

JAVASCRIPT

Introduzione

JAVASCRIPT → PARTE 1 → INTRODUZIONE

CC BY

UN PO' DI STORIA

1995

La Netscape Communication sviluppa un linguaggio di scripting inizialmente chiamato **Mocha**, poi **LiveScript**, poi **JavaScript**.

Java allora era sinonimo di tecnologia avanzata e di successo

JavaScript è un linguaggio nato per rendere il web un ambiente dinamico.

JAVASCRIPT

UN PO' DI STORIA

1995

La Netscape Communication sviluppa un linguaggio di scripting inizialmente chiamato **Mocha**, poi **LiveScript**, poi **JavaScript**.

JavaScript è un linguaggio nato per rendere il web un ambiente dinamico.

Dennis Ritchie 1941-2011

Ken Thompson

C

C++

UNIX

Java

ORACLE

CISCO

amazon

YAHOO!

Google

JAVASCRIPT

UN PO' DI STORIA

1995

La Netscape Communication sviluppa un linguaggio di scripting inizialmente chiamato **Mocha**, poi **LiveScript**, poi **JavaScript**.

JavaScript è un linguaggio nato per rendere il web un ambiente dinamico.

JAVASCRIPT

1996

Dato il successo di JavaScript Microsoft decise di realizzare **JScript**.

1999

Nasce così la necessità di uno standard: **ECMAScript**.

2011

Viene rilasciata l'ultima versione di **ECMAScript**.

IL BROWSER

JavaScript è un linguaggio (di **scripting**) **interpretato**.



Javascript è un linguaggio nato per rendere il web un ambiente dinamico.

JAVASCRIPT

Il programma ospite che interpreta gli script è il **browser**.



Il **browser** offre inoltre un'API ben definita (denominata **BOM** o **DOM**) per **usare** l'ambiente di esecuzione (ovvero il browser stesso).



Il **browser** inoltre offre **strumenti di diagnostica** per aiutare il programmatore a scovare gli errori.



IL BROWSER

L'esecuzione all'interno di un programma ospite però presenta alcuni inconvenienti.



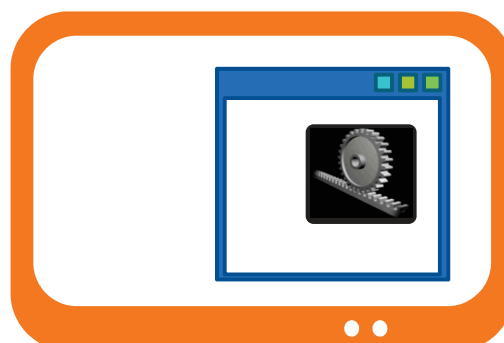
Javascript è un linguaggio nato per rendere il web un ambiente dinamico.

JAVASCRIPT

Esistono molti browser e non tutti i produttori seguono pedissequamente le specifiche (**incompatibilità**).



Tutti i browser consentono all'utente di **disabilitare** l'interprete javascript.



JAVASCRIPT

Strumenti

COSA CI SERVE?



DOVE INSERISCO IL MIO CODICE ?

Possiamo inserire il nostro codice Javascript in un documento HTML (o PHP) all'interno di un elemento **script** (che può essere annidato sia in **head** che in **body**).

```
<!-- omissis -->
```

```
<head>
```

```
<script>
```

```
// Istruzione 1
```

```
// Istruzione 2
```

```
// ...
```

```
// Istruzione n
```

```
</script>
```

```
<!-- omissis -->
```

```
<!-- omissis -->
```

```
<body>
```

```
<script>
```

```
// Istruzione 1
```

```
// Istruzione 2
```

```
// ...
```

```
// Istruzione n
```

```
</script>
```

```
<!-- omissis -->
```



+



DOVE INSERISCO IL MIO CODICE ?

Possiamo inserire il nostro codice Javascript in un documento HTML (o PHP) all'interno di un elemento **script** (che può essere annidato sia in **head** che in **body**).

L'elemento **script** dispone degli attributi **type** e **language** e possiamo trovarlo in una qualsiasi delle forme che seguono.

```
<!-- omissis -->
```

```
<script type="application/x-javascript">
```

```
/* ... */
```

```
</script>
```

```
<!-- omissis -->
```

```
<!-- omissis -->
```

```
<script type="text/javascript">
```

```
/* ... */
```

```
</script>
```

```
<!-- omissis -->
```

```
<!-- omissis -->
```

```
<script language="JavaScript">
```

```
/* ... */
```

```
</script>
```

```
<!-- omissis -->
```

DOVE INSERISCO IL MIO CODICE ?

Possiamo inserire il nostro codice Javascript in un documento HTML (o PHP) all'interno di un elemento **script** (che può essere annidato sia in **head** che in **body**).

L'elemento **script** dispone degli attributi **type** e **language** e possiamo trovarlo in una qualsiasi delle forme che seguono.

Esiste inoltre l'attributo **src** con il quale si può indicare un file esterno (.js).

```
<!-- omissis -->
<script src="pippo.js">
  // nessun contenuto
</script>
<!-- omissis -->
```

```
// Questo file si chiama pippo.js
/*
  Qui ci sono le istruzioni
  Javascript.
*/
```

DOVE INSERISCO IL MIO CODICE ?

Inoltre, ma inizialmente non lo faremo, sarebbe possibile eseguire codice Javascript anche in seguito al verificarsi di un **evento** o in risposta a un **link**.

```
<!-- omissis -->

<!-- omissis -->
```

```
<!-- omissis -->
<a href="javascript: /*codice Javascript*/ ">
<!-- omissis -->
```

COME LAVOREREMO NOI ?

Noi – salvo ove diversamente indicato – produrremo dei file **.js** con il codice Javascript, richiamandoli in **head** o in **body** a seconda dei casi.



```
<!-- omissis -->
  <head>
    <script src="pippo.js">
    </script>
  <!-- omissis -->
```

```
<!-- omissis -->
  <body>
    <script src="pippo.js">
    </script>
  <!-- omissis -->
```

JAVASCRIPT

Disclaimer

DISCLAIMER

Questo slideshow
non è un manuale!



<http://it.wikipedia.org/wiki/Javascript>



<http://it.wikibooks.org/wiki/Javascript>



JAVASCRIPT

Dichiarazione e assegnazione
Espressioni

LE VARIABILI: DICHIARAZIONE E ASSEGNAZIONE

JavaScript è un linguaggio **debolmente tipizzato**, per cui *non* occorre specificare il **tipo**.

La dichiarazione avviene mediante l'uso del costrutto **var**.

L'**assegnazione** di un **valore** avviene mediante l'utilizzo dell'operatore **=**

Dichiarazione ed **assegnazione** possono avvenire contestualmente.

```
var x;           // dichiaro una variabile
x = 3.14;       // assegno la costante 3,14
var y = 1976;   // dichiaro e inizializzo una variabile
```

Quali **valori** possiamo assegnare ad una variabile ?

ASSEGNAZIONE DI VALORI

$$\langle \text{variabile} \rangle = \left\{ \begin{array}{l} \langle \text{operando} \rangle \\ \langle \text{espressione} \rangle \end{array} \right.$$

OPERANDO

`<variabile>` = { `<operando>`
`<espressione>`

<code><operando></code>
<code><costante></code> <code><variabile></code> <code><risultato di una funzione></code>

OPERANDO: COSTANTE

`<variabile>` = { `<operando>`
`<espressione>`

<code><operando></code>
<code><costante></code> <code><variabile></code> <code><risultato di una funzione></code>

```
var num;  
num = 10;
```

```
const VOTOMASSIMO = 10;  
var voto = VOTOMASSIMO;
```

NOTA: le costanti
sono **sempre** in maiuscolo!



OPERANDO: COSTANTE

`<variabile>` = { `<operando>`
`<espressione>`

```
var num;
num = 10;
```

```
const VOTOMASSIMO = 10;
var voto = VOTOMASSIMO;
```

```
var num;
num = Math.PI;
```

`<operando>`
`<costante>`
`<variabile>`
`<risultato di una funzione>`

Con **Math.PI** si indica la costante pi greco di **Math**.

Per il momento vediamo **Math** come un *contenitore*, all'interno del quale si possono trovare sia costanti che sotto-programmi.

In Javascript i sotto-programmi sono detti **funzioni**.

OPERANDO: COSTANTE

`<variabile>` = { `<operando>`
`<espressione>`

```
var num;
num = 10;
```

```
const VOTOMASSIMO = 10;
var voto = VOTOMASSIMO;
```

```
var num;
num = Math.PI;
```

```
var nome;
nome = "Alessandro";
```

`<operando>`
`<costante>`
`<variabile>`
`<risultato di una funzione>`

OPERANDO: VARIABILE

`<variabile>` = { `<operando>`
`<espressione>`

<code><operando></code>
<code><costante></code>
<code><variabile></code>
<code><risultato di una funzione></code>

```
var num = 10;
var altroNumero;
altroNumero = num;
```

OPERANDO: RISULTATO DI UNA FUNZIONE

`<variabile>` = { `<operando>`
`<espressione>`

<code><operando></code>
<code><costante></code>
<code><variabile></code>
<code><risultato di una funzione></code>

```
var num;
num = Math.random();
```

Math.random() è una funzione che produce un numero casuale e si trova nel contenitore **Math**.

NOTA: le funzioni hanno **sempre** le parentesi!

ESPRESSIONE

`<variabile> = { <operando>
<espressione>`

<operando>
 <costante>
 <variabile>
 <risultato di una funzione>

<espressione>
 <operando> <operatore> <operando>
 <operatore> <operando>

OPERATORI ARITMETICI

`<variabile> = { <operando>
<espressione>`

<operando>
 <costante>
 <variabile>
 <risultato di una funzione>

<espressione>
 <operando> <operatore> <operando>
 <operatore> <operando>

```
var a = Math.random();
var b = Math.random();
var c = a+b;
```

In un'espressione più complessa un operando può essere sostituito da un'altra espressione.

```
var a = Math.random();
var b = Math.random();
var c = (a*b) + (a*b);
```

Op. aritmetici binari	
+	Addizione
-	Sottrazione
*	Moltiplicazione
/	Divisione
%	Modulo

ESEMPI

```
var x = 10;
var y = x;
x = x + 1;
y = 10;
y = y + x;
var z = y / 7;
// quanto vale z?
```

```
// 3
```

```
var x = 10;
var y = x;
x = x + 1;
y = 10;
y = y + x;
var z = y / 4;
// quanto vale z?
```

```
// 5.25
```

```
var raggio, circ;
const PIGRECO = 3.14;
raggio = 3.5;
circ = 2 * PIGRECO * raggio;
// quanto vale circ ?
```

```
// 21.98
```

```
var alfa = 0.50;
var beta = 50.0;
var gamma = (alfa + beta) / 5;
gamma = gamma - ( (alfa*2) / 10);
alfa = gamma % 5;
// quanto vale alfa?
```

```
// 0
```

Per vedere il valore di una variabile si può usare l'istruzione.

```
alert(<nome variabile>);
```



ESERCIZI

OPERATORI ARITMETICI

`<variabile>` = { `<operando>`
`<espressione>`

<operando>
<costante> <variabile> <risultato di una funzione>

<espressione>
<operando> <operatore> <operando> <operatore> <operando>

Op. aritmetici binari	
+	Addizione
-	Sottrazione
*	Moltiplicazione
/	Divisione
%	Modulo

Op. aritmetici unari	
-	Negativo
++	Autoincremento
--	Autodecremento

OPERATORE UNARIO -

`<variabile>` = { `<operando>`
`<espressione>`

<operando>
<costante> <variabile> <risultato di una funzione>

<espressione>
<operando> <operatore> <operando> <operatore> <operando>

```
var num;
num = 10;
num = -num;           // num vale -10
```

```
const MAX = 10;
var num;
num = -MAX;          // num vale -10
```

Op. aritmetici binari	
+	Addizione
-	Sottrazione
*	Moltiplicazione
/	Divisione
%	Modulo

Op. aritmetici unari	
-	Negativo
++	Autoincremento
--	Autodecremento

OPERATORI UNARI DI AUTOINCREMENTO E AUTODECREMENTO

`<variabile> = { <operando>
<espressione>`

```
var x;  
x = 10;  
x++; // x vale 11
```

```
var x;  
x = 10;  
x--; // x vale 9
```

<operando>
<costante> <variabile> <risultato di una funzione>

<espressione>
<operando> <operatore> <operando> <operatore> <operando>

Op. aritmetici binari	
+	Addizione
-	Sottrazione
*	Moltiplicazione
/	Divisione
%	Modulo

Op. aritmetici unari	
-	Negativo
++	Autoincremento
--	Autodecremento

OPERATORI LOGICI

`<variabile> = { <operando>
<espressione>`

```
var bello = true;  
var ricco = true;  
var fidanzato = (bello && ricco);
```

```
var bello = true;  
var ricco = true;  
var moroso = (bello || ricco);
```

```
var bello = true;  
var brutto = !(bello);
```

<operando>
<costante> <variabile> <risultato di una funzione>

<espressione>
<operando> <operatore> <operando> <operatore> <operando>

Op. aritmetici binari	
+	Addizione
-	Sottrazione
*	Moltiplicazione
/	Divisione
%	Modulo

Op. aritmetici unari	
-	Negativo
++	Autoincremento
--	Autodecremento

Op. logici	
&&	And
	Or
!	Not

OPERATORI CONDIZIONALI

`<variabile> = { <operando>
<espressione>`

```
var io = 10;
var tu = 12;

var sonoPiuGrande = (io > tu);
```

Le **condizioni** sono espressioni che assumono un valore booleano.

```
var a = 10;
var b = 3;
var c = -5;

var aMaggiore = (a>b) && (a>c);
```

<operando>
<costante> <variabile> <risultato di una funzione>

<espressione>
<operando> <operatore> <operando> <operatore> <operando>

Op. aritmetici binari	
+	Addizione
-	Sottrazione
*	Moltiplicazione
/	Divisione
%	Modulo

Op. aritmetici unari	
-	Negativo
++	Autoincremento
--	Autodecremento

Op. logici	
&&	And
	Or
!	Not

Op. condizionali	
==	Uguale
!=	Diverso
< e >	Minore e maggiore
<=	Minore o Uguale
>=	Maggiore o Uguale

OPERATORI DI AUTO-ASSEGNAZIONE

Una volta calcolato il valore di un'espressione questo è sempre assegnato ad una variabile. Se l'espressione coinvolge tale variabile allora possiamo usare gli operatori di auto-assegnazione.

```
var x = 10;
var y = 3;
y += x; // y = y+x
```

Op. di auto-assegnazione	
+=	Addizione a se stessa
-=	Sottrazione a se stessa
*=	Moltiplicazione a se stessa
/=	Divisione a se stessa
%=	Modulo a se stessa

OPERATORE DI CONCATENAMENTO E DI AUTO-CONCATENAMENTO

```
var s1 = "Ciao";
var s2 = "Alessandro";
var s3 = s1+s2;    // s3 vale "CiaoAlessandro"
```

```
var s1 = "Ciao";
var s2 = "Alessandro";
s1 += s2;         // s1 vale "CiaoAlessandro"
```

```
var s1 = "€ ";
var tot = 8.50;
var s2 = s1+tot;  // s2 vale "€ 8.5"
```

```
var s1 = "€ ";
var tot = 8.50;
s1 += tot;        // s1 vale "€ 8.5"
```

```
var tot = 15;
var s1 = " kg";
var s2 = tot+s1;  // s2 vale "15 kg"
```

```
var tot = 15;
var s1 = " kg";
tot += s1;        // tot vale "15 kg"
```

Op. di concatenamento

+	Concatena i due operandi
+=	Concatena a se stesso

Questi due operatori si applicano quando almeno uno dei due operandi è di tipo stringa.

ESEMPI

```
var x = 10;
var y = x;
x = (x+1) * y;
x += 10;
y += x + 1;
// quanto vale y?
```

```
// 131
```

```
var x = 10;
var y = x;
x = x + 1;
x += 10;
y += x + 1;
x = y / 4;
// quanto vale x?
```

```
// 8
```

```
var x = 10;
var y = x;
x = x + 1;
x += 10;
y += x + 1;
x = y / 4;
x++;
// quanto vale x?
```

```
// 9
```

```
var x = 10;
var y = x;
x = x + 1;
x += 10;
y += x + 1;
x = y / 5;
x++;
// quanto vale x?
```

```
// 7.4
```

ESEMPI

```
var x = 10;
const Y = x++;
var z = x+Y;
// quanto vale z?
```

```
// 21
```

```
var x = 10;
const Y = x--;
var z = x+Y;
// quanto vale z?
```

```
// 19
```

```
const TOTGIORNIMESE = 30;
var gg = 100;

var totGiorniAvanzati = gg % TOTGIORNIMESE;
// quanto vale totGiorniAvanzati ?
```

```
// 10
```

```
const TOTOREGIORNO = 24;
var ore = 100;

var totOreAvanzate = ore % TOTOREGIORNO;
// quanto vale totOreAvanzate ?
```

```
// 4
```

```
var nome = "Alessandro";
var anno = 1976;
var frase = nome + " è nato nel " + anno;
// quanto vale frase?
```

```
// Alessandro è nato nel 1976
```

```
var nome = "Ale";
var anno = 1976;
var frase = nome + " è nato nel " + anno;
frase += " e ha " + (2013-anno) + " anni.";
// quanto vale frase?
```

```
// Ale è nato nel 1976 e ha 37 anni
```

ESEMPI

```
var x = 10;
var y = -x;
var z = (x>y);
// quanto vale z?
```

```
// true
```

```
var x = 10;
var y = -x;
var z = (5*y)+(-x/5);
var xy = (x>y);
var xz = (x>z);
var xPiuGrandeDiTutti = (xy && xz);
// quanto vale xPiuGrandeDiTutti?
```

```
// true
```

```
var x = true;
var y = false;
var z = (x || y) && ( ! (x && y) );
// quanto vale z?
```

```
// true
```

```
var x = false;
var y = true;
var z = (x || y) && ( ! (x && y) ) && x;
// quanto vale z?
```

```
// false
```

ESERCIZI



JAVASCRIPT

Dichiarazione e assegnazione: le espressioni
Operatori di autoincremento e autodecremento
[corso avanzato]

OPERATORI UNARI DI AUTOINCREMENTO E AUTODECREMENTO

AUTOINCREMENTO

`<variabile> = { <operando> / <espressione> }`

<operando>
 <costante>
 <variabile>
 <risultato di una funzione>

<espressione>
 <operando> <operatore> <operando>
 <operatore> <operando>

Op. aritmetici binari

+	Addizione
-	Sottrazione
*	Moltiplicazione
/	Divisione
%	Modulo

Op. aritmetici unari

-	Negativo
++	Autoincremento
--	Autodecremento

```

var x;
x = 10;
x++; // x vale 11

var x;
x = 10;
x--; // x vale 9
    
```

DIAPPOSITIVA 34 ALESSANDRO URSOMANDO

Abbiamo già incontrato l'operatore di **autoincremento**

Non abbiamo ancora detto che questo operatore (oltre a lavorare sulla variabile a cui viene applicato) **restituisce un valore.**

OPERATORI UNARI DI AUTOINCREMENTO E AUTODECREMENTO

```

var x;
x = 10;
x++; // x vale 11
// e il valore restituito è 10
    
```

```

var x;
x = 10;
++x; // x vale 11
// e il valore restituito è 11
    
```

```

var x;
x = 10;
y = ++x; // x e y valgono ambedue 11
y = x++; // x vale 12 ma y vale 11
    
```

Abbiamo già incontrato l'operatore di **autoincremento**

Non abbiamo ancora detto che questo operatore (oltre a lavorare sulla variabile a cui viene applicato) **restituisce un valore.**

OPERATORI UNARI DI AUTOINCREMENTO E AUTODECREMENTO

```
var x;  
x = 10;  
x++; // x vale 11  
      // e il valore restituito è 10
```

```
var x;  
x = 10;  
++x; // x vale 11  
      // e il valore restituito è 11
```

```
var x;  
x = 10;  
y = ++x; // x e y valgono ambedue 11  
y = x++; // x vale 12 ma y vale 11
```

L'operatore di auto-incremento **prefisso** (messo prima) prima esegue l'incremento e poi restituisce il valore (della variabile a cui è applicato).

L'operatore di auto-incremento **postfisso** (messo dopo) prima restituisce il valore e dopo esegue l'incremento.

Un ragionamento analogo vale per l'operatore di **autodecremento**.

JAVASCRIPT

Math (e la funzione random)

MATH

Negli esempi di quanto visto finora abbiamo incontrato la costante **Math.PI** e la funzione **Math.random()**.

Ecco altri strumenti di **Math** che possono esserci utili.

E	numero di Eulero (~2.718)
PI	il valore di p greco (~3.14)

abs(x)	valore assoluto
ceil(x)	arrotonda all'intero superiore: $\lceil x \rceil$
exp(x)	e elevato a x
floor(x)	arrotonda all'intero inferiore: $\lfloor x \rfloor$
log(x)	logaritmo in base e
max(x,y,z,...,n)	massimo
min(x,y,z,...,n)	minimo
pow(x,y)	x elevato a y
random()	casuale
round(x)	arrotonda
sqrt(x)	radice quadrata

RANDOM

La funzione **Math.random()** restituisce un numero **reale** a caso nell'intervallo]0,1[.

```
var x = Math.random(); // un reale tra 0 e 1 (p.e. 0,56423)
```

Simbologia intervalli

[a,b]	Estremi compresi
]a, b[Estremi esclusi
[a,b[Estremo iniziale incluso
]a,b]	Estremo finale incluso

RANDOM

La funzione **Math.random()** restituisce un numero **reale** a caso nell'intervallo]0,1[.

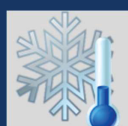
```
var x = Math.random(); // un reale tra 0 e 1 (p.e. 0,56423)
```

Ma noi (di certo) avremo bisogno di valori casuali di altro tipo.



ESEMPIO 1

Un colore del semaforo a caso.



ESEMPIO 2

Una temperatura terrestre a caso.

Come ottenere ciò che ci serve a partire da ciò che abbiamo?

RANDOM



ESEMPIO 1

Un colore del semaforo a caso.

Dovremo **mappare** ciascuno dei valori possibili (gli infiniti numeri reali in]0,1[) in uno dei valori desiderati (verde, giallo e rosso)

0,00000001
0,00000002
0,00000003
0,00000004
0,00000005



0,33333333
0,33333334
0,33333335
0,33333336
0,33333337



0,99999995
0,99999996
0,99999997
0,99999998
0,99999999



In questo caso basterà moltiplicare il valore casuale per 3 e trascurare la parte decimale: otterremo valori nell'intervallo [0,2] ai quali abbineremo i colori.

Vediamo qualche esempio.

x	x * 3	⌊ x * 3 ⌋	colore
0.012	0.036	0	rosso
0.350	1.050	1	giallo
0.412	1.236	1	giallo
0.781	2.343	2	verde
0.854	2.562	2	verde

RANDOM

La funzione **Math.random()** restituisce un numero **reale** a caso nell'intervallo]0,1[.

```
var x = Math.random(); // un reale tra 0 e 1 (p.e. 0,56423)
```

In generale, se vogliamo un valore a caso tra **k** valori possibili dobbiamo moltiplicare il valore casuale per **k** e trascurare la parte decimale.

```
var x = Math.floor(Math.random()*k); // [0, k-1]
```

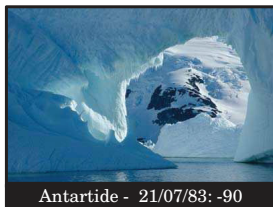
RANDOM



ESEMPIO 2

Una temperatura terrestre a caso.

Dovremo **mappare** ciascuno dei valori possibili (gli infiniti numeri reali in]0,1[) in uno dei valori desiderati [-90; +54].



In questo caso i valori desiderati sono 145 (90 negativi, lo zero e 54 positivi). Moltiplicando per 145 il valore casuale distribuirò gli infiniti valori reali possibili in 145 "pacchetti" di valori la cui parte intera sarà tra 0 e 144. Sommando il valore iniziale dell'intervallo otterrò valori all'interno dell'intervallo.

x	x * 145	⌊ x * 145 ⌋	⌊ x * 145 ⌋ - 90
0.012	1.74	1	-89
0.350	50.75	50	-40
0.412	59.74	59	-31
0.781	113.245	113	23
0.854	123.83	123	33

RANDOM

La funzione **Math.random()** restituisce un numero **reale** a caso nell'intervallo]0,1[.

```
var x = Math.random(); // un reale tra 0 e 1 (p.e. 0,56423)
```

In generale, se vogliamo un valore a caso tra **k** valori possibili dobbiamo moltiplicare il valore casuale per **k** e trascurare la parte decimale.

```
var x = Math.floor(Math.random()*k); // [0, k-1]
```

Concludendo, per ottenere un intero nell'intervallo **[a,b]** di dimensione **k**: moltiplico il valore casuale per **k**, trascuro la parte decimale e sommo **a**.

```
var x = Math.floor(Math.random()*k) + a; // [a, b]
```

ESEMPI

```
var raggio = 3.5;
var circonferenza;
circonferenza = 2 * Math.PI * raggio;
// quanto vale circonferenza ?
```

```
// 21.991148575128552
```

```
var raggio = 3.5;
var circonferenza;
circonferenza = 2 * Math.PI * raggio;
circonferenza = Math.round(circonferenza);
// quanto vale circonferenza ?
```

```
// 22
```

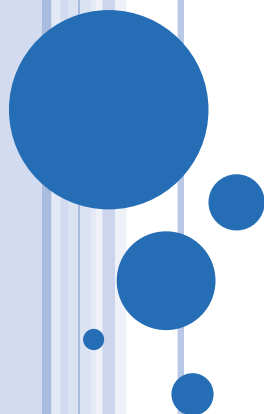
```
var numCasuale = Math.random();
numCasuale = numCasuale*10;
numCasuale = Math.floor(numCasuale);
// quanto vale numCasuale ?
```

```
// un numero a caso tra 0 e 9
```

```
var numCasuale =
    Math.floor(Math.random()*10);
// quanto vale numCasuale ?
```

```
// un numero a caso tra 0 e 9
```

ESERCIZI



Output

JAVASCRIPT

OUTPUT

JavaScript dispone di due tipi di output.
Mediante **pagina WEB** e mediante **finestra di popup**.



```
document.write("<h1>Hello World!</h1>");
```

```
alert("Ciao Mondo");
```

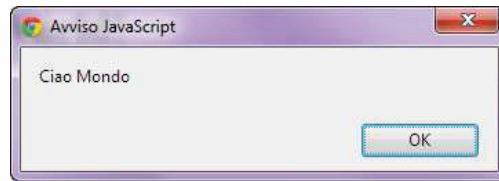
```
var saluto="<h1>Hello World!</h1>";  
document.write(saluto);
```

```
var saluto="Ciao Mondo";  
alert(saluto);
```

```
var saluto="Hello World!";  
document.write("<h1>"+saluto+"</h1>");
```

```
var saluto="Ciao";  
alert(saluto+"Mondo");
```

Hello World!



OUTPUT

JavaScript dispone di due tipi di output.
Mediante **pagina WEB** e mediante **finestra di popup**.



```
document.write(100);
```

```
alert(100);
```

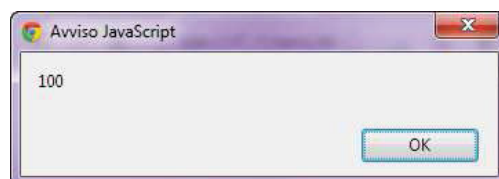
```
var num = 100;  
document.write(num);
```

```
var num = 100;  
alert(num);
```

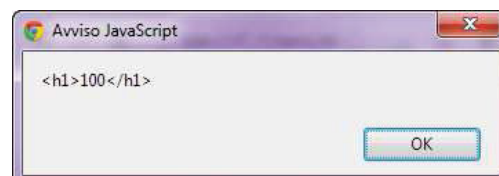
```
var num = 100;  
document.write("<h1>"+num+"</h1>");
```

```
var num = 100;  
alert("<h1>"+num+"</h1>");
```

100



100



ESEMPI

```
var numCasuale = Math.random();
numCasuale = numCasuale*6;
numCasuale = Math.floor(numCasuale);
numCasuale = numCasuale+35;
document.write("Hai la febbre a: " + numCasuale + ".");
// cosa appare a video ?
```

```
// una pag WEB che presenta
// un messaggio a caso
// tra i seguenti:
```

```
// Hai la febbre a: 35.
// Hai la febbre a: 36.
// Hai la febbre a: 37.
// Hai la febbre a: 38.
// Hai la febbre a: 39.
// Hai la febbre a: 40.
```

```
var numCasuale = Math.floor(7*Math.random());
numCasuale++;
var msg = "Oggi è il " + numCasuale + "° giorno.";
document.write(msg);
// cosa appare a video ?
```

```
// una pag WEB che presenta
// un messaggio a caso
// tra i seguenti:
```

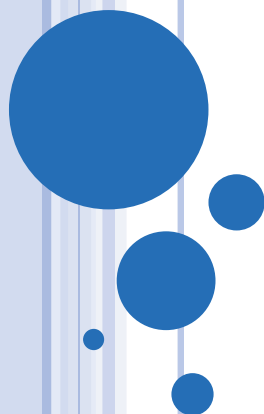
```
// Oggi è il 1° giorno.
// Oggi è il 2° giorno.
// Oggi è il 3° giorno.
// Oggi è il 4° giorno.
// Oggi è il 5° giorno.
// Oggi è il 6° giorno.
// Oggi è il 7° giorno.
```

ESEMPI

```
var x = 10;
var y = x;
var z = x*y;
alert(x);
alert(Math.floor(z*z/1000));
x = -z;
y *= x;
var msg = '<a href="http:\\www.mille.it\\">';
msg += Math.abs(y);
msg += "</a>";
document.write(msg);
// cosa appare a video?
```

```
// due popup che mostrano il valore 10
// e una pagina WEB che mostra un link sotto il testo 1000
```

ESERCIZI



Selezione

JAVASCRIPT

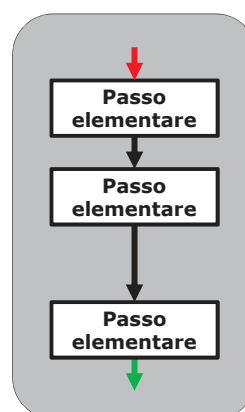
SEQUENZA

Pur senza renderlo esplicito abbiamo fin qui adoperato la più banale delle strutture di controllo del flusso: la **sequenza**.



I passi vengono eseguiti uno dopo l'altro incondizionatamente

SEQUENZA



SELEZIONE

Vediamo in che modo Javascript propone le strutture di selezione.



I passi vengono eseguiti solo dopo avere verificato una condizione

SELEZIONE

SELEZIONE

```
if (condizione) {  
    // istruzione/i  
}
```

```
if (condizione) {  
    // istruzione/i  
} else {  
    // istruzione/i  
}
```

```
switch (espressione) {  
    case valore1:  
        // istruzione/i  
        break;  
    case valore2:  
        // istruzione/i  
        break;  
    // ...  
    case valoreN:  
        // istruzione/i  
        break;  
    default:  
        // istruzione/i  
}
```

La **condizione** va sempre
in parentesi.



Le parentesi **graffe** non sono
indispensabili in presenza
di una singola istruzione.



ESEMPI

```
var x = (Math.floor(Math.random()*37));  
var msg = "La pallina si è fermata sul numero: "+x;  
if (x==0) {  
    msg += "\nVince il banco!";  
}  
alert(msg);  
// cosa appare a video?
```

```
// un popup che mostra il risultato  
// di una giocata alla roulette
```


ESERCIZI

ESEMPI

```
var x = (Math.floor(Math.random()*101))-50;
if (x < 0) {
  alert(x+" è un numero negativo");
} else {
  alert(x+" è un numero positivo");
}
// cosa appare a video?
```

```
// un popup che mostra un numero casuale tra -50 e 50 e
// che ci dice se è negativo o positivo
```

ESEMPI

```
document.write("<h1>");
document.write("Il tuo voto è: ");
var x = (Math.floor(Math.random()*8))+3;
document.write(x);
document.write("</h1>");

if (x >= 6) {
    document.write("positivo");
} else {
    document.write("negativo");
    document.write("<h1>");
    document.write("Il tuo voto di recupero è: ");
    var y = (Math.floor(Math.random()*8))+3;
    document.write(y);
    document.write("</h1>");
}
// che cosa appare a video ?
```

```
// Una pagina WEB che mostra un voto e - nel caso
// questo sia negativo - anche un voto di recupero
```

ESERCIZI

ESEMPI

```
var x = (Math.floor(Math.random()*3));
document.write("<h1>SEMAFORO</h1>");
if (x == 0) {
    document.write("rosso");
} else {
    if (x == 1) {
        document.write("giallo");
    } else {
        document.write("verde");
    }
}
// che cosa appare a video ?
```

```
// Una pagina WEB che mostra il colore di un semaforo a caso
```

ESEMPI

```
var x = (Math.floor(Math.random()*16));
document.write("<h1>Una cifra esadecimale</h1>");
if (x < 10) {
    document.write(x);
} else {
    if (x == 10) {
        document.write("A");
    } else {
        if (x == 11) {
            document.write("B");
        } else {
            if (x == 12) {
                document.write("C");
            } else {
                if (x == 13) {
                    document.write("D");
                } else {
                    if (x == 14) {
                        document.write("E");
                    } else {
                        document.write("F");
                    }
                }
            }
        }
    }
}
// che cosa appare a video ?
```

```
// Una pagina WEB che mostra una cifra esadecimale
```


ESERCIZI


SELEZIONE


```
if (condizione) {  
    // istruzione/i  
}
```


```
if (condizione) {  
    // istruzione/i  
} else {  
    // istruzione/i  
}
```

```
switch (espressione) {  
    case valore1:  
        // istruzione/i  
        break;  
    case valore2:  
        // istruzione/i  
        break;  
    // ...  
    case valoreN:  
        // istruzione/i  
        break;  
    default:  
        // istruzione/i  
}
```

L'**espressione** va sempre in parentesi e può essere sostituita da una semplice variabile. 

Le parentesi **graffe** sono obbligatorie a contenere tutti i casi e sono non richieste altrove. 

Se lo stesso blocco di istruzioni va eseguito per più di un caso, si possono elencare tutte le clausole **case** senza alcuna istruzione (né break) e far seguire a tale elenco il blocco di istruzioni. 

La clausola **default** viene eseguita nel caso in cui l'espressione non valga nessuno dei valori gestiti. 

ESEMPIO

```
var x = (Math.floor(Math.random()*16));
document.write("<h1>Una cifra esadecimale</h1>");
switch (x) {
  case 10:
    document.write("A");
    break;
  case 11:
    document.write("B");
    break;
  case 12:
    document.write("C");
    break;
  case 13:
    document.write("D");
    break;
  case 14:
    document.write("E");
    break;
  case 15:
    document.write("F");
    break;
  default:
    document.write(x);
}
// che cosa appare a video ?
```

```
// una pagina WEB che mostra una cifra esadecimale a caso.
```

ESEMPIO

```
var mese = Math.floor(Math.random()*12)+1;
switch(mese) {
  case 1:
  case 3:
  case 5:
  case 7:
  case 8:
  case 10:
  case 12:
    document.write(31);
    break;
  case 2:
    document.write(28);
    break;
  case 4:
  case 6:
  case 9:
  case 11:
    document.write(30);
    break;
}
```

```
// la quantità di giorni di un mese a caso
```

ESEMPIO

```
var mese = Math.floor(Math.random()*12)+1;
switch(mese) {
  case 2:
    document.write(28);
    break;
  case 4:
  case 6:
  case 9:
  case 11:
    document.write(30);
    break;
  default:
    document.write(31);
}
```

```
// la quantità di giorni di un mese a caso
```

ESEMPIO

```
var orario = Math.floor(Math.random()*7) + 8;
var msg = "<h1>Elenco pazienti prenotati (a partire dalle ore ";
msg += orario;
msg += "</h1>";
document.write(msg);
switch (orario) {
  case 8:
    document.write("8.00: Romolo <br/>");
  case 9:
    document.write("9.00: Numa Pompilio<br/>");
  case 10:
    document.write("10.00: Tullo Ostilio<br/>");
  case 11:
    document.write("11.00: Anco Marzio<br/>");
  case 12:
    document.write("12.00: Tarquinio Prisco<br/>");
  case 13:
    document.write("13.00: Servio Tullio<br/>");
  case 14:
    document.write("14.00: Tarquinio il superbo<br/>");
}
```

```
// l'elenco dei pazienti a partire da un certo orario
```

ESEMPIO

```
var x = 10;  
var y = Math.pow(2,x);  
y /= 10;  
if (Math.floor(y) > 15) {  
    y = -1;  
} else {  
    y += x;  
} // quanto vale y?
```

```
// -1
```

ESEMPIO

```
var x = 0;  
var y = x*2;  
if ( (x<0) || (x>0) )  
    y = 100;  
else  
    y = -100;  
// quanto vale y?
```

```
// -100
```

ESEMPIO

```
var n1 = (Math.random()*10)+1;
var n2 = (Math.random()*10)+1;
var n3 = Math.floor(n1+n2);
var x = 0;
if ( (n2<0) && (x>=0) && (n3>n2) )
    x++;
else
    x *= 2;
    x = Math.pow(2,x);
// quanto vale x?
```

```
// 1
```

ESEMPIO

```
var n1 = (Math.random()*10)+1;
var n2 = (Math.random()*10)+1;
var n3 = Math.floor(n1+n2);
var x = 0;
if ( (n2>0) && (x>=0) && (n3>n2) )
    x++;
else
    x *= 2;
    x = Math.pow(2,x);
// quanto vale x?
```

```
// 2
```


ESEMPIO

```
var n1 = (Math.random()*10)+1;
var n2 = (Math.random()*10)+1;
var n3 = Math.floor(n1+n2);
var x = 0;
if ( (n2>0) && (x>=0) && (n3>n2) ) {
  n3 = x++;
  if (n3-50 < 0)
    x = 100;
} else {
  x *= 2;
  x = Math.pow(2,x);
}
// quanto vale x?
```

```
// 100
```

```
// quanto vale n3?
```

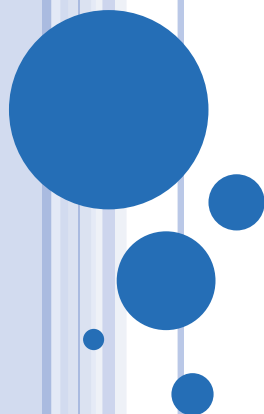
```
// 0
```

ESEMPIO

```
var n1 = (Math.random()*10)+1;
var n2 = (Math.random()*10)+1;
var n3 = Math.floor(n1+n2);
var x = 0;
if ( (n2>0) && (x>=0) && (n3>n2) ) {
  n3 = x++;
  if (n2-50 < 0)
    x = 100;
} else {
  x *= 2;
  x = Math.pow(2,x);
}
// quanto vale x?
```

```
// 100
```

ESERCIZI



Input

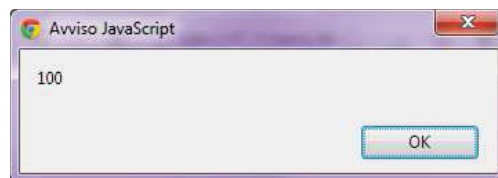
JAVASCRIPT

INTERAZIONE CON L'UTENTE

Abbiamo già visto il modo in cui mostrare un popup di output a video.



```
alert(100);
```



Vediamo adesso come usare dei popup che consentano all'utente di fornire delle informazioni.

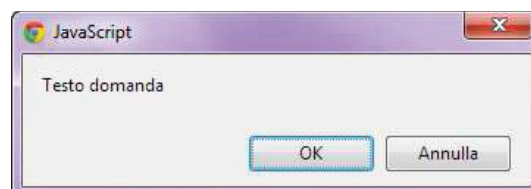


INTERAZIONE CON L'UTENTE

Il comando **confirm** propone un **popup** con il **testo** indicato e **due pulsanti**.



```
var risp = confirm("Testo domanda");
```



Se l'utente clicca su **<OK>** restituisce **true** altrimenti restituisce **false**.



```
if (risp == true) {  
    alert("Hai cliccato su <OK>.");  
} else {  
    alert("Hai cliccato su <Annulla>.");  
}
```

ESEMPIO

```
var msg = "Confermi la distruzione del computer?";
var risposta = confirm(msg);
msg = "<h1>";
if (risposta == true) {
    msg += "Computer distrutto!";
} else {
    msg += "Saggia scelta!";
}
msg += "</h1>";
document.write(msg);
// Che cosa appare a video ?
```

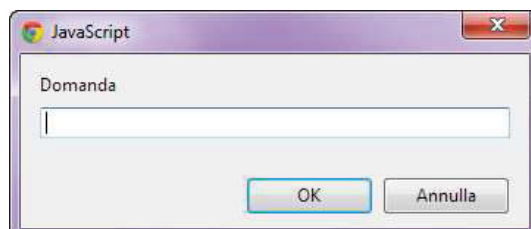
```
// Un popup con una domanda e poi una pagina WEB
// che mostra una scritta in h1 diversa in base alla
// scelta fatta dall'utente
```

INTERAZIONE CON L'UTENTE

Il comando **prompt** propone un **popup** con il **testo** indicato e **una casella di testo**.



```
var risp = prompt("Domanda");
document.write("<p>Hai risposto: <b>"+risp+"</b></p>");
```



Se l'utente clicca su **<OK>** restituisce **il testo inserito dall'utente** altrimenti restituisce **null**.



Il comando accetta anche un secondo parametro che viene utilizzato come risposta predefinita.



ESEMPI

```
var risp = prompt("Come ti chiami?", "Alessandro");
if ( (risp!=null) && (risp!="") ) {
    document.write("<h1>Ciao "+risp+"</h1>");
}
// che cosa appare a video ?
```

```
// Un popup che chiede il nome all'utente e che poi
// mostra una pagina di benvenuto personalizzata
// se questi ha fornito il suo nome.
```

ESEMPI

```
var str1 = prompt("Inserisci un numero");
if ( (str1!="") && (str1!=null) && !isNaN(str1)) {
    var str2 = prompt("Inserisci un numero");
    if ( (str2!="") && (str2!=null) && !isNaN(str2)) {
        var str3 = prompt("Inserisci un numero");
        if ( (str3!="") && (str3!=null) && !isNaN(str3)) {
            var num1 = Number(str1);
            var num2 = Number(str2);
            var num3 = Number(str3);
            var media = (num1+num2+num3)/3;
            alert(media);
        }
    }
}
// che cosa appare a video ?
```

```
// Tre popup di input che chiedono un numero
// e poi un popup che mostra la media dei tre numeri inseriti
```

JAVASCRIPT

Tipi e conversioni

TIPI

Javascript è un linguaggio *debolmente* tipizzato:
la variabile deve essere dichiarata ma non bisogna specificare il tipo.

```
var x;
```

- boolean
- string
- number
- function.

TIPI

Per il momento tralasciamo
completamente il tipo **function**.

TIPI

Javascript è un linguaggio *debolmente* tipizzato:
la variabile deve essere dichiarata ma non bisogna specificare il tipo.

```
var x;
```

- boolean
- string
- number
- function.

TIPI

- da numerico a stringa
- da stringa a numerico

CONVERSIONI

Javascript **non** distingue tra
numeri interi e in virgola mobile.

CONVERSIONI NUMERO → STRINGA

Per convertire un numero in una stringa ci
sono due metodi:

- concatenargli una stringa costante vuota
- usare la funzione **String()**

```
var numero = 100;  
var str1 = numero+"";           // variabile di tipo string contiene "100"  
var str2 = String(numero);     // variabile di tipo string contiene "100"
```

```
var numero = 3.14;  
var str1 = numero+"";           // variabile di tipo string contiene "3.14"  
var str2 = String(numero);     // variabile di tipo string contiene "3.14"
```

CONVERSIONI STRINGA → NUMERO

Number() converte la stringa nel numero corrispondente. La stringa deve contenere solo caratteri relativi al numero. Se contiene delle lettere la conversione non riesce.

Number()

Una variabile che non contiene un valore numerico vale **NaN**.



isNaN(x)	restituisce <i>true</i> se la variabile x vale NaN
typeof(x)	restituisce il <i>tipo</i> della variabile x

```
var str1 = "3.14";           // variabile di tipo string contiene "3.14"
var num1 = Number(str1);    // variabile di tipo number contiene 3.14
var str2 = "100";          // variabile di tipo string contiene "100"
var num2 = Number(str2);    // variabile di tipo number contiene 100
var str3 = "1.50 euro";     // variabile di tipo string contiene "1.50 euro"
var num3 = Number(str3);    // variabile di tipo number vale NaN
var str4 = "190 cm";        // variabile di tipo string contiene "190 cm"
var num4 = Number(str4);    // variabile di tipo number vale NaN
var str5 = "Esercizio 12";  // variabile di tipo string contiene "Esercizio 12"
var num5 = Number(str5);    // variabile di tipo number vale NaN
```

ESEMPI

```
var parteIntera = Math.floor(100*Math.random());
var parteDecimale = Math.floor(100*Math.random());
var numero = parteIntera + parteDecimale/100;
var numConvStringa = numero+"";
alert(numConvStringa);
alert(typeof(numConvStringa));
// cosa appare a video?
```

```
// due popup:
// il primo con un numero compreso tra 0.0 e 99.99
// ed il secondo con la scritta "string"
```

```
var parteIntera = Math.floor(100*Math.random());
var parteDecimale = Math.floor(100*Math.random());
var numero = parteIntera + parteDecimale/100;
alert(numero);
alert(typeof(numero));
// cosa appare a video?
```

```
// due popup:
// il primo con un numero compreso tra 0.0 e 99.99
// ed il secondo con la scritta "number"
```


ESERCIZI

ESERCIZIO



JAVASCRIPT

Dichiarazione e assegnazione: le espressioni
L'operatore modulo

L'OPERATORE MODULO

L'operatore **modulo** restituisce il resto di una divisione.

$$\begin{array}{r} 38 : 3 = 12 \\ - 8 \\ \hline 2 \end{array}$$

Viene usato principalmente per:

- scoprire se un numero è pari o dispari
- fare i calcoli relativi a date e orari
- organizzare in classi omogenee degli elementi ben distribuiti

Dividere un qualsiasi numero x per un certo numero n , darà come resto:

- 0 se x è un multiplo di n
- un numero compreso tra 1 e $(n-1)$ altrimenti
- darà x se x è minore di n

ESEMPI

```
var risp = prompt("Inserisci un numero");
if ((risp!="" && (risp!=null)) {
  var num = Number(risp);
  if (!isNaN(num)) {
    if (num%2==0) {
      document.write("<h1>"+num+" è pari.</h1>");
    } else {
      document.write("<h1>"+num+" è dispari.</h1>");
    }
  }
}
// che cosa appare a video ?
```

```
// Un popup che chiede un numero e poi una pagina
// WEB che dice se quel numero è pari o dispari
```

ESEMPI

```
var totSecondi    = Math.floor(Math.random()*10000);
var secondi       = totSecondi % 60;
var totMinuti     = Math.floor(totSecondi / 60);
var minuti        = totMinuti % 60;
var ore           = Math.floor(totMinuti / 60);

var messaggio     = "<p>" + totSecondi;
messaggio += " secondi corrispondono a: <br/>"
messaggio += ore + " ore <br/>";
messaggio += minuti + " minuti <br/>";
messaggio += secondi + " secondi </p>";
document.write(messaggio);
// cosa appare a video?
```

```
// una pagina WEB che mostra una quantità di secondi
// casuale (tra 0 e 9999) e che poi definisce tale
// quantità di secondi a quante ore, minuti e secondi
// (rimanenti) corrisponde, stampandole a video.
```

ESERCIZI

JAVASCRIPT

Iterazione

CICLO DI RIPETIZIONE SU CONTATORE



I passi elementari
vengono eseguiti
più volte

ITERAZIONE

Il ciclo di ripetizione su contatore
(**for**) si usa tutte le volte che sappiamo già a priori
quante volte va eseguito il ciclo.

```
for (inizializzazione; condizione; modifica) {  
    // istruzione/i  
}
```

ESEMPI

```
for (var i=0; i<10; i++) {  
    document.write("Ciao<br/>");  
}  
// cosa appare a video?
```

```
// una pagina WEB che mostra dieci righe con la scritta  
// "Ciao"
```

```
for (var i=0; i<10; i++) {  
    document.write(i+"<br/>");  
}  
// cosa appare a video?
```

```
// una pagina WEB che mostra i numeri da 0 a 9 su 10 righe  
// differenti.
```

ESEMPI

```
for (var i=50; i>=10; i--) {  
    if ( (i%10) == 0 )  
        document.write(i);  
}  
// cosa appare a video?
```

```
// una pagina WEB che mostra la stringa 5040302010
```

ESEMPI

```
var i = 10;
for (i=5; i<=100; i*=2) {
    document.write("<p>" + i + "</p>");
}
```

```
//cosa appare a video?
```

```
// Cinque paragrafi contenenti 5, 10, 20, 40, 80
```

ESEMPI

```
var i = 10;
var somma = 0;
for (i=5; i<=100; i*=2) {
    somma++;
}
document.write(somma+i);
//cosa appare a video?
```

```
// 165
```

ESEMPI

```
var i, j;
for (i=1; i<=10; i++) {
  for (j=1; j<=10; j++) {
    document.write(i*j + "-");
  }
  document.write("<br/>");
} // cosa appare a video?
```

```
// la tavola pitagorica
```

ESEMPI

```
var i, j;
document.write("<table>");
document.write("<tbody>");
for (i=1; i<=10; i++) {
  document.write("<tr>");
  for (j=1; j<=10; j++) {
    document.write("<td style='border: 1px red solid;'>"+i*j+"</td>");
  }
  document.write("</tr>");
}
document.write("</tbody>");
document.write("</table>");
// cosa appare a video?
```

```
// la tavola pitagorica ben formattata
```

ESEMPI

```
for (var i=10; i<=5; i++) {  
    alert(i);  
}
```

```
// cosa appare a video?
```

```
// niente
```

ESEMPI

```
for (var i=10; i>=5; i++) {  
    alert(i);  
}
```

```
// cosa appare a video?
```

```
// il codice produce un ciclo infinito  
// presentando quindi un'infinità di popup.  
// Ciascuno di essi mostrerà un numero (da 10 in poi).
```


ESEMPI

```
for (i=1; i<11; i++) {  
    alert(i);  
}
```

```
// dov'è l'errore?
```

```
// non ho dichiarato la variabile i
```

ESEMPI

```
var x=0;  
var y=100;  
var y+=x+1;  
if (y>0)  
    for (var i=1; i<=y; i++)  
        document.write("ciao<br/>");  
// dov'è l'errore?
```

```
// ho dichiarato due volte la variabile y
```

ESEMPI

```
var x=0;
var y=100;
y+=x+1;
if (y>0)
    for (var i=1; i<=y; i++)
        document.write("ciao<br/>");
// che cosa appare a video?
```

```
// 101 stringhe con la scritta «ciao»
```

ESERCIZI

CICLO DI RIPETIZIONE SU CONDIZIONE CON CONDIZIONE IN CODA



I passi elementari vengono eseguiti più volte

ITERAZIONE

```
do {
  // istruzione/i
} while (condizione);
```

Il ciclo di ripetizione su condizione si usa quando non sappiamo a priori **quante** volte va eseguito il ciclo, ma sappiamo **quando** va eseguito.

In particolare, il ciclo su condizione con verifica della condizione in coda (**do..while**) si usa nei seguenti casi.

- ❑ Quando ha senso porsi la domanda **solo dopo** avere già eseguito una prima volta il ciclo.
- ❑ Quando c'è un obiettivo da raggiungere mediante uno o più **tentativi**.

ESEMPI

```
document.write("<h1>Produzione di numeri casuali</h1>");
do {
  var x = (Math.floor(Math.random()*100))+1;
  document.write("<p>Ho prodotto: <b>"+x+"</b></p>");
} while (x!=100);
// che cosa appare a video?
```

```
// la scritta indicata e poi una serie imprecisata
// di numeri casuali compresi tra 1 e 100.
// L'ultimo numero è 100.
```

ESEMPI

```
do {  
  var x = prompt("Inserisci un numero da 1 a 10");  
} while ( (x == null) || (x == "") || isNaN(x) || (Number(x)>10) || (Number(x)<1) )  
document.write("<h1>OK</h1>");  
// che cosa fa questa porzione di codice?
```

```
// effettua l'input controllato
```

ESERCIZI

CICLO DI RIPETIZIONE SU CONDIZIONE CON CONDIZIONE IN TESTA



I passi elementari vengono eseguiti più volte

ITERAZIONE

```
while (condizione) {
  // istruzione/i
}
```

Il ciclo di ripetizione su condizione si usa quando non sappiamo a priori **quante** volte va eseguito il ciclo, ma sappiamo **quando** va eseguito.

In particolare, il ciclo su condizione con verifica della condizione in testa (**while**) si usa quando ha senso porsi la domanda **prima** di eseguire la prima volta il ciclo.

ESEMPI

```
var num = prompt("Inserisci un intero positivo");
// supponiamo l'utente abbia seguito le indicazioni
var totCifre = 0;
while (num > 0) {
  num = Math.floor(num / 10);
  totCifre++;
}
document.write("<h1>Il tuo numero era composto da "+totCifre+" cifre.</h1>");
// che cosa fa questa porzione di codice?
```

```
// conta il numero di cifre di un intero dato in input
```

ESEMPI

```
// produco un orario casuale
var h = Math.floor(Math.random()*24);
var m = Math.floor(Math.random()*60);

// preparo e stampo l'orario di benvenuto
var msg = "Sei arrivato alle ore "+h+":";
if (m<10)
    msg += "0";
msg += m;
msg = "<h1>"+msg+"</h1>"
document.write(msg);

// aggiungo una quantità di minuti a caso
// finchè non ottengo un orario o clock
while (m != 0) {
    var mCasuali = Math.floor(Math.random()*10);
    h = h + Math.floor((mCasuali+m) / 60);
    h %= 24;
    m = ((mCasuali+m) % 60);
}

// messaggio
document.write("<h1>Sei uscito alle "+(h+1)+" :00</h1>");
```

ESERCIZI

JAVASCRIPT

Tabella di traccia

TABELLA DI TRACCIA

La tabella di traccia è uno strumento con il quale **simuliamo** l'esecuzione dell'algoritmo. Questa attività ci permette di valutare la correttezza del nostro lavoro.

TABELLA DI TRACCIA

TABELLA DI TRACCIA

La tabella di traccia è uno strumento con il quale **simuliamo** l'esecuzione dell'algoritmo. Questa attività ci permette di valutare la correttezza del nostro lavoro.

TABELLA DI TRACCIA

```
1 var x1 = 10;
2 const x2 = x1;
3 x1++;
```

istruz.	x1	x2
1	10	
2		10
3	11	

Per prima cosa si numerano le istruzioni per poterci fare riferimento.

Poi si crea una tabella dove tenere traccia dell'evoluzione di tutte le variabili.

TABELLA DI TRACCIA

La tabella di traccia è uno strumento con il quale **simuliamo** l'esecuzione dell'algoritmo. Questa attività ci permette di valutare la correttezza del nostro lavoro.

TABELLA DI TRACCIA

```
1 var x = prompt("Inserisci un intero");
  // Supponiamo l'utente inserisca 12
2 if (x % 2 != 0) {
3   alert("dispari")
  } else {
4   alert("pari")
  }
```

istruz.	output	x	x%2!=0
1	inser..	12	
2			falso
4	pari		

Numeriamo solo le righe che contengono istruzioni.

Nella tabella teniamo traccia anche dell'output prodotto..

In caso di input dobbiamo ipotizzare noi un valore.

..e del valore assunto durante l'esecuzione da tutte le condizioni.

TABELLA DI TRACCIA

La tabella di traccia è uno strumento con il quale **simuliamo** l'esecuzione dell'algoritmo. Questa attività ci permette di valutare la correttezza del nostro lavoro.

TABELLA DI TRACCIA

```
1  var i = 0;  
   do {  
2     var x = Math.floor(Math.random()*101);  
3     i++;  
4   } while (x != 0);  
5  alert(i);
```

Anche nel caso di produzione di numeri casuali si procede per ipotesi.

istruz.	i	x	x != 0	output
1	0			
2		88		
3	1			
4			vero	
2		44		
3	2			
4			vero	
2		22		
3	3			
4			vero	
2		0		
3	4			
4			falso	
5				4