

# LOGICHE E SINTASSI DELLA PROGRAMMAZIONE

---

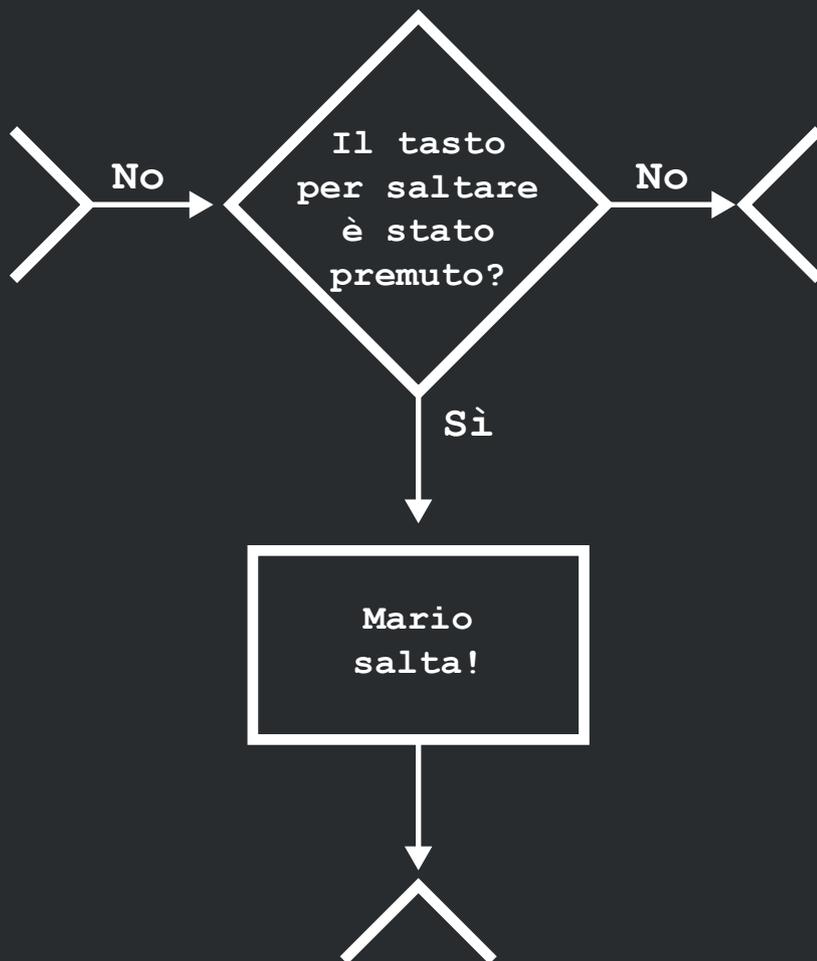
## 6. Le Funzioni

Corso di **Fabio Lettieri**

Accademia di Belle Arti di Brera  
Anno Accademico 2018/2019

## I BLOCCHI DI ISTRUZIONI

In programmazioni esistono degli elementi che servono a definire blocchi di istruzioni sotto un unico nome. In questo modo è possibile riutilizzare quel blocco di comandi più volte nel programma, richiamando semplicemente il suo nome.



**MARIO SALTA!**

Il blocco affermativo "**Mario salta!**" rappresenta un insieme di istruzioni che servono ogni qualvolta il personaggio deve saltare.

Infatti i blocchi affermativi servono a definire in programmazione o l'assegnazione di un nuovo valore ad una variabile o il **richiamo di una funzione**.

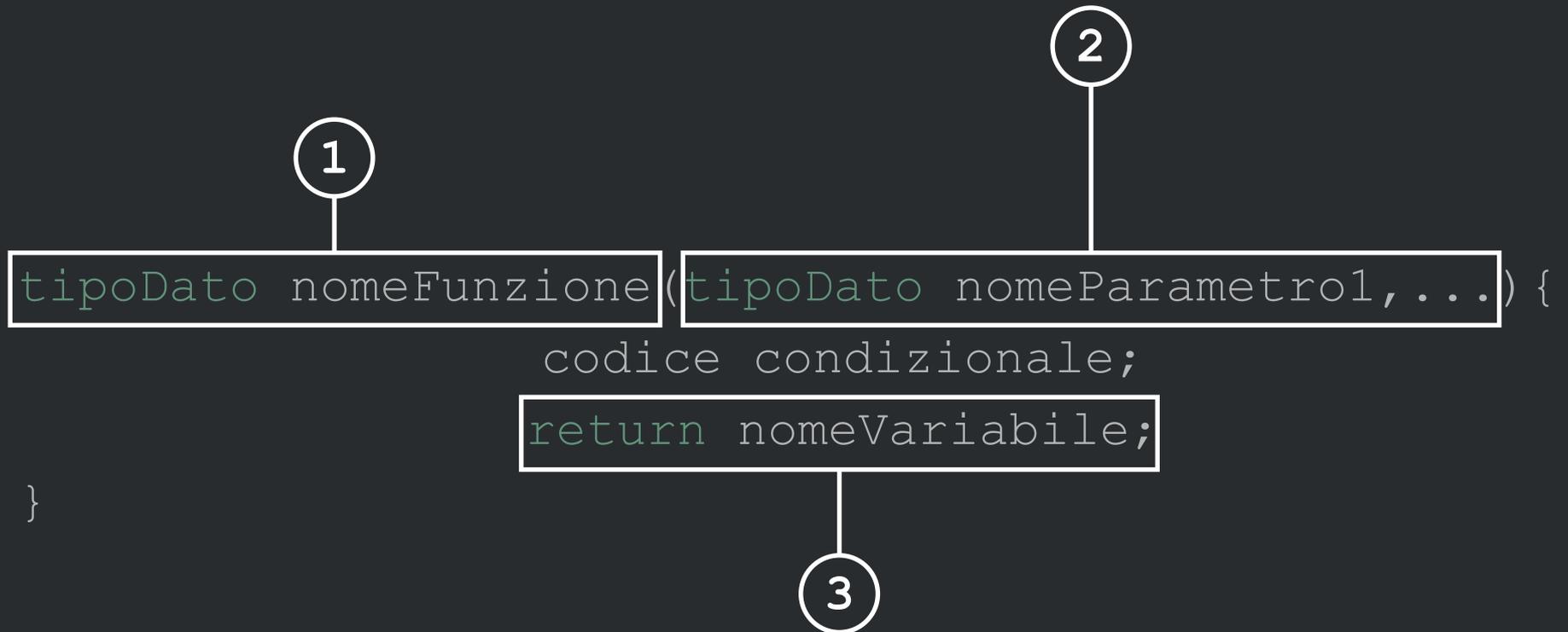
## DEFINIZIONE DI UNA FUNZIONE

```
tipoDato nomeFunzione(tipoDato nomeParametro1,...) {  
    codice condizionale;  
    return nomeVariabile;  
}
```

Le **funzioni** si definiscono inizialmente in modo molto simile alle variabili, quindi si definisce il tipo di dato che la funzione debba restituire e gli si assegna un nome. A differenza delle variabili si aggiunge però un nuovo tipo di dato, ossia il **void** (vuoto).

La parola chiave **return** serve a definire nella funzione la restituzione di un dato che viene definito a fianco e che deve essere dello stesso tipo specificato nella **dichiarazione**. Nel caso sia `void`; la keyword `return` può essere omessa.

I **parametri** all'interno delle parentesi tonde non sono obbligatori da inserire, ma servono per quei dati che saranno passati alla funzione in modo tale che possa poi svolgere il proprio compito.



1. DICHIARAZIONE FUNZIONE

2. DICHIARAZIONE PARAMETRI (non obbligatoria)

3. RESTITUZIONE VALORE DELLA FUNZIONE (da non mettere se funzione **void**)

# FUNZIONI PREDEFINITE IN PROCESSING

Le funzioni predefinite sono quelle funzioni realizzate da altri programmatori che servono a svolgere determinati compiti e che facilitano il creative coding, ovvero l'utilizzo della programmazione in ambiti creativi.

## funzione setup



`void setup()` è la funzione che serve a inizializzare alcuni elementi fondamentali del codice come la creazione della finestra di rendering dello sketch oppure per caricare file, e altro ancora.

Quindi contiene tutti quei comandi preliminari per la partenza dell'algoritmo, in quanto questa funzione viene richiamata una volta sola all'avvio del programma realizzato.

## funzione draw



`void draw()` è la funzione che viene richiamata ad ogni ciclo di rendering e serve per aggiornare in maniera ripetuta i vari dati che devono essere aggiornati e che continuano a mutare a seconda di varie condizioni presenti all'interno della funzione. Se non viene modificato il framerate dello sketch, ogni ciclo di rendering equivale ad un sessantesimo di secondo.

La funzione draw necessita per il corretto funzionamento del codice la creazione della funzione setup.

**P.S.** All'interno dello stesso algoritmo può esserci solo una funzione setup e una funzione draw.

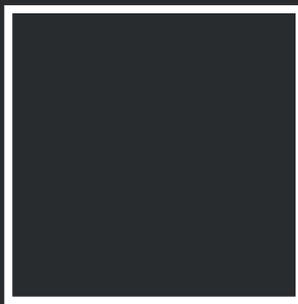
## ALTRE FUNZIONI PREDEFINITE



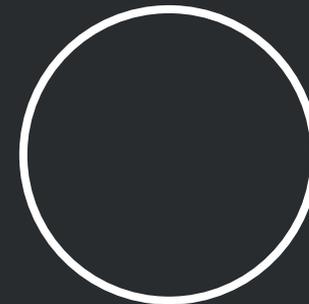
`point(x, y);`



`line(xA, yA, xB, yB);`



`rect(x, y, width, height);`



`ellipse(x, y, width, height);`

# STRUTTURA GENERALE DI UN CODICE

