

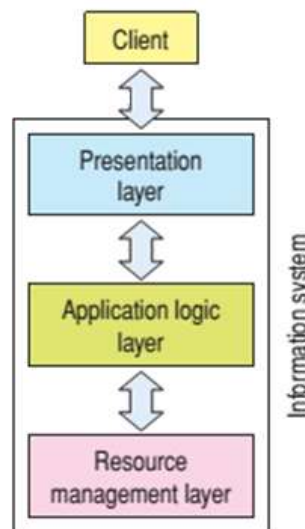
# 1.Applicazioni Web

## 1.1 I 3 livelli del Web

Il **Web** è il più grande **sistema distribuito di tipo informativo**.

Il modello di riferimento del Web è il **client/server**: il processo client chiede un servizio, tipicamente una pagina HTML, al processo server tramite il protocollo HTTP.

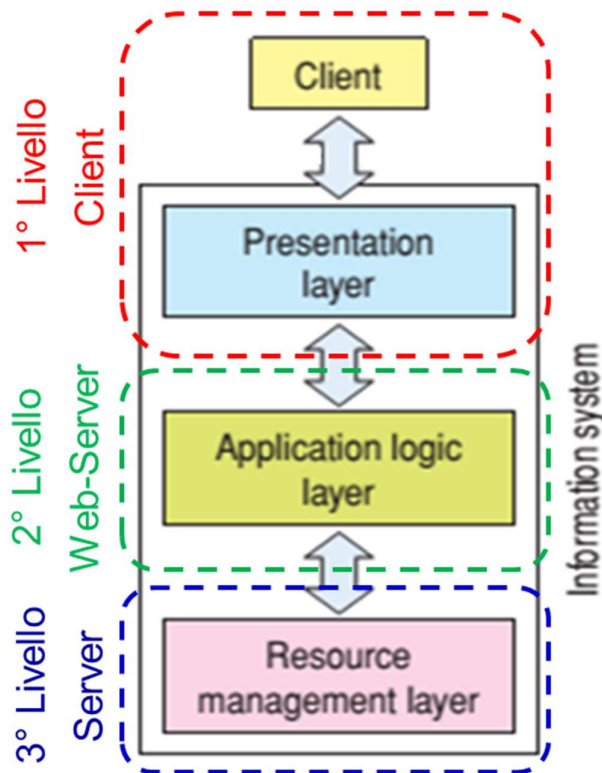
Il web è un'**architettura a 3 livelli**. Come abbiamo già visto i livelli logici di sistema informatico sono i seguenti:



Nel web tipicamente questi 3 livelli logici sono implementati su 3 macchine distinte:

1. **Primo livello: Client** → il computer del client tipicamente contiene il Presentation Layer;
2. **Secondo livello: Web-Server** → un computer, chiamato Web-Server, o anche Application-Server, esegue la logica applicativa (gli algoritmi), implementa quindi l'Application Logic Layer. Viene chiamato anche Middleware;

3. **Terzo livello: Server** → un computer mantiene e gestisce le risorse dell'applicazione. Le risorse sono memorizzate in:
- File system
  - Data base



Si noti che il linguaggio **HTML** non è un linguaggio di programmazione vero e proprio ma è un linguaggio di **markatura**. L'HTML è soltanto una sorta di "impaginatore" che permette di impostare la posizione di testo ed immagini in una pagina web, di inserire tabelle, frame, liste puntate e numerate e così via, oltre a stabilire le proprietà caratteristiche (ampiezza, colore, allineamento, ecc...) di ciascun elemento. **HTML** non ha in sé né strutture in grado di interagire con l'utente né strutture decisionali, è **completamente statico**.

Se la **pagina** è **statica**, ossia è composta solo da codice HTML, il Web-Server si limita prelevare la pagina (ed eventuali contenuti statici come

immagini) dal Server e restituirla al client che la visualizza così com'è senza compiere alcuna operazione.

Se abbiamo la necessità di aggiungere degli **elementi dinamici** alla pagina HTML dobbiamo utilizzare dei **linguaggi di programmazione** all'interno del codice HTML, linguaggi che dovranno essere **interpretati ed eseguiti**.

Ma chi interpreta ed esegue gli script inseriti nel codice HTML?

Possono essere interpretati ed eseguiti o dal client o dal web-server, a seconda del linguaggio di programmazione utilizzato.

Una pagina web quindi può essere:

A. **Completamente statica** → la pagina HTML non contiene script né lato client né lato server: non viene compiuta nessuna elaborazione, la pagina HTML viene visualizzata dall'utente così com'è memorizzata nel file system.

**B. Dinamica**

a. ***Solo lato server*** → la pagina contiene degli script lato server ma non lato client: prima di inviare la pagina al client il web-server esegue il codice all'interno della pagina. Il client mostra all'utente la pagina ricevuta dal web-server senza compiere alcuna operazione.

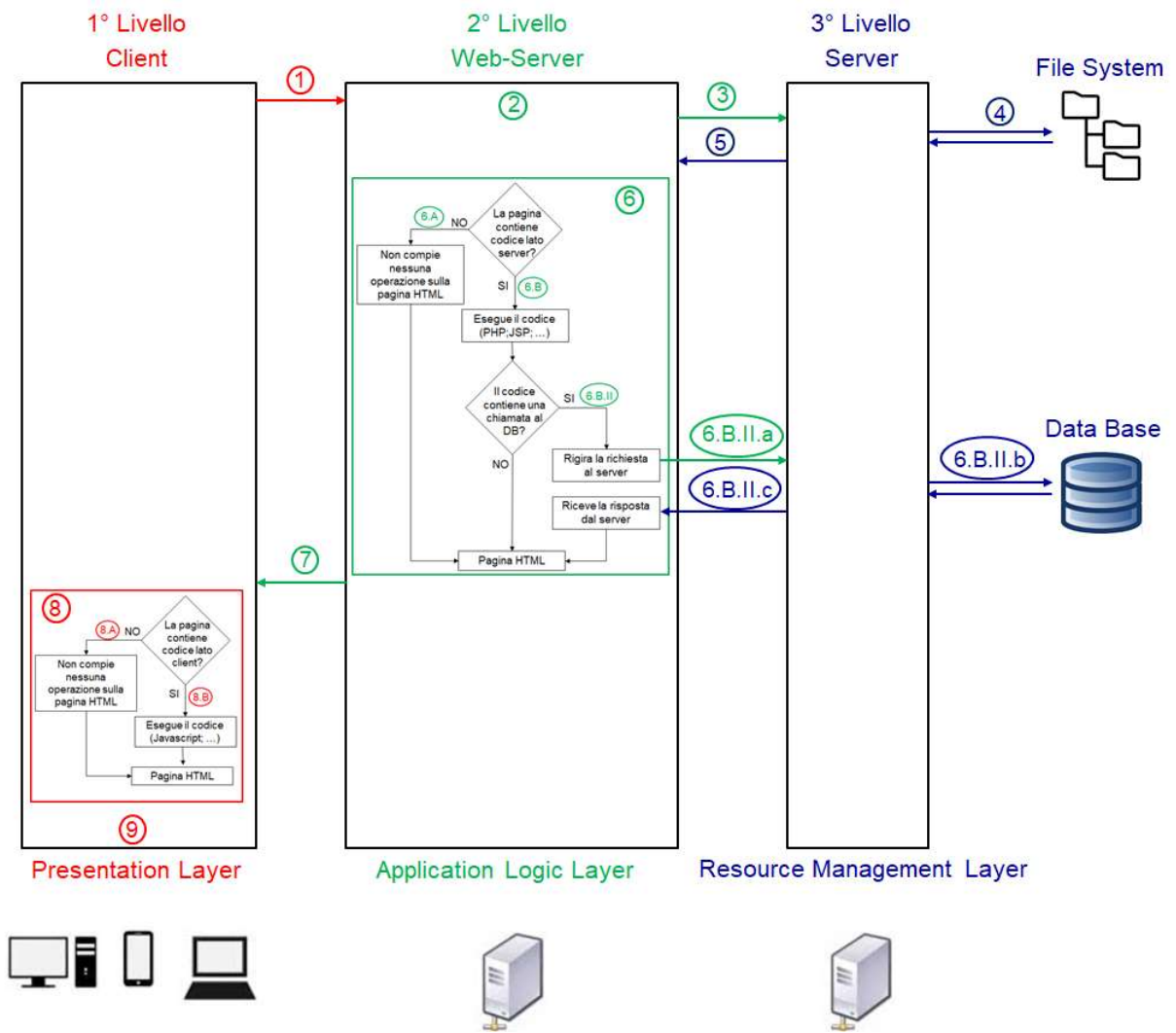
b. ***Solo lato client*** → la pagina HTML contiene degli script lato client ma non lato server: il server invia al client la pagina HTML così com'è memorizzata nel file system. Il client però prima di visualizzarla all'utente esegue lo script lato client (es: javascript).

c. ***Sia lato server che lato client*** → la pagina HTML contiene sia script lato server che lato client: il web-server esegue il codice della pagina richiesta ed invia l'eseguibile al client. Il client prima di mostrare la pagina all'utente compie a sua volta l'elaborazione del codice all'interno della pagina.

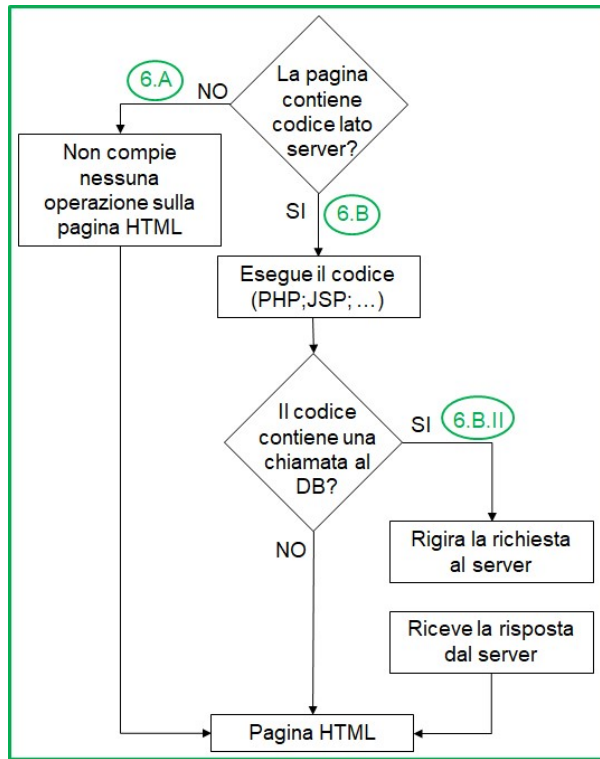
Vediamo ora nel dettaglio i passi compiuti quando l'utente richiede l'apertura di una pagina web:

1. Il client chiede una pagina HTML al server;
2. il web-server, che sta in ascolto sulla porta 80, riceve la richiesta del client e stabilisce una connessione con il client;
3. il web-server chiede la pagina al server;
4. il server preleva la pagina dal file system;
5. il server restituisce la pagina al web-server;
6. il web-server verifica se il documento HTML richiesto ha del codice lato server o meno. Riesce a capirlo dall'estensione del file: se ha estensione HTML allora non contiene codice eseguibile, altrimenti (ad esempio ha estensione PHP) contiene codice eseguibile lato server.
  - A. **Se non ha codice lato server** →
    - I. Il web-server non compie nessuna operazione sulla pagina HTML ricevuta dal server;
  - B. **se ha codice lato server** →
    - I. il web-server esegue il codice della pagina;
    - II. Se contiene una chiamata al Data Base →
      - a. Il web-server rigira la richiesta al server;
      - b. Il server compie le operazioni richieste sul Data Base ed eventualmente restituisce il risultato;
      - c. Il web-server riceve il risultato dal server ed eventualmente continua la sua esecuzione con le informazioni ottenute;
      - d. L'eseguibile ottenuto dall'elaborazione della pagina è una pagina HTML.
7. Il web-server invia la pagina HTML al client;
8. il client riceve la pagina HTML dal web-server e verifica se contiene codice lato client:
  - A. **se non ha codice lato client** →
    - I. il client non compie nessuna operazione sulla pagina HTML ricevuta dal web-server
  - B. **se ha codice lato client** →
    - I. il client esegue il codice
9. il client mostra la pagina HTML all'utente

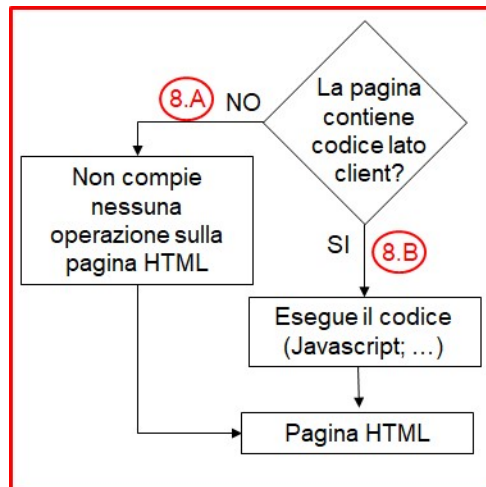
La seguente figura mostra graficamente i passaggi descritti sopra:



Di seguito il dettaglio del punto 6:



Ed il dettaglio del punto 8:



I principali client, che interpretano l'HTML, sono i browser:

- Internet explorer
- Google Chrome
- Mozilla Firefox
- ...

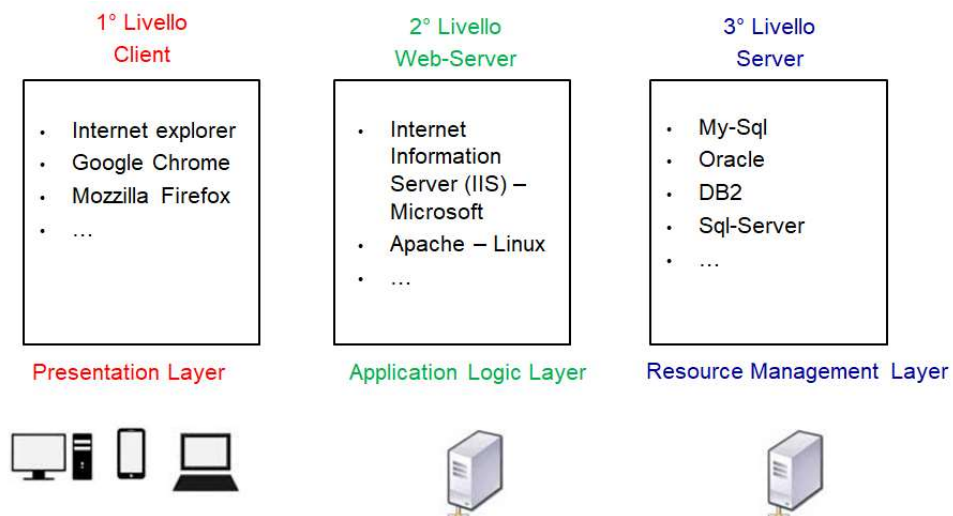
I principali web-server sono:

- Internet Information Server (IIS) – Microsoft
- Apache – Linux
- ...

I principali server sono:

- My-Sql
- Oracle
- DB2
- Sql-Server
- ...

Ricapitolando:



## 1.2 Programmazione lato client

Come visto nella programmazione lato client gli script vengono eseguiti dal browser.

Il server invia al client una pagina HTML contenente degli script che il browser dovrà essere in grado di eseguire. Gli script lato client possono inoltre contenere **istruzioni concrete per il browser**, indicando ad esempio come devono reagire a precise azioni dell'utente o ad un click su un pulsante.

Ad esempio lato client si possono implementare dei controlli formali sui dati inseriti dall'utente in una pagina form. Al click sul pulsante per esempio di "Invio", il browser verifica che i dati inseriti nel form siano coerenti col formato richiesto, o che tutti i campi siano valorizzati. In questo caso la convalida è affidata totalmente al client (browser dell'utente), i dati non vengono spediti al server finché non sono validi. Il server non interagisce in nessun modo e si limita ad attendere i dati convalidati per poterli manipolare.

Le principali tecnologie utilizzate per la creazione di pagine web dinamiche lato client sono:

- JavaScript → rilasciato nel 1995 si è diffuso velocemente ed è diventato così un linguaggio di scripting universale per tutti i browser comuni.
- Adobe Flash Player → onnipresente piattaforma client-side, ma ormai obsoleta

## 1.3 Programmazione lato server

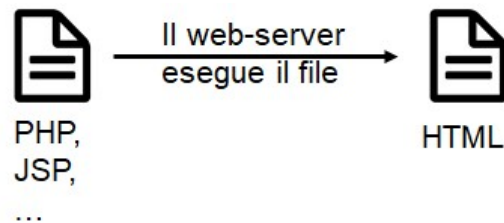
Nell'ambito della programmazione Web si definiscono linguaggi lato server quei linguaggi di programmazione che vengono interpretati ed elaborati dal web-server il quale, successivamente, invia i risultati al client (il browser dell'utente).

Questa volta quindi è l'applicazione web-server che interpreta ed esegue il codice, e non il browser.



Il web-server ricevuto il file dal server, controlla se ha elementi dinamici lato server o meno. Riesce a comprenderlo osservando l'estensione del file: se ha estensione **HTML** allora non contiene elementi dinamici lato server; se non ha estensione HTML ma ad esempio **PHP** o **ASP**, allora sa che contiene all'interno del codice che deve eseguire ed interpretare prima di passarlo al client.

L'eseguibile di un file dinamico lato server è un file HTML.



Il file HTML così ottenuto viene inviato al client.

Si noti che la pagina HTML inviata al client conterrà solo codice HTML ed eventualmente codice JavaScript da eseguire lato client, ma non codice lato server. Se l'utente prova a vedere il codice della pagina visualizzata sul proprio browser vedrà solo codice HTML ed eventualmente Javascript. Il codice lato server viene volutamente mascherato all'utente.

Questo è molto importante ed è necessario per proteggere il codice lato server:

- gli utenti più esperti non possono rubare il codice: la scrittura del codice può costare anche molti giorni di lavoro;
- o manipolarlo: si pensi se fosse possibile manipolare il codice ad esempio facendogli fare le operazioni desiderate sul Data Base, si pensi al sito di una banca!!

La **sicurezza** quindi è l'aspetto più importante della programmazione lato server.

Per accedere al Data Base, sia in lettura che scrittura, quindi è necessario passare per il codice lato server.

Spesso il compito degli script lato server è quello di raccogliere i giusti dati da un database ed inserirli nel sito web.

Ma la programmazione lato server è necessaria anche quando si ha la necessità di eseguire operazioni particolari prima dell'invio della pagina al client, operazioni che determinano i contenuti e la struttura della pagina stessa.

Ad esempio la programmazione lato server può essere utilizzata per popolare dei campi della pagina HTML con valori prelevati da un Data Base. Si pensi ad esempio al listino prezzi di una ditta, dove i prezzi possono subire spesso variazioni nel tempo. Per far sì che la pagina HTML mostrata all'utente abbia sempre i prezzi aggiornati basterà modificare i prezzi sul data base (è assolutamente sconsigliato mettere i valori dei prezzi cablati all'interno del codice!!!).

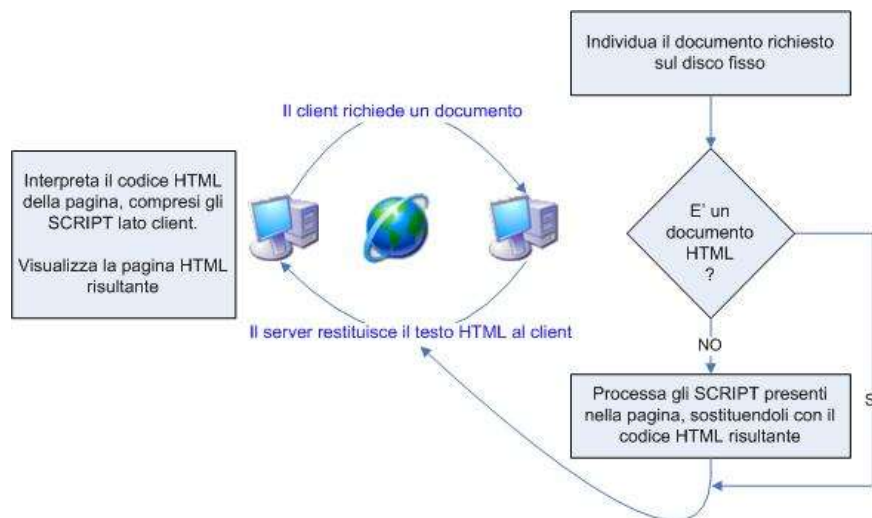
Le principali tecnologie utilizzate per la creazione di pagine web dinamiche lato client sono:

- PHP (Hypertext Preprocessor) → linguaggio di programmazione rilasciato nel 1995. Gratuito ed open source. (pagine con estensione .PHP);
- ASP (Active Server Pages) → Soluzione proprietaria Microsoft (file con estensione .asp);
- Servlet → componente software scritto in Java che produce contenuto Web dinamico;
- JSP (Java Server Page) → Pagina JSP convertita e compilata in una Servlet Java alla prima richiesta di accesso (file con estensione .jsp);
- AJAX (Asynchronous JavaScript And XML - JavaScript asincrono ed XML) → tecnica di sviluppo di applicazioni Web interattive;

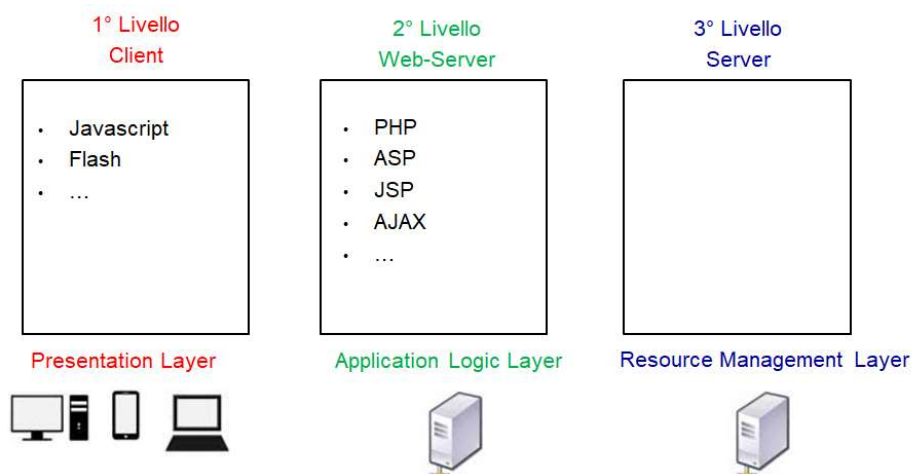
## 1.4 Programmazione lato cliente e lato server a confronto: pro e contro

Per prima cosa si noti che una pagina può contenere sia script lato server, che lato client, ma gli script non possono interagire tra loro, o condividere variabili, in quanto vengono eseguiti in due momenti distinti e successivi.

L'immagine seguente mostra una visione ad alto livello di cosa accade quando un client chiede una pagina HTML al server:



Nell'immagine seguente vengono riepilogati i principali linguaggi di programmazione lato cliente e lato server:



La seguente tabella mostra le differenze tra le due tipologie di programmazione Web, lato client e lato server, evidenziando in verde i pro ed in rosso i contro delle due scelte architetture:

	<b>Tecnologie Lato Client</b>	<b>Tecnologie Lato Server</b>
<b>Dipendenza dal browser client</b>	<p style="text-align: center;"><b>SI</b></p> <p>Non tutti i browser sono in grado di eseguire il codice lato client</p>	<p style="text-align: center;"><b>NO</b></p> <p>l'esecuzione avviene sul server, il client si deve limitare a leggere il documento HTML inviatogli</p>
<b>Codice disponibile</b>	<p style="text-align: center;"><b>SI</b></p> <p>il codice è visibile, basta visualizzare il contenuto HTML della pagina per poter "copiare" o "manipolare" il codice utilizzato dal programmatore</p>	<p style="text-align: center;"><b>NO</b></p> <p>la pagina interpretata dal client non contiene codice, che risiedendo sul server è impossibile da analizzare</p>
<b>memorizzare dati (su file o su Data Base)</b>	<p style="text-align: center;"><b>NO</b></p> <p>non è possibile memorizzare sul server dati relativi agli utenti</p>	<p style="text-align: center;"><b>SI</b></p> <p>È possibile memorizzare i dati inviati dagli utenti mediante le form su file o in basi di dati anche di notevoli dimensioni</p>
<b>Possibile manomissione del codice</b>	<p style="text-align: center;"><b>SI</b></p> <p>non è conveniente fare convalide da considerare completamente attendibili, dal momento che utenti esperti potrebbero modificare la pagina in modo da eludere certi controlli</p>	<p style="text-align: center;"><b>NO</b></p> <p>le operazioni svolte lato server sono occultate al client, che ne può solo constatare i risultati</p>
<b>Velocità</b>	<p style="text-align: center;"><b>SI</b></p> <p>l'esecuzione del codice è veloce ed immediata, dal momento che è lo stesso computer client ad interpretarlo</p>	<p style="text-align: center;"><b>NO</b></p> <p>il caricamento delle pagine risulterà leggermente più lento, dal momento che ad ogni richiesta del client</p>

	<b>Tecnologie Lato Client</b>	<b>Tecnologie Lato Server</b>
		corrisponderà l'invio della richiesta al server che dovrà interpretare degli script prima dell'invio della pagina risultante
<b>Sovraccaricamento del Server</b>	NO Poiché l'esecuzione dello script avviene totalmente sul lato client senza interessare il server	SI Poiché il server deve eseguire gli script. Questo potrebbe causare un abbassamento delle prestazioni intese come tempi di risposta del server ai client
<b>Utilizzo della banda di connessione</b>	NO Gli script vengono eseguiti direttamente dal client (browser) e perciò si evita lo scambio di nuove pagine HTML tra client e server. Anche nel caso in cui la connessione alla rete non sia attiva in quel momento, la pagina potrà aggiornarsi lo stesso.	SI Il client deve inviare la richiesta al server attraverso la rete ed il server deve restituire la risposta sempre tramite la rete. Questo significa un utilizzo maggiore della banda e soprattutto presuppone che sia attiva e funzionante una connessione di rete.
<b>Dinamica delle pagine</b>	SI Pagine dinamiche nel browser (popup di dialogo con l'utente)	NO Pagine generate dinamicamente dal server ma statiche nel browser