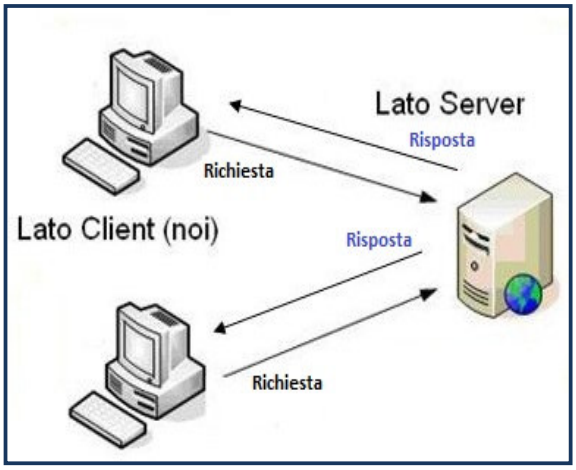
JAVASCRIPT

Introduzione

JavaScript è un *linguaggio di scripting* che viene utilizzato soprattutto nella programmazione per il web e consiste in un linguaggio formale che fornisce al **browser** determinate istruzioni da compiere.

Una pagina scritta in **HTML** è infatti *statica*, in quanto una volta che la pagina è stata interpretata dal browser la disposizione degli elementi rimane immutata, così come il suo contenuto.

Tramite il JavaScript, invece, è possibile conferire *dinamicità* (pagine *dinamiche*) alle pagine web permettendo, ad esempio di modificare i contenuti in base a input dell'utente o creare semplici applicazioni per il Web. In questi casi, quindi il contenuto della pagina non è prefissato, ma viene (parzialmente o interamente) generato in tempo reale.



Per essere più precisi, si parla in genere di linguaggi per creare pagine dinamiche ***lato client*** o ***lato server***. Proviamo a spiegare le differenze senza addentrarci in troppi particolari.

Quando si digita un indirizzo Internet di una determinata pagina accadono diversi eventi che tutti insieme rassomigliano ad una commedia con copione e protagonisti.

I protagonisti:

- **CLIENT** (noi), il computer dell'utente che richiede la pagina Web.
- Gli Internet Service Provider (ISP) che forniscono il collegamento Internet a Client e a Server
- **SERVER**, il computer sul cui disco rigido è memorizzato il sito Internet che contiene la pagina richiesta.

Il copione:

- **CLIENT** (noi) digita l'indirizzo Internet (per es.: www.trenitalia.it) sulla finestra del proprio browser
- La richiesta viene inviata al **SERVER** che contiene la pagina e durante questa fase l'ISP che fornisce il collegamento Internet al Client interpreta l'indirizzo Internet (www.trenitalia.it) e lo trasforma da alfanumerico nel corrispondente indirizzo IP numerico (193.138.162.107) per portare a destinazione la richiesta (server)
- Sul **SERVER** viene ricercata la pagina richiesta e, se esiste, viene inviata al CLIENT
- **CLIENT** interpreta tramite il browser il codice HTML e formatta sullo schermo la pagina richiesta

Nella commedia descritta abbiamo supposto di trattare solo pagine scritte in codice HTML. Se la pagina, invece, contiene anche codici diversi (ad esempio codici di scripting) essi vengono trattati nei due modi seguenti a seconda che siano:

- **LATO CLIENT**: il browser residente nel computer client interpreta i codici HTML e i codici di scripting (ad esempio JavaScript, le Applets Java, …); in questo caso la pagina viene elaborata direttamente **sul computer client.**
  In questo caso quindi, **le elaborazioni avvengono sul nostro computer.**
- **LATO SERVER**: l'interprete residente nel server interpreta i codici di scripting (ad esempio VBScript, ASP, Perl, CGI….) che elaborano i dati; i risultati dell'elaborazione vengono trasformati in codice HTML puro ed inviati al computer client che tramite il browser formatta la pagina a video.
  In questo caso quindi, **le elaborazioni avvengono sul server** e al client arriva la pagina tradotta in HTML.

- **Breve storia di JavaScript**

Nel **1995** la **Netscape Communications Corporation** decise di dotare il proprio browser **Netscape Navigator 2.0** di un linguaggio di scripting che permettesse ai web designer di interagire con i diversi oggetti della pagina (immagini, form, link, ecc.), ma soprattutto con le applet Java (programmi che permettono di interagire con l'utente).

Brendan Eich venne incaricato del progetto e inventò **LiveScript** (chiamato così ad indicare vivacità e dinamicità) che fu incluso nel browser Netscape Navigator. In quello stesso anno Netscape era particolarmente vicina all'azienda **Sun Microsystems** (ideatrice del linguaggio **Java** che rappresentava una delle tecnologie più avanzate dell'epoca), con cui aveva stretto una partnership.

Le due aziende il 4 dicembre 1995 annunciarono la nascita del linguaggio LiveScript, descrivendolo come «complementare all'HTML e a Java». La versione beta di Netscape Navigator 2.0 incorporava quindi LiveScript, ma - in omaggio a Java - Netscape decise di ribattezzare il nuovo linguaggio di scripting **JavaScript**.

La versione 2.0 del browser Netscape Navigator ebbe anche per questo un grande successo.

Cosa stava facendo intanto la **Microsoft**?

Nel **1996** la **Microsoft Corporation** rispose a JavaScript in due modi:

- con una propria versione di JavaScript, sotto molti aspetti simile all'originale, chiamata **JScript** (siamo nel luglio 1996)
- con l'introduzione di **VBScript** (che è una versione semplificata di **Visual Basic**) all'interno di Internet Explorer 3

JScript era dunque la versione di **JavaScript** supportata da Internet Explorer, ma aveva notevoli differenze con la versione sviluppata dalla Netscape.

Netscape e Sun ritennero necessario standardizzare il JavaScript e si affidano all'European Computer Manufacturers Association (**ECMA**). La standardizzazione incominciò nel novembre 1996 e fu adottata nel giugno 1997 da **ECMA** e nell'aprile 1998 da **ISO** (International Organization for Standardization, una delle più prestigiose organizzazioni internazionali che si occupano di standard) dando luogo allo standard chiamato **ECMAScript** siglato come **ECMA 262**.

Attualmente siamo alla quinta versione di ECMAScript (ECMA-262 Edition 5.1 giugno 2011). **ECMAScript** è dunque figlio di JavaScript. E oggi quando si dice JavaScript, JScript ed ECMAscript sostanzialmente si indicano tre varietà dello stesso linguaggio.

Bisogna poi tener conto che differenti browsers o versioni diverse dello stesso del browser, implementano differenti versioni di JavaScript, quindi il modo di interpretare determinati costrutti potrebbe variare da una sottoversione del browser all'altra.

Tutto questo tuttavia non ci deve minimamente preoccupare: si tratta della normale evoluzione (e crescita) di un linguaggio di scripting che si adatta alle diverse esigenze sopraggiunte dei programmatori e del commercio.

Per completezza bisogna inoltre notare, che JavaScript può anche essere utilizzato per scrivere anche delle applicazioni lato server, nel caso che il web server lo consenta. Tuttavia, ad oggi, l'utilizzo di JavaScript lato server è per lo più un caso sporadico.

- **Aspetti strutturali di JavaScript**

  Le caratteristiche principali di JavaScript sono quelle di:

  - Essere un **linguaggio di scripting**, come già detto. È quindi un linguaggio che viene utilizzato per scrivere un programma all'interno di un altro programma.
    L'idea di base è che *il programma ospite* (quello che ospita ed esegue lo script) consenta allo script l'accesso ad operazioni specifiche, le cui implementazioni sono a carico del programma ospite stesso. Il *programma ospite* per uno script JavaScript è quello del **browser**.
  - Avere le funzionalità tipiche dei **linguaggi di programmazione** ad alto livello (consente l'uso delle strutture di controllo: selezioni, cicli, la definizione di funzioni, etc.) pur non essendo un linguaggio di programmazione.
  - Essere un **linguaggio interpretato**: il codice non viene compilato, ma interpretato. In JavaScript lato client, l'interprete è incluso nel browser che si sta utilizzando. Il codice viene eseguito direttamente sul client e non sul server. Il vantaggio di questo approccio è che, anche con la presenza di script particolarmente complessi, il server non viene sovraccaricato a causa delle richieste dei clients. Di contro, nel caso di script che presentino un sorgente particolarmente grande, il tempo per lo scaricamento può diventare abbastanza lungo.
  - Essere un **linguaggio orientato agli oggetti** come i linguaggi Java e C++. (Consente di definire oggetti software in grado di interagire gli uni con gli altri attraverso lo scambio di messaggi).
  - Un **linguaggio guidato dagli eventi** come i linguaggi Java e C++. (Il flusso del programma è largamente determinato dal verificarsi di eventi esterni come ad esempio la creazione di un file in una certa cartella o la pressione di un tasto del mouse o della tastiera).
  - Essere un **linguaggio case-sensitive**, cioè fa differenza tra maiuscole e minuscole. La sintassi è molto semplice e relativamente simile a quella del C, del C++ e del Java.