



Mauro Bogliaccino

# Corso Javascript

## JavaScript nel browser

---

## BOM- Browser Object Model - Window Object

- L'oggetto **Window** è il più alto livello di oggetti JavaScript che corrispondono alla finestra del browser.
- Window fornisce funzioni per finestre di dialogo.
- BOM (Gli oggetti del browser)
- DOM (Attraversamento e manipolazione)
- Eventi (tipologia e gestione)

---

### Funzioni per dialog

- `alert(messaggio)`
- `res=confirm(messaggio)`
- `res=prompt(messaggio, valore_default)`

---

## API Window

oggetto JS | descrizione

`location`|array oggetto che rappresenta la url attuale. `history`|back - forward - go `navigator`| `screen`|fornisce informazione relativa al browser e alla risoluzione.

---

## DOM

- DOM (Document Object Model), è una interfaccia di programmazione per documenti in HTML e XML.
- Fornisce una rappresentazione strutturata del documento (in forma di una struttura di albero), e definisce un modo per accedervi,
- Inoltre può modificarne stile e contenuto.
- DOM permette un accesso alla struttura di una pagina HTML mediante la mappatura degli elementi di questa pagina in un albero di nodi.
- Ogni elemento se converte in un nodo e ogni porzione di testo in un nodo di testo.

---

## API DOM

- La manipolazione del contenuto nel browser dipende da un insieme di funzioni e attributi che son previsti per l'oggetto document, che è di tipo HTMLDocument.
- Rappresentazione di un documento HTML
- Notazione di navigazione
- Gerarchia di classi DOM
- Fonte: http://stackoverflow.com
- Maggiori informazioni: http://www.javascriptkit.com/domref/

---

## Selezionare elementi - Selettori (I)

Sono disponibili molti metodi di selezione di elementi.

- `document.getElementById(id)`
- `document.getElementsByName(name)`
- `document.getElementsByTagName(tagname)`
- `document.getElementByClassName(classname)`

---

## Selettori (II)

Addizionalmente ci sono selettori di tipo CSS.

- `document.querySelector(selector)`
- `document.querySelectorAll(selector)`

---

## Attraversare il DOM - Traversing

Una volta stabilito un punto di partenza, si può percorrere la ramificazione

- `parentNode`
- `childNodes`
- `firstChild`
- `lastChild`
- `nextSibling`
- `previousSibling`

---

## Tipi di nodi

i nodi hanno alcune proprietà che si possono consultare, per vederne le caratteristiche.

- `nodeType`
- `nodeValue`
- `nodeName`

---

## Traversing (II)

- `children`

- `firstElementChild`
- `lastElementChild`
- `nextElementSibling`
- `previousElementSibling`
- `childElementCount`

---

# Manipolare elementi del DOM

Una volta visti i metodi per selezione e introspezione, vediamo come manipolare il DOM.

---

## Attributi

Sono disponibili metodi specifici per l'accesso agli attributi di un elemento.

- `getAttribute()`
- `setAttribute()`
- `hasAttribute()`
- `removeAttribute()`

---

## Contenuti (I)

Sono disponibili metodi per la manipolazione del contenuto.

- `innerHTML`
- `outerHTML`
- `insertAdjacentHTML()`
- `beforebegin()`
- `afterbegin()`
- `beforeend()`
- `afterend()`

---

## Contenuti (II)

per accedere al contenuto di un elemento, esistono due metodi

- textContent, per HTMLElement
- data, per Node

---

## Creazione di nodi

Sono disponibili tre metodi per la creazione di nuovi nodi.

- `createElement()`
- `createTextNode()`
- `cloneNode()`

## Manipolazione di elementi

Ci sono metodi per la manipolazione di nodi dell'albero.

- `appendChild()`
- `insertBefore()`
- `removeChild()`
- `replaceChild()`

---

## Posizionamento (I)

Sono disponibili vari metodi utili per conoscere le posizioni degli elementi, E le loro dimensioni.

- `w.pageXOffset`
- `w.pageYOffset`
- `w.innerWidth`
- `w.innerHeight`
- `getBoundingClientRect()`
- `document.elementFromPoint(x,y)`
- `scrollTo(x,y)`
- `scrollBy(x,y)`
- `scrollIntoView(x,y)`

---

## Gestione dei Form HTML

- I moduli html si possono accedere come qualsiasi altro elemento
- Inoltre esiste il parametro `document.forms` che permette di accedere attraverso gli attributi name e id
- Per accedere agli elementi del form, si utilizza la variabile 'elements'.
- `document.forms.name.element.input`

---

## Stili

- Per la modifica di proprietà CSS, si usa la propietà 'style'.
- Inoltre è possibile strutturare un documento in modo che scambiarsi delle classi.
- Questi riflettono cambi nella presentazione di documenti.
- `document.querySelector('selector').style`

---

## Eventi

Un evento è qualcosa che scatena una azione specifica nel browser.

---

## Un evento accade quando:

- Termina il caricamento di un elemento della pagina.

- L'utente esegue azioni col mouse o la tastiera.
- Eventi associati ad una pagina
    - Tastiera, mouse, dispositivi Touch, click
    - Device-independent input events
    - User interface events: focus, change, submit
    - State-change events, Cambio di stato in generale
    - API-specific events
    - DnD, Players
    - Timers and error handlers: temporizzatori, errori

---

## Cattura di eventi

Sono disponibili due metodi per registrare eventi.

```
b.onclick=function(){console.log('asdf')};
```

```
b.addEventListener('click',function(){
    console.log('asd')
},false);
```

---

## Tipologia di Eventi Javascript

| EVENTO | DESCRIZIONE |
|---|---|
| `blur`, `focus` | Inviati ad un elemento quando rispettivamente perde il focus od ottiene il focus. |
| `focusin`, `focusout` | Inviato ad un elemento se esso o un suo discendente rispettivamente ottiene o perde il focus |
| `load` | Inviato ad un elemento quando esso e tutti i suo discendenti sono stati completamente caricati |
| `resize` | Inviato all'elemento windows quando la finestra del browser ha cambiato dimensioni |
| `scroll` | Inviato ad un elemento quando l'utente ha effettuato lo scroll in un differente punto dell'elemento stesso |
| `unload` | Inviato all'oggetto window quando l'utente naviga fuori dalla pagina |
| `click`, `dbclick` | Inviati ad un elemento quando il mouse è sopra di esso e viene effettuato un click o un doppio click |
| `mousedown`, `mouseup` | Inviati ad un elemento quando il mouse è sopra di esso e viene rispettivamente premuto o rilasciato il bottone del mouse |

| EVENTO | DESCRIZIONE |
| --- | --- |
| `mouseover`, `mouseout`, `mousemove` | Eventi inviati all'elemento in cui il puntatore del mouse entra (mouseover), esce (mouseout), si sta mouvendo (mousemove) |

## Tipologia di Eventi Javascript

| EVENTO | DESCRIZIONE |
| --- | --- |
| `keydown`,`keypress`, `keyup` | Eventi inviati quando un tasto viene premuto (keydown) è stato rilasciato (keyup) o è stato premuto (keypress) |
| `select` | Inviato ad un elemento quando viene selezionato del test all'interno di esso. |
| `change` | Evento inviato ad un elemento che ha cambiato il proprio valore. |
| `submit` | Evento inviato quando l'utente tenta di fare il submit di un form |

Maggiori informazioni: