

Corso di JavaScript

2 - Concetti generali

M. Malatesta 2-Concetti generali-02

1
03/02/2017

Prerequisiti

- Conoscenza HTML
- Programmazione ad oggetti (Java, C++)
- Concetto di programmazione ad eventi

M. Malatesta 2-Concetti generali-02

2
03/02/2017

Introduzione

In questa Unità introduciamo alcuni strumenti generali come le variabili, i tipi di dato, le conversioni di tipo e le istruzioni di input.

M. Malatesta 2-Concetti generali-02

3
03/02/2017

Le variabili

```
<HTML>
<HEAD>
<TITLE>Le variabili</TITLE>
<SCRIPT Language="JavaScript">
  var X=1,Y,
  Y="sono nello script "
  alert (Y+'HEAD e X vale '+X)
</SCRIPT>
</HEAD>
<BODY>
  <P>Doppio di X</P>
  <SCRIPT Language="JavaScript">
    X=2*X; alert (Y+'BODY e X vale '+X)
  </SCRIPT>
</BODY>
</HTML>
```

Le variabili sono utilizzate con la tecnica di collegamento posticipato (*late binding*), ossia ricevono il tipo al momento dell'assegnazione

Volendo scrivere più istruzioni sulla stessa riga, occorre usare il ";" come separatore

M. Malatesta 2-Concetti generali-02

4
03/02/2017

Conversioni di tipo

Poiché il tipo di una variabile può cambiare dinamicamente (durante l'esecuzione di un programma) è necessario distinguere due tipi di conversione:

- **Conversione implicita**
- **Conversione esplicita**

M. Malatesta 2-Concetti generali-02

5
03/02/2017

Conversioni di tipo

La **conversione implicita** avviene quando si trattano espressioni con operandi di tipo diverso.

Ad esempio:

$a = 2 + \text{"15"} \rightarrow a = \text{"215"}$

$a = \text{"1"} + 21 \rightarrow a = \text{"121"}$

(nelle stringhe prevale l'operatore "+")

$a = 3 * \text{"2"} \rightarrow a = 6$

$a = \text{"5"} * 4 \rightarrow a = 20$

(nei numeri prevale l'operatore "*")

M. Malatesta 2-Concetti generali-02

6
03/02/2017

Conversioni di tipo

La **conversione esplicita** avviene quando si usano appositi metodi di conversione.

Ad esempio:

a = **String** (12) → a = "12" (converte da numerico a stringa)

Per convertire da stringa a numerico si usano le seguenti funzioni

- **parseInt()**
- **parseFloat()**

Conversioni di tipo

- **parseInt()** estrae un intero all'inizio di una stringa, scartando i caratteri, ignorando i decimali e l'eventuale virgola
 - **parseInt** ("39 gradi")=39
 - **parseInt** ("2abc") = 2
 - **parseInt** ("39 gradi",16)=27 (il secondo parametro, facoltativo, è la base numerica di arrivo);
- **parseFloat** opera allo stesso modo, ma conserva l'eventuale virgola presente e il segno.
 - **parseFloat** ("39,2 a") = 39.2

Conversioni di tipo

```
<HTML>
  <BODY><P>Conversioni di tipo</P>
  <SCRIPT Language = "JavaScript">
    alert("La conversione parseInt('29a') vale “ + parseInt("29a")+“\n”)
    alert("La conversione parseFloat('23abc.5') vale” +
      parseFloat("23abc.5")+“\n”)
    alert("La conversione parseFloat('23.5abc') vale” +
      parseFloat("23.5abc")+“\n”)

    alert("typeof 5 = “ + typeof 5 + “\n”)
    alert("typeof '5a' = “ + typeof ("5a"))
  </SCRIPT></BODY>
</HTML>
```

M. Malatesta 2-Concetti generali-02

9
03/02/2017

eval() e *typeof*

L'operatore *eval(espressione)* restituisce il tipo di dato di espressione, come mostrato sotto (in cui ad esempio, si ha a=2, b=3).

– **eval** (“a+b+4”) = 9

In qualsiasi momento, comunque, si può verificare il tipo della variabile tramite l'operatore **typeof** (i valori restituiti sono: **string**, **boolean**, **number**, **function**).

- **var** prova=new Function() → **typeof** prova = “object”
- **var** numero=1 → **typeof** numero = “number”
- **var** carattere="Salve" → **typeof** carattere = “string”

M. Malatesta 2-Concetti generali-02

10
03/02/2017

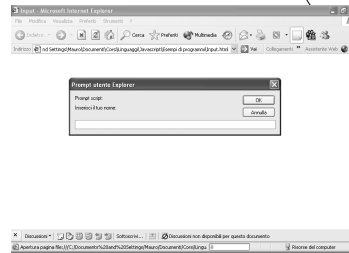
Istruzione di input

Ad esempio:

```
<HTML><HEAD><TITLE>Input</TITLE>
<SCRIPT Language = "JavaScript">
    var X = prompt("Inserisci il tuo nome: ", "")
    document.write("Ciao " + X + ", benvenuto in JavaScript")
</SCRIPT>
</HEAD>
</HTML>
```

Input mediante finestra grafica

Aprendo la pagina web l'input avviene mediante una finestra come indicato in figura



M. Malatesta 2-Concetti generali-02

11
03/02/2017

Istruzione di input

```
<HTML>
<HEAD>
<TITLE>Addizionatore</TITLE>
    <SCRIPT Language="JavaScript">
        var X = prompt("Primo numero: ")
        var Y = prompt("Secondo numero: ")
        var Z = parseInt(X) + parseInt(Y)
    </SCRIPT>
</HEAD>
<BODY>
    <SCRIPT Language="JavaScript">
        alert("La somma e' " + Z)
    </SCRIPT>
</BODY>
</HTML>
```

Questo script esegue l'addizione tra due numeri.

Si noti la necessità di conversione a intero con **parseInt()**

M. Malatesta 2-Concetti generali-02

12
03/02/2017

Script associati a link

L'esecuzione di uno script può essere associata ad un link sulla pagina Web nel seguente modo.

```
<HTML><HEAD><TITLE>Script con link</TITLE>
<SCRIPT Language = "JavaScript">
  var X = prompt("Inserisci il tuo nome: ", "")
</SCRIPT>
</HEAD>
<BODY>
  <P><A HREF="JavaScript:alert('Ciao ' + X +' benvenuto in
    JavaScript')">Ciao</A>
</BODY>
</HTML>
```

Apici doppi e semplici (v. seguito)

Link allo script

Per comodità le istruzioni possono essere continuate a capo

M. Malatesta 2-Concetti generali-02

13
03/02/2017

Un esempio di script su evento

L'esecuzione di uno script può essere associato ad un evento come il passaggio del mouse nel seguente modo.

```
<HTML>
<HEAD><TITLE>Tipi di script</TITLE></HEAD>
<BODY><p>
  <SCRIPT LANGUAGE="JavaScript">alert("Script interno")</SCRIPT>
  <a href="javascript:alert('Script su link')">Clicca qui</a>
  <a href="#" onmouseover="alert('Script su evento')">Passa il
    mouse</a> </BODY>
</HTML>
```

Un esempio di gestore di evento **onmouseover** che consente di chiamare la funzione al passaggio del mouse

Osservare gli apici semplici e doppi

M. Malatesta 2-Concetti generali-02

14
03/02/2017

Commenti

I commenti, come accennato, sono rappresentabili in due modi

- */*commento*/*
Il commento può essere posto su più righe o su una riga singola, mentre *non è accettato dall'interprete il commento annidato.*
- *// commento*
Commenta tutti i caratteri fino alla fine della riga

M. Malatesta 2-Concetti generali-02

15
03/02/2017

Commenti

- I commenti Javascript non possono essere inseriti al di fuori dei tag <SCRIPT>...</SCRIPT>, altrimenti HTML li considererà come parte del testo
- Viceversa, i tag di commento HTML non si possono utilizzare all'interno dello script. L'unico commento ammissibile è quello che consente **di racchiudere tutto lo script nei tag di commento di HTML**, ad esempio:

```
<SCRIPT Language = "JavaScript">  
  <!-- alert("Welcome!");  
  //-->  
</SCRIPT>
```

In tal modo si maschera il codice JavaScript ad eventuali browser che non lo leggono, per evitare che l'HTML lo possa considerare come testo e quindi visualizzarlo.

M. Malatesta 2-Concetti generali-02

16
03/02/2017

Apici (' ') e doppi apici (“ ”)

Un'altra osservazione riguarda gli apici semplici e doppi.

- Con i doppi apici racchiudiamo le parti di *codice* JavaScript e le *stringhe*
- Occorre prestare attenzione a *non annidare stringhe racchiuse da apici dello stesso tipo*
- Se si desidera che *in una stringa appaiano apici doppi o singoli* come parte integrante della stringa stessa, si fanno precedere da una barra rovesciata (\).

Argomenti

- Le variabili
- Conversioni di tipo
- **eval()** e **typeof**
- Istruzione di input
- Script associati a link
- Un esempio di script su evento
- Commenti
- Apici (' ') e doppi apici (“ ”)

Altri testi di riferimento

- P.Gallo, F.Salerno – HTML, CSS e JavaScript – ed. Minerva Italica
- E.A. Vander Veer, JavaScript for dummies - ed. Apogeo