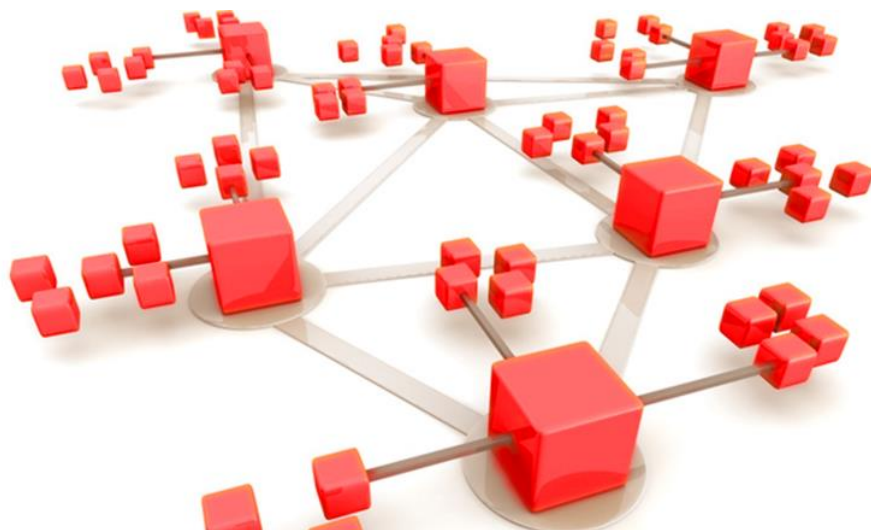


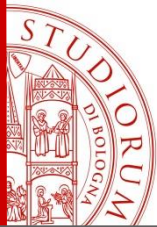


**1 domanda
bonus** **BONUS**

Architettura del Web



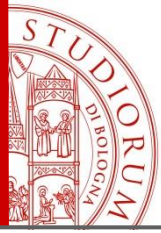
lezione uno



Argomenti

- Architettura del Web
 - Principi architetturali:
 - Tecnologie in evoluzione
 - Modelli per il web dinamico
 - Solution Stack (piattaforme)





Cosa è il web?



Cosa è il Web?

- Proviamo a definirlo attraverso l'esperienza
 - Come lo usiamo
 - Come funziona
 - Cosa fa
 - ...



Tecnologie web

- Collochiamo anche le tecnologie e i tool che vedremo durante il corso:

tecnologie



- HTTP
- HTML
- URI/URL
- CSS
- PHP
- JS
- JSON

build

- Bootstrap
- JQuery
- Atom
- W3School





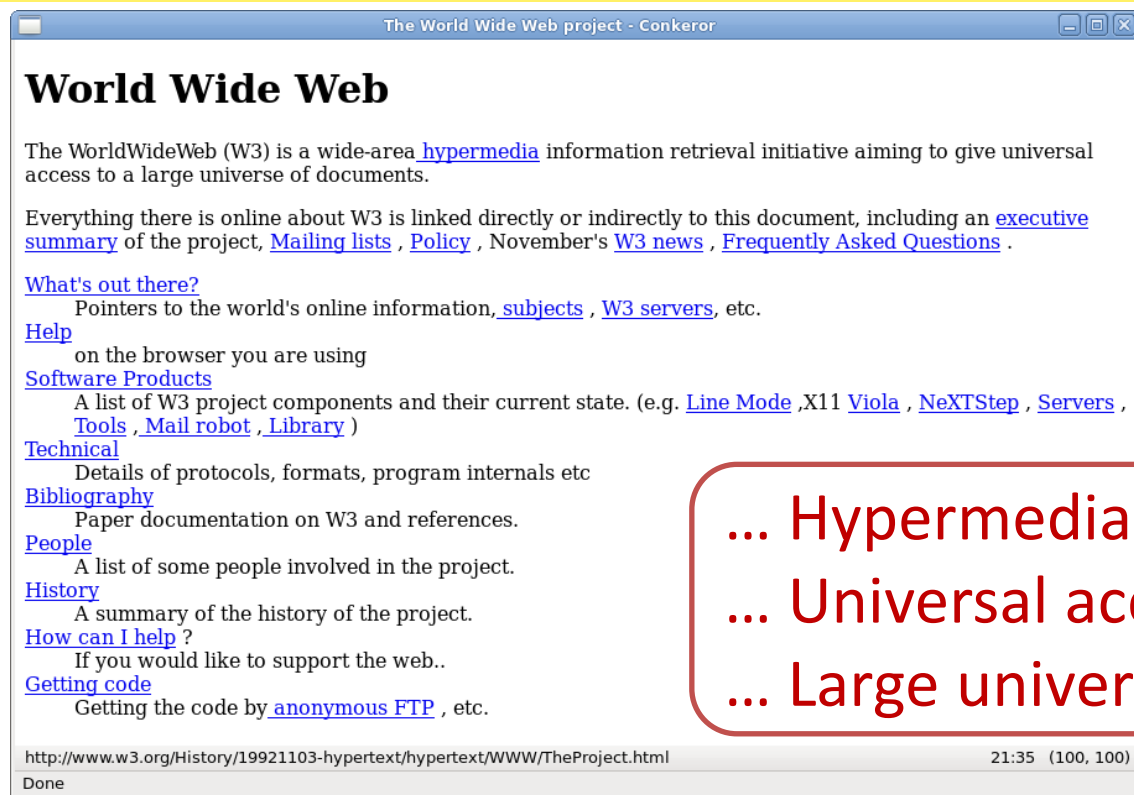
Da Wikipedia

- “*The **World Wide Web** (abbreviated WWW or the Web) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet*”
- Questa definizione viene a sua volta da un documento del consorzio che standardizza il web, il **W3C**
- La prima definizione di WWW era contenuta nella **prima pagina web**, pubblicata il **6 agosto 1991**



La prima pagina...

- Il **6 agosto 1991** è il giorno in cui viene pubblicata la prima pagina ed è considerato come **data di nascita** del WWW
- Il progetto è ovviamente precedente, del **1989**.



... Hypermedia
... Universal access
... Large universe of documents

La nascita del web

- Le 3 tecnologie fondamentali che ancora oggi (ovviamente molto migliorate e integrate da altre) sono alla base del funzionamento del Web erano pronte entro l'ottobre del 1990:



HTML

HyperText Markup
Language

Formato per le
pagine

URI

Uniform Resource
Identifier

Identificazione
delle risorse

HTTP

Hypertext Transfer
Protocol,

Interazione C/S
sopra TCP

- Inoltre furono predisposti:
 - un primo editor/browser, WorldWideWeb.app,
 - ed il primo server Web, httpd.



Principi architetturali del WWW

- **Formato**: la disponibilità di apposite **applicazioni** sul computer ricevente e la corretta identificazione del **formato** dati con cui la risorsa è stata comunicata permette di accedere al suo contenuto in maniera chiara e non ambigua. I formati sono molti, tra cui **HTML**, PNG, RDF, ecc.)
- **Identificazione**: Il WWW è uno spazio informativo (per umani e applicazioni) in cui ogni elemento di interesse è chiamato **risorsa** ed è identificato da un identificatore globale chiamato **URI**.
- **Interazione**: Un **protocollo** di comunicazione chiamato **HTTP** permette di scambiare messaggi su una rete informatica (in particolare TCP/IP)

HTML

HyperText
Markup
Language

Formato per le
pagine

URI

Uniform
Resource
Identifier

Identificazione
delle risorse

HTTP

Hypertext
Transfer
Protocol,

Interazione
C/S sopra TCP



HTML

- **HTML** è un **linguaggio di markup** progettato per dare formato a documenti scientifici con struttura ipertestuale.
 - Il contenuto del documento è inframezzato da elementi di **markup** (chiamati **tag**) che ne definiscono struttura e semantica.
 - I tag sono stringhe contenute tra **<** e **>**.
 - Il linguaggio è evoluto dal 1998 (W3C standard 5.3).

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Story</title>
</head>
<body>
<h1>My Story</h1>
<p>Once upon a time,
...</p>
</body>
</html>
```



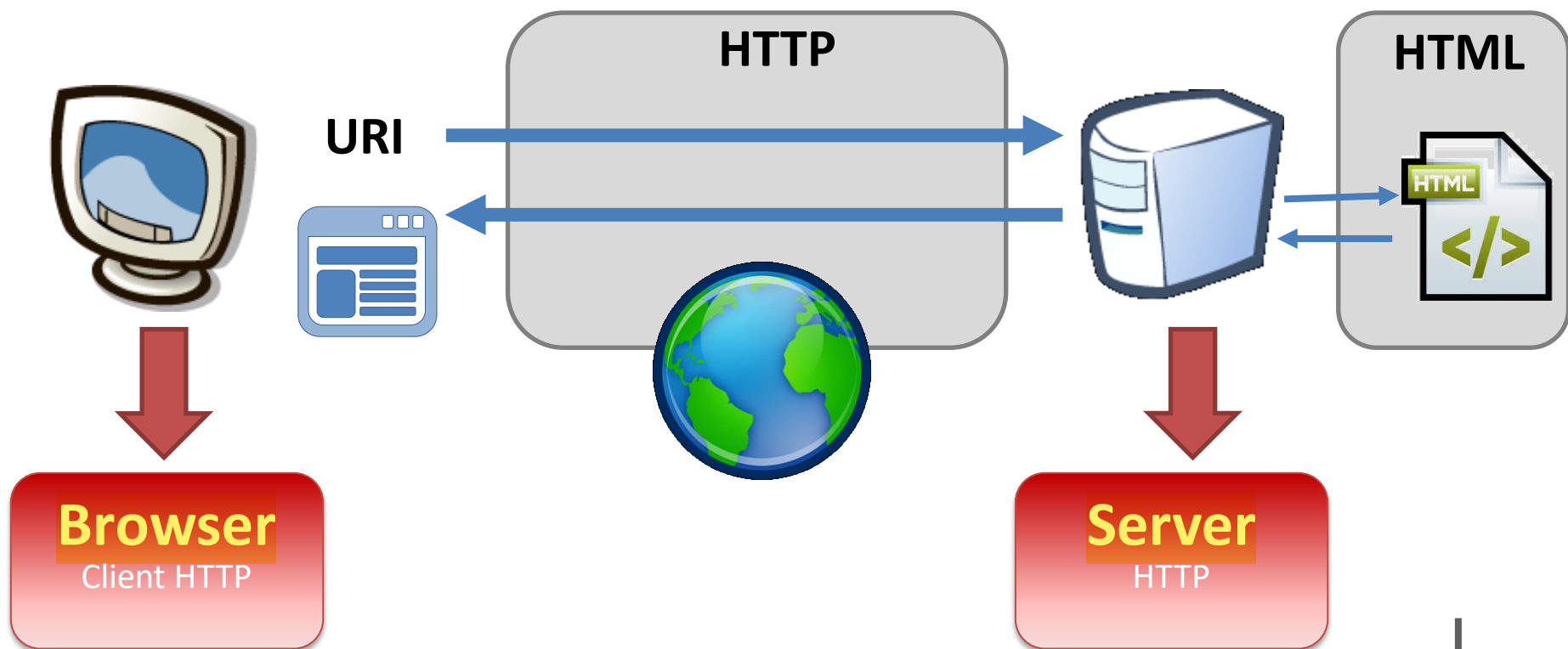
CSS

- Gli aspetti presentazionali della pagina sono gestiti attraverso un linguaggio specifico ne definisce gli **stili**, **CSS (Cascading Style Sheets)**.
 - Separazione presentazione (CSS) e contenuto (HTML)
 - CSS permette di controllare le caratteristiche presentazionali dei documenti HTML.
 - La definizione di CSS1 è del 1996 e il supporto dei browser è iniziato da IE3 (1996) e Netscape Communicator 4 (1997).
 - Ora siamo alla versione 3, quella che vedremo insieme.



Client e Server

- Il web si basa su un protocollo Internet di livello applicazione (**HTTP**) basato sul modello **client** e **server**:



Browser

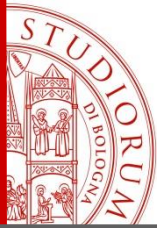
- Il **client** o **browser** è un visualizzatore di documenti ipertestuali e multimediali in HTML:
 - Inizia l'interazione (come client).
 - Può visualizzare testi, immagini e semplici interfacce grafiche.
 - Può agire da editor, ma solo localmente.
 - I browser hanno anche:
 - **plug-in** che permettono di visualizzare ogni tipo di formato speciale,
 - un linguaggio di programmazione interno (**Javascript**).



Server

- Il **server** è una applicazione in grado di rispondere a richieste di risorse locali (file, record di database, ecc.) individuate da un identificatore univoco:
 - La funzione primaria del server web è rispondere alle richieste di pagine effettuate dai client.
 - Il server web può collegarsi ad applicazioni server-side ed agire da tramite tra il browser e l'applicazione in modo che il browser diventi l'interfaccia di una applicazione che gira altrove.

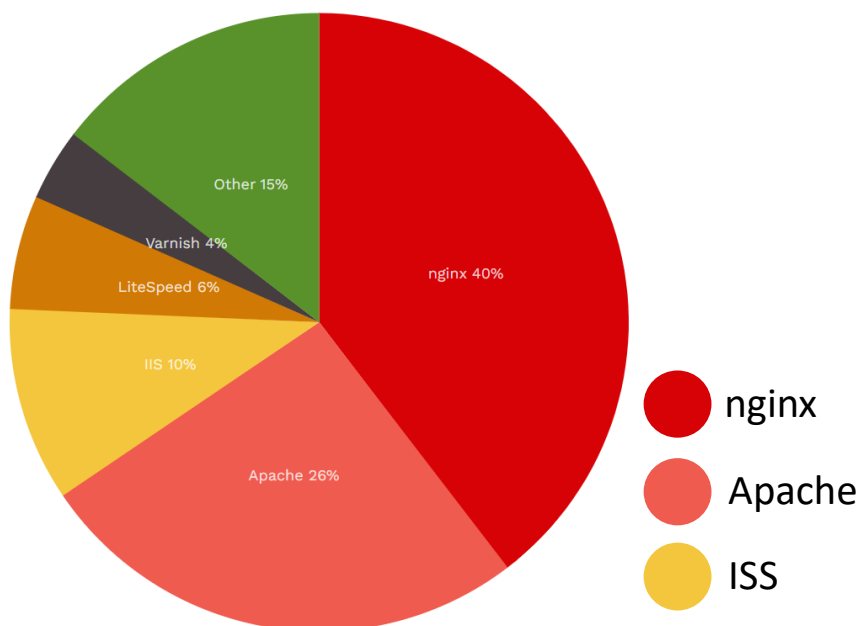




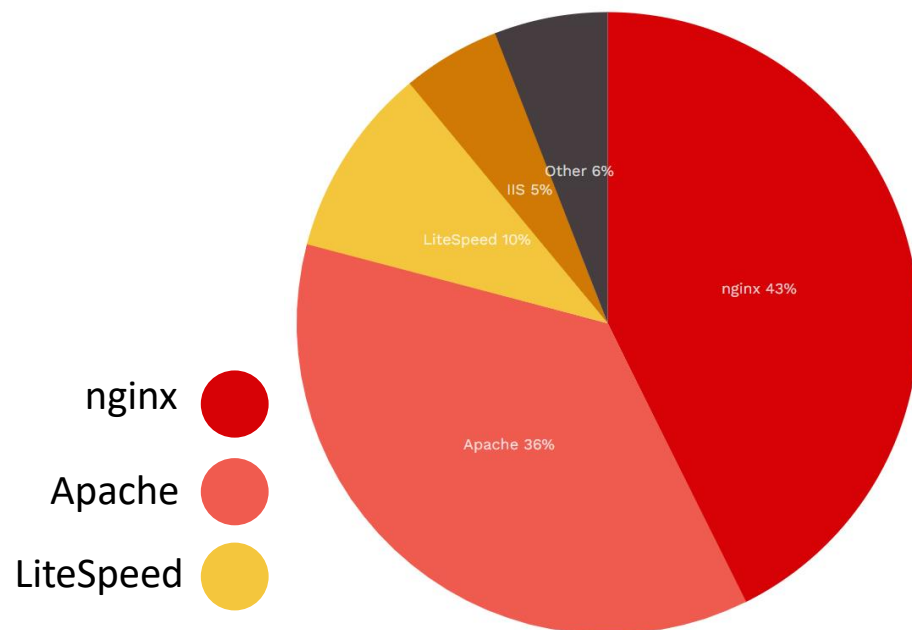
Per la cronaca

- Qualche statistica di diffusione dei server (settembre 2023, fonte <http://trends.builtwith.com/web-server>):

top 1M siti



tutta Internet



Prima domanda



BONUS

DOMANDA 1:

Nell'architettura del Web:

- ☐ le applicazioni web sono eseguite solo sui server
- ☐ il client inizia sempre l'interazione con i server
- ☐ I client sono in grado di visualizzare autonomamente tutti i formati
- ☐ il server inizia sempre l'interazione con i client

Prima domanda

BONUS

DOMANDA 1:

Nell'architettura del Web:

☐ Le applicazioni web sono eseguite solo sui server

☒ **il client inizia sempre l'interazione con i server**

☐ I client sono in grado di visualizzare autonomamente tutti i formati

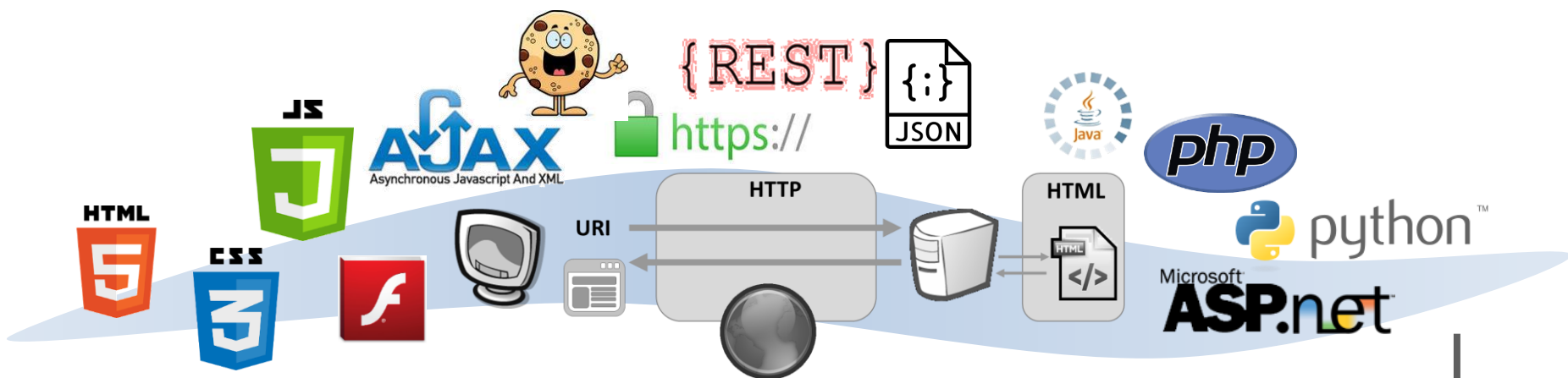
No, per alcuni formati usano plug-in

☐ il server inizia sempre l'interazione con i client

No, il server risponde alle richieste dei client

Evoluzione

- L'architettura del Web attuale è basata sulla struttura originaria ma nel tempo si è arricchita per fornire:
 - servizi avanzati agli utenti e
 - strumenti efficaci agli sviluppatori.
- L'evoluzione ha riguardato numerosi aspetti critici come la **complessità** (ricchezza) dei contenuti, la **dinamicità** dei contenuti, la **sicurezza** delle transazioni, la **distribuzione** dei servizi, l'**espressività** dei linguaggi, ecc.

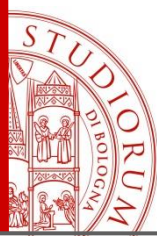


HTTP: PULL e PUSH

- **Modello di interazione C/S:**

- il modello di funzionamento di HTTP prevede che l'interazione tra client e server **sia iniziata esclusivamente dal client**. Quindi il client attira a sé (**pull**) i contenuti richiesti. Questo impone che il client (o, più precisamente, l'utente) richieda ogni volta l'informazione.
- In certi altri casi, invece, si vuole prediligere la immediata disponibilità di informazione anche se l'utente non le ha richieste. Quindi il server spinge al client (**push**) i contenuti. Questo è il funzionamento di molte applicazioni di **casting** (webcasting, podcasting, streaming, ecc.) e anche di tutti i sistemi di **instant messaging**.
- Sistemi più recenti utilizzano un modello **pseudo-push** basato su polling (pool): un processo automatico, ad intervalli regolari, interroga il server (**poll**) per sapere se ci sono nuovi contenuti. Viene usato, per esempio da RSS o dalla mail.





I cookie sono usati principalmente per
Gestione sessione: mantenere l'utente autenticato, contenuto del carrello, preferenze in corso.

Personalizzazione: lingua, tema UI, impostazioni di layout.

Tracciamento: analisi dei comportamenti, profilazione pubblicitaria.

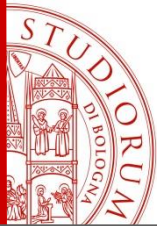


Cookies

Il browser, a sua volta, lo rimanda al server ad ogni richiesta successiva allo stesso dominio, permettendo di mantenere uno "stato" in un protocollo (HTTP) che per impostazione predefinita è stateless

- Il termine **cookie** (anche magic cookie) indica un blocco di dati opaco (cioè non interpretabile) lasciato dal server in consegna ad un richiedente per poter ristabilire in seguito il suo diritto alla risorsa richiesta:
 - I cookies sono usati nella gestione delle sessioni
 - Ogni server lascia in consegna suoi cookies e associa a questi dati ad informazioni sulla transazione.
 - Ogni volta che il browser accederà al server che ha lasciato i cookie, rifornirà i dati del cookie che permettono al server di identificare il richiedente, e creare così un profilo specifico.





Programmare il web

- Le applicazioni per il Web possono essere realizzate programmando **lato client** (codice che gira nel browser) o **lato server**.



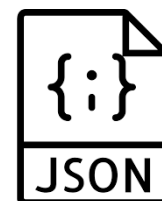


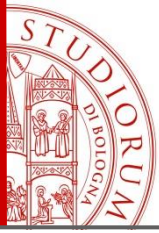
Javascript

- **JavaScript** ,è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client:
 - È stato introdotto da Netscape come LiveScript, (poi rinominato Javascript)
 - è un marchio Oracle, il linguaggio standard è ECMAScript, standardizzato da ECMA.
 - un linguaggio interpretato debolmente tipizzato, debolmente orientato agli oggetti.
 - Funziona anche lato server, sostanzialmente da sempre, le prime implementazioni sono sui server Netscape. L'uso lato server è stato *revitalizzato* (lo stack MEAN si basa su questo linguaggio sia a lato client che server).
 - L'ultima standardizzazione è di giugno 2022, ECMA-262 Edition 13.



- **AJAX** (Asynchronous JavaScript and XML), un gruppo di tecnologie utilizzate per la realizzazione di **RIA** (Rich Internet Application) ovvero applicazioni web client-side e server-side fortemente interattive:
 - Lo sviluppo AJAX si basa su uno **scambio di dati in background** fra web browser e server, che consente l'aggiornamento dinamico di una pagina web senza esplicito ricaricamento da parte dell'utente.
 - Nonostante l'acronimo, l'uso di XML non è indispensabile, si usa per esempio **JSON** (in una variante che è chiamata AJAJ) e la richiesta al server non deve necessariamente essere asincrona.





Java client-side: le applet



- Le **applet** Java sono applicazioni scritte in Java (introdotte nel 1995) che vengono incapsulate all'interno di pagine web (con `<object>`) e fatte eseguire nel browser.
- Hanno avuto grande sviluppo in passato ma ad oggi sono considerate obsolete:
 - Sono **deprecate** da Java 9 in (2017)
 - Sono state rimosse completamente a partire da Java SE 11 (18.9) nel 2018.

Altri oggetti dinamici

- Alcune plug-in del browser, ne estendono la funzionalità per inserire specifici oggetti dinamici.
 - Con questo meccanismo sono inclusi oggetti **Adobe Flash** o **MS Silverlight**, che sono stati usati per lungo tempo per realizzare player di streaming audio/video.
 - Questi strumenti consentono di creare applicazioni interattive lato client grazie alla presenza di un linguaggio di scripting interno.
 - In particolare Flash usa ActionScript, che è ispirato a Javascript mentre Silverlight usa proprio Javascript.
 - Questo tipo di strumenti sta andando in disuso con l'avvento di HTML5.

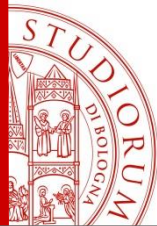




Flash

- YouTube ha sostituito il player Flash con uno HTML5 nel gennaio 2015
- A causa di attacchi e vulnerabilità note, alcuni browser hanno disabilitando il supporto automatico a Flash (rimosso da Chrome nel 2021)
- *“daily Chrome users who have loaded at least one page containing Flash content has gone down from over 80% in 2014 to 8% in 2018”.*





Framework di sviluppo

Una libreria è come una scatola di attrezzi (martello, cacciavite, sega). Puoi scegliere quali attrezzi usare e come usarli per costruire la tua casa nel modo che preferisci. Un framework è come un prefabbricato o un progetto architettonico dettagliato. Ti fornisce una struttura di base con delle regole su come assemblare le diverse parti. Devi seguire il progetto del framework, ma puoi personalizzare alcune sezioni (ad esempio, il colore delle pareti, l'arredamento).

- I **framework** sono librerie che rendono più ricco, sofisticato e semplice l'uso di una tecnologia, come un linguaggio server-side, un linguaggio client-side o le specifiche grafiche di una pagina web.
 - **Server-side** esistono dalla fine degli anni novanta, e hanno reso la programmazione a tre livelli drasticamente più facile.
 - **Client-side** si sono sviluppate a partire dal 2002, su CSS e Javascript, con scopi molto difforni.
- Si può utilizzare un **full-stack framework** che fornisce piattaforme completamente integrate oppure un **glue-framework** che usa contestualmente strumenti non completamente integrati





Web solution stack

- Un **solution stack** è un insieme di componenti o sottosistemi software che sono necessari per creare una piattaforma completa in modo che nessun software aggiuntivo sia **indispensabile** allo sviluppo di applicazioni.
- Per esempio per lo sviluppo Web un solution stack è **solitamente** costituito da 4 elementi: sistema operativo, web server, database, e linguaggio di programmazione.

LINGUAGGIO DI
PROGRAMMAZIONE

DATA BASE MS

WEB SERVER

SISTEMA
OPERATIVO

LAMP

- Uno dei solution stack più noti è LAMP, in cui:
 - Linux è il sistema operativo.
 - Apache è il server web.
 - Il DBMS è MySQL.
 - Il linguaggio di programmazione comunemente è PHP ma vengono usati anche Perl e Python.
- Tutto il solution stack è completamente opensource.



PHP



MySQL



Apache



Linux



WAMP

- Variante Windows di LAMP (che useremo anche noi):
 - **W**indows è il sistema operativo.
 - **A**pache è il server web.
 - Il DBMS è **M**ySQL.
 - Il linguaggio di programmazione comunemente è **P**HP ma vengono usati anche **P**erl e **P**ython.



PHP



MySQL

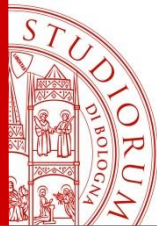


Apache



Windows



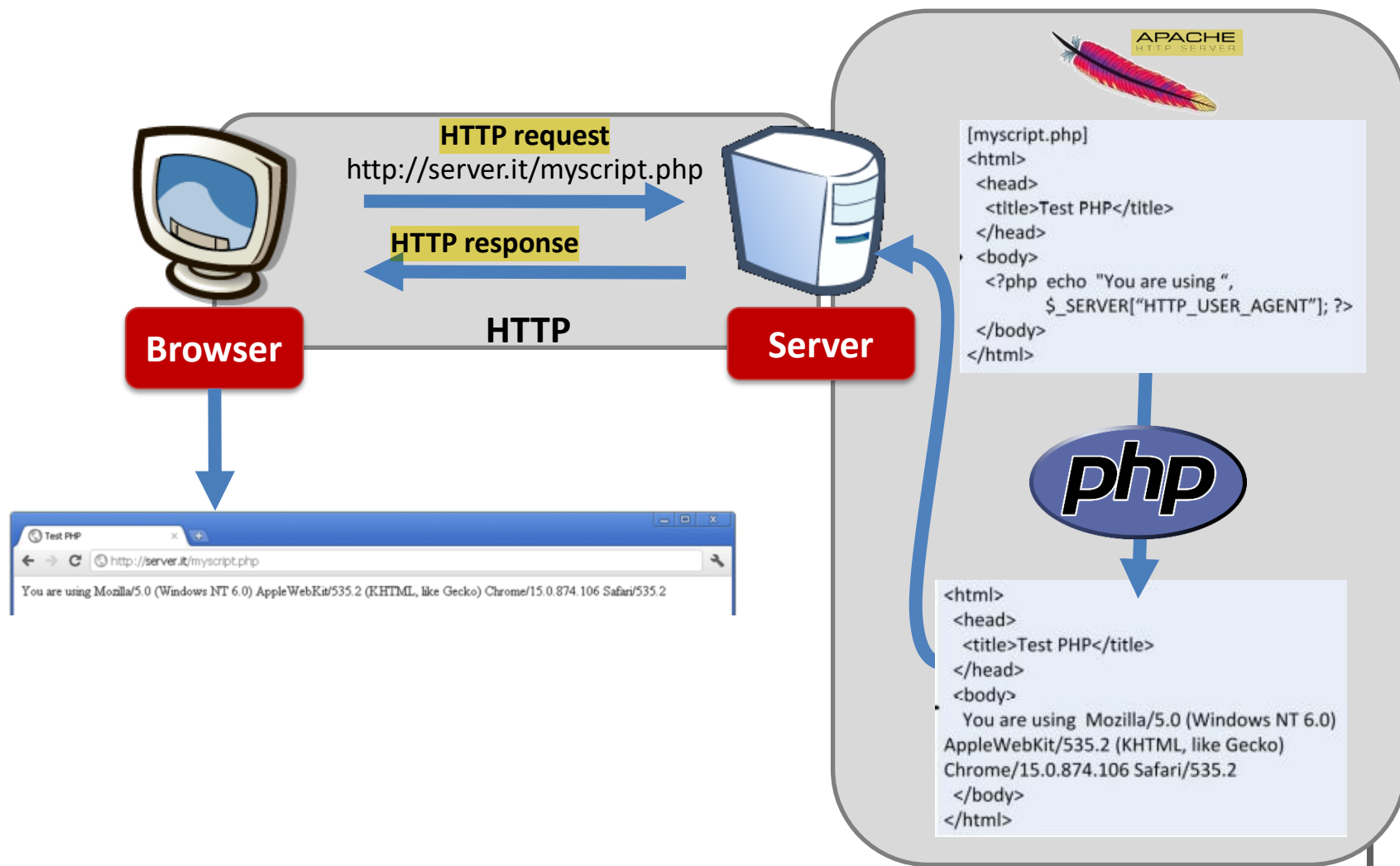


PHP

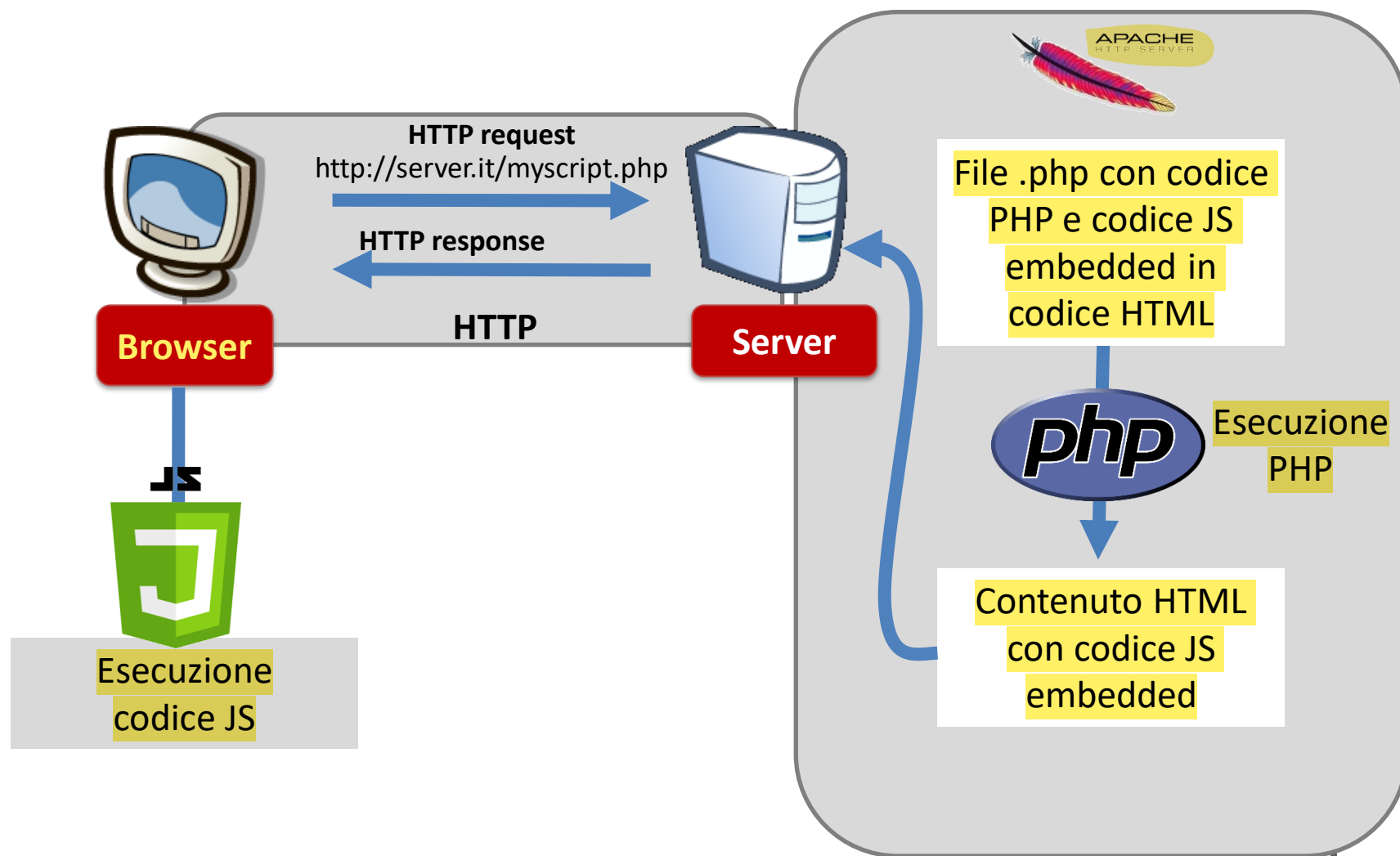


- PHP è un **linguaggio di scripting, interpretato**, originariamente concepito per la programmazione di pagine web dinamiche **lato server**.
 - L'interprete PHP è un software libero distribuito sotto la PHP License.
 - L'acronimo è attualmente espanso come nella forma ricorsiva **PHP Hypertext Preprocessor** (originariamente era inteso come acronimo di «Personal Home Page»)
 - Citando Wikipedia «Un esempio di software scritto in PHP è MediaWiki, su cui si basano i progetti wiki della Wikimedia Foundation come Wikipedia»

Architettura PHP



E se c'è anche JS?



WISA

- L'architettura composta solo da prodotti Microsoft è:
 - **W**indows è il sistema operativo.
 - **I**nternet Information Services è il server web.
 - Il DBMS è Microsoft **S**QL Server.
 - Il linguaggio di programmazione è **A**SP.NET.



ASP



SeQueL



IIS



Windows

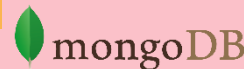


MEAN

- **MEAN** è un solution stack con una struttura diversa.
- Non fa riferimento a sistema operativo ma in compenso specifica componenti per lo sviluppo client.
- In particolare:
 - **M**ongoDB, come data Base (NoSQL database).
 - **E**xpress.js, come framework di sviluppo JavaScript lato **server**.
 - **A**ngular JS, come framework di sviluppo JavaScript lato **client**.
 - **N**ode.js, ambiente di esecuzione per applicazioni **server-side** (permette di eseguire codice Javascript server-side all'esterno del browser).

me  

MongoDB



Express

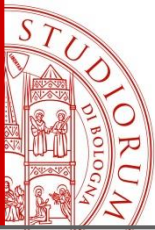
express

Angular



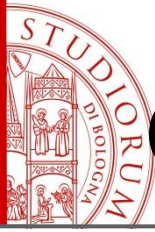
Node





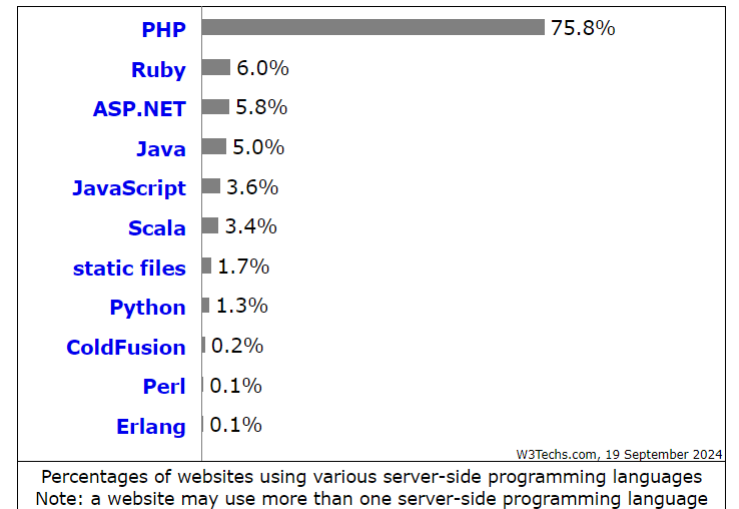
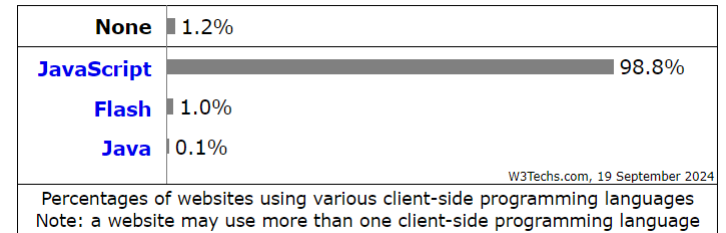
Confronto MEAN-L(W)AMP

- Il confronto tra **MEAN** e **L(W)AMP** evidenzia molte differenze:
 - LAMP e WAMP fanno riferimento a uno specifico sistema operativo. MEAN è più moderno, nasce **multiplatforma**.
 - MEAN usa Node.js come **ambiente di esecuzione delle applicazioni server-side**, esegue codice JS al di fuori del browser (paradigma: «**JavaScript everywhere**»).
 - MEAN usa un DB **non relazionale** (MongoDB) al posto di MySQL, DB relazionale di L(W)AMP.
 - MEAN fornisce **due supporti di programmazione** (uno **client** e uno **server**, Angular e Express) basati sullo stesso linguaggio **Javascript**. L(W)AMP si occupano invece di definire solo lo stack a lato server, usando per programmare PHP/Python e Perl.

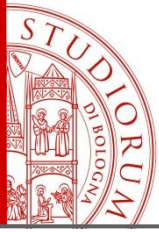


Concludendo... quali tecnologie Web?

- Quelle indispensabili a **creare pagine**:
 - **HTML5**: per strutturare le pagine e definirne il contenuto
 - **CSS3**: per definirne il layout
- Tra le varie tecnologie per **programmare**:
 - **JS**: client side
 - **PHP**: server side
- Altre tecnologie e tool...



<http://w3techs.com/technologies/>



Domande?

