

Tecnologie Web: Javascript & DOM

Prof. Raffaele Montella, PhD

raffaele.montella@uniparthenope.it

Sommario

- Introduzione
- Tipologie di oggetti HTML5
- Gestione degli eventi
- Oggetto Window
- Oggetto History
- Oggetto Navigator
- Interfaccia Element
- Oggetto document
- Conclusioni

Introduzione

- JavaScript è un linguaggio di scripting client side
- Permette di creare pagine web attive
- E' stato introdotto con Netscape Navigator 2.0 nel 1995
- E' indipendente dalla piattaforma
- L'interprete JavaScript gira all'interno del browser

Introduzione

- E' un linguaggio di programmazione orientato agli oggetti
- E' basato sui prototipi
 - Non esiste una distinzione fra classi ed istanze: solo oggetti
 - Si basa sul concetto di prototipo: un oggetto funge da prototipo da cui prendere le proprietà iniziali di un nuovo oggetto
 - Qualsiasi oggetto può specificare le suo proprietà alla creazione e al runtime
- Non è correlato in alcun modo a Java!

Tipologie di oggetti HTML5

- Nativi
 - Messi a disposizione dal linguaggio
- Appartenenti al BOM (Browser Object Model)
 - Forniti dall'ambiente del browser
 - Permettono di interagire con quest'ultimo
- Appartenenti al DOM (Document Object Model)
 - Operano sugli elementi presenti all'interno della pagina visualizzata dal browser.
 - Gli oggetti appartenenti al DOM appaiono integrati nel BOM

Gestione degli eventi

- Il motore di JavaScript invoca i metodi degli oggetti del DOM in relazione all'interazione che l'utente ha con la pagina o con componenti di essa
- Con JavaScript è possibile ridefinire il comportamento di default degli oggetti del DOM in modo da programmarne le funzionalità

Gestione degli eventi

- Tipi di evento:
 - Documento: apertura e chiusura di pagine web
 - Link: interazioni ipertestuali
 - Form:
 - Eventi dovuti alla pressione di bottoni, caselle di controllo, bottoni di invio e di reset
 - Eventi di disattivazione, attivazione e modifica di campi di testo, aree di testo e liste di selezione
 - Selezione in campi ed aree di testo
 - Caricamento immagini
 - Ingresso o uscita del cursore dal box di un elemento

Eventi

- Alcuni eventi gestiti dal motore JavaScript

Evento	Attributo	Lancio
Blur	onBlur	L'input di un elemento di un form non è più evidenziato
Change	onChange	Cambia il valore di un campi di testo, di un'area di testo o di un elemento di una lista di selezione
Click	onClick	Si fa click su di un elemento
Focus	onFocus	L'input di un elemento di un form viene evidenziato
Load	onLoad	Un documento viene caricato correttamente
Mouse over	onMouseOver	Il puntatore del mouse entra nel box di un elemento
Mouse out	onMouseOut	Il puntatore del mouse esce dal box di un elemento
Select	onSelect	Si seleziona un campo di input di un form
Submit	onSubmit	Si invia un form
Unload	onUnLoad	Viene chiuso il documento corrente, oppure viene caricato un nuovo documento

Oggetto Window

- L'oggetto window è un oggetto del BOM e rappresenta la finestra del browser
- E' caratterizzato da un insieme di proprietà, metodi e gestori di eventi
- E' globale
- Non è necessario usare il nome per accedere a proprietà e metodi

Oggetto Window

- Principali proprietà
 - `defaultStatus`:
stringa contenente il messaggio predefinito visualizzato nella barra di stato della finestra
 - `frames`:
array contenente i frame associati alla finestra
 - `parent`:
fa riferimento alla finestra o al frame il cui frameset contiene il frame corrente
 - `self`: la finestra corrente
 - `status`: messaggio di stato della finestra corrente

Oggetto Window

- Principali metodi
 - `alert()`: crea una finestra di dialogo contenente un messaggio ed il pulsante OK
 - `close()`: chiude la finestra corrente
 - `confirm()`: è simile ad `alert()` ma visualizza i pulsanti OK e Cancel. Se si seleziona OK restituisce `true`, altrimenti `false`
 - `Open(url [, titolo, attributi])`: apre una nuova finestra

Oggetto Window

- Principali eventi
 - onLoad: intercetta il caricamento della finestra
 - onUnload: intercetta la chiusura della finestra
- Esempio:

```
...  
<body onLoad="alert( 'Benvenuto !' );"  
onUnload="alert( 'Arrivederci' );" >  
...  
</body>
```

Oggetto History

- Fornisce la possibilità di accedere allo storico delle pagine visitate
- Possiede una sola proprietà che fornisce il numero delle pagine dello storico (length)
- Implementa i metodi: back(), forward(), go(int)

- Esempi:

```
<a href="#" onClick="history.back()">Indietro</a>
```

```
<a href="#" onClick="history.forward()">Avanti</a>
```

```
<a href="#" onClick="go(-2)">Indietro di 2</a>
```

Oggetto Navigator

- E' un oggetto BOM che rappresenta il browser
- E' accessibile come proprietà dell'oggetto Window
- Consente di accedere alla configurazione del browser e del sistema operativo
- Proprietà
 - appName: nome del browser
 - appVersion: versione del browser
 - mimeTypes: array con i tipi di MIME supportati
 - oscpu: stringa che identifica processore e CPU
 - plugins: array con i plug-in installati nel browser
 - userAgent: header HTTP dell'agente browser

Interfaccia Element

- Ogni elemento di un documento HTML5 implementa l'interfaccia element del DOM
- Offre proprietà e metodi definiti in Element
- Con DOM e JavaScript è possibile modificare il documento
- In questo modo la pagina può essere attiva in maniera asincrona (AJAX)

Interfaccia Element

- Metodi:
 - `e.getElementsByTagName("tag")`: restituisce un array di tutti gli elementi di tipo tag nel sottoalbero XHTML avente come radice l'elemento e
`aTags = el.getElementsByTagName("a");`
 - `attr = e.getAttribute(attrName)`: legge il valore di un attributo dell'elemento e
 - `e.setAttribute(attrName, attrValue)`: imposta il valore dell'attributo dell'elemento

Interfaccia Element

- Metodi:
 - `var newElement = el.appendChild(child)`: aggiunge un nuovo nodo figlio (dopo tutti gli altri)
- Esempio: Crea un nuovo paragrafo e lo aggiunge in coda

```
var p=document.createElement("p");  
el.appendChild(p)
```
- Metodi correlati: `insertBefore`, `replaceChild`, `removeChild`

Interfaccia Element

- Attributi:
 - e.innerHTML: permette di leggere o scrivere l'intero contenuto XHTML del nodo

```
var d = document.getElementById("d");
dump(d.innerHTML);
document.body.innerHTML="";
```
 - e.style: restituisce un oggetto che rappresenta le proprietà di stile CSS associate all'elemento

```
e.style.marginTop="20px";
```

 - Per ogni attributo CSS ha una proprietà con lo stesso nome
 - In questo modo è possibile implementare GUI sofisticate

Oggetto document

- L'oggetto document è l'oggetto DOM che rappresenta l'intero documento HTML5 caricato in una finestra del browser
- E' un oggetto accessibile globalmente
- document rappresenta il nodo body del documento XHTML
`anchorTags = document.getElementsByTagName("a");`
- Fornisce un insieme aggiuntivo di attributi e metodi utili

Oggetto document

- A partire dalla versione DOM sono disponibili array che contengono tutti gli elementi di un certo tipo che compaiono nel documento (mantenendo l'ordine)
 - applets
 - anchors
 - embeds
 - forms
 - images
 - links
- Esempio
`document.forms[0].TestoPrimoForm`
`document.forms["PrimoForm"].TestoPrimoForm`
- `document.body`: oggetto che rappresenta il body del documento XHTML
 - E' manipolabile come ogni altro nodo tramite l'interfaccia element

Oggetto document

- Altri attributi

- cookie

Legge/Scrive cookie

- lastModified

Data ultima modifica

- location

URI del documento

- referrer

URL di provenienza

- title:

Titolo del documento

Oggetto document

- Metodi:
 - `document.getElementById("id")`: restituisce un elemento di un documento XHTML in base al suo id
`menu = document.getElementById("MenuLaterale")`
 - `document.write("text")`: inserisce una stringa di testo nel documento nel punto in cui si trova la chiamata
 - `document.createElement("tag")`: crea un nuovo elemento da associare ad un nodo preesistente

Oggetto document

- La modifica di un documento HTML5 è alla base dello sviluppo di pagine web attive
- La modifica diretta del testo vanifica la verifica del documento HTML5
- Il W3C scoraggia l'uso di questo approccio
- Il W3C incoraggia l'interazione attraverso il DOM

Conclusioni

- JavaScript è utilizzato in differenti contesti
- Molto spesso come “lingua franca” dei linguaggi di scripting
- È adoperato spesso insieme a librerie come jQuery
- Funziona in ambito Web Client, Web Server, IoT & Mobile.