



Fondamenti di Informatica

Alma Artis
Francesca Pratesi (ISTI, CNR)

Esercizi



ESERCIZI DEL 29 NOVEMBRE

Operatore XOR

- L'operatore booleano binario $\text{XOR}(a,b)$ calcola
 - *true*, quando solamente uno tra *a* e *b* è *true*,
 - *false*, in tutti gli altri casi.
- Si scriva un programma che definisca una funzione XOR che, avendo come parametri due valori booleani *a* e *b*, restituisca il risultato di $\text{XOR}(a,b)$.
- Si completi il programma invocando la funzione precedentemente definita per tutte le possibili combinazioni di valori booleani per i suoi parametri, e si stampino i risultati ottenuti.
- Si definisca una funzione *XOR_senza_if*, con funzionamento simile alla funzione XOR (stesso input e output) ma che non faccia uso dell'istruzione *if*



Possibili soluzioni

```
function XOR(a,b){  
  if(a==false && b==false)  
    return false;  
  if(a==true && b==true)  
    return false;  
  if(a==true && b==false)  
    return true;  
  if(a==false && b==true)  
    return true;  
}
```

```
function XOR_2(a,b){  
  if( (a==false && b==false) || (a==true && b==true) )  
    return false;  
  if( (a==true && b==false) || (a==false && b==true) )  
    return true;  
}
```

```
console.log(XOR_2_bis(false,false)); // false
```



Uso di booleani negli if

- Quando abbiamo una variabile booleana da testare, essa può essere usata direttamente in una condizione booleana

```
// a è una variabile booleana con un valore assegnato in precedenza  
if(a==true) console.log(true);
```

è equivalente a

```
var a = true;  
if(a) console.log(true);
```



Possibili soluzioni (2)

```
function XOR_bis(a,b){  
  if(!a && !b) // a == false  
    return false;  
  if(a && b)  
    return false;  
  if(a && !b)  
    return true;  
  if(!a && b)  
    return true;  
}
```

```
function XOR_2_bis(a,b){  
  if( (!a && !b) || (a && b) )  
    return false;  
  if( (a && !b) || (!a && b) )  
    return true;  
}
```





ESERCIZI

Esercizi - 1

- Scrivere lo stato del seguente programma nei punti evidenziati dai commenti

```
function potenza(base, esponente){  
    // 2  
    var risultato = base**esponente;  
    // 3  
    return risultato;  
}  
var base = 5;  
var esp = 3;  
// 1  
var elev = potenza(base,esp);  
// 4  
var risultato = elev*2;  
// 5  
console.log(risultato);
```



Math

- La funzione di libreria
 - `Math.random()`
- genera numeri casuali compresi tra 0 e 1

Nota: lo 0 è compreso, l'1 no!

- `Math.floor`, `Math.ceil`, `Math.round` sono funzioni che prendono come parametro un decimale e restituiscono un intero (rispettivamente, il troncamento, l'intero superiore e l'arrotondamento)



Esercizi - 2

- Scrivere una funzione *genera_random* che genera un numero random e lo restituisce
- Scrivere una funzione *genera_random10* che genera un numero naturale random compreso tra 0 e 10 e lo restituisce
- Scrivere una funzione *genera_randomX* che prende come parametro un numero naturale x e genera un numero naturale random compreso tra 0 e x
- Scrivere una funzione *genera_random_intervallo* che genera un numero naturale casuale compreso tra un valore minimo e un valore massimo, passati come parametro, e restituisce il numero generato



Esercizi - 3

- Scrivere una funzione *stampa_media* che dati 5 numeri ne stampi la somma e la media.
- Scrivere una funzione *calcola_media_random* che genera 5 numeri casuali compresi tra 0 e un intero ricevuto in input e ne calcola la media.
- Scrivere una funzione che prenda come parametro un numero di tre cifre e restituisca true se il numero è “palindromo”, cioè se invertito rimane lo stesso.
- Scrivi un programma che dato un numero di secondi, calcoli la quantità di ore, minuti e secondi corrispondenti e poi stampi il risultato. L'output avrà solo numeri interi. Ad esempio:
 - Input: 12560
 - Output: “3 ore, 29 minuti e 20 secondi”.

