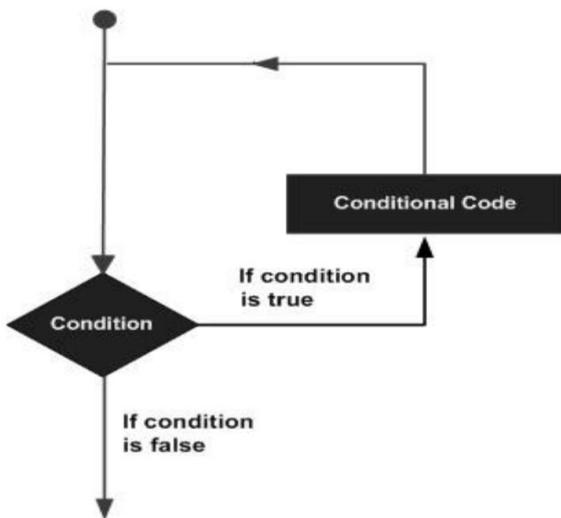


Arduino – Ciclo “while, do...while, for”

I linguaggi di programmazione forniscono varie strutture di controllo che consentono percorsi di esecuzione più complicati. Una istruzione loop (ciclo) ci consente di eseguire una dichiarazione o un gruppo di istruzioni più volte e in seguito è la forma generale di un'istruzione loop nella maggior parte dei linguaggi di programmazione



Il linguaggio di programmazione C fornisce i seguenti tipi di loop per gestirne i requisiti.

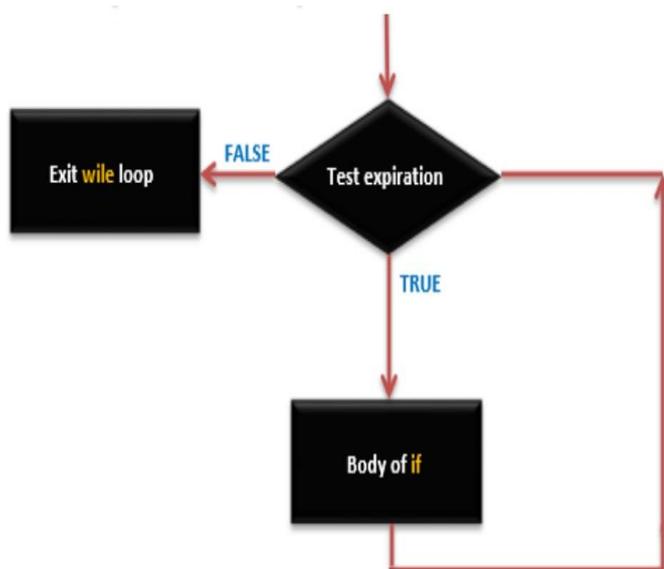
1. Ciclo “ while ”

while andrà in loop (ciclo) continuamente, ed infinitamente, fino a quando l'espressione all'interno delle parentesi, () diventa falsa. Qualcosa deve cambiare la variabile testata, o il ciclo *while* non uscirà mai.

SINTASSI DEL CICLO “ WHILE ”

```
while(espressione){  
    Blocco di dichiarazioni;  
}
```

SEQUENZA DI ESECUZIONE



Esempio

```
/* Definizione della variabile globale */  
int Pulsante = 5;  
int LED = 13;  
Void setup () {
```

```

pinMode(Pulsante, INPUT);
pinMode(LED, OUTPUT);
}
Void loop () {
while(digitalRead(Pulsante) == HIGH){
digitalWrite(LED,HIGH);
delay(1000);
digitalWrite(LED,LOW);
delay(1000);
}
}

```

2. Ciclo "do...while"

Il ciclo *do...while* è simile al ciclo *while*. Nel ciclo *while*, la condizione di continuazione del ciclo viene testata all'inizio, prima di eseguire il corpo del loop. L'istruzione *do ... while* verifica la condizione di continuazione del ciclo dopo aver eseguito il corpo del loop. Pertanto, il corpo del loop verrà eseguito almeno una volta.

Quando un *do...while* termina, l'esecuzione continua con la proposizione dopo la clausola *while*. Non è necessario usare parentesi graffe nella dichiarazione *do ... while* se c'è una sola istruzione nel corpo. Tuttavia, le parentesi graffe sono solitamente incluse per evitare confusione tra le dichiarazioni *while* e *do ... while*.

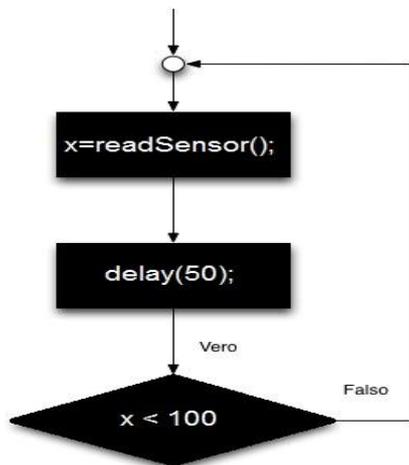
SINTASSI DEL CICLO "DO...WHILE"

```

do{
    Blocco di dichiarazioni;
}
while(espressione);

```

SEQUENZA DI ESECUZIONE



Esempio

```

do
{
x=readSensor(); // assegna il valore di readSensor() a x
delay(50); // pausa di 50 millisecondi
} while (x < 100); // cicla fino a quando x < 100

```

3. Ciclo " for "

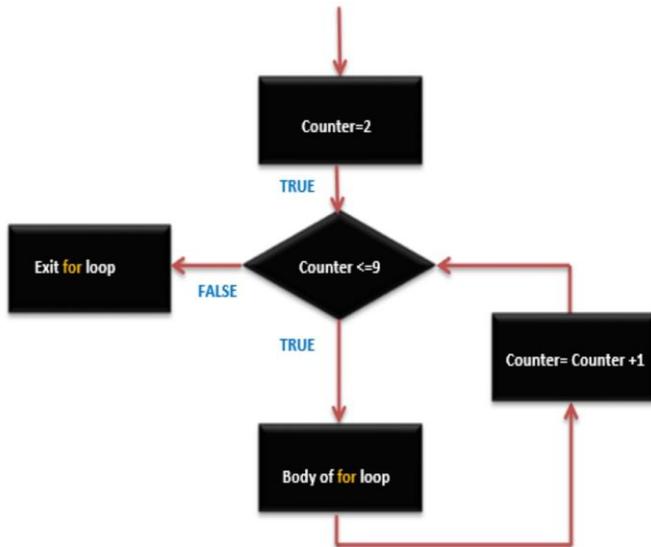
Un ciclo *for* esegue le istruzioni un numero predeterminato di volte. L'espressione di controllo per il ciclo viene inizializzata, testata e manipolata interamente all'interno delle parentesi del ciclo *for*. È facile eseguire il debug del comportamento di loop della struttura in quanto è indipendente dall'attività all'interno del ciclo.

Ogni ciclo *for* ha fino a tre espressioni, che ne determinano l'operazione. L'esempio seguente mostra in generale la sintassi del ciclo. Si noti che le tre espressioni nelle parentesi di argomento ciclo *for* sono separate da punto e virgola.

SINTASSI DEL CICLO " FOR "

```
for ( inizializza; controlla; incrementa o decrementa ) {  
  
    // blocco di istruzioni  
  
}
```

SEQUENZA DI ESECUZIONE



Esempio

```
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    for (int counter=2;counter<9;counter++)  
        //il blocco di istruzioni verrà eseguito 7 volte  
        {  
            Serial.println(counter);  
            delay (500);}  
}
```

4. Ciclo " for " nidificato

Il linguaggio C consente di utilizzare un ciclo all'interno di un altro ciclo. Il seguente esempio illustra il concetto.

SINTASSI DEL CICLO " FOR " NIDIFICATO

```
for ( inizializza; controlla; incrementa o decrementa ) {  
  
    // blocco di istruzioni  
  
    for ( inizializza; controlla; incrementa o decrementa ) {  
  
        // blocco di istruzioni  
  
    }  
  
}
```

```
}  
}
```

Esempio

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  for (int counter = 2; counter <= 9; counter++){  
    Serial.println(counter); //il blocco di istruzioni verrà eseguito 7 volte  
    delay (500);  
    for (int i = 0; i <= 99; i++){  
      Serial.println(i); //il blocco di istruzioni verrà eseguito 100 volte  
      delay (250);  
    }  
  }  
}
```

5. Ciclo " infinito "

È il ciclo che non ha condizione di terminazione, quindi il loop diventa infinito.

SINTASSI DEL CICLO " INFINITO "

Usando il ciclo " for "

```
for(;;){  
  // blocco di istruzioni  
}
```

Usando il ciclo " while "

```
while(1){  
  // blocco di istruzioni  
}
```

Usando il ciclo " do...while "

```
do{  
  // blocco di istruzioni  
}  
while(1);
```

Esempio

```
int counter = 0;  
  
void setup() {  
  
  Serial.begin(9600);  
}  
  
void loop() {
```

```
for(;;){  
  counter++;  
  Serial.println(counter);//il blocco di istruzioni verrà eseguito all'infinito  
  delay (500);  
}  
Serial.println("fine");//Noterete che la scritta "fine" non appare mai, perché  
//il ciclo for va all'infinito.  
}
```