

Istruzioni di ripetizione



Ripetizione del codice

- Ma se devo fare la stessa operazione per tante volte, cosa faccio? Riscrivo n volte lo stesso pezzo di codice?
- Non diventa un po' noiosa come operazione?
- Per fortuna c'è una soluzione

```
int num = 0;
if(num < 10){
    num += 1;
}
if(num < 10){
    num += 1;
}
if(num < 10){
    num += 1;
}
if(num < 10){
    num += 1;
}
if(num < 10){
    num += 1;
}
```

Istruzione while

Viene utilizzata per ripetere un blocco di codice fino a quando la condizione di controllo presente risulta non verificata (ossia *false*)

È possibile, nel caso in cui la condizione sia subito *false*, che le istruzioni all'interno del corpo del *while* non vengano mai eseguite

```
while(condizione){  
    blocco_di_istruzioni;  
}
```

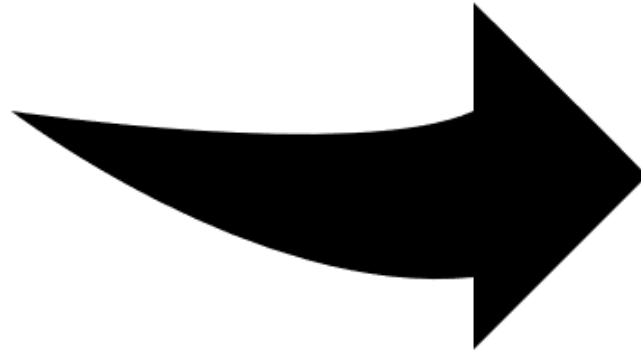
Valuta la condizione scritta tra parentesi, chiamata **condizione di controllo** e, se risulta verificata (*true*), vengono eseguite le istruzioni all'interno delle parentesi graffe

Una volta eseguite tutte le istruzioni, la condizione di controllo viene nuovamente valutata

- Se vera, riesegue le istruzioni all'interno della parentesi graffa
- Se falsa, esce dal blocco *while* ed esegue l'istruzione successiva

Come funziona il while?

```
int num = 0;
if(num < 10){
    num += 1;
}
if(num < 10){
    num += 1;
}
if(num < 10){
    num += 1;
}
if(num < 10){
    num += 1;
}
if(num < 10){
    num += 1;
}
```



```
int num = 0;
while(num < 10){
    num += 1;
}
```

Esempio

Esempio 2

Scrivere un programma che calcola la tabellina di un numero inserito dall'utente

```
#include <iostream>

using namespace std;

int main(int argc, char const *argv[])
{
    int num, tab;

    cout << "Inserisci un numero" << endl;
    cin >> num;

    int i = 1;
    while(i < 11){
        tab = num * i;
        cout << num << "\t" << i << "\t" << tab << endl;
        i += 1;
    }
    return 0;
}
```

Esercizi

01

Scrivere un programma che somma uno ad un numero inserito da tastiera finché il numero non è maggiore di 27

02

Scrivere un programma che stampa 15 volte la frase «Devo studiare di più il C++»

03

Scrivere un programma che stampi a video la sequenza di Fibonacci

Ma...

E se noi volessimo essere sicuri di eseguire almeno una volta le istruzioni dentro il corpo del *while*

Possiamo usare il costrutto *do...while*

```
do{  
    blocco_di_istruzioni;  
} while (condizione);
```


Come funziona il do...while?

Funziona allo stesso modo del ciclo *while*, ma garantisce che almeno una volta il codice del corpo venga eseguito

Esempio

- Scrivere un programma che calcola la media dei voti inseriti dall'utente fino a che egli non ci dà il segnale che ha inserito tutti i voti

```
#include <iostream>

using namespace std;

int main(int argc, char const *argv[])
{
    double voto;
    int numVoti = 0;
    int somma = 0;

    double media;

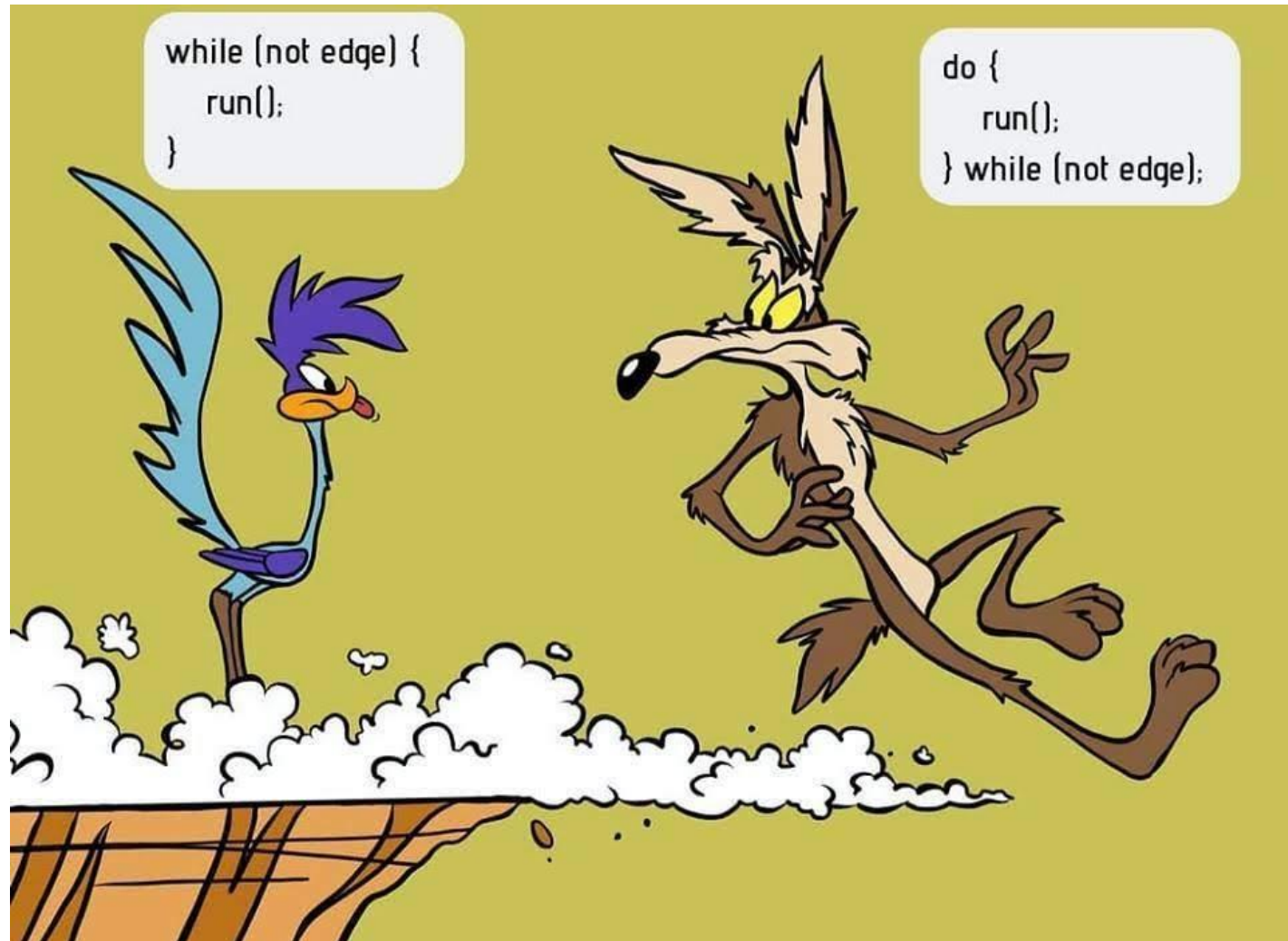
    do{
        cout << "Scrivi un voto. Digita 0 per terminare" << endl;
        cin >> voto;
        if (voto != 0){
            numVoti += 1;
            somma += voto;
        }
    } while (voto != 0);

    media = somma / numVoti;

    cout << "La media dei tuoi voti \212: " << media << endl;

    return 0;
}
```

Differenza tra i while



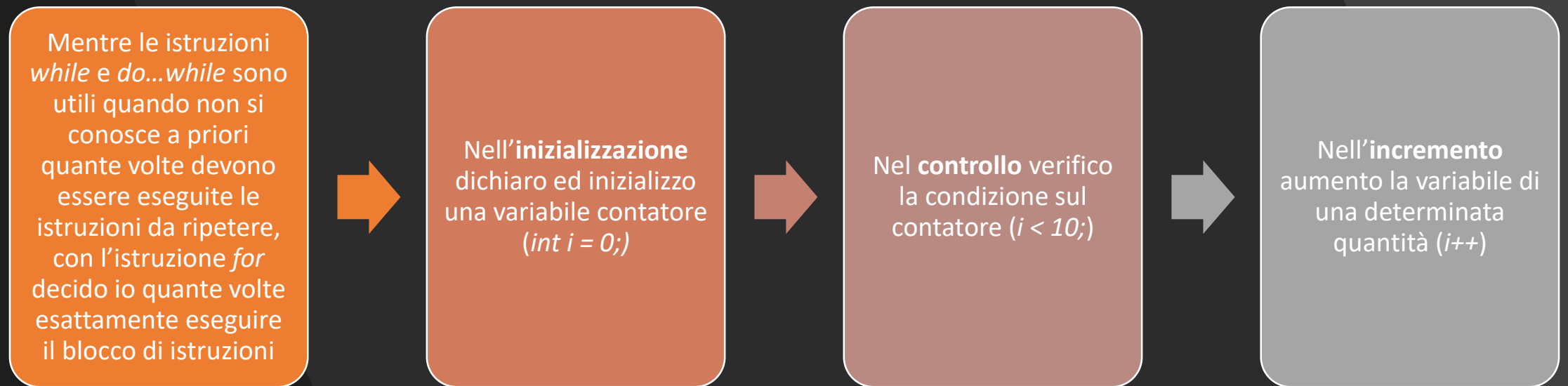
Un altro ma...

E se volessi ripetere delle istruzioni
un numero ben definito di volte?

Posso usare il costrutto *for*

```
for (inizializzazione; controllo; incremento){  
    blocco di istruzioni;  
}
```

Come funziona il ciclo for?



Esempio

Scrivere un programma che calcola il fattoriale di un numero inserito a tastiera

```
#include <iostream>

using namespace std;

int main(){
    int num;
    int fatt = 1;

    cout << "Scrivi un numero" << endl;
    cin >> num;

    for (int i = 1; i <= num; i++){
        fatt *= i;
    }

    cout << "Il fattoriale \212: " << fatt << endl;
}
```

Provate a riscrivere l'esempio del fattoriale tramite il costrutto *while*

Che differenze vedete?

Tempo di pratica

