

JavaScript

JavaScript è un linguaggio di scripting sviluppato per dare interattività LATO CLIENTE alle pagine HTML. Esso rende le pagine Web **attive**.

JavaScript è un linguaggio interpretato, object-based ma non class-based. E' inoltre debolmente tipizzato. Viene eseguito da un interprete contenuto all'interno del browser. Permette:

- accedere e modificare elementi della pagine HTML.
- reagire ad eventi generati dall'interazione con l'utente.
- validare i dati inseriti.
- interagire con il browser.

Viene inserito all'interno della pagina HTML tramite il tag `<script>` :



La più semplice modalità di utilizzo di JS consiste nell'inserire nel corpo della pagina **script** che generano *dinamicamente* parti della pagina HTML. Bisogna tener presente

però che questi script vengono eseguiti solo una volta durante il caricamento della pagina e quindi non si ha interattività con l'utente.

La pagina corrente è rappresentata dall'oggetto **document**. Per scrivere sulla pagina si utilizzano i metodi `document.write()` e `document.writeln()`.

Per quanto riguarda le informazioni del browser si utilizza invece l'oggetto **navigator** con i suoi metodi:

- `appName` : nome in codice del browser. (*poco utile*)
- `appName` : nome del browser
- `appVersion` : versione del browser
- `cookieEnabled` : cookies abilitati o meno
- `platform` : piattaforma per cui il browser è stato compilato
- `userAgent` : stringa passata dal browser come header user-agent

Le proprietà dello schermo si rilevano invece tramite l'oggetto **screen** che possiede proprietà come `screen.width` e `screen.height`.

Interattività

Per avere una reale interattività bisogna però utilizzare il meccanismo degli **eventi**. JS consente di associare script agli eventi causati dall'interazione dell'utente con la pagina HTML, tramite gli attributi collegati agli elementi della pagina. Questi script prendono il nome di **event handlers**.

Evento	Applicabilità	Occorrenza	Event handler
Abort	Immagini	L'utente blocca il caricamento di un'immagine	onAbort
Blur	Finestre e tutti gli elementi dei form	L'utente toglie il focus a un elemento di un form o a una finestra	onBlur
Change	Campi di immissione di testo o liste di selezione	L'utente cambia il contenuto di un elemento	onChange
Click	Tutti i tipi di bottoni e i link	L'utente 'clicca' su un bottone o un link	onClick
DragDrop	Finestre	L'utente fa il drop di un oggetto in una finestra	onDragDrop
Error	Immagini, finestre	Errore durante il caricamento	onError
Focus	Finestre e tutti gli elementi dei form	L'utente dà il focus a un elemento di un form o a una finestra	onFocus
KeyDown	Documenti, immagini, link, campi di immissione di testo	L'utente preme un tasto	onKeyDown
KeyPress	Documenti, immagini, link, campi di immissione di testo	L'utente digita un tasto (pressione + rilascio)	onKeyPress
KeyUp	Documenti, immagini, link, campi di immissione di testo	L'utente rilascia un tasto	onKeyUp

Evento	Applicabilità	Occorrenza	Event handler
Load	Corpo del documento	L'utente carica una pagina nel browser	onLoad
MouseDown	Documenti, bottoni, link	L'utente preme il bottone del mouse	onMouseDown
MouseMove	Di default nessun elemento	L'utente muove il cursore del mouse	onMouseMove
MouseOut	Mappe, link	Il cursore del mouse esce fuori da un link o da una mappa	onMouseOut
MouseOver	Link	Il cursore passa su un link	onMouseOver
MouseUp	Documenti, bottoni, link	L'utente rilascia il bottone del mouse	onMouseUp
Move	Windows	La finestra viene spostata	onMove
Reset	Form	L'utente resetta un form	onReset
Resize	Finestre	La finestra viene ridimensionata	onResize
Select	Campi di immissione di testo (input e textarea)	L'utente seleziona il campo	onSelect
Submit	Form	L'utente sottomette il form	onSubmit
Unload	Corpo del documento	L'utente esce dalla pagina	onUnload

La sintassi per agganciare un gestore di evento ad un evento è `<tag eventHandler=" JS code">`.

utile definire delle funzioni in testata, piuttosto che inserire istruzioni in sequenza

Un documento può contenere oggetti **form**, che possono essere referenziati con il loro nome o mediante il vettore `forms[]` esposto da document. Gli elementi del form possono essere referenziati con il loro nome o mediante il vettore `elements[]`.

Metodi di **form**:

- `blur()` toglie il focus al controllo
- `focus()` dà il focus al controllo
- `click()` simula il click del mouse sul controllo

Eventi di **form**:

- `onblur` quando il controllo perde focus
- `onfocus` quando il controllo prende il focus
- `onclick` quando l'utente clicca sul controllo

Validazione del form

Generalmente si valida il form in due momenti:

- durante l'inserimento utilizzando l'evento **onChange()** sui vari controlli
- al momento del submit utilizzando l'evento **onClick()** del bottone di submit o l'evento **onSubmit()** del form

Document DOM

L'oggetto document rappresenta la pagina web. Se vogliamo accedere ad un qualunque elemento in una pagina HTML, dobbiamo sempre passare attraverso questo oggetto.

Selezione elementi

La selezione elementi avviene grazie alle chiamate a `querySelector()` e `querySelectorAll()`.

```
document.querySelector(#id)           //trova il primo elemento con l'id specificato
document.querySelector(tagName)       //trova il primo elemento con il tag specificato
document.querySelectorAll(.className) //trova tutti gli elementi con la classe specificata
```

NOTA: entrambe le funzioni restituiscono una NodeList, che va iterata tramite un `forEach`

Changing HTML

Tramite la funzione `innerHTML` otteniamo facilmente il contenuto di un elemento (che, volendo, rimpiazziamo)

```
var stringa = document.querySelector("#text").innerHTML;
//stringa ora contiene il valore dell'elemento referenziato dalla querySelector
stringa.innerHTML += " " + stringa.toUpperCase();
//scrivo in append la stringa in upperCase
```

Changing CSS

Tramite la funzione `style` andiamo a cambiare il CSS dell'elemento referenziato:

```
document.querySelector("#text").style.color = "blue";  
document.querySelector("#text").style.fontSize = "30pt";
```

EventListener

Un event listener è un ascoltatore che si focalizza su una specifica azione ed esegue una parte di codice da noi specificata:

```
document.querySelector("#btn").addEventListener("click", function()  
document.querySelector("p").style.color = "red";  
});  
//premendo il bottone btn andiamo a cambiare il colore a tutti i
```

Eventi tipo:

- *mouseover*
- *mouseout*
- *click*
- *dblclick*
- *keypress*
- *focus*
- *blur*
- *submit*
- *etc...*

jQuery