Oggetti in Javascript

Un oggetto in JavaScript è una **collezione di valori**[1] con nome (*named values*); si fa riferimento a una **proprietà** di un oggetto con uso di "*dot notation*": il nome dell'oggetto seguito dal punto (.) e dal nome della proprietà. Ad esempio immagine.altezza oppure immagine.larghezza

Se un oggetto contiene una funzione, allora la funzione prende il nome di **metodo** dell'oggetto ed il nome della proprietà diventa il nome del metodo. Una variabile è fondamentalmente la stessa cosa di una proprietà di un oggetto

Le principali famiglie di oggetti in Javascript:

- ***"Riflessi" da HTML*** (creati automaticamente dal browser per ogni ***elemento*** di un documento HTML)
- ***Interni*** o integrati (non legati al browser né al documento HTML: Array, String, Date, Math)
- ***Browser-dipendenti***[2] (detti oggetti Navigator, "riflessi" dall'ambiente del browser)
- ***Definiti dall'utente*** richiedono:
  - prima la definizione di una ***classe*** cioè una famiglia di oggetti con le stesse caratteristiche o **attributi** e comportamenti o **metodi** tra i quali il **costruttore** che servirà ad inizializzare i valori degli attributi
  - poi la creazione (*istanza* della classe con occupazione di memoria) con operatore **new**

L'oggetto globale

Quando l'interprete JavaScript va in esecuzione una delle prime cose che fa è creare un ***oggetto globale***.Le proprietà di questo oggetto globale sono le variabili dichiarate all'interno di programmi JavaScript.

Quando si dichiara una variabile (**var**) si sta definendo una proprietà dell'oggetto globale; l'interprete JavaScript inizializza questo oggetto con altri valori e funzioni predefinite.

Per far riferimento a questo oggetto in codice JavaScript che non fa parte di funzioni possiamo usare la parola chiave ***this***. All'interno di funzioni, this ha un significato differente (è un riferimento all'oggetto corrente).

- In JavaScript **lato client** l'oggetto ***Window*** serve da oggetto globale: è il padre della gerarchia che contiene ***document*** (che contiene oggetti "riflessi" da HTML quali ***link***, … ***image*** e ***form*** che è contenitore di ***elements*** quali text, button etc…). Ogni discendente è una proprietà dell'antenato.
  - Per far riferimento a ***this*** – inteso come oggetto globale - possiamo usare la *proprietà* auto-referenziale *window* dell'oggetto globale Window

Fra gli utilizzi maggiori di JavaScript ci sono il *controllo della visualizzazione* di pagine HTML, la *creazione dell'interfaccia* per l'inserimento dei dati e la *verifica delle scelte* fatte dell'utente.
Esistono quindi diversi oggetti, metodi e proprietà per creare tali script. La prima necessità per utilizzare facilmente gli oggetti in qualsiasi linguaggio, è quella di capirne la struttura, in modo da aver ben chiari i vari livelli gerarchici e i legami tra i diversi livelli.

---

[1] Una specie di struttura in C: insieme di elementi eterogenei.

[2] Oggetti "di livello massimo" nella gerarchia degli oggetti JavaScript: Window, document, location, navigator e history

Oggetti JavaScript

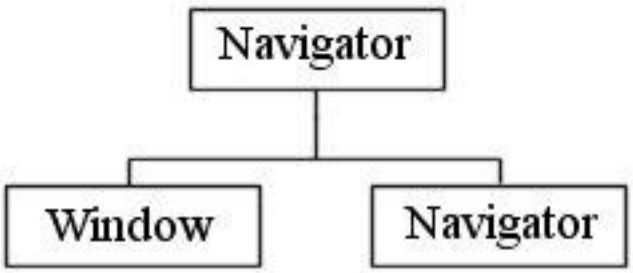
Quando programmate con JavaScript dovete immaginare che la pagina HTML sia formata da vari elementi in varia relazione fra loro. Il browser infatti (con all'interno il documento HTML) può essere infatti "sezionato" in vari elementi:

- prima di tutto c'è il **browser** stesso (l'**oggetto navigator**)
- poi la **finestra** che contiene tutto quanto (l'**oggetto window**)
- eventuali frames (l'oggetto window.frames)
- il documento HTML vero e proprio (document)
- i moduli per raccogliere l'input dell'utente (document.forms["nomeForm"])
- le immagini (document.images["nomeImmagine"])
- i cookie (document.cookie["nomeCookie"])
- i livelli
- le applet (document.applets["nomeApplet"])
- la barra degli indirizzi (location)
- la barra di stato, nella parte bassa del browser (status) e via di seguito.

Tutti gli oggetti che vediamo nel browser sono in relazione gerarchica fra di loro (ci sono elementi-padre ed elementi-figli) e tramite JavaScript - utilizzando la corretta sintassi - è possibile interrogare questi elementi, leggerne le proprietà e in taluni casi anche cambiare il valore di queste proprietà.

Una pagina, anche vuota, possiede almeno due oggetti fondamentali:

- **Navigator**: contiene proprietà per il nome e la versione di Navigator (browser) utilizzata, per i tipi MIME supportati ed i plug-in installati;
- **Window:** è la finestra del browser con una serie di proprietà e metodi che si applicano alla intera finestra; è l'oggetto di più alto livello per ogni finestra in un documento a frame



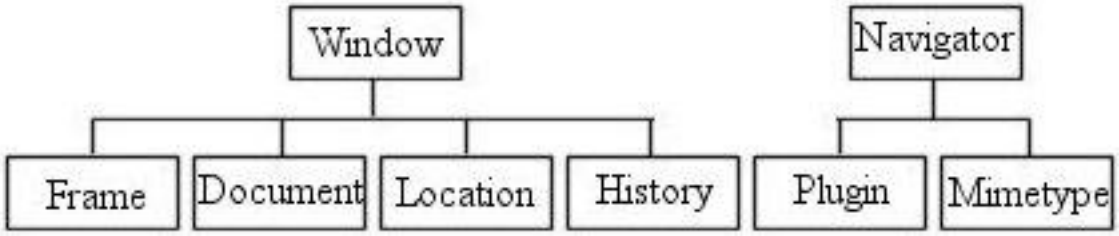
Alcuni di questi oggetti possono contenerne altri che a loro volta possono contenerne altri, sempre in forma gerarchica dal padre al figlio.

Ogni pagina può contenere i seguenti oggetti:

- *document*: contiene proprietà basate su valori globali del documento come titolo, colore di background, collegamenti e forms.
- *location*: contiene proprietà basate sull'URL corrente;
- *history*: contiene proprietà circa gli URL che il browser ha richiesto in precedenza.

A seconda del contenuto, poi, un documento potrà avere altre proprietà, ad esempio tanti oggetti di tipo *form* per ogni corrispondente form nella pagina.

Lo schema gerarchico è il seguente:



Come si può osservare, la struttura degli oggetti di JavaScript, per costruire una pagina Web, parte dai due oggetti fondamentali per poi strutturarsi a più livelli.

L'oggetto Navigator è il più semplice avendo una struttura con solo due oggetti figli che sono rispettivamente l'oggetto Plugin e MimeType.

La struttura si complica per l'oggetto window anche perchè alcuni suoi oggetti figli possono contenere oggetti di tipo padre.

Un esempio sono i frame che sono simili alle finestre e possono a loro volta contenere altri frame e cosi via, fino a ottenere una struttura molto complessa.

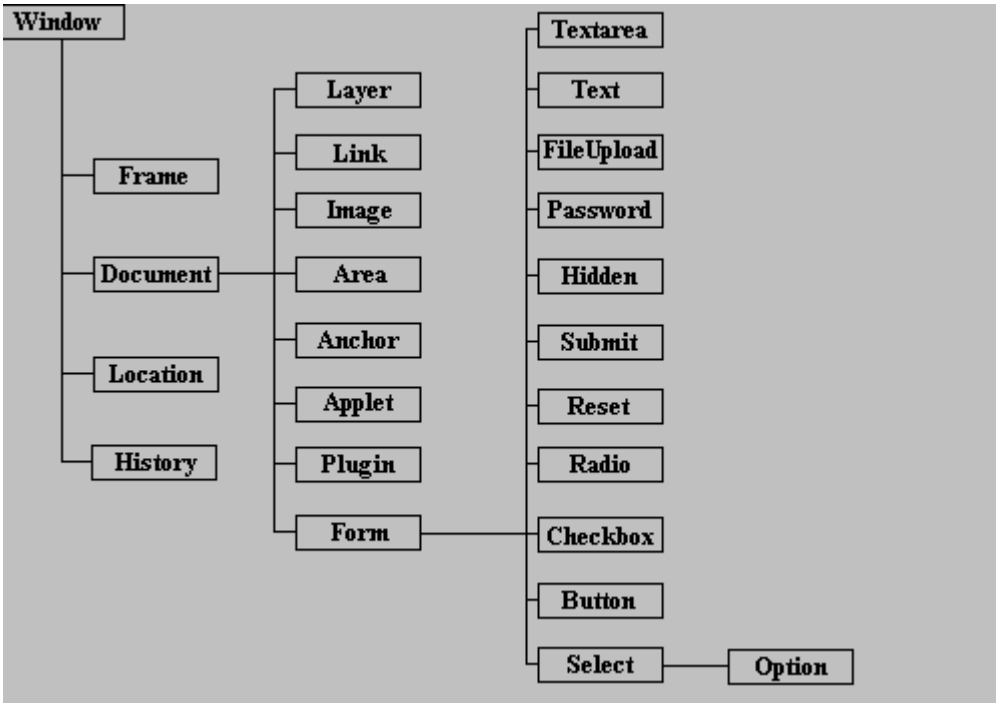
La creazione di script passa attraverso la conoscenza di questa ***struttura gerarchica*** perchè un oggetto fa riferimento a quello di livello precedente; ad esempio, un pulsante sarà gerarchicamente così definito:

window.document.form.button

anche se spesso si possono tralasciare i primi due oggetti: si possono sottintendere i primi due oggetti quando sono univocamente definiti nella struttura dello script.

Alcuni oggetti sono realizzabili utilizzando semplici tag HTML, ma per un uso più approfondito e versatile bisogna utilizzare comandi JavaScript; ad esempio, per il caricamento dei frame è possibile utilizzare i tag ma si può caricare un solo frame per volta, mentre con le istruzioni è possibile cambiare tutti i frame presenti nella pagina.

La struttura completa dell'**oggetto window**:



Oggetti di tipo frame e window

L'oggetto window è il padre di tutti gli oggetti del browser. È possibile creare finestre multiple anche sfruttando i comandi di JavaScript. Gli oggetti di tipo frame, definiti dai tag FRAME, hanno le stesse proprietà di quelli di tipo window e differiscono solo nel modo in cui sono visualizzati.

Gli **oggetti window** hanno numerosi ***metodi*** utili tra cui:

- ***open*** e ***close***: apre e chiude una finestra. Si può specificare la dimensione della finestra, il suo contenuto e la presenza di vari attributi quali la barra dei pulsanti.
- ***alert***: visualizza una finestra di avvertimento con un messaggio specificato dall'utente.
- ***confirm***: visualizza una finestra di conferma con i pulsanti OK e CANCEL.
- ***prompt***: visualizza un dialog box con un campo per l'inserimento di un valore.
- ***blur*** e ***focus***: rimuove il fuoco o da il fuoco ad una finestra.
- ***scroll***: sposta una finestra alla coordinate specificate.
- ***setTimeout***: valuta una espressione dopo un tempo specificato.

Due utili *proprietà* sono invece *location* e *status*. La prima serve a ridirigere il browser verso un altro URL mentre la seconda consente di scrivere sulla barra di stato che è presente nella maggioranza dei browser.

Oggetti di tipo document

L'oggetto document è uno dei più importanti in quanto dispone dei ***metodi write*** e ***writeln*** che consentono di scrivere sulla pagina HTML. Ogni pagina ha un solo oggetto document che contiene tutta una serie di *proprietà* riguardanti i colori del background, del testo, dei link: *bgColor*, *fgColor*, *linkColor*. La proprietà *lastModified* contiene la data dell'ultima modifica alla pagina, *referrer* l'URL precedentemente visitato e *URL* contiene l'URL del documento.

Oggetti di tipo Form

Ogni oggetto FORM nel documento crea un oggetto di tipo Form. Oltre che con il nome, è possibile accedere alle form usando l'array *forms*, che è una proprietà dell'oggetto document. Esso contiene in ordine tutte le form presenti nella pagina, dalla prima (document.forms[0]) all'ultima. Allo stesso modo i vari elementi di una form come campi testi, radio button, ecc., possono essere riferiti utilizzando il campo array *elements* della form a cui appartengono. Ad esempio, document.forms[0].elements[0] rappresenta il primo oggetto della prima form di un documento. Ogni elemento di una form ha poi una proprietà *form* che è un riferimento all'oggetto padre.

Oggetto location

L'oggetto location ha proprietà basate sull'URL corrente. Per esempio *hostname* è il nome (comprensivo di dominio) del server che contiene la pagina HTML

- ***reload***: forza il ricarimento della pagina nella finestra corrente
- ***replace***: mette l'URL specificata sopra la posizione corrente della history.

Oggetto history

L'oggetto history contiene una lista di stringhe rappresentante gli URL che il browser ha visitato. È possibile accedere all'URL corrente, successivo e precedente tramite le proprietà *current*, *next* e *previous*. Gli altri valori sono accessibili tramite l'array history. È possibile inoltre dirigere il browser verso una precisa entry dell'history tramite il metodo ***go***. Ad esempio history.go(-2) carica la pagina che è due posizioni indietro nella history mentre history.go(0) ricarica la pagina corrente.

Oggetto navigator

L'oggetto navigator contiene informazioni circa la versione di Navigator in uso. Per esempio la proprietà *appName* specifica il nome del browser mentre *appVersion* contiene la sua versione. Il ***metodo*** javaEnabled consente invece di sapere se Java è abilitato.

Per approfondire:

oggetto Navigatore – Navigator
oggetto Navigatore – Document
oggetto Navigatore – Location
oggetto Navigatore – History
oggetto Navigatore – Form