

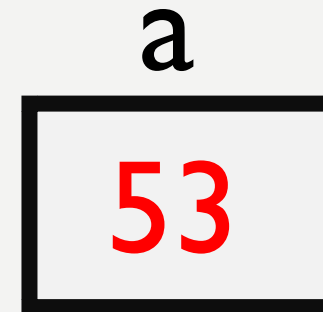


# **VARIABLE ET CONDITIONS**

# VARIABLES

## Quezaco?

Une variable est une boîte dans laquelle on peut mettre n'importe quelle valeur.  
On appelle **nom de la variable** le nom correspondant au nom de la boîte et **valeur de la variable** ce qu'il y a à l'intérieur de la boîte.



# VARIABLES

## Déclaration

La déclaration d'une variable consiste à créer une boîte avec un nom. Pour ce qui est de la valeur à l'intérieur, ça peut être différent d'un langage à l'autre.

Dans Godot, pour déclarer une variable, il faut utiliser le mot clé « var » pour indiquer que c'est une variable et ensuite donner le nom de la variable.



**GODOT**

```
5 >| var a
6 >| print("La valeur de a est égal à : " , a)
```

```
La valeur de a est égal à : Null
```

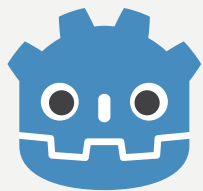
a



# VARIABLES

## Assignment

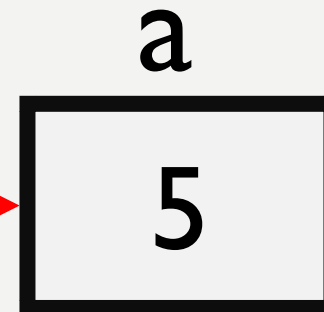
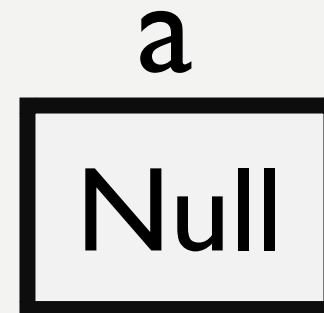
Assigner une valeur à une variable est l'action de stocker une valeur dans la variable en écrasant la valeur précédente stockée.



**GODOT**

```
5 >| var a
6 >| a = 5
7 >| print("La valeur de a est égal à : " , a)
```

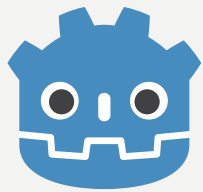
```
La valeur de a est égal à : 5
```



# VARIABLES

## Astuce

Assigner une valeur à une variable est l'action de stocker une valeur dans la variable en écrasant la valeur précédente stockée.



**GODOT**

```
5 >| var a = 5
6 >| print("La valeur de a est égal à : " , a)
```

```
La valeur de a est égal à : 5
```

a

5

# VARIABLES

## Assignation



**GODOT**

```
5 >| var a = 4
6 >| var b = 8
7 >| a = b
8 >| print("La valeur de a est égal à : " , a)
9 >| print("La valeur de a est égal à : " , b)
```

a



b



# VARIABLES

## Assignation



**GODOT**

```
5 >| var a = 4
6 >| var b = 8
7 >| a = b
8 >| print("La valeur de a est égal à : " , a)
9 >| print("La valeur de a est égal à : " , b)
```

```
La valeur de a est égal à : 8
La valeur de b est égal à : 8
```

a



b



a



b



# VARIABLES

## Assignation



**GODOT**

```
5 >| var a = 4
6 >| var b = 8
7 >| a = b + a - 6
8 >| b = c
9 >| print("La valeur de a est égal à : " , a)
10 >| print("La valeur de b est égal à : " , b)
```

a



b





# VARIABLES

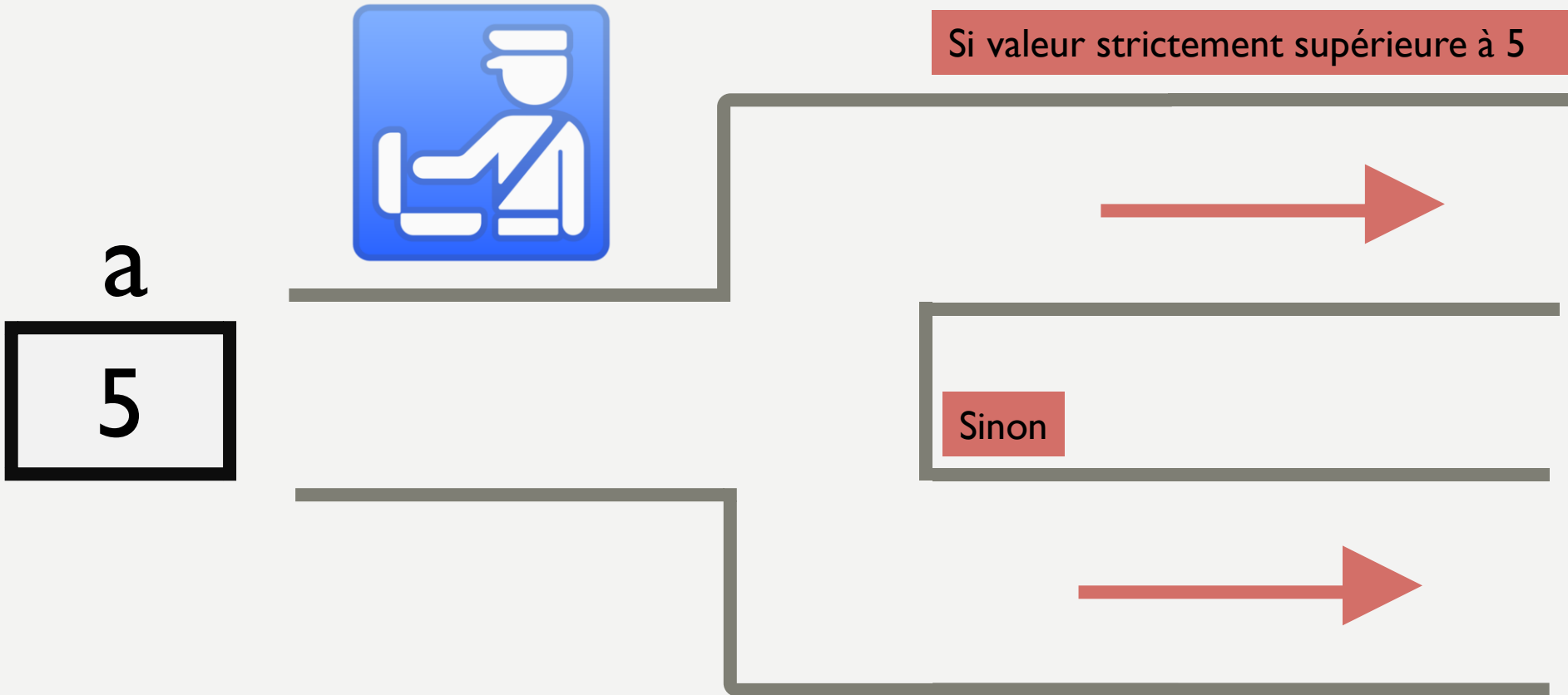
## Exercices

- Créer une nouvelle variable qui s'appelle « direction » et qui a comme valeur `Vector2(12,-45)`
  - Créer une nouvelle variable qui s'appelle « point » et qui a comme valeur `Vector2(250,250)`
  - Faire une addition de ces deux variables et stocker le résultat dans une variable appelée « destination »
  - Afficher le résultat
- 
- Créer une nouvelle variable qui s'appelle « resultat » et qui a comme valeur la différence entre le `Vector2(48,67)` et `Vector2(-7,1)` et l'addition du résultat avec le `Vector2(9,12)`

# CONDITIONS

# Quezaco?

Une condition est une expression visant à réaliser des actions différentes en fonction d'une valeur.



# CONDITIONS

## Exemple

Dans Godot, pour réaliser une condition, le mot clé « if » doit être indiqué en premier suivi de la condition. Enfin « : » doit être ajouté juste à la fin de la ligne. Les actions à réaliser lorsque la condition est respectée doivent être écrites en dessous du if décalé d'une tabulation vers la droite.



**GODOT**

```
5 >| var a = 5
6 v>| if a == 5:
7 >| >| print("La valeur de a est bien égale à 5")
8
```

# CONDITIONS

## Exemple

Si la condition n'est pas respectée, on peut tout de même réaliser des actions grâce au mot clé « else »



**GODOT**

```
5 >| var a = 4
6 v>| if a == 5:
7 >| >| print("La valeur de a est bien égale à 5")
8 v>| else:
9 >| >| print("La valeur de a n'est pas égale à 5")
```

# CONDITIONS

## Comparaisons

$==$

$<$

$<=$

$!=$

$>$

$>=$

# CONDITIONS

## Exemple



**GODOT**

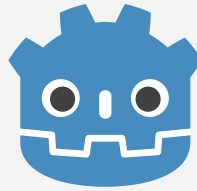
```
5 >| var a = 4
6 ✓ >| if a == 5:
7 >| >| print("La valeur de a est bien égale à 5")
8 ✓ >| if a <= 4:
9 >| >| print("Oui!")
```

**==**

**<=**

# CONDITIONS

## Vecteur

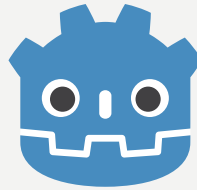


GODOT

```
5  >|  var a = Vector2(5,8)
6  v >|  if a == Vector2(5,8):
7  >|    >|  print("YES -> 1")
8  v >|  if a.x == 8:
9  >|    >|  print("YES -> 2")
10 v >|  if a.y <= 8:
11 >|    >|  print("YES -> 3")
12 v >|  if Vector2(5,8) == a:
13 >|    >|  print("YES -> 4")
14 v >|  if Vector2(5,4) != a:
15 >|    >|  print("YES -> 5")
16 v >|  if a < Vector2(5,9):
17 >|    >|  print("YES -> 6")
18 v >|  if a == 9:
19 >|    >|  print("YES -> 7")
```

# CONDITIONS

## Vecteur



GODOT

```
5 >| var a = Vector2(5,8)
6 v>| if a == Vector2(5,8):
7 >| >| print("YES -> 1")
8 v>| if a.x == 8:
9 >| >| print("YES -> 2")
10 v>| if a.y <= 8:
11 >| >| print("YES -> 3")
12 v>| if Vector2(5,8) == a:
13 >| >| print("YES -> 4")
14 v>| if Vector2(5,4) != a:
15 >| >| print("YES -> 5")
16 v>| if a < Vector2(5,9):
17 >| >| print("YES -> 6")
18 v>| if a == 9:
19 >| >| print("YES -> 7")
```

```
YES -> 1
YES -> 3
YES -> 4
YES -> 5
YES -> 6
```



# CONDITIONS

## Exercices

- Créer une nouvelle variable qui s'appelle « test » et qui a comme valeur 8
  - Afficher la variable « test » si sa valeur est strictement supérieure à 5, autrement afficher « NON »
- Créer deux nouvelles variables qui s'appelle « test0 » et qui a comme valeur 8 et « test1 » qui a comme valeur 3
  - Si la multiplication de « test0 » avec 3 est inférieure ou égale à « test1 », afficher le résultat
- Créer une nouvelle variable qui s'appelle « vecteur0 » et qui a comme valeur Vector2(0,10)
  - Si la valeur x de « vecteur0 » est supérieure ou égale à 0 alors afficher « x is good »
  - Sinon si la valeur de y est différente de 15, alors afficher « y is good »
  - Sinon afficher « nothing is good »

# FONCTIONS

Quezaco?

Une fonction peut être vu également comme une boîte avec un nom. Cette fois-ci l'intérieur de la boîte contient une série d'instructions qui peut être par exemple :

- Déclaration de variables
- Assignment de variables
- Conditions
- Appel d'autres fonctions

somme

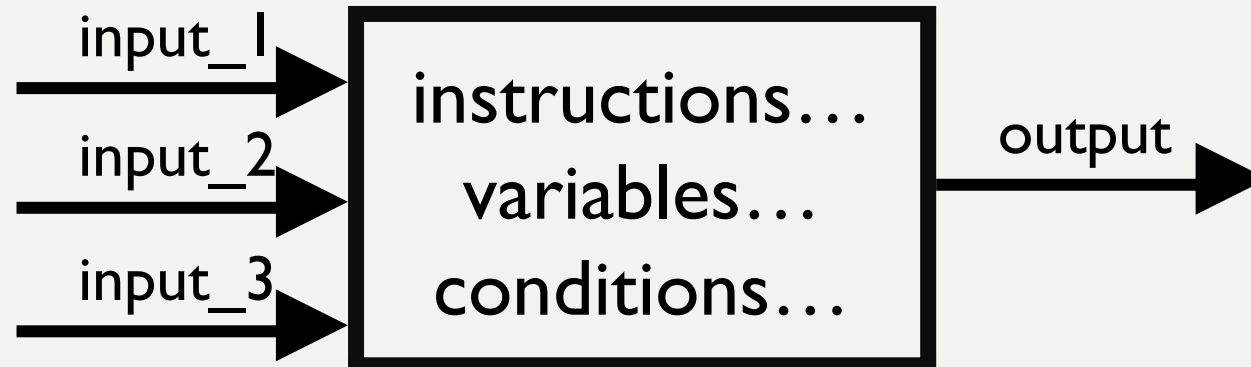
instructions...  
variables...  
conditions...

# FONCTIONS

## Quezaco?

Une fonction a comme particularité qu'elle peut prendre des valeurs en entrée qu'on appelle des **paramètres d'entrée** et va peut être produire une valeur en sortie. Le terme « peut-être » est employé, car parfois aucune valeur en sortie n'est produite, ou alors aucune valeur en entrée n'est donnée, voire même les deux cas à la fois.

## somme



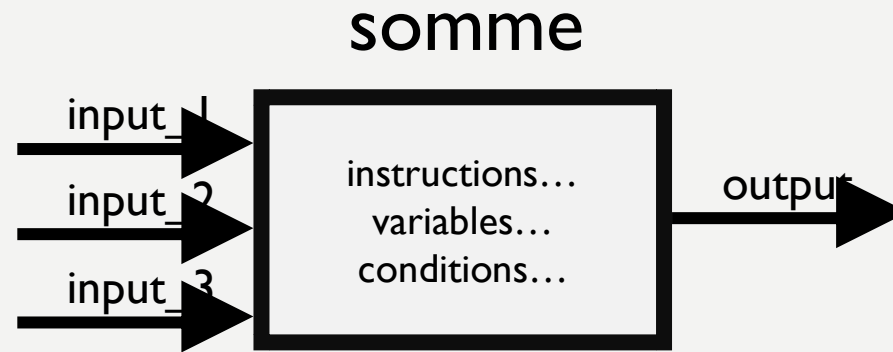
# FONCTIONS

## Signature



**GODOT**

```
4 func somme(var input_1, var input_2, var input_3):  
5     » # coeur de la fonction
```



Une fonction dans Godot s'écrit avec un premier mot clé « func » pour spécifier qu'on va écrire une fonction. Ensuite le nom de la fonction est donnée. Ensuite tous les paramètres d'entrée doivent être placés entre ( ) et séparés par des « , » Enfin « : » doit être placé juste à la fin de la ligne pour placer le cœur de la fonction. Toute cette ligne est appelée la signature de la fonction, car elle regroupe toutes les informations qui décrivent une fonction.

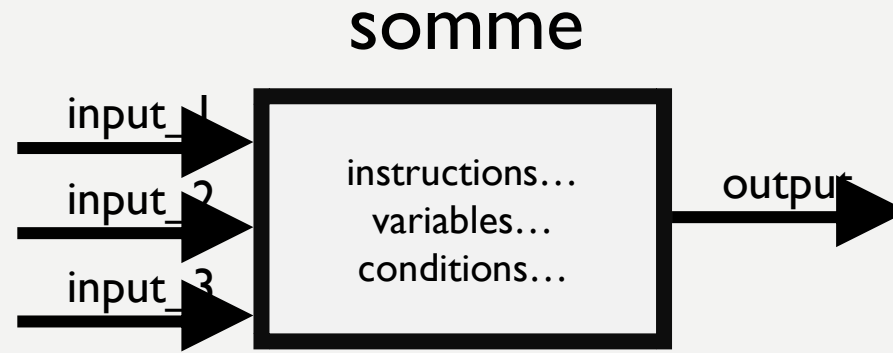
# FONCTIONS

Retour d'une  
valeur



GODOT

```
4 ▾ func somme(var input_1, var input_2, var input_3):  
5   »   var result = input_1 + input_2 + input_3  
6   »   return result #output  
7
```



Si une fonction doit retourner une valeur de sortie, alors le mot clé « return » doit être écrit suivi de la valeur à retourner.

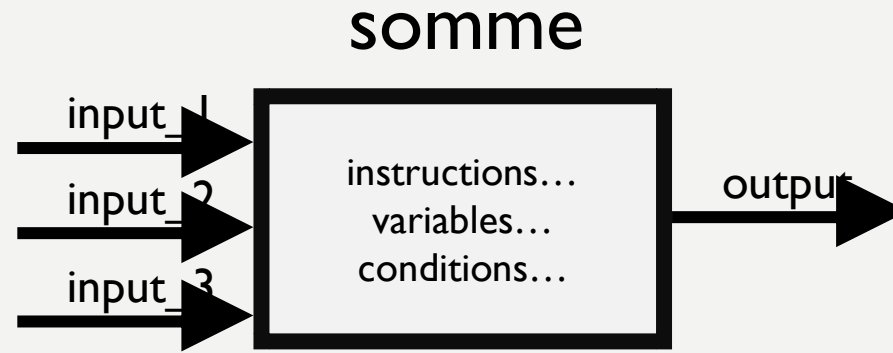
# FONCTIONS

## Appel d'une fonction



GODOT

```
4 v func somme(var input_1, var input_2, var input_3):  
5   >| var result = input_1 + input_2 + input_3  
6   >| return result #output  
7  
8  
9 v func _ready():  
10  >| var result = somme(4, 5, 6)  
11  >| print("La somme est égale à " , result)
```



Pour appeler une fonction, il faut écrire le nom de la fonction suivi des (). Si des valeurs doivent être fournies en entrée de la fonction appelée, alors elles doivent être ajoutées entre les ( ) dans le même ordre que dans la signature de la fonction.

Enfin si la fonction renvoie une valeur en sortie, il suffit de créer par exemple une variable à laquelle on affecte l'appel de la fonction.

# CONDITIONS

## Exercices

- Réaliser la fonction « somme » qui va faire l'addition de deux valeurs en entrée.
  - Appeler cette fonction avec deux entier en entrée et afficher le résultat en sortie
  - Appeler cette fonction avec deux Vector2 en entrée et afficher le résultat en sortie
- 
- Réaliser la fonction « print » qui va afficher « La valeur vaut : » + la valeur du paramètre d'entrée
- 
- Réaliser la fonction qui va s'activer lorsque la personne appuie sur la touche espace et qui fait disparaître ou apparaître la ballon (reprendre la scène avec le ballon)