**Storia ed evoluzione del linguaggio**

JavaScript è stato ideato nel 1995 da Netscape. Da subito JavaScript aggiunse alle pagine HTML la possibilità di essere modificate in modo dinamico, in base all’interazione dell’utente con il browser. Questo grazie alle funzionalità di calcolo e di manipolazione dei documenti che era possibile effettuare anche senza coinvolgere il server. In sostanza è un linguaggio di programmazione composto da righe di testo che vengono inserite all’interno di pagine web e memorizzate in file con estensione .html. Quando si vuole visualizzare questo tipo di pagine, il browser restituisce il contenuto completo del documento contenente sia i tag HTML che gli scritti JS.

Gli strumenti di lavoro

Come per qualsiasi linguaggio di programmazione, per iniziare a lavorare con JavaScript c’è bisogno di almeno tre strumenti:

* un editor;
* un interprete o compilatore;
* un debugger.

Come editor è naturalmente possibile utilizzare un comune editor di testo. Quello che noi utilizzeremo prende il nome di Visual Studio Code.

Il compilatore è un programma che traduce le istruzioni scritte in un [linguaggio di programmazione](http://www.dacrema.com/Informatica/linguaggi_programmazione.htm).

Un debugger integrato consente di analizzare il codice durante l’esecuzione. In assenza di un ambiente di sviluppo integrato, possiamo comunque ricorrere a un comune browser, dal momento che i più diffusi prevedono ormai un ambiente di debugging.

Naturalmente Internet Explorer e Firefox hanno ambienti analoghi tutti e due i browser prevedono un ambiente per l’esecuzione diretta di codice JavaScript, la cosiddetta console. In questa finestra speciale possiamo scrivere codice JavaScript per vederne l’immediata esecuzione.

La stessa console può essere utilizzata come ambiente per la visualizzazione dell’output di uno script, grazie all’omonimo oggetto console.

È molto comune utilizzare nel codice il metodo console.log per tenere traccia dell’evoluzione del programma o per verificare rapidamente il valore di oggetti e variabili.

Inserire codice JavaScript in una pagina HTML

Vediamo quindi come è possibile inserire codice JavaScript in una pagina HTML. Esistono essenzialmente tre modi per farlo:

* inserire **codice inline**;
* scrivere **blocchi di codice nella pagina**;
* importare file con codice **JavaScript esterno**.

### Codice inline

Il primo approccio, l’inserimento di codice *inline*, consiste nell’inserire direttamente le istruzioni JavaScript nel codice di un elemento HTML, assegnandolo ad un attributo che rappresenta un evento. Chiariamo il concetto con un esempio:

<button type="button" onclick="alert('Ciao!')">Cliccami</button>

Abbiamo assegnato all’attributo onclick dell’elemento button la stringa alert('Ciao!'). L’attributo onclick rappresenta l’evento del clic sul pulsante del mouse, quindi in corrispondenza di questo evento verrà analizzato ed eseguito il codice JavaScript assegnato. Nel caso specifico verrà visualizzato un box con la scritta Ciao!

Un altro approccio per l’inserimento di codice inline, utilizzabile però soltanto con i link, è quello mostrato nel seguente esempio:

<a href="javascript:alert('Ciao!')">Cliccami</a>

In questo caso indichiamo al browser di interpretare il link come la richiesta di esecuzione del codice JavaScript specificato invece che come un collegamento ad un’altra pagina.

### Blocchi di codice, il tag <script>

L’approccio inline può risultare immediato perché mette direttamente in relazione il codice da eseguire con un elemento HTML. Risulta però scomodo quando il codice da eseguire è più complesso o abbiamo necessità di definire variabili e funzioni. In questi casi possiamo ricorrere al tag **<script>** per inserire blocchi di codice in una pagina HTML, come nel seguente esempio:

<script>alert('Ciao!')</script>

Quando il parser HTML (richiede una pagina Web da un server Web, proprio come si digita indirizzi Web nella barra degli indirizzi del browser. Il server invia il codice HTML al parser, che poi esegue la scansione attraverso la pagina, alla ricerca di tag e il testo. Esso controlla il file per assicurarsi che ha tag HTML e nel giusto ordine. Il parser poi organizza, visualizza e spigola informazioni dalla pagina web) del browser esamina la pagina, riconosce il tag <script> come blocco di codice e ne passa il contenuto all’ engine JavaScript presente nei browser web permette di interagire con il contenuto dei documenti web, interagire con l'utente e reagisce agli eventi che vengono generati.

Possiamo inserire blocchi di codice (e i relativi tag <script>) nella sezione <head> o nella sezione <body> della pagina HTML.

### JavaScript esterno

Il terzo approccio, quello più consigliato, consiste nel collegare alla pagina HTML, codice **JavaScript presente in un file esterno**. Questa tecnica permette di agganciare script e librerie in modo detto *non intrusivo*, con il vantaggio di una separazione netta tra la struttura del documento e il codice, come accade per i fogli di stile CSS, che separano struttura e presentazione.

Per **inserire un file JavaScript esterno** ci serviamo sempre del tag <script> in cui specificando l’attributo **src**, come mostrato dal seguente esempio:

<script src="codice.js"></script>.