




Il linguaggio JavaScript

Laura Farinetti
Dip. Automatica e Informatica
Politecnico di Torino
farinetti@polito.it

L. Farinetti - Politecnico di Torino



JavaScript

- Linguaggio di scripting sviluppato da Netscape
- Scopo: aggiungere **interattività** alle pagine Web
 - HTML è **statico**
 - gli script CGI aggiungono interattività ed “effetti speciali” alle pagine HTML ma sono eseguiti dal **Web server**
 - gli script JavaScript sono eseguiti dal **client**

L. Farinetti - Politecnico di Torino

2



JavaScript

- Introdotta nei browser Netscape a partire da **Navigator 2**
- Il codice JavaScript è parte **integrante** del documento HTML (richiede browser compatibile)
- Codice **interpretato** dal client
- Permette al client di intraprendere azioni direttamente senza passare dalla rete

L. Farinetti - Politecnico di Torino

3



JavaScript

- Fornisce un insieme di **funzioni e comandi**
 - per eseguire calcoli matematici, manipolare stringhe, gestire finestre di pop-up, emettere suoni, verificare gli input dell'utente, ...
- Il codice che esegue queste azioni può essere **incluso** in una pagina ed **eseguito** quando la pagina è caricata
- Inoltre permette di scrivere funzioni ed associarle ad **eventi** specificati

L. Farinetti - Politecnico di Torino

4

JavaScript in HTML

- È possibile includere codice in linguaggio JavaScript all'interno di un documento HTML in **due modi**:
 - come insieme di istruzioni e funzioni usando il **tag <SCRIPT>**
 - come handler di eventi usando un generico **tag HTML**

Istruzioni JavaScript

- Per inserire istruzioni Javascript in una pagina HTML si usa il tag **<SCRIPT>**

```
<SCRIPT LANGUAGE="JavaScript">  
  istruzioni JavaScript ...  
  ...  
</SCRIPT>
```

- Un documento può avere **più tag <SCRIPT>**, ciascuno con un numero a piacere di istruzioni



Istruzioni JavaScript

- I tag `<SCRIPT>` possono essere inseriti in `<HEAD>` o in `<BODY>`
- Uno script deve essere “caricato” prima di essere eseguito
 - gli script in `<HEAD>` vengono “caricati” all’apertura della pagina
 - gli script in `<BODY>` vengono “caricati” quando si incontra lo script
- Lo stesso documento può avere tag sia in `<HEAD>` che in `<BODY>`

L. Farinetti - Politecnico di Torino

7



Istruzioni JavaScript

- Explorer 4.x supporta il 99% delle istruzioni, ma le versioni precedenti...
- Per far sì che i browser che non sono in grado di interpretare i comandi JavaScript li ignorino:

```
<SCRIPT>
<!--
    istruzioni JavaScript ...
-->
</SCRIPT>
```

L. Farinetti - Politecnico di Torino

8

Esempio

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
<!--
    /* Inizia a nascondere lo script
    per i browser incompatibili. */
    document.write("Ciao a tutti!!!")
// Fine del codice. -->
</SCRIPT>
</HEAD>
<BODY>
    Ecco fatto!!!
</BODY>
</HTML>
```

L. Farinetti - Politecnico di Torino

9


Istruzioni JavaScript

- Invece di scrivere le istruzioni JavaScript nel file HTML, è possibile includere un [file esterno](#)
- Utile per condividere funzioni in pagine diverse

```
...
<SCRIPT SRC="common.js">
...
</SCRIPT>
...
```

L. Farinetti - Politecnico di Torino

10




Le variabili

- Dichiarazione

```
var nome
var nome = "Andrea"
var num = 10
var num2
```
- Utilizzo

```
nome = "Marina"
num = 10
num = num2 - 5
```

L. Farinetti - Politecnico di Torino 11



Le variabili

- Possono essere **locali** o **globali**
- Locali
 - definite **all'interno** di una funzione
 - possono essere usate solo **all'interno** di quella funzione
- Globali
 - definite **fuori** dalle funzioni
 - possono essere usate in **qualsiasi** funzione

L. Farinetti - Politecnico di Torino 12



Gli operatori di assegnazione

Operatore	Esempio	Significato
=	i=5	
+=	i+=5	i=i+5
-=	i-=5	i=i-5
=	i=5	i=i*5
/=	i/=5	i=i/5
%=	i%=5	i=i%5
++	i++	i=i+1
--	i--	i=i-1

L. Farinetti - Politecnico di Torino

13

Le funzioni

- Funzioni **predefinite**

```
alert("Messaggio da visualizzare")
```

- Funzioni **definite dall'utente**

```
function somma(a,b)
{
  c = a+b
  return c
}
...
s = somma(num1, num2)
```

L. Farinetti - Politecnico di Torino

14

Passaggio dei parametri

```
function area(base, altezza)
{
  var area = base*altezza
  alert("L'area vale "+area)
}
...
area(4,3)
...
area(16,2)
```

```
function quadrato(n)
{
  temp= n*n
  return temp
}
...
var q=quadrato(5)
document.write(q)
```

L. Farinetti - Politecnico di Torino

15

Costrutti condizionali

- Istruzione if

```
if (condizione)
{
  istruzioni da eseguire se la cond. è vera
}

if (condizione)
{
  istruzioni da eseguire se la cond. è vera
}
else
{
  istruzioni da eseguire se la cond. è falsa
}
```

L. Farinetti - Politecnico di Torino

16

Gli operatori relazionali

Operatore	Significato
==	uguale
!=	diverso
>	maggiore
>=	maggiore o uguale
<	minore
<=	minore o uguale
 	or logico
&&	and logico

L. Farinetti - Politecnico di Torino

17

Costrutti condizionali

- Istruzione **switch**

```
switch (espressione)
{
  case val1:
    istruzioni da eseguire se espress. = val1
    break

  case val2:
    istruzioni da eseguire se espress. = val2
    break

  default:
    istruzioni da eseguire se espress. è
    diversa sia da val1 che da val2
}
```

L. Farinetti - Politecnico di Torino

18

I cicli

- Ciclo **while**

```
while (condizione)
{
  istruzioni da eseguire finché
  la condizione è vera
}
```

- Ciclo **do .. while**

```
do {
  istruzioni da eseguire finché
  la condizione è vera
} while (condizione)
```

I cicli

- In un ciclo **do .. while** le istruzioni sono eseguite almeno una volta

- Ciclo **for**

```
for (inizializz.; condizione; incremento)
{
  istruzioni da eseguire
}

for (i=0; i<=5; i++)
{
  alert("Messaggio")
}
```

Note

- JavaScript è **case sensitive**
 - attenzione nell'utilizzo dei nomi di variabili e funzioni
- **Commenti**

```
// commento su un'unica linea  
/* commento che può estendersi  
su più linee */
```

Oggetti JavaScript

- JavaScript è un **linguaggio ad oggetti**
 - metodi
 - proprietà: attributi
- **Esempio:**

```
document.write("Ciao a tutti!!!")
```

- document: oggetto
- write: metodo

Oggetto document

Proprietà	Significato
<code>bgColor</code>	modifica il colore dello sfondo
<code>fgColor</code>	modifica il colore del testo
<code>lastModified</code>	data dell'ultima modifica
<code>referrer</code>	URL della pagina che contiene il link che ha condotto al documento

Oggetto document

Metodo	Significato
<code>write</code>	scrive quanto specificato
<code>writeln</code>	scrive su una nuova linea

```
<script>
  var esempio
  esempio=document.lastModified
  document.write("Ultima modifica: "+esempio)
</script>
```

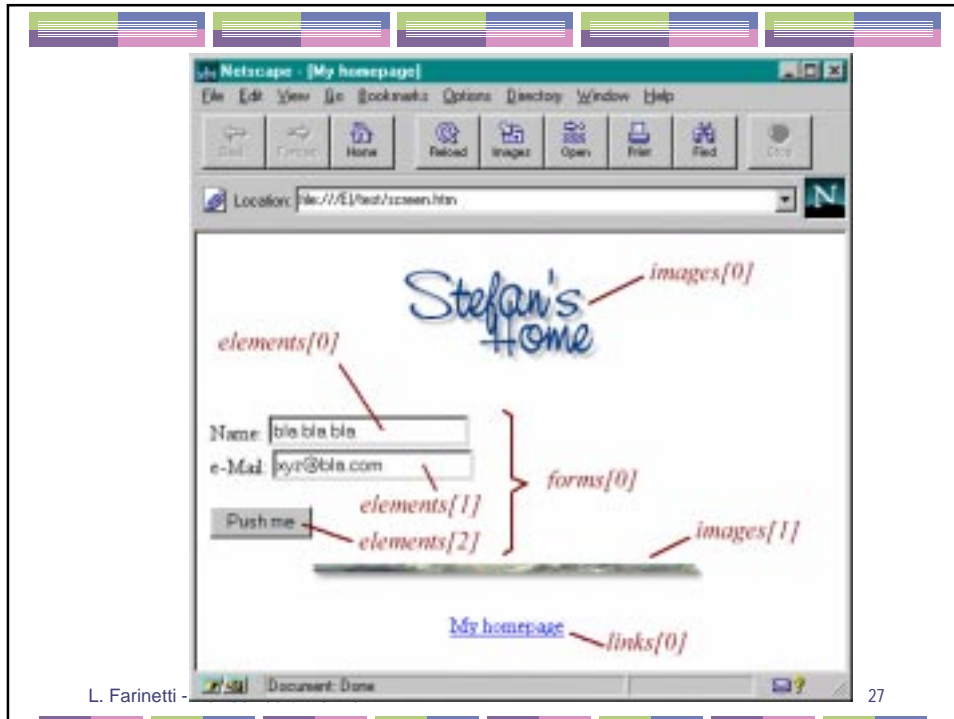
Metodi per la manipolazione delle stringhe

Metodo	Significato
<code>length()</code>	restituisce la lunghezza della stringa
<code>indexOf()</code>	restituisce l'indice della prima occorrenza di un carattere nella stringa
<code>lastIndexOf()</code>	come <code>indexOf()</code> , ma lavora da destra a sinistra
<code>match()</code>	come <code>indexOf()</code> , ma restituisce il carattere
<code>substr()</code>	restituisce una sottostringa; <code>substr(14,7)</code> restituisce 7 caratteri a partire dal 14esimo
<code>substring()</code>	restituisce una sottostringa; <code>substring(7,14)</code> restituisce i caratteri dal 7mo al 14esimo
<code>toLowerCase()</code>	restituisce la stringa in minuscolo
<code>toUpperCase()</code>	restituisce la stringa in maiuscolo

Metodi per la manipolazione delle stringhe

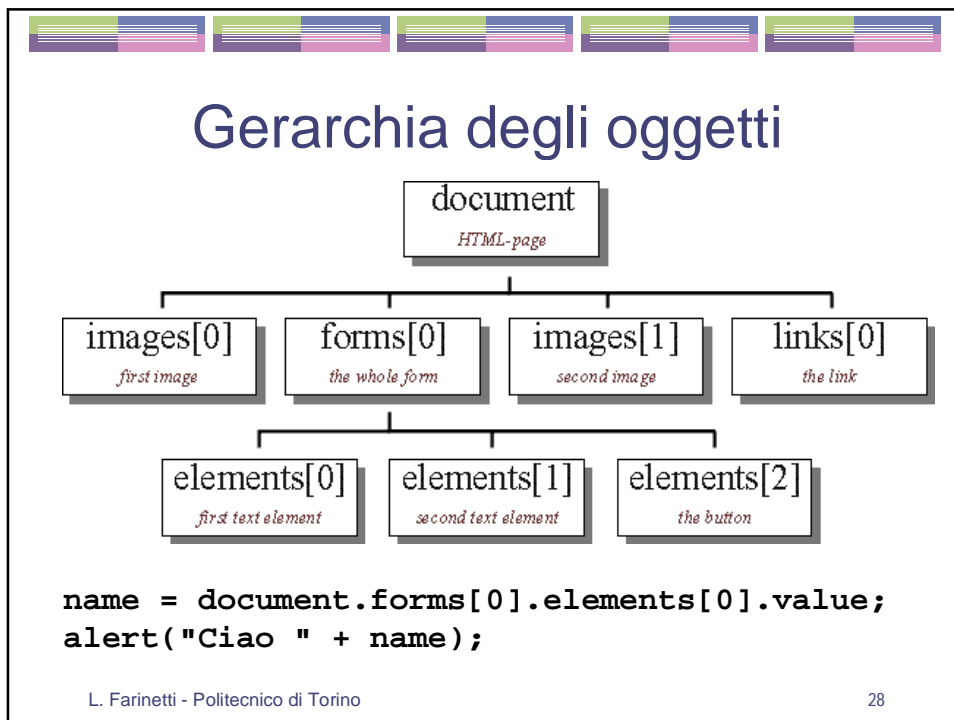
- Esempio:

```
<html>
<body>
<script language="JavaScript">
  var stringa=("Salve a Tutti!")
  document.write(stringa.toLowerCase())
  document.write(stringa.toUpperCase())
  document.write(stringa.substr(8,5))
</script>
</body>
</html>
```



L. Farinetti

27



28

Gerarchia degli oggetti

- Usando gli attributi name degli oggetti si **semplificano** i riferimenti

```
<form name="myForm">  
<input type="text" name="name" value="">  
...
```

```
name = document.myForm.name.value;
```

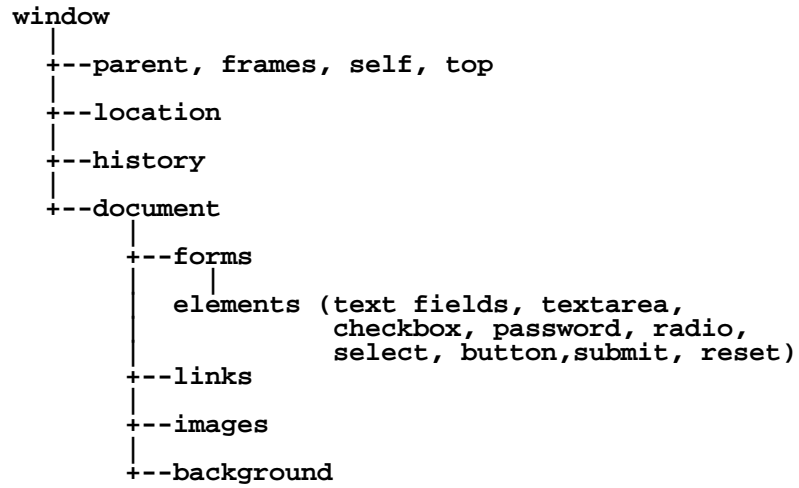
invece di

```
name = document.forms[0].elements[0].value;
```

Gli oggetti

- Ogni pagina contiene gli oggetti:
 - **window**: il livello più alto della gerarchia; contiene proprietà che si applicano all'intera finestra
 - **location**: contiene le proprietà dell'URL corrente
 - **history**: contiene proprietà relative alle URL visitate in precedenza
 - **document**: contiene proprietà relative al contenuto del documento corrente, come il titolo, il background, le form,...

Gerarchia degli oggetti



L. Farinetti - Politecnico di Torino

31

Esempio

```
location.href = "http://www.polito.it"
document.title = "A Simple Document"
document.fgColor = #000000
document.bgColor = #ffffff
history.length = 7
```

L. Farinetti - Politecnico di Torino

32

Handler di eventi

- **Evento**: conseguenza di un'azione eseguita dall'utente (es: button click)
- Un handler di eventi è **incapsulato** in un documento HTML come **attributo** di un tag a cui si assegna il codice JavaScript da eseguire
- La sintassi generica è:
`<TAG-HTML eventHandler="JavaScript Code">`

Handler di eventi

Handler	Significato
onClick	fa eseguire codice JavaScript quando un oggetto (link, bottone di una form, ...) viene "cliccato"
onLoad	fa eseguire codice JavaScript subito dopo che una pagina o un'immagine è stata caricata
onmouseover	fa eseguire codice JavaScript quando si passa con il mouse sopra un link
onmouseout	fa eseguire codice JavaScript quando si passa con il mouse fuori da un link
onUnload	fa eseguire codice JavaScript subito dopo che l'utente lascia la pagina

Esempi

```
<form>
<input type="button" value="Ciao"
  onClick="alert('Ciao!')">

<input type=button value="Poli"
  onClick="location.href='http://www.polito.it';">
</form>

<BODY onload="alert('La pagina &egrave;
  stata caricata!')">

<BODY onunload="alert('Grazie della
  tua visita, a presto!')">
```

L. Farinetti - Politecnico di Torino

35

Esempio

```
<form>
<input type="radio" name="r"
  onclick="document.bgColor='#FF0000'">Rosso
<input type="radio" name="r"
  onclick="document.bgColor='#00FF00'">Verde
<input type="radio" name="r"
  onclick="document.bgColor='#0000FF'">Blu<P>
<input type="reset" value="Bianco"
  onclick="document.bgColor='#FFFFFF'">
</form>
```

L. Farinetti - Politecnico di Torino

36

Esempio


```
<html>
<head>
<script language="JavaScript">
  function calcolo() {
    var x= 10; var y= 5
    var somma = x + y
    alert(somma) }
</script>
</head>
<body>
<form>
<input type="button" value="Calcola"
  onClick="calcolo()">
</form>
</body>
</html>
```

L. Farinetti - Politecnico di Torino 37

Finestre di popup

```
window.alert()
```

```
<form>
<input type="button" value="Ciao"
  onClick="alert('Benvenuto!')">
</form>
```




L. Farinetti - Politecnico di Torino 38

Finestre di popup

`window.confirm()`

```
<script>  
var x=window.confirm("Vuoi procedere?")  
if (x)  
    window.alert("Bene!")  
else  
    window.alert("Peccato")  
</script>
```



L. Farinetti - Politecnico di Torino 39

Finestre di popup

`window.prompt()`

```
<script>  
var y=window.prompt("Come ti chiami?")  
window.alert("ciao, "+y)  
</script>
```



L. Farinetti - Politecnico di Torino 40

Gli array (vettori)

- Dichiarazione

```
var nomi = new Array(3)
```

- Assegnazione

```
nomi[0] = "Laura"  
nomi[1] = "Anna"  
nomi[2] = "Margherita"
```

- Uso

```
nonna = nomi[2]
```

Esempio

```
<html>  
<body>  
<script language = "JavaScript">  
  var nomi = new Array(3)  
  nomi[0] = "Laura"  
  nomi[1] = "Anna"  
  nomi[2] = "Margherita"  
  for (i=0;i<3;i++)  
  {  
    document.write(nomi[i] + "<br>")  
  }  
</script>  
</body>  
</html>
```

Esempio

```
<html>
<body>
<script language = "JavaScript">
  var nomi = new Array("Laura","Anna",
    "Margherita")
  for (i=0;i<nomi.length;i++)
  {
    document.write(nomi[i] + "<br>")
  }
</script>
</body>
</html>
```

L. Farinetti - Politecnico di Torino

43

Funzioni matematiche (metodi)

Metodo	Significato
<code>max()</code>	restituisce il maggiore fra due numeri
<code>min()</code>	restituisce il minore fra due numeri
<code>random()</code>	restituisce un numero casuale compreso fra 0 e 1
<code>round()</code>	restituisce il numero arrotondato all'intero più vicino


```
<script language = "JavaScript">
  num = Math.random()*10
  document.write(Math.round(num))
</script>
```

L. Farinetti - Politecnico di Torino

44

Esempio: controllo di validità


```
<html>
<head>
<script language="JavaScript">
function valida()
{
  x = document.modulo
  val = x.numero.value
  if (val>=1 && val<=5)
  { return true }
  else
  { alert("Deve essere compreso fra 1 e 5")
    return false }
}
</script>
</head>
```



L. Farinetti - Politecnico di Torino

45

Esempio: controllo di validità



```
<body>
<form name="modulo" action="prossima.htm"
  onsubmit="return valida()">

Inserisci un numero compreso fra 1 e 5:
<input type="text" name="numero">

<input type="submit" value="Invia">

</form>
</body>
</html>
```

L. Farinetti - Politecnico di Torino


46

Animazione di immagini

- Preload delle immagini



```
<head>
<script language="JavaScript">
  image01= new Image(42,42)
  image01.src="rosso.gif"
  image02= new Image(42,42)
  image02.src="verde.gif"
</script>
</head>
```





rosso.gif



verde.gif

L. Farinetti - Politecnico di Torino 47

Animazione di immagini

- Inserimento dell'immagine "animata"



```
<body>
<a href="vuoto.htm" onmouseover=
"document.images['esempio'].src=image02.src"
onmouseout=
"document.images['esempio'].src=image01.src">
  </a>
</body>
```

L. Farinetti - Politecnico di Torino 48



Puntatori utili

- <http://www.w3schools.com/js/>
- Esempi
 - http://www.w3schools.com/js/js_examples.asp
 - <http://javascript.internet.com>