



CORSO FULL STACK WEB DEVELOPER



Voglio che lei scriva:  
*Non devo dire bugie.*

Quante volte? - *chiese Harry.*

Oh, quanto ci vuole  
perché il messaggio penetri!





# JS - Il Ciclo While



# Ciclo while

## Come si scrive un ciclo while?

Se non sappiamo quante volte deve ripetersi il nostro ciclo, possiamo utilizzare un **while**.

- keyword:  
while
- condizione:  
poi si imposta la condizione da verificare per poter eseguire il codice, che sarà ripetuto finché la condizione tra parentesi è vera.
- Istruzioni per terminare il ciclo:  
All'interno delle parentesi graffe dobbiamo modificare la condizione in modo da renderla falsa quando vogliamo terminare il ciclo

```
1 while (condizione) {  
2     // codice da eseguire  
3  
4     // istruzioni per terminare il ciclo  
5  
6 }  
7
```

## ATTENZIONE

La condizione prima o poi deve diventare falsa, altrimenti avremo un **ciclo infinito!**





## Ciclo while vs ciclo for

Quando usiamo il ciclo for e quando il while?

Il **ciclo for** è pratico per eseguire cicli a conteggio, perché permette di indicare in modo compatto **contatore**, **condizione** ed **incremento**.

Il **ciclo while** è pratico per eseguire cicli la cui condizione non dipende per forza da un contatore.



## Ciclo while vs ciclo for

Quando usiamo il ciclo for e quando il while?

A parte ciò il ciclo for ed il ciclo while sono equivalenti:

**tutto ciò che può essere realizzato con un ciclo for può essere realizzato anche con un ciclo while.**

È sempre possibile *tradurre* un ciclo for in un ciclo while e viceversa.

Andrà quindi scelto caso per caso il ciclo più comodo (o più affine al proprio modo di ragionare).



# LIVE CODING



# Ciclo do-while

Il ciclo **do-while** è molto simile al ciclo while.

La differenza sostanziale è che il ciclo while può anche non essere mai eseguito se la condizione risulta falsa fin dall'inizio.

**Il ciclo do-while viene sempre eseguito almeno una volta anche se la condizione risulta falsa fin dall'inizio.** Questo perché la condizione viene verificata solo dopo che il blocco di codice è già stato eseguito.

```
1 do {  
2  
3   // codice da eseguire  
4  
5   // istruzioni per terminare il ciclo  
6  
7 } while (condizione);
```





# Cicli

## Time to Recap

Servono a ripetere un blocco di codice più volte, c'è sempre un modo per trasformare un ciclo in un altro

```
1 for (let i = 0; i < 10; i++) {  
2  
3   // istruzioni da eseguire  
4  
5 }
```

### Ciclo for

Utile se il ciclo deve girare per un numero conosciuto di volte (es. *per attraversare array con la proprietà .length*)

```
1 while (condizione) {  
2  
3   // codice da eseguire  
4  
5   // istruzioni per terminare il  
   ciclo  
6  
7 }
```

### Ciclo while

Utili se la condizione da verificare non è un contatore oppure non si sa a priori quante volte deve ripetersi il ciclo

```
1 do {  
2  
3   // codice da eseguire  
4  
5   // istruzioni per terminare il  
   ciclo  
6  
7 } while (condizione);
```

### Ciclo do-while

Simile al while ma controlla la condizione solo dopo aver eseguito



# LIVE CODING



# ESERCITAZIONE