



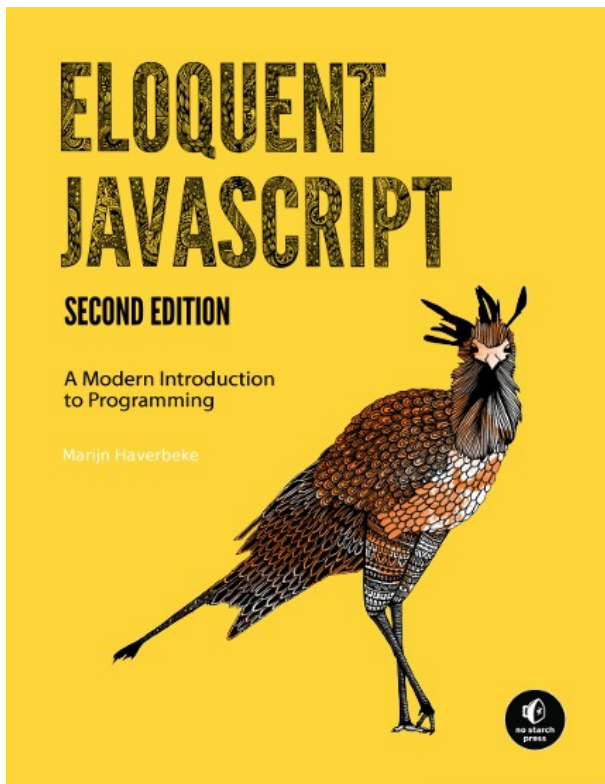
# Fondamenti di Informatica

Alma Artis

Francesca Pratesi (ISTI, CNR)

Logica e istruzioni condizionali

# Libri e Riferimenti



- Capitolo 1
- Capitolo 2

Eloquent Javascript – Second Edition  
Marijn Haverbeke  
Licensed under CC license.  
Available here: <http://eloquentjavascript.net/>



# **ESERCITAZIONE**

# Proposizioni logiche

- Scrivere un programma che stampi il cubo del numero 244.
- Scrivere un programma che stampi le potenze di 2 partendo dall'esponente 0 fino all'esponente 5. Il programma deve quindi stampare il risultato delle seguenti espressioni  $2^0$ ,  $2^1$ ,  $2^2$ ,  $2^3$ ,  $2^4$ ,  $2^5$ .
- Scrivere un programma che calcoli e stampi il lato di un quadrato con perimetro 70 cm.
- Scrivere un programma che calcoli e stampi l'area di un cerchio con raggio 10.5 m.



# Possibili soluzioni

```
/*
```

Scrivere un programma che stampi il cubo del numero 244.

```
*/
```

```
console.log("Esercizio 1");  
console.log(244**3);  
console.log(Math.pow(244,3));  
console.log(Math.pow(3,244)); // questo è sbagliato
```

```
/*
```

Scrivere un programma che stampi le potenze di 2 partendo dall'esponente 0 fino all'esponente 5. Il programma deve quindi stampare il risultato delle seguenti espressioni  $2^0$ ,  $2^1$ ,  $2^2$ ,  $2^3$ ,  $2^4$ ,  $2^5$ .

```
*/
```

```
console.log("Esercizio 2");  
console.log(2**0);  
console.log(2**1);  
console.log(2**2);  
console.log(2**3);  
console.log(2**4);  
console.log(2**5);
```



# Possibili soluzioni

```
/*
```

```
Scrivere un programma che calcoli e stampi il lato di un quadrato con perimetro 70 cm.
```

```
*/
```

```
console.log("Esercizio 3");
```

```
console.log(70/4);
```

```
// oppure
```

```
var perimetro = 70;
```

```
console.log("Il lato di un quadrato con perimetro "+perimetro+" cm è "+perimetro/4);
```

```
// oppure
```

```
console.log(perimetro/4);
```

```
/*
```

```
Scrivere un programma che calcoli e stampi l'area di un cerchio con raggio 10.5 m.
```

```
*/
```

```
console.log("Esercizio 4");
```

```
console.log(Math.PI); // per vedere quanto vale il pi greco
```

```
//  $A = \pi * r^2 \rightarrow r = \sqrt{A/\pi}$ 
```

```
console.log(3.14*10.5**2); // soluzione approssimata
```

```
// oppure
```

```
console.log(Math.PI*(10.5**2)); // più precisa
```

```
// oppure alternativa con variabili; le parentesi non sono necessarie ma è possibile metterle per chiarezza
```

```
var raggio = 10.5;
```

```
console.log(Math.PI*raggio**2);
```





# **INTRODUZIONE ALLA LOGICA**

# Proposizioni logiche

- Si dice **proposizione logica** una frase alla quale sia possibile attribuire un **valore di verità secondo un criterio oggettivo**
- Stabilire il valore di verità (o valore logico) di una proposizione, significa stabilire se questa è vera oppure falsa:
  - assegneremo il valore V/T a un'affermazione vera,
  - assegneremo il valore F a un'affermazione falsa.
- Esempi
  - Sono proposizioni logiche:
    - Il leone è un pesce  $\rightarrow$  F
    - Il ferro è un metallo  $\rightarrow$  V
    - il numero 2 è un numero pari  $\rightarrow$  V
  - Non sono proposizioni logiche:
    - Smettila di piangere! (Frase esclamativa, non ha valore di verità)
    - Marco è andato al cinema? (Frase interrogativa, non ha valore di verità)
    - Questo libro è interessante (Frase il cui valore di verità è soggettivo)





# I tre principi della logica

- La logica classica si basa su 3 principi fondamentali:
- **Principio di identità**: ogni oggetto del pensiero logico è uguale solamente a sé stesso
- **Principio di non contraddizione**: una proposizione logica non può essere contemporaneamente vera e falsa
- **Principio del terzo escluso**: una proposizione logica è sicuramente vera o falsa, non esiste una terza opzione



# Altri tipi di logica

- La logica a due valori (vero e falso) non è l'unica possibile
- Ad esempio, esistono:
  - la logica a tre valori: true, false, null
  - la logica fuzzy, in cui ogni proposizione ha un grado di verità



# Rappresentazione logica

- È conveniente rappresentare ogni proposizione (semplice) con un simbolo
  - $p$  = Oggi vado all'Università
  - $q$  = Oggi vado al mare
  - $s$  = Oggi c'è il sole
- Possiamo così ragionare in modo più astratto



# Proposizioni semplici o composte

- Si dicono **proposizioni semplici** o atomiche le proposizioni costituite da un soggetto e un predicato oppure da un soggetto e un oggetto collegato da un predicato.
- Nella realtà si usano proposizioni più complesse che si ottengono collegando le proposizioni semplici mediante particelle linguistiche come non, e, o, o...o, se...allora, se e solo se. Ognuna di queste particelle si dice connettivo logico.
- Si dicono **proposizioni composte** o molecolari le proposizione ottenute collegando due o più proposizioni semplici mediante connettivi logici.



# Connettivi Logici

- I connettivi logici rappresentano le particelle che permettono di indicare alcune condizioni che legano proposizioni semplici
  - non, e, o, o...o, se...allora, se e solo se
- Hanno degli specifici simboli in logica
  - $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\oplus$ ,  $\rightarrow$ ,  $\leftrightarrow$
- Permettono di creare proposizioni logiche composte
- Sono universali e indipendenti dal linguaggio
- Hanno un significato preciso



# Le tabelle di verità

- Le tabelle di verità sono tabelle usate nella logica per **determinare se**, attribuiti i valori di verità alle proposizioni che la compongono, **una determinata proposizione è vera o falsa**
- La tabella di verità quindi si applica a qualsiasi operatore logico
- Le condizioni di verità o falsità di qualunque enunciato che si ottiene applicando quell'operatore è determinato interamente ed esclusivamente da quelle degli enunciati più semplici a cui si applica
- Per costruirle dobbiamo prevedere tutte le combinazioni dei valori vero/falso di tutti gli enunciati semplici



# Negazione

- La negazione di una proposizione  $p$  è la proposizione "**non**  $p$ ", risulta **vera se  $p$  è falsa e falsa se  $p$  è vera**.
- La negazione si indica con il simbolo  $\neg$  anteposto alla proposizione da negare.
- Esempio. Se  $p$ : "Oggi vado all'università", allora  $\neg p$ : "Oggi non vado all'università".

$p$	$\neg p$
T	F
F	T



# Congiunzione

- La congiunzione di due proposizioni  $p$  e  $q$  è la proposizione " $p$  e  $q$ ", risulta **vera solo se le due proposizioni sono entrambe vere**. Basta che una delle due sia falsa perché sia falsa
- La congiunzione si indica con il simbolo  $\wedge$ .
- Esempio. Se  $p$ : "Oggi vado all'università" e  $q$ : "Oggi vado al mare" allora  $p \wedge q$ : "Oggi vado all'università e al mare".

<b>p</b>	<b>q</b>	<b><math>p \wedge q</math></b>
T	T	T
T	F	F
F	T	F
F	F	F



# Disgiunzione

- La disgiunzione di due proposizioni  $p$  e  $q$  è la proposizione " $p \vee q$ ", essa è **falsa solo se le due proposizioni sono entrambe false**. Basta che una delle due sia vera perché sia vera.
- La disgiunzione si indica con il simbolo  $\vee$ .
- Esempio. Se  $p$ : "Oggi vado all'università" e  $q$ : "Oggi vado al mare" allora  $p \vee q$ : "Oggi vado all'università o al mare".

<b>p</b>	<b>q</b>	<b><math>p \vee q</math></b>
T	T	T
T	F	T
F	T	T
F	F	F

# Disgiunzione esclusiva

- La disgiunzione esclusiva di due proposizioni  $p$  e  $q$  è la proposizione " $\bullet p \bullet q$ ", essa è vera solo se una proposizione è vera e l'altra è falsa.
- La disgiunzione esclusiva si indica con il simbolo  $\oplus$ .
- Esempio. Se  $p$ : "Oggi vado all'università" e  $q$ : "Oggi vado al mare" allora  $p \oplus q$ : "Oggi o vado all'università o vado al mare"

<b>p</b>	<b>q</b>	<b><math>p \oplus q</math></b>
T	T	F
T	F	T
F	T	T
F	F	F

# Implicazione

- L'implicazione di due proposizioni  $p$  e  $q$  è la proposizione "**se  $p$  allora  $q$** ", essa risulta **falsa solo se  $p$  è vera e  $q$  è falsa**.
- L'implicazione si indica con il simbolo  $\rightarrow$  o  $\Rightarrow$ .
- Esempio. Se  $p$ : "Oggi c'è il sole" e  $q$ : "Oggi vado al mare" allora  $p \rightarrow q$ : "Se oggi c'è il sole allora vado al mare".

<b>p</b>	<b>q</b>	<b><math>p \rightarrow q</math></b>
T	T	T
T	F	F
F	T	T
F	F	T

# La doppia implicazione

- La doppia implicazione di due proposizioni  $p$  e  $q$  è la proposizione " $p$  **se e solo se**  $q$ ", è vera solo se  $p$  e  $q$  sono entrambe vere o entrambe false.
- La doppia implicazione si indica con il simbolo  $\leftrightarrow$  o  $\Leftrightarrow$ .
- Esempio. Se  $p$ : "Oggi vado al mare" e  $q$ : "Oggi c'è il sole" allora  $p \leftrightarrow q$ : "Oggi vado al mare se e solo se c'è il sole"

$p$	$q$	$p \leftrightarrow q$
T	T	T
T	F	F
F	T	F
F	F	T

Infatti:

$p$	$q$	$p \rightarrow q$	$q \rightarrow p$	$(p \rightarrow q) \wedge (q \rightarrow p)$
T	T	T	T	T
T	F	F	T	F
F	T	T	F	F
F	F	T	T	T

# Uso di tabelle di verità

- Le tabelle di verità consentono di verificare (in modo formale) quando una certa proposizione (complessa quanto si vuole) è vera o falsa
- $(\neg p \vee (p \wedge q)) \oplus \neg(p \rightarrow q)$

		1	2	3	4	5	
<b>p</b>	<b>q</b>	<b><math>\neg p</math></b>	<b><math>p \wedge q</math></b>	<b><math>1 \vee 2</math></b>	<b><math>p \rightarrow q</math></b>	<b><math>\neg(p \rightarrow q)</math></b>	<b><math>3 \oplus 5</math></b>
T	T	F	T	T	T	F	T
T	F	F	F	F	F	T	T
F	T	T	F	T	T	F	T
F	F	T	F	T	T	F	T

# Equivalenza di espressioni logiche

- Due proposizioni sono equivalenti se e solo se hanno gli stessi valori di verità, ovvero la loro tavola di verità coincide.
- Alcune equivalenze sono:
  - $\neg\neg p = p$
  - $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$
  - $p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$
  - $(p \Rightarrow q) = (\neg q \Rightarrow \neg p)$  (proposizione contronominale)
  - $\neg(p \wedge q) = \neg p \vee \neg q$
  - $\neg(p \vee q) = \neg p \wedge \neg q$

} Leggi di De Morgan



# Esercizi per casa

- Dimostrare le seguenti equivalenze con le tabelle di verità
  - $\neg\neg p = p$
  - $p \wedge (q \vee r) = (p \wedge q) \vee (p \wedge r)$
  - $p \vee (q \wedge r) = (p \vee q) \wedge (p \vee r)$
  - $(p \Rightarrow q) = (\neg q \Rightarrow \neg p)$
  - $\neg(p \wedge q) = \neg p \vee \neg q$
  - $\neg(p \vee q) = \neg p \wedge \neg q$



# Sommario

- Proposizioni logiche
- Connettivi logici
- Tabelle di verità







# **OPERATORI ARITMETICI**

# Operatori compositi

Operatore	Esempio	Equivalente a
<code>+=</code>	<code>x+=y</code>	<code>x=x+y</code>
<code>-=</code>	<code>x-=y</code>	<code>x=x-y</code>
<code>*=</code>	<code>x*=y</code>	<code>x=x*y</code>
<code>/=</code>	<code>x/=y</code>	<code>x=x/y</code>



# Incremento e decremento

- È possibile incrementare/decrementare di 1 una variabile
- Con gli operatori ++ e --

```
x=2;  
x++;  
console.log(x); // stampa 3
```

```
x--;  
console.log(x); // stampa 2
```



# Incremento e decremento (2)

- Esiste anche una funzionalità diversa degli operatori di incremento/decremento, dove il valore viene modificato prima dell'uso

```
x=2;  
console.log(x++); // stampa 2  
console.log(x); // stampa 3  
  
console.log(++x); // stampa 4
```





# **ISTRUZIONI CONDIZIONALI**

# Istruzioni Condizionali

- Tutti i programmi in JavaScript visti finora prevedono istruzioni eseguite in sequenza
- Con questi programmi è possibile solo affrontare problemi che richiedano l'esecuzione di una sequenza di calcoli
- A volte è necessario eseguire del codice solo se si verifica una certa condizione
  - Es. Se la temperatura ambientale scende sotto i 18 gradi accendo il riscaldamento



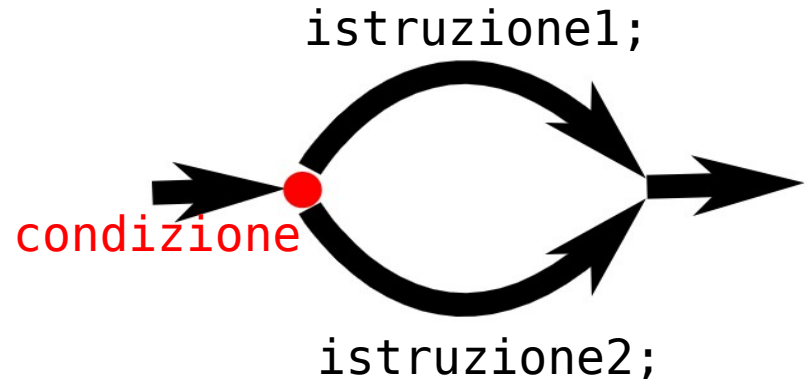
# Istruzioni Condizionali (2)

- Per poter eseguire una istruzione (o un blocco di istruzioni) solo in determinati casi è necessario introdurre le istruzioni condizionali
- Istruzioni condizionali: una istruzione viene eseguita solo se una specifica condizione è verificata
- Condizione: espressione booleana



# Istruzione IF-THEN-ELSE

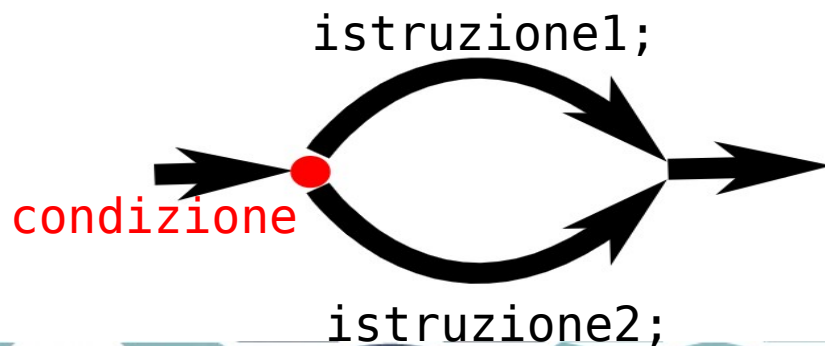
- Sintassi:  
if (condizione)      ← Espressione booleana  
    istruzione1;   ← Ramo «then»  
else  
    istruzione2;   ← Ramo «else»





# Istruzione IF-THEN-ELSE (2)

- Semantica:
  - viene valutata l'espressione booleana «condizione»
  - se «condizione» è vera allora viene eseguita l'«istruzione1», altrimenti («condizione» è falsa) viene eseguita l'«istruzione2»
  - successivamente, l'esecuzione riprende dall'istruzione successiva al blocco if-then-else



# Esempio

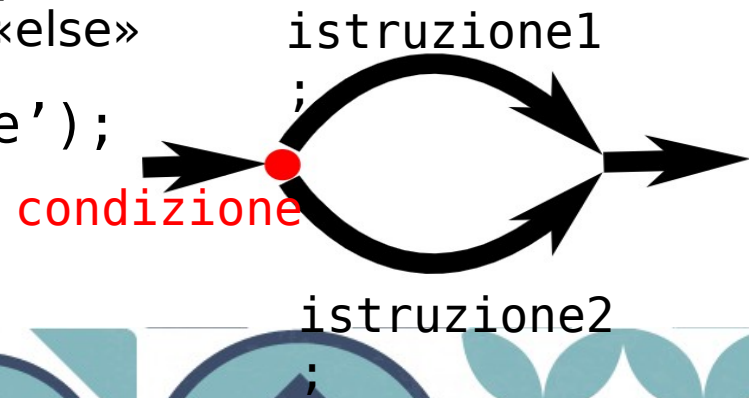
- Scrivere un programma che stampi la stringa "fa caldo" se la temperatura è superiore a 30 gradi e "si sta bene" altrimenti

```
•   var t=27;  
    if (t>30)  
        console.log('Fa caldo');  
    else  
        console.log('Si sta bene');
```

Espressione booleana

Ramo «then»

Ramo «else»



# Istruzione IF semplice (o IF-THEN)

- In alcuni casi (quando non bisogna fare nulla se la condizione è false), non è necessario prevedere un ramo else
- Sintassi:  
if (condizione)      ← Espressione booleana  
    istruzione1;      ← Ramo «then»



# Cosa può contenere la condizione?

- Come detto prima, la condizione è una qualsiasi espressione booleana



# Cosa può contenere l'istruzione dei rami THEN ed ELSE?

- Una qualsiasi istruzione:
  - sia tra quelle viste finora (es. una stampa, un assegnamento)
  - sia tra quelle che vedremo nelle prossime lezioni
- Nota: l'istruzione può essere anche un if-then o un if-then-else!
  - In questo caso si parla di **if annidati**
- Ogni ramo può essere composto anche da più istruzioni, che vengono eseguite normalmente
  - In questo caso è necessario inserire il **blocco di codice** tra parentesi graffe



# ESEMPIO

- Scrivere un programma che dati un mese (espresso in numeri da 1 a 12) e un anno stampi il mese successivo
  - *es: se abbiamo Gennaio 2012, il mese successivo è Febbraio 2012;*
  - *es: se abbiamo Dicembre 2022, il mese successivo è Gennaio 2013*



# ESEMPIO

- Scrivere un programma che dati un mese (espresso in numeri da 1 a 12) e un anno stampi il mese successivo
  - es: se abbiamo Gennaio 2012, il mese successivo è Febbraio 2012;
  - es: se abbiamo Dicembre 2022, il mese successivo è Gennaio 2013

```
var mese = 10;
var anno = 2011;
var mese_successivo, anno_successivo;
/* se il mese è dicembre allora il prossimo mese è gennaio e l'anno aumenta di uno */
if(mese==12){
    mese_successivo = 1;
    anno_successivo = anno+1;
}
/* altrimenti l'anno resta lo stesso e il mese aumenta di uno */
else{
    mese_successivo = mese+1;
    anno_successivo = anno;
}
console.log('Dato il '+mese+' '+anno+', il mese successivo è '+mese_successivo+' '+anno_successivo);
```



# ESEMPIO

- Scrivere un programma che dati un mese (espresso in numeri da 1 a 12) e un anno stampi il mese successivo
  - es: se abbiamo Gennaio 2012, il mese successivo è Febbraio 2012;
  - es: se abbiamo Dicembre 2022, il mese successivo è Gennaio 2013

```
var mese = 10;
var anno = 2011;
var mese_successivo, anno_successivo;
/* se il mese è dicembre allora il prossimo mese è gennaio e l'anno aumenta di uno */
if(mese==12){
    mese_successivo = 1;
    anno_successivo = anno+1;
}
/* altrimenti l'anno resta lo stesso e il mese aumenta di uno */
else{
    mese_successivo = mese+1;
    anno_successivo = anno;
}
console.log('Dato il '+mese+' '+anno+', il mese successivo è '+mese_successivo+' '+anno_successivo);
```

Nota: Le parentesi graffe sono necessarie per indicare che ogni ramo è composto da due istruzioni; inoltre, ho aggiunto una tabulazione all'inizio delle righe dei due rami per rendere più chiaro sono blocchi interni





# Indentazione

- Indentare il codice significa utilizzare degli spazi o tabulazioni per distinguere gerarchicamente le diverse parti del codice
- In alcuni linguaggi (es. Python) l'indentazione ha un significato ben preciso, nel senso che guida l'esecuzione
- In Javascript/Java per Processing, non viene codificato dall'interprete in nessun modo, ma ha come scopo solo quello di **migliorare la leggibilità e la comprensibilità** del codice



# ESEMPIO

- Scrivere un programma che stampi la stringa "fa caldo" se la temperatura è superiore a 25 gradi; "fa freddo" se la temperatura è inferiore a 15 gradi; "si sta bene" altrimenti

```
console.log("Esercizio temperatura");  
var temperatura = 30;  
if (temperatura>25)  
    console.log("fa caldo");  
else // temperatura sotto 25 gradi  
    if (temperatura<15)  
        console.log("fa freddo");  
    else  
        console.log("si sta bene");
```

L'indentazione del secondo if rende più chiaro che siamo all'interno del ramo else del primo if



# Ambiguità degli IF annidati

- Es. come viene interpretato il seguente programma?

```
var a=-1;  
var b=2;  
if(a>=0)  
  if(b>=0)  
    console.log("prima stampa");  
  else  
    console.log("seconda stampa");
```



# Ambiguità degli IF annidati (2)

- Es. come viene interpretato il seguente programma?

```
var a=-1;  
var b=2;  
if(a>=0)  
if(b>=0)  
console.log("prima stampa");  
else  
console.log("seconda stampa");
```

```
var a=-1;  
var b=2;  
if(a>=0)  
if(b>=0)  
    console.log("prima stampa");  
else  
    console.log("seconda stampa");
```

```
var a=-1;  
var b=2;  
if(a>=0)  
if(b>=0)  
    console.log("prima stampa");  
else  
    console.log("seconda stampa");
```



# Ambiguità degli IF annidati (3)

- Ogni else fa sempre riferimento all'if più vicino
- Affinché un else corrisponda ad un altro if occorre usare i blocchi
- Se si scrivono if annidati, il consiglio è di mettere le parentesi graffe per rendere esplicito a quale if ci stiamo riferendo
- ```
if(a >= 0){  
    if(b >= 0)  
        console.log("b positivo");  
}  
else  
    console.log("a negativo");
```



# IF senza THEN

- Il ramo THEN non può essere vuoto!
- Se in un qualsiasi momento scrivete un IF-THEN-ELSE con il ramo ELSE ma senza il ramo THEN
  - invertite i due rami
  - Invertite la condizione

```
if(a >=0){  
  }  
else  
  console.log("fai qualcosa");
```

```
if(a < 0){  
  console.log("fai qualcosa");
```

