

Appunti di fondamenti dei sistemi web

Appunti fondamenti dei sistemi web

Prime lezioni

Per iniziare direi che la definizione di **Web** è necessaria, dove infatti definiamo uno spazio dove sono presenti documenti e altri tipi di risorse identificate da un **URL** (Uniform Resource Locators) e collegate tra di loro con dei link ipertestuali. Tutto questo è accessibile dall'internet tramite richieste client-server. La prima pagina web nasce il 6 Agosto 1991 e al suo interno si poteva trovare definizione del web rilasciata dal consorzio W3. Entrate in vigore nel 1990 sono le tre tecnologie fondamentali del web, ovvero l'**HTML** (HyperText Markup Language), l'**URI** (Uniform Resource Identifier) e l'**HTTP** (Hypertext Transfer Protocol). Queste tre caratteristiche si sono evolute nel tempo ma rimangono la base di tutto il web odierno, fornendo un formato delle pagine, un identificativo delle risorse e l'interazione tra client e server.

Il primo browser è stato creato nel 1991 dalla CERN, anche se non aveva interfaccia grafica. Il primo vero browser con interfaccia grafica è stato *Mosaic* nel 1992. Nel 1994 troviamo **Netscape** che diventa il browser più popolare sul mercato. Nel 1995 Microsoft lancia il suo *Internet Explorer* e poi è iniziata la **guerra dei browser**. Questa guerra era data dall'introduzione di tag personali per HTML, così da imporre limitazioni agli altri browser. Tutto questo ha portato alla fine alla creazione di un consorzio per la gestione degli standard web chiamata **W3C** che gestisce i linguaggi web. Questo sviluppo continuo è fatto dal *WHAT WG*, ovvero il gruppo di lavoro delle tecnologie web.

Con **URI** definiamo la versione generica dell'indirizzo di un file, mentre l'**URL** (Uniform Resource Locator) è la parte che identifica la risorsa tramite un indirizzo IP. Un URI ha uno schema ben preciso che deve essere rispettato per identificare una risorsa.

- Un URI è diviso in:

URI → **schema : [// authority] path [? query] [# fragment]**

L'*authority* individua l'organizzazione dello spazio dei nomi, il *path* è la parte identificativa della risorsa, la *query* è un insieme di parametri che indica l'output di una query sul motore di ricerca e il *fragment* indica una risorsa secondaria come un link a una pagina.

Ecco alcune definizioni fondamentali per iniziare:

- Con **HTML** definiamo un linguaggio di *markup*, ovvero un linguaggio di formattazione del testo che usufruisce di tag (<>) per delineare la struttura.

- Con **CSS** (Cascading Style Sheets) definiamo un linguaggio specifico a gestire gli aspetti presentazionali della pagina, controllandone le caratteristiche quali colore, forma, dimensione ecc...
- Con **Browser** definiamo l'applicativo lato client che permette di visualizzare documenti ipertestuali e multimediali, inoltre è questo che inizia la comunicazione con il server per la richiesta di pagine HTML.
- Il **Server** è l'applicazione in grado di gestire e rispondere alle richieste dei client tramite individuazione di identificatore quale URL.

Il modello client server prevede tre tipologie di interazioni per la gestione delle richieste delle pagine HTML come:

- Il **pull** è l'azione da parte di un client di richiedere una risorsa al server;
- Il **push** invece è l'azione da parte del server di *spingere* al client i contenuti anche se l'utente non gli ha richiesti;
- Lo **pseudo-push** è un modello basato sul polling, quindi sulla richiesta al server con intervalli regolari per sapere se ci sono novità.

Con il termine **cookie** definiamo un *blocco di dati* che il server consegna ad un richiedente per poter continuare ad usare la sua sessione utilizzata all'ultimo accesso. Questo vuol dire che ogni volta che il browser richiede a un server una risorsa avrà anche accesso ai cookie lasciati in precedenza.

Programmazione web

Le applicazioni Web possono essere realizzate programmando sia lato client che lato server, questo dipende dalla tecnologia utilizzata e dalle necessità. Negli anni di vita di internet si sono alternate molte tecniche diverse di approcciarsi a questo argomento.

- Una delle principali oggi è **Javascript**, ovvero un linguaggio di scripting orientato agli oggetti e debolmente tipizzato utilizzato soprattutto nella programmazione lato client.
- Possiamo anche avere il metodo **AJAX** (Asynchronous JavaScript and XML), ovvero l'utilizzo di un gruppo di tecnologie per la rappresentazione di applicazioni client-side e server-side interattive. Il tutto si basa sull'interazione tra browser e server che si scambiano dati in background così da aggiornare dinamicamente le pagine web.

Con **meta-linguaggio di markup** definiamo un tipo di linguaggio per definire linguaggio di markup come il linguaggio xml, dtd o json.

- Le **applet** Java sono applicazioni scritte in Java che vengono incapsulate all'interno di pagine web e fatte eseguire nel browser. In generale questo metodo non è più utilizzato da anni.

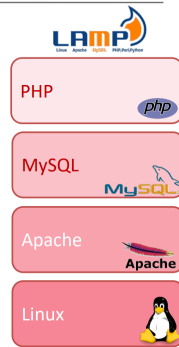
Per migliorare lo sviluppo di applicazioni web e facilitare l'integrazioni di tutte le parti (client e server side) si possono utilizzare dei **framework**, ovvero delle librerie che rendono più ricco l'uso della tecnologia grazie a piattaforme integrate. Possiamo avere *full-stack framework* che forniscono piattaforme di sviluppo, oppure *glue-framework* che utilizza strumenti non completamente integrati.

Con **solution stack** invece definiamo un insieme di componenti software necessari per sviluppare applicazioni web. Per esempio dal punto di vista web è necessario avere quattro layer, ovvero il sistema operativo, il web server, il DBMS e il linguaggio di programmazione.



LAMP

- Uno dei solution stack più noti è LAMP, in cui:
 - **L**inux 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.
- Tutto il solution stack è completamente opensource.



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.

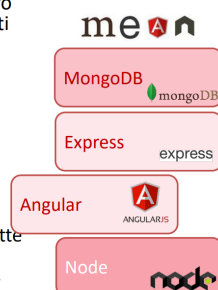


PHP è un linguaggio di scripting interpretato lato server che permette di creare pagine web dinamiche. Uno tra gli esempi più comuni di utilizzo php sono le wiki.



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).



Tutti questi sono dei tipi di approcci allo sviluppo web e, in particolare vediamo che in base alle esigenze abbiamo diversi modi di agire. La tipologia **MEAN** è una delle più recenti che non si basa su un sistema operativo, essendo portabile ovunque, ma si focalizza su entrambi client e server side.

HTML

Il linguaggio HTML è formato da tag che permettono di creare elementi nella pagina. Tutti questi tag sono divisi in categorie che permettono di fare specifiche informazioni.

- **Metadati** sono gli elementi che descrivono un dato, come per esempio un'informazione sulla parola più ricercata su un sito;
- **Interactive** sono gli elementi che permettono di visualizzare contenuti multimediali;

- **Embedded** permettono di includere elementi espressi in un altro tipo di linguaggio, come immagini vettoriali o file audio;
- **Phrasing** che definisce il testo del documento;
- **Flow** che definisce la maggior parte dei tag del body;
- **Heading** che definisce la testa;
- **Section** permette la divisione della pagina.

In una pagina HTML possiamo trovare tanti tags descrittivi (metadati) sempre nella head di una pagina come:

Nome	Descrizione
<base>	Un file unico per indicare il documento che verrà visualizzato tramite URL
<meta>	Usato per aggiungere metadati al documento
<link>	Permette di collegare file esterni, come script o fogli di stile CSS
<style>	Per utilizzare linguaggio CSS dentro il file HTML, questo sovrascrive i fogli di testo

Quando andiamo a scrivere una pagina web dobbiamo fare attenzione ai caratteri che vengono utilizzati nella trascrizione (parsing) della pagina. Il più usato e consigliato è l'UTF-8 perché si basa sull'UNICODE a 32 bit, ma con differenza che non appesantisce la memoria perché non ha una lunghezza fissa, varia da 1 a 4 byte.

Libreria comandi

Nome	Descrizione
<a>	Tag che permette di poter inserire link a pagine oppure a sezione della pagina
	Tag per lista puntata non ordinata
	Tag per lista ordinata con numeri
	Tag interno alla lista che permette di inserire un nuovo elemento della lista
<video>	Tag per inserire video all'interno di una pagina, ha sempre bisogno dell'attributo <i>src</i> per indicare la risorsa
<audio>	Tag per inserire file audio con lo stesso funzionamento del tag video
<source>	Tag interno a video e audio che permette di inserire più risorse per evitare che non siano supportate dalla pagina
<table>	Tag che indica l'inizio di una tabella
<tr>	Tag interno a una tabella che indica la riga
<th>	Tag interno a tr che permette di mettere una header in grassetto
<td>	Tag interno a tr che permette di scrivere all'interno
<caption>	Tag che aggiunge una didascalia sopra una tabella

Inserimento dati

Per poter rendere la pagina web interattiva tramite controlli di input dell'utente. Questa sezione si apre tramite il tag **<form>** che permette di processare dati lato client e server. Il form al suo interno necessita di più attributi fondamentali per la gestione dell'input quali **action e method**. L'action specifica l'URL dell'applicazione serve e method invece specifica in che modo verranno inviati i dati. Il method ha due opzioni, il *get* e il *post*. Il primo invia i dati raccolti in modo testuale nella sezione query dell'URL, ma questo comporta una scarsa sicurezza dal punto di vista di dati troppo lunghi o sensibili. Il post invece invia i dati nel body del messaggio HTTP e non ha restrizioni di grandezza. Inoltre è molto più sicuro perchè si basa sull'incapsulamento, ma richiede sempre un protocollo HTTPS per essere molto più sicuro. Per poter *inviare* i dati dobbiamo usare un bottone tramite il tag **<input>**, essenziale anche per poter creare delle caselle di testo.