

Variabili

Una variabile è un contenitore, dentro cui possiamo mettere un valore.

Tipi

Le variabili sono di diversi tipi, in base ai dati che vogliamo memorizzare.

- `int` : numeri interi (1, 2, 14, -100, ...)
- `char` : carattere (a, b, c, !, L, K, ...)
- `float` : numeri decimali (1.0, 4.932, 3.14159265, ...)

Creazione di una variabile

Esempio con `int` :

```
int main () {  
    int pippo; // creo la variabile  
    pippo = 4; // assegno il valore 4 alla variabile pippo  
  
    int pluto = 9; // creo la variabile e la assegno direttamente  
  
    cout << pippo << endl << pluto << endl;  
}
```

C++

Output:

```
4  
9
```

Esempio con `float` :

```
int main () {  
    float paperino; // creo la variabile  
    paperino = 1.2; // assegno il valore 4 alla variabile pippo  
  
    float paperone = 1000000.4123; // creo la variabile e la assegno direttamente  
  
    cout << paperino << endl << paperone << endl;  
}
```

C++

Output:

```
1.2
1000000.4123
```

Esempio con `char` :

```
int main () {
    char topolino; // creo la variabile
    topolino = 'T'; // assegno il valore 4 alla variabile pippo

    char basettoni = 'B'; // creo la variabile e la assegno direttamente

    cout << topolino << endl << basettoni << endl;
}
```

C++

Output:

```
T
B
```

Leggere e scrivere sulla console

Esempio di lettura e scrittura valori:

```
int main () {
    int A, B, C; // creo le variabili

    cin >> A >> B >> C; // leggo i valori e li metto nelle variabili

    cout << "A: " << A << endl; // stampo la stringa "A: ", il valore di A, mando a
    cout << "B: " << B << endl;
    cout << "C: " << C << endl;
}
```

C++

Output:

```
1
2
3
A: 1
B: 2
C: 3
```

Quando eseguo il programma non succede niente: il cursore rimane fermo ad aspettare che io inserisca i numeri. Quando ho inserito tutti i 3 numeri, il programma prosegue e stampa i valori.

if

Tradotto letteralmente "se". Se la condizione è vera, il codice dentro le parentesi graffe viene eseguito, altrimenti no.

Esempio:

```
int main () {
    int numeroRuote;
    cout << "Inserisci il numero delle ruote:";
    cin >> numeroRuote;

    if (numeroRuote == 4) {
        cout << "auto" << endl;
    }

    if (numeroRuote > 4) {
        cout << "camion" << endl;
    }

    if (numeroRuote == 2) {
        cout << "moto" << endl;
    }
}
```

C++

Output:

```
Inserisci il numero di ruote: 4
auto
```

```
Inserisci il numero di ruote: 30
camion
```

Inserisci il numero di ruote: 2
moto

Inserisci il numero di ruote: 3

for

Il for si usa per eseguire piu' volte un pezzo di codice.

```
for (int i=0; i<5; i++) { // i++ aumenta i di 1
    cout << "ciao, ";
}
```

C++

Output:

ciao, ciao, ciao, ciao, ciao,

La struttura base del for è:

```
for (inizializzazione; condizione; incremento)
```

C++

quindi:

```
for (int i=0; i<4; i++) {
    cout << i << endl;
}
cout << "fine" << endl;
```

C++

- crea `i` e la pone a `0`
- controlla la condizione
 - `i<4` ?
 - `i` vale `0` , `0<4` allora la condizione è **vera**
 - eseguo il codice dentro al `for`
- stampo il valore di `i`
- incremento `i` (ovvero eseguo `i++`)
 - ora `i` vale `1`

- controllo la condizione
 - `i < 4` ?
 - `i` vale `1` , `1 < 4` allora la condizione è **vera**
 - eseguo il codice dentro al `for`
- stampo il valore di `i`
- incremento `i` (ovvero eseguo `i++`)
 - ora `i` vale `2`
- controllo la condizione
 - `i < 4` ?
 - `i` vale `2` , `2 < 4` allora la condizione è **vera**
 - eseguo il codice dentro al `for`
- stampo il valore di `i`
- incremento `i` (ovvero eseguo `i++`)
 - ora `i` vale `3`
- controllo la condizione
 - `i < 4` ?
 - `i` vale `3` , `3 < 4` allora la condizione è **vera**
 - eseguo il codice dentro al `for`
- stampo il valore di `i`
- incremento `i` (ovvero eseguo `i++`)
 - ora `i` vale `4`
- controllo la condizione
 - `i < 4` ?
 - `i` vale `4` , `4 == 4` allora la condizione è **FALSA**
 - NON eseguo il codice dentro al `for`
- eseguo il codice *dopo* il `for`
 - stampo `fine`