

Fondamenti di Java Script parte due

La parola chiave che noi usiamo per dire a Java Script che stiamo lavorando con una variabile é var (che possiamo anche omettere ma che dal punto della scrittura del codice é un toccasana!). Una variabile contiene un valore ad esempio var x = 100; cioè alla variabile X abbiamo ASSEGNATO the value 100. Occorre fare sempre attenzione al modo in cui le variabili vengono scritte, dire lastName non é la stessa cosa che scrivere lastname. Assegnare un valore in JS si fa tramite il segno = detto anche operatore di assegnazione, ad esempio x=y. Per stampare a video il valore di una variabile possiamo usare il metodo document.write, figlio dell' oggetto DOCUMENT del DOM ad esempio:

```
var x = 8;  
document.write(x);
```

Se tu hai il simbolo x sparso in tutta la pagina, cambiando il valore della variabile automaticamente cambierai tutti i simboli convenzionali corrispondenti.

```
var x = 8;  
document.write(X);
```

per tornare al tema della trascrizione case sensitive, nell' esempio riportato il metodo write NON stamperà il valore numerico intero otto.

il nome della variabile:

- **non deve coincidere con una delle parole chiave del linguaggio;**
- non può iniziare con un numero;
- non può contenere uno spazio;
- non può contenere caratteri speciali come, ad esempio, il punto (.) ed il trattino (-).

abstract	do	if	package	throw
boolean	double	implements	private	throws
break	else	import	protected	transient
byte	extends	in	public	true
case	false	instanceof	return	try
catch	final	int	short	var*
char	finally	interface	static	void
class	float	long	super	while
const*	for	native	switch	with
continue	function	new	synchronized	
default	goto	null	this	

Se si lavora con variabili numeriche, oltre all'uguale è possibile usare anche altri operatori di assegnazione, e cioè:

`+=`

E' un operatore incrementale e viene usato così:

`x += y;`

Questa sintassi equivale a:

`x = x + y;`

NOTA: Questo operatore può essere utilizzato anche per le variabili stringa che, ovviamente, non vengono sommate ma concatenate.

`-=`

E' un operatore decrementale e viene usato così:

`x -= y;`

Questa sintassi equivale a:

`x = x - y;`

`*=`

E' un operatore di moltiplicazione e viene usato così:

`x *= y;`

Questa sintassi equivale a:

`x = x * y;`

`/=`

E' un operatore divisionale e viene usato così:

`x /= y;`

Questa sintassi equivale a:

`x = x / y;`

`%=`

E' un operatore modulare e viene usato così:

`x %= y;`

Questa sintassi equivale a:

`x = x % y;`

Un piccolo riassunto schematico sugli operatori di assegnazione:

Operatore	Esempio	Significato
<code>+=</code>	<code>a += b</code>	<code>a = a + b</code>
<code>-=</code>	<code>a -= b</code>	<code>a = a - b</code>
<code>*=</code>	<code>a *= b</code>	<code>a = a * b</code>
<code>/=</code>	<code>a /= b</code>	<code>a = a / b</code>
<code>%=</code>	<code>a %= b</code>	<code>a = a % b</code>
<code><<=</code>	<code>a <<= b</code>	<code>a = a << b</code>
<code>>>=</code>	<code>a >>= b</code>	<code>a = a >> b</code>
<code>>>>=</code>	<code>a >>>= b</code>	<code>a = a >>> b</code>
<code>&=</code>	<code>a &= b</code>	<code>a = a & b</code>
<code>^=</code>	<code>a ^= b</code>	<code>a = a ^ b</code>
<code>! =</code>	<code>a ! = b</code>	<code>a = a ! b</code>

I caratteri di escape per i caratteri di stringa sono contrassegnati dal backslash `\` e servono a includere magari le virgolette in una istruzione marcata dalla stessa convenzione.

Sequenza	Descrizione	Designazione standard
<code>\'</code>	Virgolette singole: per rappresentare questo carattere che altrimenti è letto come delimitatore di stringa	'
<code>\"</code>	Virgolette doppie: per rappresentare questo carattere che altrimenti è letto come delimitatore di stringa	"
<code>\&</code>	La 'E' commerciale: per rappresentare il carattere '&' altrimenti usato come operatore logico	"
<code>\b</code>	Backspace	BS
<code>\n</code>	Nuova riga (tipicamente usato per andare a capo)	NL (LF)
<code>\r</code>	Ritorno carrello (meno usato, in genere è accoppiato con n)	CR
<code>\t</code>	Tab orizzontale	HT
<code>\f</code>	Avanzamento pagina	FF
<code>\\</code>	Backslash: serve per scrivere il carattere backslash altrimenti interpretato come sequenza di escape	

Le variabili in Javascript possono assumere differenti valori:

stringa di testo;
valore numerico (intero o decimale);
valore booleano (true/false);
valore NULL;

Tipi di Dato

Javascript gestisce quattro diversi tipi di dato:

- **Numeri**
 - Non c'è distinzione tra interi e reali.
 - Sono costanti numeriche tutte le espressioni che rappresentano un numero valido, con e senza virgola.
 - (i) Le funzioni *parseInt* e *parseFloat* possono essere usate per convertire stringhe in numeri.
- **Booleani**
 - Le costanti booeane sono *true* e *false*.
- **Stringhe**
 - Le stringhe sono particolari oggetti Javascript. Possono essere create implicitamente, attraverso una costante di tipo stringa, o esplicitamente tramite il costruttore dell'oggetto *String*.
 - Sono costanti di tipo stringa i valori racchiusi tra virgolette (singole o doppie).
- **Oggetti**
 - Gli oggetti sono un tipo di dato molto comune in javascript.
 - Le variabili di tipo oggetto contengono in effetti dei *riferimenti* ad oggetti. Più variabili possono quindi fare riferimento allo *stesso* oggetto.
- **Null**
 - Il tipo nullo ha un unico valore, null.

Variabili locali e variabili globali

Le variabili possono essere locali o globali, le prime servono a dichiarare un valore che verrà conservato in memoria per l'utilizzo della funzione o della routine in cui viene dichiarata, le seconde, al contrario, vengono dichiarate all'inizio di un blocco di script e possono essere utilizzate da una molteplicità di funzioni e routine.

Vediamo un esempio di variabile locale:

```
function mia_funzione() {  
  var miaVariabile = 123;  
  alert(miaVariabile);  
}
```

In questo caso la nostra variabile *miaVariabile* esiste solo all'interno della funzione.

Vediamo adesso un esempio di variabile globale:

```
var miaVariabile = 123;  
  
function mia_funzione() {  
  alert(miaVariabile);  
}
```

In questo caso la variabile miaVariabile sarà accessibile ovunque ed è quindi definita "globale".

Operatori di confronto:

Operatore	Esempio	Significato
==	x==y	verifica se x è uguale a y, è diverso da x=y
!=	x!=y	verifica se x è diverso da y
<	x<y	verifica se x è minore di y
>	x>y	verifica se x è maggiore di y
<=	x<=y	verifica se x è minore o uguale di y
>=	x>=y	verifica se x è maggiore o uguale di y
? :	x=(y<0)? -y:y	come un ciclo if. La riga chiede: y è minore di zero? se sì x=-y altrimenti x=y

Fate attenzione a non confondere x==y con x=y.
== è un operatore di confronto.
= è un operatore di assegnazione.

Per dire il valore di X coincide perfettamente con il valore di Y come tipo di dato e valore si usa il simbolo === (tre uguali).

OPERATORI LOGICI

Gli operatori logici sono conosciuti anche come operatori booleani e possono ritornare i valori vero o falso

Operatori logici:

Operatore	Significato	Esempio
&&	And	if (x==y && a!=b)
::	Or	if (x==y :: a!=b)
!	Not	if (!x)