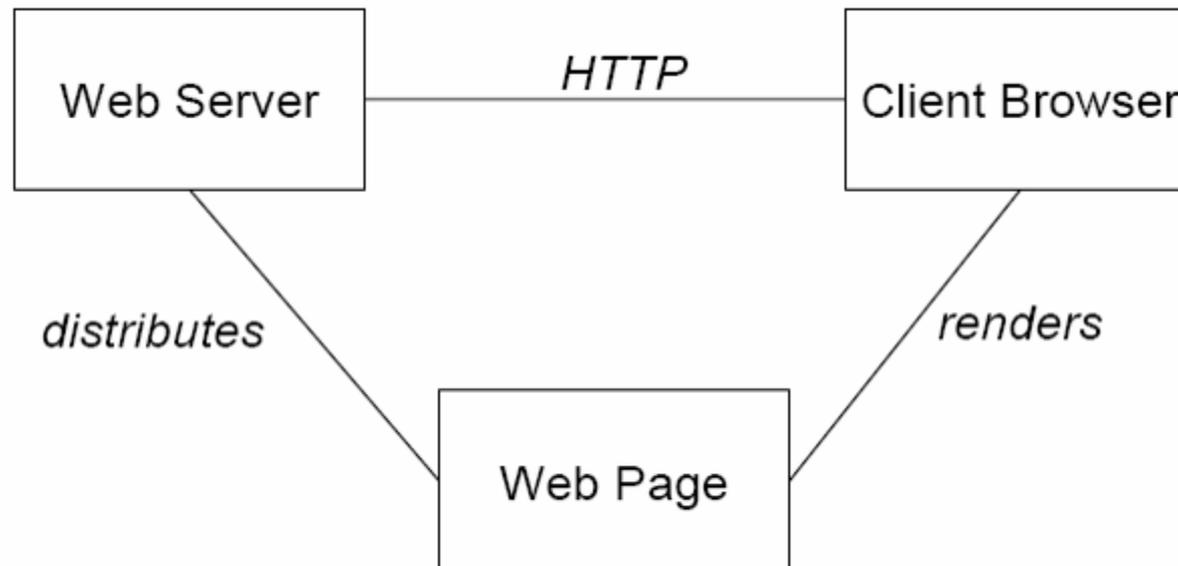
---

## **Documenti statici:**

- Caratteristiche
  - Il documento visualizzato dal browser corrisponde a un file acceduto dal web server
  - Formato HTML
  - Il contenuto e' fisso e determinato dall'autore al momento della creazione
- Vantaggi
  - Facili da creare o clonare
  - Non richiedono skill di programmazione: editor WYSIWYG
- Svantaggi
  - Interazione utente limitata alla funzione di download dei documenti
  - Poco pratici per aggiornamento frequente di informazioni

# Web Programming

- Componenti principali per documenti statici
  - un web server
  - una connessione di rete
  - uno o più client browser



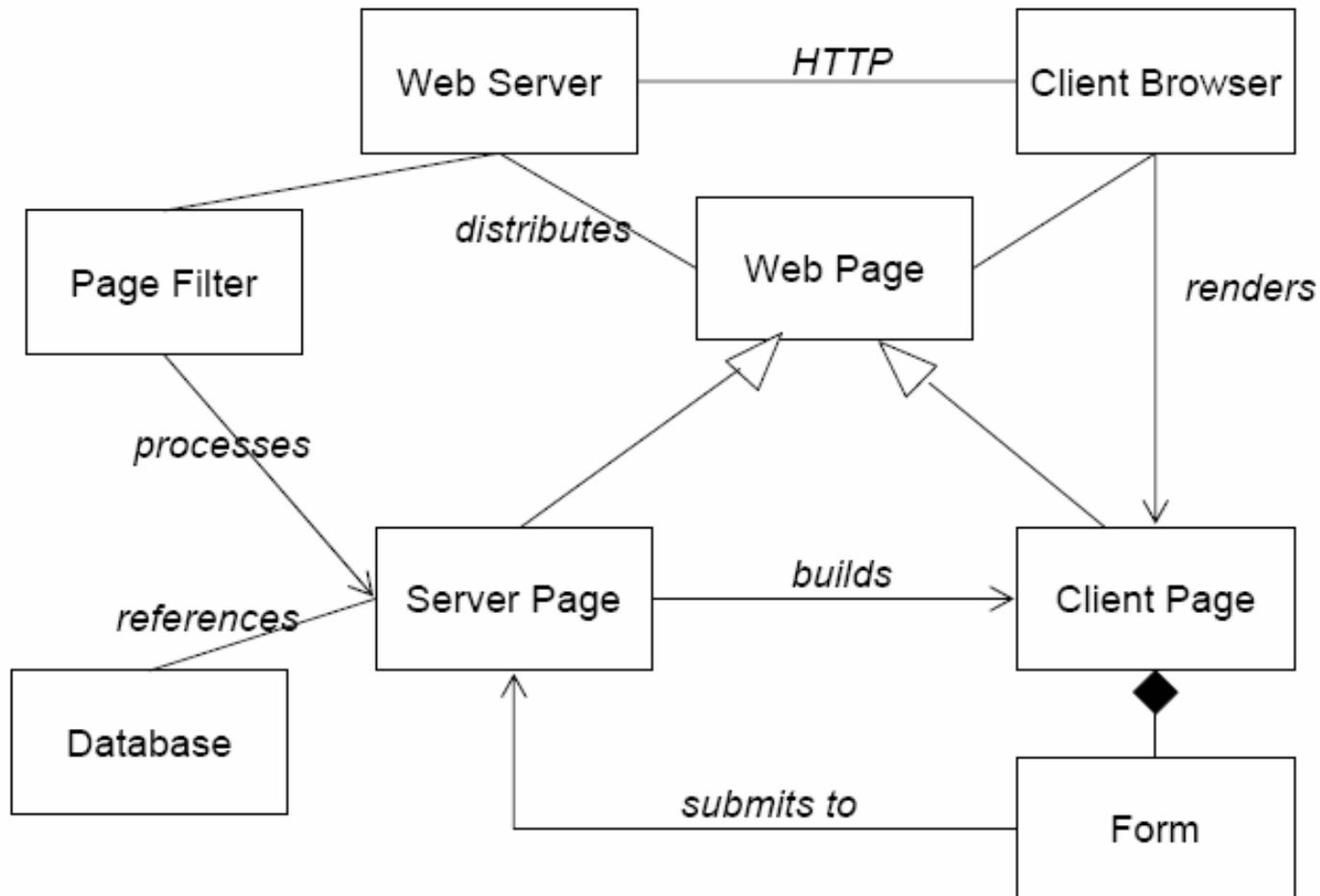
# Web Programming

## Documenti dinamici

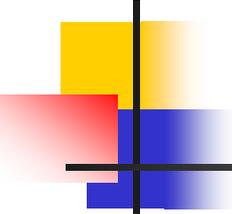
- Caratteristiche
  - Il documento visualizzato dal browser non corrisponde a un file acceduto dal web server
  - Il server filtra le richieste del browser e invoca un programma (compilato o interpretato) il cui output è restituito al browser
  - Il contenuto della risposta può variare tra una richiesta e l'altra
- Vantaggi
  - Generano il contenuto dipendentemente dall'input dell'utente e dallo stato
  - Svantaggi
  - Necessita' di programmare per la creazione automatica del contenuto (qualsiasi linguaggio interpretato o eseguito dal server)
  - L'input può essere validato solo dal server

# Web Programming

## Componenti principali per documenti dinamici



# Web Programming



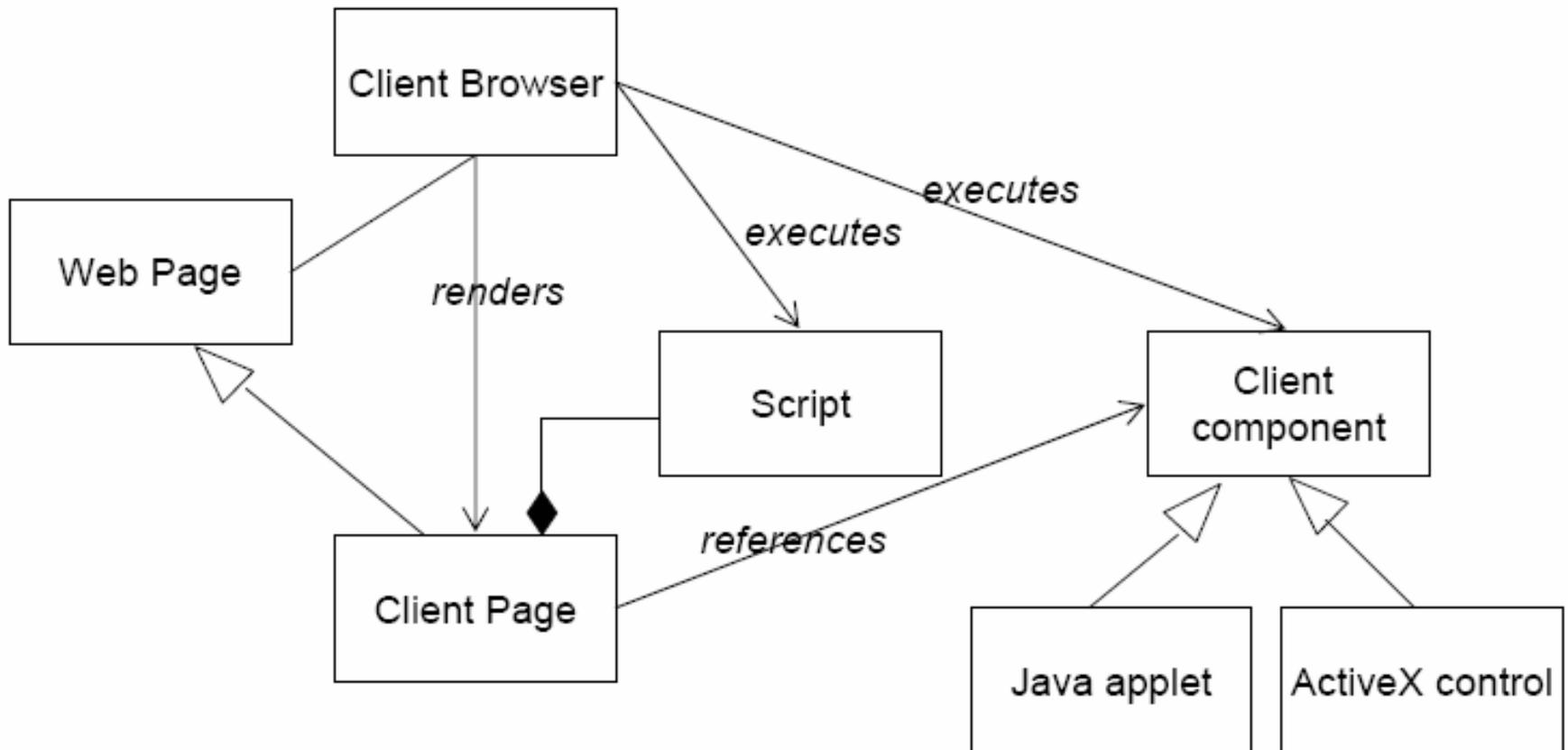
---

## Documenti attivi

- Caratteristiche
  - Il documento visualizzato dal browser contiene un programma (compilato o interpretato) che il browser è in grado di eseguire
  - Client dinamico:
    - Il programma interagisce con l'utente tramite eventi attivati da tastiera e mouse;
    - Il display è aggiornato senza che sia richiesta una connessione al server
    - il programma può leggere/scrivere file locali (non sempre per ragioni di sicurezza) e contattare altri servizi Internet
- Vantaggi
  - Interfacce utente più avanzate (es. toolbar)
  - Maggiori funzionalità senza attendere le risposte dal server
- Svantaggi
  - Browser più complesso
  - Nessuna garanzia che il programma possa essere eseguito su tutti i client

# Web Programming

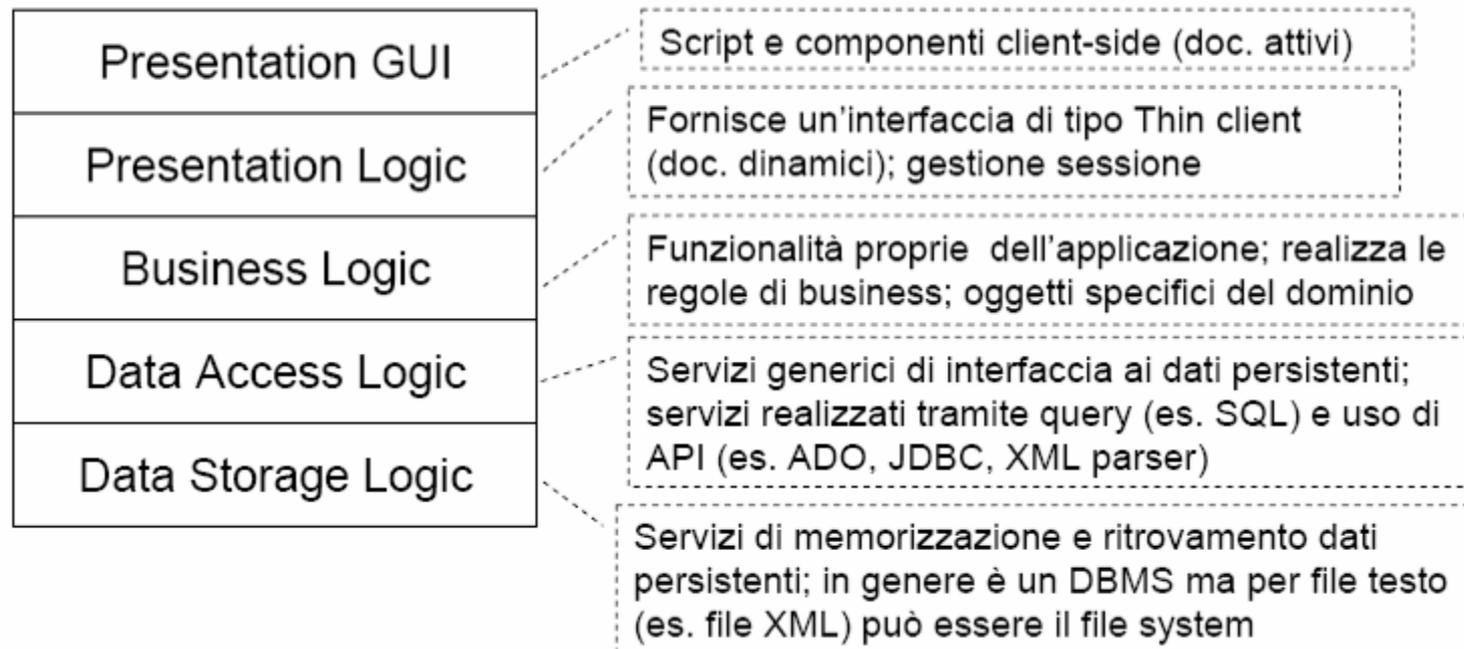
- Componenti principali per documenti attivi

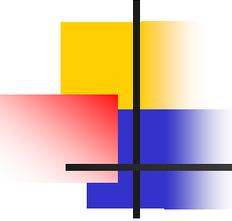


# Web Programming

## Stratificazione logica di un'applicazione web

- Ogni livello offre servizi al livello superiore e nasconde i servizi del livello inferiore (simile al modello ISO/OSI delle reti)
- Non è detto che tutti i livelli siano presenti in quanto necessari
- Le entità di livelli diversi dovrebbero essere distinguibili nel codice sorgente (in file separati o in sezioni separate di un file)





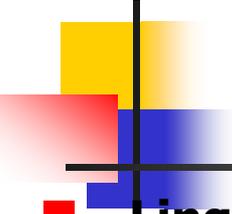
# Web Programming

---

## **Principi di progettazione di siti Web:**

- Progettazione centrata sull'utente e sulle sue esigenze
- Architettura dell'informazione, ossia come organizzare i contenuti di pagine e siti
- Usabilità, ossia semplicità ed efficacia di accesso ai contenuti
- Accessibilità, ossia garantire l'accesso a tutti (disabili, utenti con pc vecchi, connessioni lente, dispositivi portatili ecc.)

# Web Programming



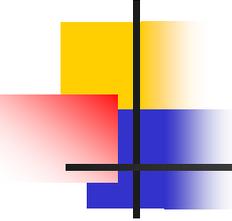
---

## ■ Linguaggi lato client:

- HTML
- CSS
- Javascript, VBscript
- Plug-in, ActiveX, Applet Java
- DHTML, SHTML, XHTML
- XML e RDF

## ■ Linguaggi lato server:

- Codice imperativo con output in HTML
  - Programmi CGI (C, C++, Perl, Python, ...)
  - Programmi FastCGI
  - Java servlets
- Codice HTML con istruzioni imperative
  - Active Server Pages (ASP)
  - Cold Fusion
  - PHP
  - Java Server Pages (JSP)
- ASP.NET con code-behind



# Web Programming

---

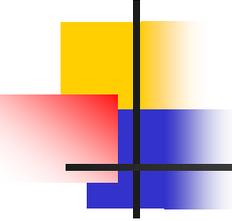
## 2. Linguaggi di programmazione Client-side:

### ■ HTML

- L'HTML un linguaggio di formattazione di documenti, non necessariamente da visionare su Internet
- Un documento HTML è un file di testo (file ASCII) contenente dei comandi per formattazione del contenuto, inserimento parti multimediali, link ipertestuali ecc. Un file HTML viene interpretato dal browser che effettua il rendering della pagina

### ■ CSS

- CSS è una raccomandazione del World Wide Web Consortium (W3C), con versioni CSS1 (1996), CSS2 (1998), e CSS3 (2001).
- I fogli di stile forniscono un mezzo per separare l'aspetto grafico di una pagina WEB dal contenuto della stessa, ponendo la prima nei fogli di stile, e lasciando il secondo entro il file HTML.



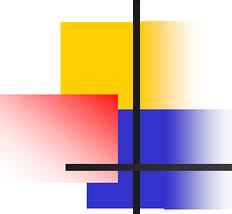
# Web Programming

---

## 2. Linguaggi di programmazione Client-side...

### ■ JavaScript

- Javascript è un linguaggio di programmazione, ideato da Netscape (non SUN); non ha quasi niente in comune con Java
- Javascript è un linguaggio interpretato, object-based piuttosto che object-oriented, nel senso che non permette una completa definizione di classi, ereditarietà ecc., piuttosto opera sull'insieme limitato di oggetti predefiniti che rappresentano le diverse componenti della pagina (DOM)
- **Vbscript** è un linguaggio simile a Javascript proposto dalla Microsoft (VB sta per Visual Basic)
- **plug-in**: programmi aggiuntivi sviluppati da terze parti che ampliano le funzionalità del browser, permettendo di vedere le pagine web che li richiedano: Macromedia flash (file .swf), Apple Quicktime (file .mov)...
- **ActiveX** è il nome di una tecnologia sviluppata dalla Microsoft che estende le funzioni di un'applicazione. I controlli ActiveX forniti con Internet Explorer vengono installati ed eseguiti all'apertura di una pagina web che li contiene. L'utilità di tali controlli è simile a quella dei plug-in, ossia possibilità di visualizzare pagine con contenuti particolari, ad esempio con un grafico o un foglio elettronico incorporati nel codice HTML.

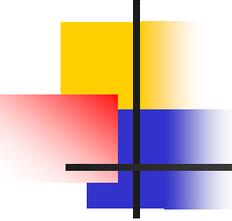


# Web Programming

---

## 2. Linguaggi di programmazione Client-side...

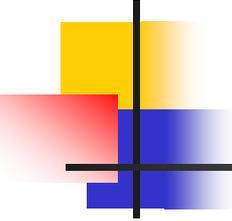
- **Applet Java**, applicazioni scritte in linguaggio Java che possono anche essere molto complesse
- **DHTML, SHTML, XHTML** sono tutte varianti dell'HTML che introducono funzioni specifiche (SHTML) e/o propongono un approccio diverso da quello dell'HTML puro
- **XML e RDF** sono i linguaggi di base del cosiddetto "Semantic Web";
  - XML permette la descrizione della struttura di un documento usando i tag, come in HTML, ma a differenza di questo i tag descrivono la struttura (non l'aspetto) e sono personalizzabili.
  - RDF è un linguaggio per la descrizione di risorse (una pagina web o una sua parte), che utilizza XML e che serve per potenziare l'efficacia dei motori di ricerca



# DHTML

---

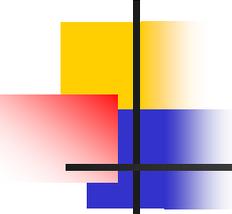
- DHTML=DOM+Javascript+CSS
- DOM = Document Object Model (modello ad oggetti della pagina WEB)
- Javascript (EcmaScript) linguaggio di programmazione per la scrittura degli script
- CSS = Cascading Style Sheet (fogli di stile a cascata)



# DHTML - DOM

---

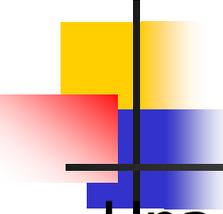
- Il DOM definisce gli oggetti di una pagina WEB
- Ogni oggetto ha delle proprietà che lo descrivono.
- Ogni oggetto ha dei metodi che permettono di inviare dei messaggi allo stesso per comandargli di fare qualcosa
- notazione
- standardizzazione



# DHTML - DOM

---

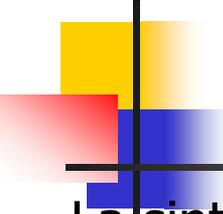
- Window (parent, top, self, opener)
  - frames[i]
    - document
      - images[i]
      - links[i]
      - forms[i]
        - elements[i]
        - button
        - checkbox
        - image
        - radio
        - reset
        - select
          - options[i]
        - submit
        - text/textarea
    - location



# SHTML

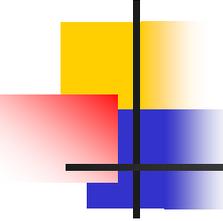
---

- Una pagina in **SHTML** (estensione .shtml) è una pagina Html arricchita da **direttive SSI**, offerte dal Web Server Apache
- Le direttive sono poste nelle pagine HTML, e valutate dal server mentre le pagine sono inviate verso il client. Questo permette di aggiungere ad una pagina HTML del **contenuto generato dinamicamente, senza dovere servire l'intera pagina** attraverso un programma CGI.
- La decisione su quando è opportuno utilizzare le direttive SSI dipende solitamente da quanto contenuto dinamico la pagina contiene, e se ha necessita' di essere rivalutato ogni volta che la pagina viene servita. SSI è il modo migliore per aggiungere piccole porzioni di informazione, ad esempio l'ora corrente. Se, invece, la quasi totalità della pagina fosse generata dinamicamente, sarebbe opportuno ricorrere al CGI (ad esempio, invocando uno script php)



# SHTML

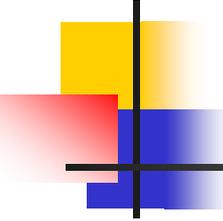
- La sintassi delle direttive SSI è la seguente:
  - <!--#element attribute=value attribute=value ... -->*
- La formattazione è identica a quella di un commento HTML, e non a caso, in quanto se le SSI non fossero correttamente abilitate, il browser le potrà ignorare. Se invece le direttive funzionano, il commento HTML che rappresenta la loro invocazione verrà sostituito dal risultato corrispondente
- Qualche esempio:
  1. *<!--#echo var="DATE\_LOCAL" -->*
  2. *<!--#config timefmt="%A %B %d, %Y" --> Today is <!--#echo var="DATE\_LOCAL" -->*
  3. *This document last modified <!--#flastmod file="index.html" -->*
  4. *<!--#include virtual="/cgi-bin/counter.pl" -->*
  5. *<!--#set var="modified" value="\$LAST\_MODIFIED" -->*
  6. *<!--#if expr="{Mac} && {InternetExplorer}" --> Apologetic text goes here <!--#else --> Cool JavaScript code <!--#endif -->*
- SSI non è un sostituto per il CGI, ma offre la possibilità di aggiungere facilmente contenuti dinamici alla pagina senza grosso sforzo



# XHTML

---

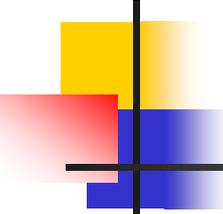
- L'**XHTML** (acronimo di **eXtensible HyperText Markup Language**) è un linguaggio di markup che associa alcune proprietà dell'XML con le caratteristiche dell'HTML: un file XHTML è un pagina HTML scritta in conformità con lo standard XML.
- Il linguaggio prevede un uso più restrittivo dei tag HTML; solo la struttura della pagina è scritta in XHTML, mentre il layout è imposto dai fogli di stile a cascata (CSS, Cascading Style Sheets).
- L'XHTML è nato ufficialmente il 26 gennaio 2000 come standard W3C, e può essere definito tecnicamente una riformulazione dell'HTML 4.01 come applicazione dell'XML 1.0, una sorta di transizione tra questi due linguaggi.



# XHTML

---

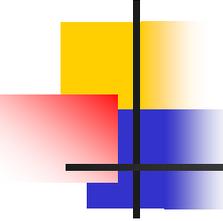
- L'XHTML è il successore diretto e la versione più aggiornata dell'HTML. La necessità di un linguaggio dotato di una sintassi meglio definita rispetto a quella dell'HTML cominciò ad essere avvertita quando si diffuse l'uso di inviare pagine web ai nuovi dispositivi apparsi sul mercato diversi dai tradizionali computer, come ad esempio piccoli apparecchi portatili, dotati di risorse hardware e software non sufficienti ad interpretare il linguaggio HTML. Va tenuto presente che più generica è la sintassi di un linguaggio di programmazione, più difficile risulta realizzare dispositivi in grado di interpretarlo correttamente. Una specifica Document Type Definition (DTD) definisce l'insieme di regole mediante le quali un dato documento può essere renderizzato (cioè *rappresentato correttamente*) dall'XHTML.



# XHTML

---

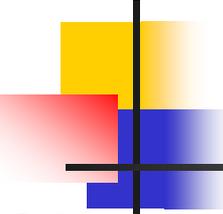
- XHTML è una riformulazione di HTML come linguaggio XML. Quindi, come è logico aspettarsi, ne esistono tre differenti DTD, corrispondenti ad altrettante DTD di HTML 4.01
  - XHTML 1.0 Transitional: nato per favorire la migrazione dal HTML 3.2, o per uso insieme a link e frame in-line
  - XHTML 1.0 Strict: separa il contenuto dal layout, cioè dalla presentazione, che ora viene definita mediante i CSS
  - XHTML 1.0 Frameset: per suddividere la finestra visualizzata dal browser in diversi frame (sottofinestre)
- L'ultima versione di XHTML è la XHTML 1.1: avendo un'unica DTD, la strict, ne viene omessa la dicitura. Non accetta né tag o attributi deprecati né la struttura a frameset. Questa versione inoltre supporta i tag del cosiddetto markup ruby, che consente di visualizzare i set di caratteri delle lingue orientali.



# XHTML

---

- Esistono altre versioni di XHTML, studiate per dispositivi portatili:
- *XHTML Basic*: versione speciale semplificata dell'XHTML, per dispositivi le cui risorse non sono in grado di supportare il set completo di comandi e tag XHTML, come ad esempio i telefoni cellulari. È da considerare come la controparte degli esistenti linguaggi WML e C-HTML.
- *XHTML Mobile Profile*: basato sulla versione *Basic* di XHTML, sviluppato da Nokia per aggiungere all' XHTML Basic funzionalità specifiche dei terminali telefonici.
- Le specifiche della versione XHTML 2.0 sono tuttora in via di definizione, non senza contrasti fra le varie parti interessate, visto che sono sorti vari problemi di compatibilità con le versioni precedenti. Ciò è dovuto al fatto che si tratta a tutti gli effetti di un nuovo linguaggio di markup, piuttosto che di un'evoluzione della precedente versione.



# Web Programming

---

## 3 Linguaggi di programmazione Server-side:

- **PHP** (acronimo ricorsivo di PHP Hypertext Preprocessor) è un linguaggio di scripting open source concepito per la realizzazione di pagine web dinamiche. Attualmente è utilizzato per sviluppare applicazioni web lato server ma può essere usato anche per script a linea di comando o applicazioni GUI standalone. Il suo nome è un acronimo ricorsivo che sta per *PHP: Hypertext Preprocessor* (PHP: preprocessore di ipertesti).
- le **Active Server Pages (ASP)** sono pagine web contenenti, oltre al puro codice HTML, degli scripts server-side per generare *runtime* il codice HTML da inviare al browser (pagine web dinamiche). I linguaggi utilizzati sono VB.NET, C# e J#. Funziona sul web server Microsoft Internet Information Services (IIS), ma esiste un modulo Apache per l'esecuzione di codice ASP. Attualmente si utilizza ASP.NET su .NET framework
- **JavaServer Pages (JSP)** è una tecnologia Java per lo sviluppo di applicazioni Web dinamiche. Si basa su un insieme di tag con cui possono essere invocate funzioni predefinite o codice Java. In aggiunta, permette di creare librerie di nuovi tag che estendono l'insieme dei tag standard. Le librerie di tag JSP si possono considerare estensioni indipendenti dalla piattaforma delle funzionalità di un Web server. Nel contesto di Java, la tecnologia JSP è correlata con i servlet. Alla prima invocazione, le pagine JSP sono tradotte in servlet. Il web server Tomcat supporta le JSP