

javascript

Costantino Pistagna

pistagna@dmi.unict.it
<http://aladino.dmi.unict.it>

Cos'è Javascript?

- Javascript è stato pensato per aggiungere interattività alle pagine HTML
- Ma non solo!
- E' un linguaggio interpretato
- Leggero
- Semplice da imparare

(c) 2009 - costantino pistagna - pistagna@dmi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Cos'è Javascript?

- Javascript viene inserito direttamente all'interno della pagina HTML
- Il codice viene eseguito senza una compilazione preliminare
- Non c'è da acquistare nessuna licenza per usarlo!

(c) 2009 - costantino pistagna - pistagna@dmi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Java e...Javascript?

- NO!
- Java e Javascript sono due linguaggi completamente differenti!
- Logica
- Design

(c) 2009 - costantino pistagna - pistagna@dmi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Java e...Javascript?

- Java (Sun Microsystems) è un linguaggio orientato agli oggetti.
 - Stessa categoria di C++ ed ObjC
- Javascript fu introdotto da netscape
 - Navigator 2.0 - Livescript

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Cosa posso farci?

- JavaScript fornisce ai programmatori web un potente strumento
- Normalmente gli autori HTML non sono programmatori
 - Javascript è un linguaggio semplice
 - Chiunque può usarne le sue funzionalità sulle pagine web!

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Cosa posso farci?

- JavaScript può aggiungere testo dinamico all'interno delle pagine HTML

```
document.write("<h1>" + name + "</h1>")
```
- Scrive una variabile di testo all'interno della pagina HTML

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Cosa può fare?

- JavaScript può reagire ad eventi
- E' possibile istruire un javascript per essere eseguito quando accade "qualcosa"
 - Una pagina che finisce di caricare
 - Il click del mouse
 - Il movimento del cursore

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Cosa può fare?

- JavaScript può leggere e scrivere elementi HTML
- E' possibile cambiare proprietà ed elementi della pagina HTML
 - Purchè sia ben strutturata!
 - xHTML...ma non solo

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Cosa può fare?

- E' possibile utilizzarlo per validare dati
 - Validare i dati di un form prima di inviarli al server
 - Risparmiamo il server da lavoro extra

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Cosa può fare?

- E' possibile interagire con il browser dell'utente
 - Può essere usato per conoscere il browser dell'utente
 - In base a questo caricare contenuti personalizzati

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Cosa può fare?

- Può essere usato per creare cookies
 - E' possibile conservare e leggere informazioni personali dell'utente direttamente dal suo computer

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Un esempio

```
<html>
<body>

<script type="text/javascript">
document.write("Hello World!")
</script>

</body>
</html>
```

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dm.unict.it - all rights reserved.

Spiegazioni

- Per inserire un Javascript all'interno di una pagina HTML, usiamo il tag `<script>`
`<script type="text/javascript">`
...
`</script>`
- Identificano dove inizia e finisce la porzione di javascript
- Viene interpretato dal motore interno del browser

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dm.unict.it - all rights reserved.

Spiegazioni

- Per inserire un Javascript all'interno di una pagina HTML, usiamo il tag `<script>`
`<script type="text/javascript">`
...
`</script>`
- Identificano dove inizia e finisce la porzione di javascript
- Viene interpretato dal motore interno del browser

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dm.unict.it - all rights reserved.

Spiegazioni

- L'istruzione **document.write** è un'istruzione standard di javascript per la scrittura a schermo
- Lo schermo è la pagina html stessa
- Il browser lo riconosce come codice valido ed esegue il codice

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dm.unict.it - all rights reserved.

Fine riga?

- Nei linguaggi tradizionali, ogni linea di istruzione deve finire con un ‘ ; ’
- In javascript sono opzionali
- Sono necessarie se si vogliono inserire più istruzioni in una singola riga.

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

...ed i vecchi browser?

- I browser che non supportano i tag `<script>` ed `</script>`
- Mostreranno il contenuto dello script a schermo senza formattarlo e/o interpretarlo.
- Per evitare questo comportamento, possiamo usare i commenti

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

...ed i vecchi browser?

- I browser che non supportano i tag `<script>` ed `</script>`
- Mostreranno il contenuto dello script a schermo senza formattarlo e/o interpretarlo.
- Per evitare questo comportamento, possiamo usare i commenti

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Dove mettere gli script?

- Gli scripts in una parte qualunque della pagina vengono eseguiti immediatamente, durante il caricamento.
- Non è sempre desiderabile!
- Potremmo volere che lo script entri in azione quando “succede qualcosa”.

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

La sezione head

- Gli script che devono essere eseguiti “on demand” vanno inseriti nella head section.
- Ci assicuriamo che lo script sarà caricato prima di qualunque altra cosa.
- In ogni caso non viene eseguito

La sezione head

- Gli script che devono essere eseguiti “on demand” vanno inseriti nella head section.
- Ci assicuriamo che lo script sarà caricato prima di qualunque altra cosa.
- In ogni caso non viene eseguito

La sezione body

- Gli script da avviare simultaneamente con il caricamento della pagina, vanno inseriti nella sezione body.
- Quando uno script viene inserito nella sezione body, verrà avviato durante il caricamento della pagina.

La sezione body

- Gli script da avviare simultaneamente con il caricamento della pagina, vanno inseriti nella sezione body.
- Quando uno script viene inserito nella sezione body, verrà avviato durante il caricamento della pagina.

Body ed Head...insieme

- E' possibile dotare il documento di un numero illimitato di scripts.
- E' possibile piazzarli contemporaneamente nella sezione head e nella sezione body.

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Body ed Head...insieme

- E' possibile dotare il documento di un numero illimitato di scripts.
- E' possibile piazzarli contemporaneamente nella sezione head e nella sezione body.

```
<html>
<head>
<script type="text/javascript">
</script>
</head>
<body>
<script type="text/javascript">
</script>
</body>
```

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Utilizzo di javascript esterni

- A volte è necessario dotare pagine diverse dello stesso javascript.
 - Riscriverlo ogni volta, è fuori discussione!
- E' possibile utilizzare un file **esterno**
 - Un semplice file di testo con suffisso **.js**

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Utilizzo di javascript esterni

- Gli script esterni non possono contenere i tags `<script>` ed `</script>`!
- Per utilizzare un file esterno, è necessario fornire l'indirizzo **src** dello script da caricare
 - All'interno del tag `<script>`

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Utilizzo di javascript esterni

- Gli script esterni non possono contenere i tags `<script>` ed `</script>`!
- Per utilizzare un file esterno, è necessario fornire l'indirizzo **src** dello script da caricare
- All'interno del tag `<script>`

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Variabili

- Una variabile è un contenitore per le nostre informazioni
- Il suo valore può cambiare nel corso dello script
- E' possibile riferirsi ad essa attraverso il suo nome.
- Leggerne o scrivere il valore

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Nomenclatura

- I nomi delle variabili sono CASE SENSITIVE
- Devono iniziare con una lettera o underscore
- JavaScript è case-sensitive!
 - Una variabile chiamata 'pippo' non è la stessa di 'PIPP0'

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Dichiarazione di variabili

- Per creare una variabile usiamo l'istruzione **var**:
`var strname = some value`
- E' possibile creare variabili **senza** l'aiuto dell'istruzione var:
`strname = some value`

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Assegnazione di un valore

- Per assegnare un valore, si usa l'operatore =

```
var strname = "Hege"
```

- Oppure

```
strname = "Hege"
```

```
var str1 = "pippo";  
var str2 = "saro";  
str1 = str2;  
str2 = "cambio";  
document.write(str1);
```

Scoping delle variabili

- Quando si dichiara una variabile all'interno di una funzione, la variabile risulta visibile dalla funzione.
- Quando si esce dalla funzione la variabile viene distrutta.
- Queste variabili sono chiamate locali.

Scoping delle variabili

- E' possibile usare lo stesso nome di variabile all'interno di funzioni differenti
- Ognuna di esse è riconosciuta correttamente **ESCLUSIVAMENTE** dalla funzione che la istanzia

Scoping delle variabili

- Se si dichiara una variabile esterna alle funzioni
 - Tutte le funzioni possono accedervi
- La vita di queste variabili
 - inizia con la dichiarazione
 - Finisce quando la pagina viene chiusa

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Istruzioni condizionali

- Molto spesso, vorremmo effettuare *azioni differenti* in base ad alcune *condizioni*
- Javascript permette di utilizzare le istruzioni condizionali

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Istruzioni condizionali

- **if** (cond) statement - solo se una particolare condizione è vera
- **if...else** statement - se la condizione risulta vera, verrà eseguita una porzione di codice, altrimenti un'altra
- **if...else if...else** statement - condizioni multiple

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

If Statement

- Se vogliamo eseguire una porzione di codice **solo se** una determinata condizione risulta esaudita.

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

If Statement

- Se vogliamo eseguire una porzione di codice **solo se** una determinata condizione risulta esaudita.

Esempio

```
<script type="text/javascript">
//Write a "Good morning" greeting if
//the time is less than 10
var d=new Date()
var time=d.getHours()

if (time<10)
{
document.write("<b>Good morning</b>")
}
</script>
```

Esempio

```
<script type="text/javascript">
//Write "Lunch-time!" if the time is 11
var d=new Date()
var time=d.getHours()

if (time==11)
{
document.write("<b>Lunch-time!</b>")
}
</script>
```

If...else Statement

- Possiamo usare **if...else**
- Se si vuole eseguire una porzione di codice quando una condizione è vera
- Ed un'altra porzione di codice quando è falsa

If...else Statement

- Possiamo usare **if...else**

```
if (condition)
{
  code to be executed if condition is true
}
```
- Se si vuole eseguire una porzione di codice quando una condizione è vera
- Ed un'altra porzione di codice quando è falsa

```
else
  code to be executed if condition is not true
```

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Esempio

```
<script type="text/javascript">
//If the time is less than 10,
//you will get a "Good morning" greeting.
//Otherwise you will get a "Good day" greeting.
var d = new Date()
var time = d.getHours()

if (time < 10)
{
  document.write("Good morning!")
}
else
{
  document.write("Good day!")
}
</script>
```

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

If...else if...else Statement

- Se si vogliono eseguire istruzioni multiple per condizioni multiple
- E' possibile annidare i costrutti if...else

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

If...else if...else Statement

- Se si vogliono eseguire istruzioni multiple per condizioni multiple
- E' possibile annidare i costrutti if...else

```
if (condition1)
{
  code to be executed if condition1 is true
}
else if (condition2)
{
  code to be executed if condition2 is true
}
else
{
  code to be executed if condition1 and
  condition2 are not true
}
```

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Esempio

```
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time<10)
{
document.write("<b>Good morning</b>")
}
else if (time>10 && time<16)
{
document.write("<b>Good day</b>")
}
else
{
document.write("<b>Hello World!</b>")
}
</script>
```

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Switch Statement

- Se le condizioni da verificare risultano troppe
- A meno solo del valore e non della logica di confronto
- E' possibile utilizzare il costrutto **switch...case**

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Switch Statement

- Se le condizioni da verificare risultano troppe
- A meno solo del valore e non della logica di confronto
- E' possibile utilizzare il costrutto **switch...case**

```
switch (n)
{
case 1:
executes code block 1
break
case 2:
execute code block 2
break
code to be executed if n is
different from case 1 and 2
}
```

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Esempio

```
<script type="text/javascript">
var d=new Date()
theDay=d.getDay()
switch (theDay)
{
case 5:
document.write("Finally Friday")
break
case 6:
document.write("Super Saturday")
break
case 0:
document.write("Sleepy Sunday")
break
default:
document.write("I'm looking forward this weekend!")
}
</script>
```

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Operatori

Operator	Description	Example	Result
+	Addition	x=2 y=2 x+y	4
-	Subtraction	x=5 y=2 x-y	3
*	Multiplication	x=5 y=4 x*y	20
/	Division	15/5 5/2	3 2,5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Operatori di assegnamento

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
=	x=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Operatori di confronto

Operator	Description	Example
==	is equal to	5==8 returns false
===	is equal to (checks for both value and type)	x=5 y="5" x==y returns true x===y returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Operatori Logici

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
	or	x=6 y=3 (x==5 y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

(c) 2009 - costantino pistagna - pistagna@dm.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Operatori Stringa

- Una stringa è un insieme di caratteri
- "Hello World!".
- L'operatore di concatenamento, si denota con il +

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Operatori Stringa

- Una stringa è un insieme di caratteri
- "Hello World!".
- L'operatore di concatenamento, si denota con il +

```
txt1="What a very"  
txt2="nice day!"  
txt3=txt1+txt2
```

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Operatori Stringa

- Una stringa è un insieme di caratteri
- "Hello World!".
- L'operatore di concatenamento, si denota con il +

```
txt1="What a very"  
txt2="nice day!"  
txt3=txt1+" "+txt2  
or  
txt1="What a very "  
txt3=txt1+txt2
```

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Operatore Condizionale

- JavaScript contiene un operatore condizionale
- Assegna un valore ad una variabile basandosi su una precisa condizione di verità

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

Operatore Condizionale

- JavaScript contiene un operatore condizionale

`variablename=(condition)?value1:value2`

- Assegna un valore ad una variabile basandosi su una precisa condizione di verità

Operatore Condizionale

- Se la variabile è uguale ad IPH...
 - allora metti la stringa “iPhone” nella variabile *browser*
- Se la variabile è diversa da IPH...
 - allora metti la stringa “Generic” nella variabile *browser*

Operatore Condizionale

- Se la variabile è uguale ad IPH...

- allora metti la stringa “iPhone” nella

`variable (browser)="IPH"? "iPhone": "Generic"`

- Se la variabile è diversa da IPH...

- allora metti la stringa “Generic” nella variabile *browser*

Alert Box

- An alert box is often used if you want to make sure information comes through to the user.
- When an alert box pops up, the user will have to click "OK" to proceed.

Alert Box

- An alert box is often used if you want to make sure information comes through to the user. `alert("sometext")`
- When an alert box pops up, the user will have to click "OK" to proceed.

Confirm Box

- A confirm box is often used if you want the user to verify or accept something.
- When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.
- If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

Confirm Box

- A confirm box is often used if you want the user to verify or accept something.
- When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed. `confirm("sometext")`
- If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

Prompt Box

- A prompt box is often used if you want the user to input a value.
- When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.
- If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Prompt Box

- A prompt box is often used if you want the user to input a value.
- When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.
- If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

JavaScript Functions

- To keep the browser from executing a script when the page loads, you can put your script into a function.
- A function contains code that will be executed by an event or by a call to that function.

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

JavaScript Functions

- You may call a function from anywhere within the page.
- Functions can be defined both in the <head> and in the <body> section of a document. However, to assure that the function is read/loaded by the browser before it is called, it could be wise to put it in the <head> section.

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

JavaScript Functions

- You may call a function from anywhere within the page.
- Functions can be defined both in the <head> and in the <body> section of a document. However, to assure that the function is read/loaded by the browser before it is called, it could be wise to put it in the <head> section.

```
<html>
<script type="text/javascript">
function displaymessage()
{
    alert ("Hello World")
}
</script>
</head>
<form>



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

(c) 2009 - costantino pistagna - pistagna@mi.unict.it - http://aladino.dmi.unict.it - all rights reserved.

How to Define a Function

- The syntax for creating a function is:

```
function functionname(var1,var2,...,varX)
{
some code
}
```

- var1, var2, etc are variables or values passed into the function. The { and the } defines the start and end of the function.

How to Define a Function

- A function with no parameters must include the parentheses () after the function name:

```
function functionname()
{
some code
}
```

The return Statement

- The return statement is used to specify the value that is returned from the function.
- So, functions that are going to return a value must use the return statement.

The return Statement

- The return statement is used to specify the value that is returned from the function.
- So, functions that are going to return a value must use the return statement.

```
function functionname()
{
x=a*b
return x
}
```