

## Introduzione ad un confronto tra JavaScript e Java

JavaScript è un semplice linguaggio ad oggetti per lo sviluppo di applicazioni Internet sia lato server che client. I programmi in JavaScript possono essere infatti inseriti direttamente all'interno delle pagine HTML e venire interpretati o direttamente dal browser (client-side JavaScript) oppure da apposite estensioni del server Web (server-side JavaScript).

I programmi **client-side** JavaScript possono rispondere agli input dell'utente e svolgere tutta una serie di compiti senza che vi sia alcun traffico di dati sulla rete. Questo consente di realizzare delle pagine Web realmente interattive e veloci. Una tipica applicazione di JavaScript è controllare che i dati inseriti in una form siano corretti prima che essi vengano spediti al server.

I programmi **server-side** sono invece per molti versi simili agli script CGI: la differenza fondamentale è che i programmi non sono separati ma inseriti dentro le pagine HTML, cosa che ne facilita lo sviluppo e il mantenimento. JavaScript per server ha inoltre alcune estensioni rispetto alla versione client che consentono di gestire strutture dati persistenti tra accessi multipli della pagina e un'interfaccia a server di database SQL. Nonostante queste ed altre differenze, la struttura fondamentale del linguaggio JavaScript è la stessa per entrambe le versioni client e server.

JavaScript supporta la maggior parte delle espressioni e dei costrutti di controllo di Java ma non è fortemente tipato; anzi, la **tipizzazione** avviene completamente **a tempo di esecuzione**: una stessa variabile può contenere in istanti di tempo diversi dati di tipo diverso.

Anche il **modello ad oggetti** è profondamente diverso tra i due linguaggi: mentre quello di Java si basa sul concetto di classe ed ereditarietà, il modello ad oggetti di JavaScript è incentrato sul concetto di istanza.

Ecco riassunte le principali differenze tra i due linguaggi:

JavaScript	Java
Interpretato dal client mentre viene letta la pagina HTML. Non eseguibile fuori dal browser	Bytecode compilato scaricato dal server ed eseguito dal client. Eseguibile all'interno di qualsiasi SO: multiplatforma. Occorre solo una macchina virtuale (non è indispensabile un browser)
Basato sugli oggetti. Usa oggetti di base, supporta oggetti estendibili ma non classi ed ereditarietà. Le proprietà degli oggetti possono essere aggiunte dinamicamente.	Orientato agli oggetti. Gli applet consistono di classi di oggetti con ereditarietà. Non si possono aggiungere proprietà dinamicamente; tutti gli oggetti sono divisi in classi.
Codice integrato e immerso nel file HTML	Applet in un file .class distinto dal file HTML
Variabili senza dichiarazione di tipo	Dichiarazione esplicita dei tipi delle variabili (tipizzazione forte)
Binding dinamico. Riferimento agli oggetti controllato a tempo di esecuzione	Binding statico. I riferimenti agli oggetti devono esistere a tempo di compilazione
Non può scrivere direttamente su hard-disk	Non può scrivere direttamente su hard-disk