

Corso di JavaScript

4 – Funzioni e strutture di controllo

M. Malatesta 4-Funzioni e strutture di controllo-03

1
03/02/2017

Prerequisiti

- Conoscenza HTML
- Architettura client-server
- Programmazione ad oggetti (Java, C++)
- Concetto di programmazione ad eventi
- Parametri formali e attuali
- Variabili locali e globali
- Valore di ritorno di una funzione

M. Malatesta 4-Funzioni e strutture di controllo-03

2
03/02/2017

Introduzione

In questa Unità illustriamo alcuni strumenti sintattici di JavaScript per la scrittura di semplici programmi, in particolare studiamo le funzioni (sottoprogrammi) sia predefiniti, che definibili dal programmatore e le strutture di controllo, per realizzare tutti i tipi di elaborazione.

M. Malatesta 4-Funzioni e strutture di controllo-03

3
03/02/2017

Funzioni

Le funzioni in JavaScript hanno il consueto significato che hanno in Java o C++. La sintassi è

```
function ident (lista_parametri)    // intestazione o firma della funzione
{
    istruzioni                      // definizione della funzione
}
dove
ident è il nome della funzione
lista_parametri è l'elenco dei parametri formali passati alla funzione
```

M. Malatesta 4-Funzioni e strutture di controllo-03

4
03/02/2017

Funzioni

Alcune caratteristiche delle funzioni JavaScript sono:

- non è possibile *definire funzioni all'interno di altre funzioni*
- nella lista dei parametri formali *non compare il tipo dei parametri*
- le funzioni sono di solito dichiarate e definite nella sezione HEAD del documento HTML

M. Malatesta 4-Funzioni e strutture di controllo-03

5
03/02/2017

Funzioni

```
<HTML><!-- Funzioni.html -->
<HEAD>
<TITLE>Funzioni</TITLE>
<SCRIPT Language = "JavaScript">
  function Saluti( )
  { alert("Ciao a tutti!") }
</SCRIPT>
</HEAD>
<BODY><FORM>
  <INPUT TYPE = "button" NAME = "SalutiBtn" VALUE = "Ciao a tutti" onClick
    = "Saluti()">
</FORM></BODY>
</HTML>
```

Questo script contiene una funzione *Saluti()* che stampa la frase "Ciao a tutti" quando viene premuto il pulsante

Si noti il gestore di evento **onClick** che consente di chiamare la funzione *Saluti()* alla pressione del pulsante (evento **Click**)

M. Malatesta 4-Funzioni e strutture di controllo-03

6
03/02/2017

Funzioni

Le funzioni possono essere realizzate in modo da operare su **parametri formali**; al momento della chiamata, questi vengono sostituiti dai **parametri attuali** per produrre il risultato.

ATTIVITA': Scrivere il file *FunzConParametri.html* contenente una funzione *ChiediDati()* che legge due numeri *d1* e *d2* e ne stampa il prodotto, calcolandolo mediante un'altra funzione con parametri *Prodotto(n1, n2)*

M. Malatesta 4-Funzioni e strutture di controllo-03

7
03/02/2017

Funzioni

```
<HTML><!-- FunzConParametri.html -->
<HEAD>
<SCRIPT Language = "JavaScript">
  function ChiediDati( )
  { var d1=prompt("Primo valore: ",0)
    var d2=prompt("Secondo numero: ",0)
    alert('Il prodotto è '+ Prodotto(d1,d2))  }
  function Prodotto(n1,n2)
  { return (n1*n2) }
</SCRIPT>
<HEAD>
<BODY><FORM>
<INPUT TYPE = "button" NAME = "calcolo" VALUE = "Prodotto"
  onClick = "ChiediDati( )">
</FORM></BODY>
</HTML>
```

All'evento **Click** si attiva la funzione di gestione **onClick** che chiama la funzione *ChiediDati()* che a sua volta chiama la funzione *Prodotto()* con parametri *n1* ed *n2*

M. Malatesta 4-Funzioni e strutture di controllo-03

8
03/02/2017

Funzioni predefinite

Funzioni matematiche:

Math.*nome(parametri)*

dove *nome* è il tipico nome di funzioni matematiche:

pow(), abs(), sin(), cos(), tan(), exp(), floor(), sqrt(), min(), max(), log(), random(), round()

Funzioni stringa:

*stringa.***nome(parametri)**

dove *nome* è il tipico nome di funzioni di stringa:

charAt(), substring(), lastIndexOf(), italics(), indexOf(), fontcolor(), fontsize(), bold(), blink(), toLowerCase(), toUpperCase()

Funzioni predefinite

ATTIVITA': Scrivere il file *tavoleNumeriche.html* che stampa le tavole aritmetiche fino al valore n (impresso da input). Ogni riga deve contenere n, il suo quadrato, il cubo, la radice quadrata e quella cubica.

OSSERVAZIONE:

Per andare a capo nel documento occorre scrivere:

```
.....  
document.write("<P>");  
istruzioni di stampa;  
document.write("</P>")  
.....
```

Variabili locali e globali

Le funzioni possono avere

- **variabili locali**
- **variabili globali**

Le **variabili locali** hanno valore solo all'interno del corpo della funzione in cui sono definite

Le variabili locali rendono una funzione indipendente dal contesto e quindi riutilizzabile anche in altre pagine HTML, soprattutto se contenuta in un file esterno con estensione .js.

Variabili locali e globali

Le **variabili globali**, che hanno valore per tutto il documento HTML, vanno dichiarate *all'inizio dello script e fuori da ogni funzione* (in pratica nel tag <SCRIPT> della sezione <HEAD>).

Le variabili globali sono dei contenitori che durano per tutta la durata della pagina e servono a trasmettere valori

- tra le funzioni e tra gli script;
- tra le varie pagine;
- al server.

Variabili locali e globali

```
<HTML>
<HEAD>
  <Script Language = "JavaScript">
    var X // variabile globale
    function Modifica( )
    { X = prompt("Inserisci un numero")*1
      alert(X)
    }
    function Azzera( )
    { X=0 }
    function Visualizza( )
    { alert(X) }
  </SCRIPT>
</HEAD>
```

ATTIVITA': Scrivere il file *VarGlobali.htm* che presenta 3 pulsanti (Azzera, Modifica e Visualizza) che attivano le corrispondenti funzioni su una variabile globale X.

- Modifica() legge X e la stampa
- Visualizza() stampa la X
- Azzera() pone 0 in X.

e verificare che la X essendo globale è accessibile a tutte le funzioni

Nella sezione HEAD si dichiarano la variabile globale X e le tre funzioni *Modifica()*, *Azzera()* e *Visualizza()*

M. Malatesta 4-Funzioni e strutture di controllo-03

13
03/02/2017

Variabili locali e globali

```
<BODY><FORM>
  <INPUT TYPE = "button" NAME = "visualizza" VALUE = "Visualizza"
    onClick = "Visualizza()">
  <INPUT TYPE = "button" NAME = "modifica" VALUE = "Leggi" onClick =
    "Modifica()">
  <INPUT TYPE = "button" NAME = "azzera" VALUE = "Azzera" onClick =
    "Azzera()">
</FORM>
</BODY>
</HTML>
```

Nella sezione BODY si creano 3 pulsanti.
Immettendo un valore tramite la funzione *Modifica()*, questo viene poi azzerato se si chiama *Azzera()*, in quanto essendo X globale, è di accesso pubblico per tutte le funzioni.

M. Malatesta 4-Funzioni e strutture di controllo-03

14
03/02/2017

Strutture di controllo

- sequenza e selezione

Le strutture di controllo sono quelle tipiche dei linguaggi Java e C++ e le richiamiamo brevemente dal punto di vista sintattico.

Sequenza	<code>{ ... }</code>
Selezione	<code>if (espr_logica) istruzione</code> <code>[else istruzione]</code>
Selezione multipla	<code>switch (espr)</code> <code>{</code> <code> case cost : istruzione; break</code> <code> case cost: istruzione; break</code> <code> </code> <code> [default] : istruzione</code> <code>}</code>

Nota: le parentesi quadre [] indicano opzione

M. Malatesta 4-Funzioni e strutture di controllo-03

15
03/02/2017

Strutture di controllo

- iterazione

Iterazione	<code>while (espr_logica)</code> <code>{ istruzione }</code> <code>for (istruzione ; espr_logica ; istruzione)</code> <code>{ istruzione }</code> <code>do</code> <code>{ istruzione }</code> <code>while (espr_logica)</code>
-------------------	--

M. Malatesta 4-Funzioni e strutture di controllo-03

16
03/02/2017

Strutture di controllo

- esempi

```
<HTML><!-- calcolo del massimo tra due valori – massimo.html -->
<HEAD>
  <SCRIPT Language = "JavaScript">
    function Max()
    { N1 = parseInt(prompt("Inserisci il primo numero",0))
      N2 = parseInt(prompt("Inserisci il secondo numero",0))
      if (N1 > N2)      Max = N1
      else             Max = N2
      alert("Il massimo è " + Max)
    }
  </SCRIPT>
</HEAD>
<BODY>
  <FORM NAME="f1">
    <INPUT TYPE="button" NAME="confronta" VALUE="Massimo" onClick =
      "Max()">
  </FORM>
</BODY>
</HTML>
```

ATTIVITA': Scrivere un file *massimo.html* contenente uno script attivato da un pulsante che esegue una funzione *Max()* che legge due valori e stampa il massimo.

M. Malatesta 4-Funzioni e strutture di controllo-03

17
03/02/2017

Strutture di controllo

- esempi

```
<HTML><!-- calcolo della somma di 5 numeri – sommaNumeri.html -->
<HEAD>
  <SCRIPT Language = "JavaScript">
    function Leggi()
    { var I = 0, S=0
      while (I < 5)
      { X = parseInt(prompt("Inserisci un numero ", 0))
        I++; S+=X
      }
      alert("La somma vale " + S)
    }
  </SCRIPT>
</HEAD>
<BODY><FORM>
  <INPUT TYPE = "button" NAME = "leggi" VALUE = "Somma" onClick =
    "Leggi()">
</FORM></BODY>
</HTML>
```

ATTIVITA': Scrivere un file *sommaNumeri.html* contenente uno script attivato da un pulsante che esegue una funzione *Leggi()* che legge 5 valori e ne stampa la somma

M. Malatesta 4-Funzioni e strutture di controllo-03

18
03/02/2017

Argomenti

- Funzioni
- Funzioni predefinite
- Variabili locali e globali
- Strutture di controllo
 - sequenza e selezione
 - iterazione
 - esempi

M. Malatesta 4-Funzioni e strutture di controllo-03

19
03/02/2017

Altri testi di riferimento

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

M. Malatesta 4-Funzioni e strutture di controllo-03

20
03/02/2017