



# Linguaggi e Paradigmi di Programmazione WEB

**Ing. Simone Giustetti**  
[www.giustetti.net](http://www.giustetti.net)

# Come Funziona il WEB

- Un sito è un insieme di file salvati su un server.
- Il server è collegato ad Internet oppure ad una rete aziendale (LAN).
- Il sito può essere visionato con un browser (Client).
- Il client invia continuamente dati al server (Richieste) .
- Il client scarica continuamente file dal server.



Il lato client è chiamato anche “**Front End**”.  
Si tratta della parte di presentazione di ogni applicazione che è gestita dal Browser.

Il lato server è chiamato anche “**Back End**”  
Contiene la logica dell'applicazione e gestisce i dati centralizzati.

L'insieme di Front e Back End viene chiamato “**Full Stack**”.



Il Front End di un sito consiste principalmente di 4 tipologie di file che sono caricati dal browser:

- CSS
- **HTML**
- JavaScript
- Multimedia

Il Back End consiste solitamente di 3 parti:

- Database Server (RDBMS)
- Linguaggio di programmazione
- **Web Server**



Una applicazione WEB è costituita da molte parti di conseguenza per la sua realizzazione sono usati molti linguaggi. Ognuno ottimizzato per un compito specifico:

- **CSS**: Formattazione e stili
- **HTML**: Contenuto e struttura della pagina
- **JavaScript**: Impaginazione dinamica / logica dell'applicazione
- **PHP**: Logica dell'applicazione
- **SQL**: Gestione dei dati



# HyperText Markup Language

Linguaggio su cui si fonda ogni sito.

Un file HTML viene sempre caricato nel browser quando si imposta un URL di una pagina.

Include l'intero contenuto di una pagina.

Usa TAG specifici per distinguere tra le diverse tipologie di contenuto.



# Cascading Style Sheets

Consente di definire gli stili degli elementi: Colore, forma, dimensione dei caratteri, font, dimensione delle immagini, ecc.

Consente di impostare la posizione di un elemento.

Consente di creare animazioni semplici ed altri effetti dinamici.



# JavaScript

Linguaggio di programmazione progettato per girare nel browser e rendere dinamica una pagina.

Risponde agli eventi.

Consente di scrivere applicazioni web contenute in una singola pagina (Videogiochi, ecc.).

Il suo uso improprio è la causa di molti problemi di sicurezza che affliggono il web (Deve essere usato con prudenza).





Linguaggio di programmazione lato server.

Supporta sia il paradigma della programmazione strutturata che quello della programmazione ad oggetti.

Supporta un gran numero di RDBMS.

Sviluppato dal 1994 ed usato da 80% dei siti con interfaccia dinamica.



# Structured Query Language

Linguaggio specializzato nella strutturazione e amministrazione di banche dati.

Lo standard dominante nell'ambito dei DBMS.

Standardizzato, ma nessuna implementazione supporta appieno uno standard. Ogni produttore distribuisce un suo dialetto specifico.

Le funzionalità di base sono comuni a tutti i dialetti.



# Strumenti di Sviluppo

Pacchetto / Programma	Linux / BSD	MacOsX	Windows	Sito Ufficiale
AMPPS	X	X	X	<a href="http://ampps.com/">http://ampps.com/</a>
LAMP	X			
MAMP		X		<a href="https://sourceforge.net/projects/mamp/">https://sourceforge.net/projects/mamp/</a>
XAMPP	X	X	X	<a href="https://www.apachefriends.org/index.html">https://www.apachefriends.org/index.html</a>
WAMP Server			X	<a href="https://www.wampserver.com/">https://www.wampserver.com/</a>



# Strumenti di Sviluppo

Pacchetto / Programma	Linux	MacOsX	Windows	Sito Ufficiale
Bluefish	X	X	X	<a href="http://bluefish.openoffice.nl">bluefish.openoffice.nl</a>
Eclipse	X	X	X	<a href="http://www.eclipse.org">www.eclipse.org</a>
Kdevelop	X	X	X	<a href="http://www.kdevelop.org">www.kdevelop.org</a>
Notepad++			X	<a href="http://notepad-plus-plus.org">notepad-plus-plus.org</a>
NetBeans	X	X	X	<a href="http://netbeans.apache.org">netbeans.apache.org</a>
Vim / Gvim	X	X	X	<a href="http://www.vim.org">www.vim.org</a>



# Strumenti di Sviluppo

Pacchetto / Programma	Linux	MacOsX	Windows	Sito Ufficiale
DBeaver	X	X	X	<a href="http://dbeaver.io">dbeaver.io</a>
LibreOffice Base	X	X	X	<a href="http://www.libreoffice.org">www.libreoffice.org</a>
MySQL	X	X	X	<a href="http://dev.mysql.com/">dev.mysql.com/</a>
MariaDB	X	X	X	<a href="http://mariadb.org/">mariadb.org/</a>



# Classificazione dei Linguaggi

**Compilati:** Necessitano di un compilatore che traduca la sintassi di alto livello in istruzioni per la macchina. L'eseguibile ottenuto è specifico per l'architettura impiegata, ha buone prestazioni e può essere eseguito autonomamente.

C, C++, Pascal, Fortran, ...

**Interpretati:** La sintassi di alto livello viene tradotta in tempo reale in istruzioni per la macchina. Il codice può girare su ogni architettura per cui esista un interprete. Non può essere eseguito autonomamente. Prestazioni potenzialmente peggiori.

Java, **JavaScript**, Perl, **Php**, Python, ...



**Programmazione Strutturata:** Un problema viene scomposto in sotto-problemi più facili da risolvere. Sfrutta variabili, strutture di controllo, procedure e/o funzioni. Ha un moderato livello di astrazione.

**Programmazione ad Oggetti:** Indirizzata alla modellazione dei dati. Consente di definire oggetti software in grado di interagire tra loro tramite messaggi o altre forme di comunicazione. Permette un elevato livello di astrazione.



**Procedurale / Imperativo:** Il programmatore ha il compito di scrivere la sequenza di operazioni a basso livello necessarie per ottenere un risultato.

C, C++, Java, **JavaScript**, Perl, **PHP**, ...

**Descrittivo / Dichiarativo:** Il programmatore descrive il risultato che vuole ottenere ed il compilatore / interprete rende il risultato richiesto nascondendo gli aspetti a basso livello.

**HTML**, SGML, **SQL**, XML, ...





# Classificazione dei Linguaggi

**Non tipizzato:** Al momento della definizione, non viene impostata una tipologia per le variabili. La tipologia è solitamente contestuale, dipendente dalle elaborazioni eseguite.

**JavaScript, PHP, ...**

**Tipizzato debole:** La tipologia delle variabili deve essere definita alla dichiarazione, ma il compilatore può eseguire conversioni implicite.

**C, SQL, ...**

**Tipizzato forte:** La tipologia delle variabili deve essere definita alla dichiarazione. Impossibile eseguire conversioni di tipo implicite.



Progettare / Scrivere un programma equivale ad elaborare un algoritmo che porti al risultato desiderato.

Un algoritmo è una sequenza ben definita di istruzioni elementari che risolvono un problema.

Al crescere della complessità di un problema, aumenta la lunghezza del programma e diventano inefficaci le operazioni di manutenzione, modifica e correzione degli errori.



# Programmazione Imperativa Strutturata

Include l'uso di **cicli** e **strutture di controllo** per regolare il flusso del programma

Un ciclo permette di eseguire una istruzione molte volte

```
do <istruzioni> while <condizione>
```

Una struttura di controllo permette di eseguire del codice se e solo se una condizione è verificata

```
if <condizione>  
then <istruzione 100>  
else <istruzione 500>
```



Consente di scomporre un algoritmo in parti più piccole, facili da modificare, correggere e mantenere.

Le parti sono chiamate **funzioni**, procedure, metodi, routine, ...

Consentono di eliminare le ripetizioni inutili di codice, abbreviare e semplificare il programma, standardizzarne alcune funzionalità.



Si suddivide il codice in un insieme di “piccoli” blocchi logici, autonomi ed indipendenti tra loro.

I blocchi secondari, chiamati moduli, possono essere sviluppati, testati e corretti singolarmente prima di essere inclusi nel programma finale.

Le interfacce tra moduli devono essere progettate e ben documentate prima di realizzare i moduli.



Si concentra sul modo migliore di rappresentare la realtà per arrivare a risolvere un problema.

È un'evoluzione ed un'enfatizzazione della programmazione imperativa modulare.

Un programma consiste in un insieme di entità, gli oggetti, che comunicano ed interagiscono tra loro.

Un oggetto lega in se un insieme di attributi e di funzionalità.



Una classe è il modello su cui sono basati gli oggetti.

Una classe è:

- Dotata di una interfaccia (La specifica) ed un corpo (L'implementazione).
- L'interfaccia nasconde i dettagli di implementazione, le strutture dati, gli algoritmi, ecc.
- L'interfaccia controlla l'accesso ai dati interni.
- Ogni oggetto ha un proprio stato che evolve solo in base alle operazioni ad esso applicate.
- Le funzionalità sono indipendenti dall'implementazione perciò è possibile modificare la seconda senza toccare la prima.



# I Vantaggi della Programmazione ad Oggetti

- **Coesione:** Una classe rappresenta un'entità finita.
- **Modularità:** Ogni classe è un modulo del programma.
- **Disaccoppiamento dei Moduli:** Sistemi complessi sono costruiti componendo operazioni sugli oggetti.
- **Nascondere l'implementazione:** Le strutture dati e le funzionalità sono accessibili solo attraverso l'interfaccia.
- **Riuso:** Possibile assemblare e distribuire librerie di classi.
- **Estensibilità:** Possibile progettare sottoclassi che estendono quelle originarie.





Tipica delle interfacce grafiche e della pagine web dinamiche.

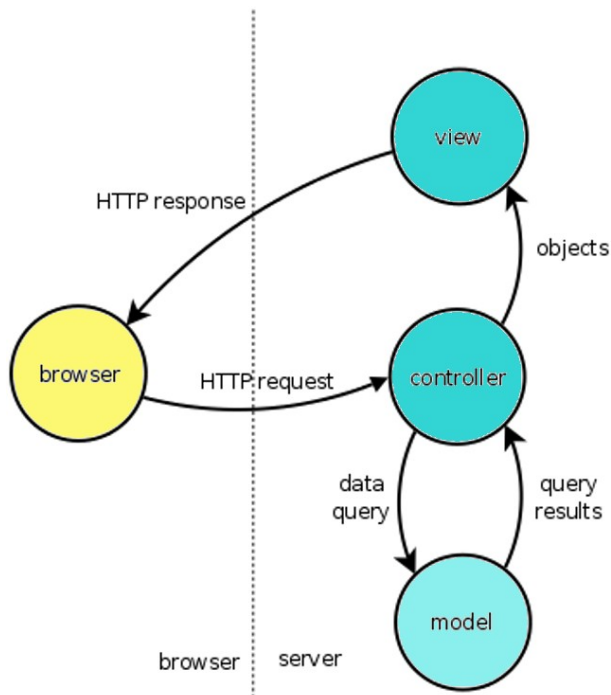
Il codice viene eseguito in risposta agli eventi causati dall'utenza, dal programma stesso o dall'ambiente esterno.

Il codice di risposta può essere progettato seguendo sia un paradigma imperativo che uno ad oggetti, ma il secondo è considerato più adatto.



# Design Pattern

Le applicazioni web sono progettate utilizzando il pattern MVC (Modello - Vista - Controllore) .



Ogni richiesta del browser viene gestita da un controllore che la elabora e reperisce le risorse necessarie per fornire la risposta.

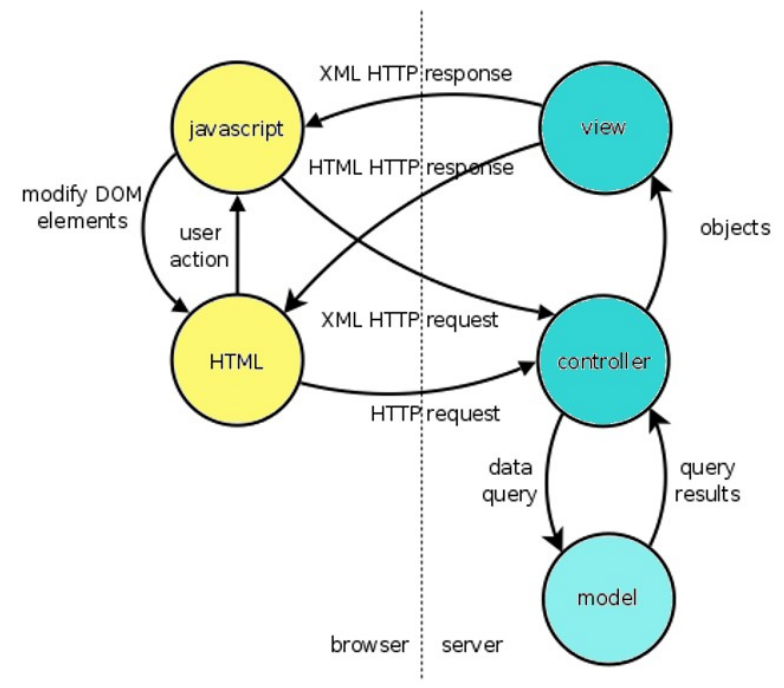
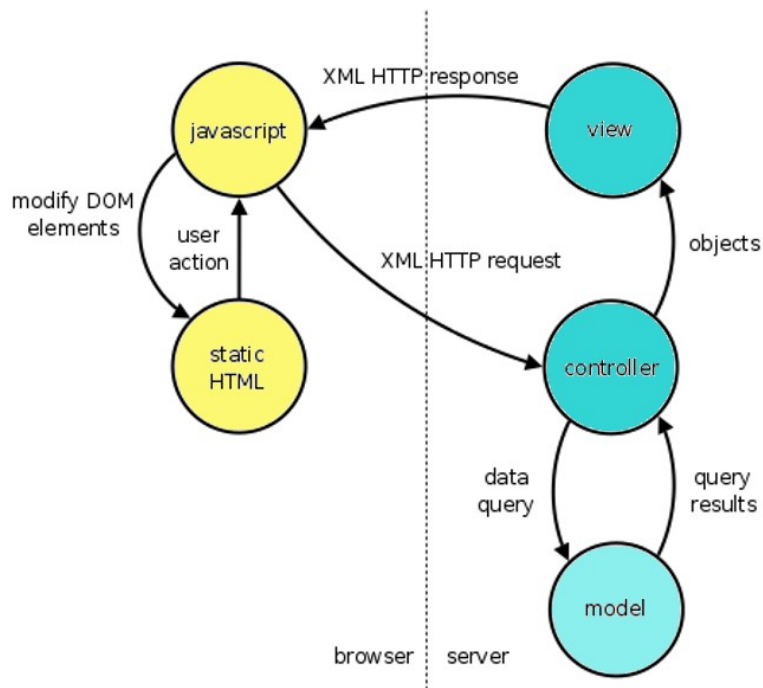
Il controllore eventualmente interroga un RDBMS direttamente oppure attraverso un modello dei dati.

Una volta raccolti i dati, il controllore carica una vista che contiene i dettagli di impaginazione.



# Design Pattern

JavaScript può essere usato in sostituzione o combinato con HTML per pilotare le richieste al server.



## Canali Web

- [www.youtube.com/channel/UCVTlvUkGslCV\\_h-nSAId8Sw](http://www.youtube.com/channel/UCVTlvUkGslCV_h-nSAId8Sw)
- [www.youtube.com/c/TraversyMedia](http://www.youtube.com/c/TraversyMedia)
- [www.youtube.com/channel/UCpOIUW62tnJTtpWFABxWZ8g](http://www.youtube.com/channel/UCpOIUW62tnJTtpWFABxWZ8g)
- [www.youtube.com/channel/UCoebwHSTvwalADTJhps0emA](http://www.youtube.com/channel/UCoebwHSTvwalADTJhps0emA)

## Manuali

- [www.php.net/docs.php](http://www.php.net/docs.php)
- [www.w3schools.com/html/](http://www.w3schools.com/html/)
- [www.w3schools.com/css/default.asp](http://www.w3schools.com/css/default.asp)
- [www.w3schools.com/js/default.asp](http://www.w3schools.com/js/default.asp)
- [developer.mozilla.org/it/docs/Learn](http://developer.mozilla.org/it/docs/Learn)



# Informazioni & Licenze

## LICENZA

Salvo dove altrimenti specificato grafica, immagini e testo della presente opera sono © Simone Giustetti. L'opera può essere ridistribuita per fini non commerciali secondo i termini della licenza:

Creative Commons Attribuzione - Non commerciale - Condividi allo stesso modo 4.0 Internazionale



È possibile richiedere versioni rilasciate sotto diversa licenza scrivendo all'indirizzo: [studiosg@giustetti.net](mailto:studiosg@giustetti.net)

## TRADEMARK

- FreeBSD è un trademark di The FreeBSD Foundation.
- Linux è un trademark di Linus Torvalds.
- Macintosh, OS X e Mac OS X sono tutti trademark di Apple Corporation.
- MariaDB è un trademark di MariaDB Corporation Ab.
- MySQL è un trademark di Oracle Corporation.
- UNIX è un trademark di The Open Group.
- Windows e Microsoft SQL Server sono trademark di Microsoft Corporation.
- Alcuni algoritmi crittografici citati nella presente opera potrebbero essere protetti da trademark.

Si prega di segnalare eventuali errori od omissioni al seguente indirizzo: [studiosg@giustetti.net](mailto:studiosg@giustetti.net)

