

Il linguaggio Javascript

Antonio Lioy
<lioy@polito.it >

Politecnico di Torino
Dip. Automatica e Informatica

Javascript

- linguaggio di script pensato per il web (ma ora adottato anche in altri ambienti, es. Adobe PDF)
- eseguibile lato client o lato server
- introdotto in pagine HTML tramite:
 - il tag <script> nell'head o nel body
 - istruzioni specificate per un event-handler
- eseguito nel momento in cui il browser, leggendo HTML, incontra il codice Javascript (sospensione temporanea dell'azione di interprete HTML)

Il tag <script>

- due parametri principali:
 - type=text/javascript
 - src=URI (lo script è presente alla URI indicata)
- esempi:

```
<script type="text/javascript">
alert("CIAO!");
</script>
```

```
<script
  type="text/javascript"
  src="esempio.js">
</script>
```

esempio.js

```
alert("CIAO!");
```

Sintassi

- Javascript è case-sensitive
- ogni istruzione è terminata da punto-e-virgola ;
- tipi di dati e corrispondenti valori base:
 - numeri decimali (es. 14, -7, 3.14, 10.7e-4), ottali (es. 016) ed esadecimali (es. 0xE)
 - valori Booleani (true false)
 - stringhe di caratteri, racchiuse tra apici doppi o singoli (es. "ciao mamma" 'ciao babbo')
 - oggetti
 - il valore speciale null (variabile non inizializzata)
 - il valore speciale undefined (variabile non definita)
 - il valore speciale NaN (Not A Number)

Commenti

- stile C++: da "//" sino a fine riga
- stile C: tutto il testo compreso tra "/*" e "*/"
- esempi:

```
<script type="text/javascript">
// commento di una sola riga in stile C++
/*
commento che occupa quattro righe
in stile C
*/
</script>
```

Variabili

- identificate tramite il loro nome:
 - deve iniziare con un carattere alfabetico, \$ o _
 - può poi contenere caratteri alfanumerici, \$ e _
 - es. costo, \$1, studente_12345, _2009q1
- non sono tipate (al contrario di Java, C, ...) ma:
 - prendono un tipo al momento della loro inizializzazione
 - possono cambiare tipo (automaticamente) per adattarsi al contesto in cui vengono usate

Creazione di una variabile

- **creazione esplicita tramite l'istruzione "var" (con o senza un valore iniziale):**
 - var totale;
 - var totale = 0;
 - var saluto = "ciao mamma!"
- **creazione implicita assegnando un valore alla variabile:**
 - totale = 0;
- **se si usa una variabile senza averle prima assegnato un valore:**
 - undefined / NaN (se dichiarata con "var")
 - runtime error (se non dichiarata)

Costanti

- **tramite l'istruzione "const" si può dichiarare una variabile con valore fisso:**
 - const iva = 0.19;
 - const autore = "A.Lioy";
- **all'interno di una stringa si possono usare:**
 - caratteri ISO-8859-1
 - caratteri Unicode
 - sequenze di escape:
 - \r \n \t \' \" \\
 - \nnn(ottale) \xNN (hex) \uNNNN (Unicode hex)
 - attenzione all'uso (se stringa usata in HTML allora può richiedere codifica, es. ` ;)

Input ed output

- **JS è un linguaggio di scripting, pensato per essere eseguito lato client (in un browser) o lato server**
- **le funzioni di I/O dipendono dall'ambiente di esecuzione**
- **per il lato client si possono usare:**
 - per l'input
 - i dati provenienti da una finestra pop-up di input
 - i dati provenienti da un form (tramite DOM)
 - per l'output
 - un finestra pop-up di output
 - la pagina HTML (mentre viene "costruita" o tramite DOM)

Pop-up di I/O

- **window.alert(*messaggio*)**
 - apre un pop-up bloccante contenente il messaggio ed un pulsante per conferma di lettura (OK)
- **window.prompt(*prompt [, valore_iniziale]*)**
 - apre un pop-up bloccante contenente il testo di prompt, un campo di input (vuoto o col valore iniziale specificato) e due pulsanti per inserire la risposta (OK) o per non fornirla (Cancel)
 - restituisce il valore introdotto, oppure null nel caso l'utente abbia premuto Cancel o chiuso il pop-up
- **a rigor di termini queste sono implementazioni JS dell'oggetto window (DOM livello 0) e relativi metodi**

Pop-up di I/O: esempio 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Pop-up di I/O: esempio 1</title>
  <script type="text/javascript">
    var n = window.prompt("Nome?", "nessuno")
    window.alert(n);
  </script>
</head>
<body>
  <p>Fine dell'esempio.</p>
</body>
```

Pop-up di I/O: esempio 2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
  <title>Pop-up di I/O: esempio 2</title>
</head>
<body>
  <p>Inizio dell'esempio.</p>
  <script type="text/javascript">
    var n = window.prompt("Nome?", "nessuno")
    window.alert("Ciao "+n);
  </script>
  <p>Fine dell'esempio.</p>
</body>
```

Output tramite HTML

- si può usare l'oggetto DOM "document" con uno dei seguenti metodi:
 - write(text)
 - inserisce il testo
 - writeln(text)
 - inserisce il testo seguito da CR LF
 - il testo è inserito nel punto ove si incontra lo script

Output tramite HTML: esempio

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Output tramite HTML: esempio</title>
</head>
<body>
<script type="text/javascript">
var n = window.prompt("Nome?", "nessuno")
document.writeln("<p>Ciao "+n+"</p>");
</script>
</body>
```

Operatori relazionali e logici

descrizione	simbolo
uguaglianza (valore)	==
identità (valore e tipo)	===
disuguaglianza (valore)	!=
non identità (valore e tipo)	!==
maggiore di / maggiore o uguale a	> >=
minore di / minore o uguale a	< <=
appartenenza	in
AND logico	&&
NOT logico	!
OR logico	

Operatori e valori Booleani

- i seguenti valori sono equivalenti a Falso:
 - false
 - 0
 - NaN
 - la stringa vuota ""
 - null
 - undefined
- qualunque altro valore è equivalente a Vero
- attenzione quindi ai confronti:
 - (27 == true) fornisce valore Vero
 - (27 === true) fornisce valore Falso

Operatori aritmetici

descrizione	simbolo
addizione	+
incremento unitario	++
sottrazione	-
decremento unitario	--
moltiplicazione	*
divisione (floating-point)	/
modulo (resto della divisione intera)	%

Operatori di assegnazione

descrizione	simbolo	esempio	equivalenza
assegnazione	=	a = 5	
assegn. con somma	+=	a += 5	a = a + 5
assegn. con sottrazione	-=	a -= 5	a = a - 5
assegn. con prodotto	*=	a *= 5	a = a * 5
assegn. con divisione	/=	a /= 5	a = a / 5
assegn. con modulo	%=	a %= 5	a = a % 5

Le stringhe di caratteri

- **particolarmente importanti perché qualunque input fornito dall'utente tramite browser è una stringa**
- **operatori:**
 - assegnazione (=)
 - confronto in ordine alfabetico (== != > >= < <=)
 - concatenazione (+ +=)
- **attenzione! se un'istruzione contiene stringhe, numeri ed il simbolo +, tutto viene trattato come stringhe; si suggerisce perciò l'uso delle parentesi:**

```
ris = "N=" + 5 + 2; // ris = "N=52"
ris = "N=" + (5 + 2); // ris = "N=7"
ris = "N=" + 5 - 2; // ris = NaN
```

Conversioni stringhe - numeri (I)

- **Number(oggetto)**
 - restituisce una rappresentazione numerica dell'oggetto o NaN
- **String(oggetto)**
 - restituisce una rappresentazione come stringa di caratteri dell'oggetto o undefined o null

```
n = Number("2"); // n = 2
n = Number("2.3"); // n = 2.3
n = Number("2,3"); // n = NaN
n = Number("2mila"); // n = NaN
n = Number("2 mila"); // n = NaN
```

Conversioni stringhe - numeri (II)

- **parseInt(stringa [, base])**
 - restituisce un numero intero o NaN
 - possibile specificare la base numerica (default: 10)
- **parseFloat(stringa)**
 - restituisce un numero floating-point o NaN
- **parseInt e parseFloat considerano solo la parte iniziale, fermandosi al primo carattere non valido**

```
n = parseInt("10",2); // n = 2
n = parseInt("2.3"); // n = 2
n = parseInt("2,3"); // n = 2
n = parseInt("2mila"); // n = 2
n = parseInt("duemila"); // n = NaN
```

Proprietà e metodi dell'oggetto String (I)

- **importanti perché usabili anche su variabili stringa**
- **length()**
 - lunghezza N della stringa (indice da 0 a N-1)
- **charAt(pos)**
 - il carattere in posizione pos
- **charCodeAt(pos)**
 - il codice numerico Unicode del carattere in posizione pos
- **indexOf(searchString [, start])**
 - posizione della stringa cercata (a partire da start o dall'inizio)
 - -1 se la stringa non viene trovata

Proprietà e metodi dell'oggetto String (II)

- **lastIndexOf(searchString [, start])**
 - posizione della stringa cercata (a partire da start o dalla fine)
 - -1 se la stringa non viene trovata
- **slice(begin [, end])**
 - crea una nuova stringa coi caratteri presenti da begin a end (escluso) oppure alle fine
 - usare end negativo per indicare posizioni dalla fine
- **substring(begin [, end])**
 - estrae i caratteri da begin a end (o alla fine)
- **substr(begin [, length])**
 - estrae i caratteri da begin per la quantità length

Proprietà e metodi dell'oggetto String (III)

- **toLowerCase()**
 - restituisce i caratteri convertiti in minuscolo
- **toUpperCase()**
 - restituisce i caratteri convertiti in maiuscolo

Test su valori errati

- non si può fare un confronto con NaN o altri valori limite, ma si possono usare funzioni che testano questi casi
- **isFinite(number)**
 - vero se il numero non è pari a +/- infinito
- **isNaN(number)**
 - vero se il numero ha un valore diverso da NaN
- **typeof(x)**
 - restituisce una stringa che esprime il tipo di dato attualmente corrispondente a X
 - risposte possibili: boolean, function, number, object, string, undefined

Controllo di flusso

- strutture di controllo utili per eseguire un programma in modo non sequenziale
 - if
 - if/else
 - while
 - do/while
 - for
 - for/in

Controllo di flusso "if" / "if-else"

- esecuzione condizionale di istruzioni in base al valore di una condizione Booleana
 - if
 - if/else

```
if ( condizione )
{
    ... istruzioni
}
```

```
if ( condizione )
{
    ... istruzioni
}
else
{
    ... istruzioni
}
```

Esempio di costrutto "if-else"

```
<script type="text/javascript">
var t_mis = window.prompt("Temperatura misurata?");
if ( t_mis <= 0 )
    alert("l'acqua e' ghiacciata");
else if ( t_mis >= 100 )
    alert("l'acqua e' vapore");
else
    alert("l'acqua e' allo stato liquido");
</script>
```

Selezione multipla: l'istruzione "switch"

- è una forma abbreviata di cascata di "if-else"
- uso di "break" per non continuare col caso successivo
- "default" se non si ricade in nessun caso esplicito

```
switch ( espressione )
{
case valore1: ... istruzioni;
    break;
case valore2: ... istruzioni;
    break;
...
default: ... istruzioni;
}
```

Esempio di costrutto "switch"

```
<script type="text/javascript">
var frutto = window.prompt("Quale frutto vuole?");
switch (frutto) {
case "pera":
    alert ("pere a 2 Euro/kg"); break;
case "mela":
    alert ("mele a 1.5 Euro/kg"); break;
case "banana":
    alert ("banane a 1 Euro/kg"); break;
default:
    alert ("spiacenti, non abbiamo "+frutto);
}
</script>
```

Controllo di flusso "while"

- struttura per ripetere un blocco di istruzioni finché una condizione è e rimane vera
- le istruzioni del ciclo possono quindi essere eseguite zero o più volte

```
while ( condizione )
{
  ... istruzioni
}
```

Esempio di ciclo "while"

```
<script type="text/javascript">
// conto alla rovescia
var x = 5;
while (x >= 0)
{
  alert(x);
  x--;
}
</script>
```

Controllo di flusso "do-while"

- struttura simile al while con la differenza che il controllo si fa alla fine del ciclo e quindi il ciclo viene sempre eseguito almeno una volta

```
do
{
  ... istruzioni
} while ( condizione );
```

Esempio di ciclo "do-while"

```
<script type="text/javascript">
var ris;
do {
  ris = window.prompt(
    "Scrivi 'ciao' o resti bloccato qui");
} while (ris != "ciao");
</script>
```

Controllo di flusso "for"

- struttura per ripetere blocchi di istruzioni finché una condizione rimane vera
- specifica:
 - un'azione di inizializzazione
 - una condizione
 - un'azione da ripetere alla fine di ogni ciclo (tipicamente un incremento/decremento dell'indice associato al ciclo)

```
for ( inizializzazione ; condizione; azione_ripetitiva )
{
  ... istruzioni_da_ripetere
}
```

Esempio di ciclo "for" numerico

```
<script type="text/javascript">
/*
calcolo della somma
dei primi 10 numeri naturali
*/
var totale = 0;
for (var i=1; i <= 10; i++)
{
  totale = totale + i;
}
alert("Somma dei numeri [1...10] = "+totale);
</script>
```