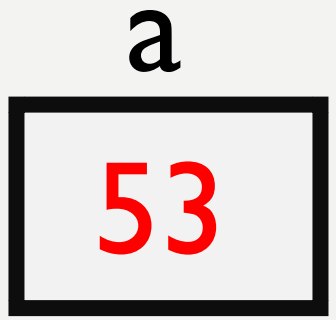
VARIABLE ET CONDITIONS

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

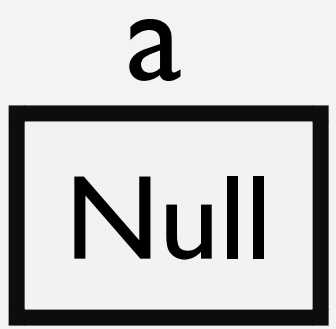
La déclaration d'une variable consiste à créer une boîte avec un nom. Pour ce qui est de la valeur à l'intérieur, çà peut être différent d'un langage à l'autre.

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
```



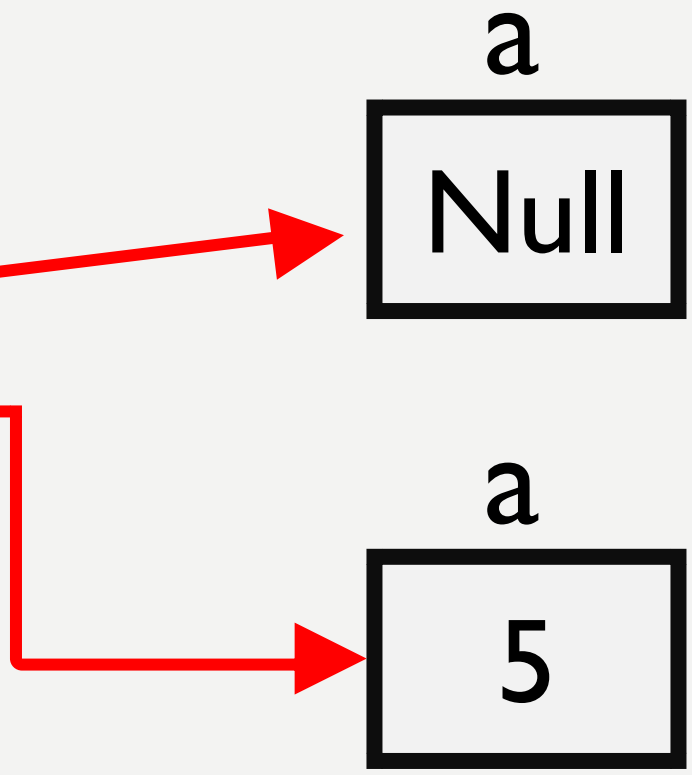
VARIABLES

Affectation

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


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

```
La valeur de a est égal à : 5
```



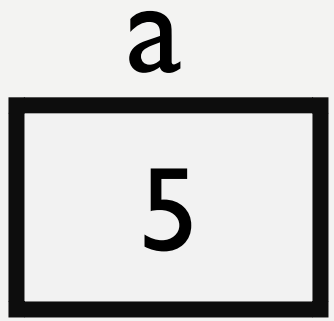
VARIABLES

Astuce

GODOT

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

```
La valeur de a est égal à : 5
```

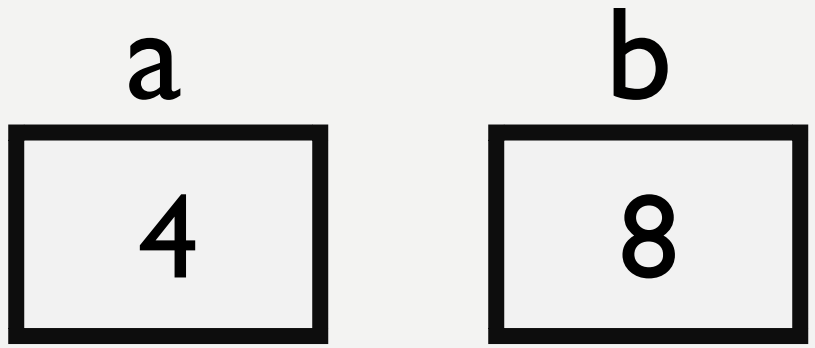


VARIABLES

Exemples

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)
```



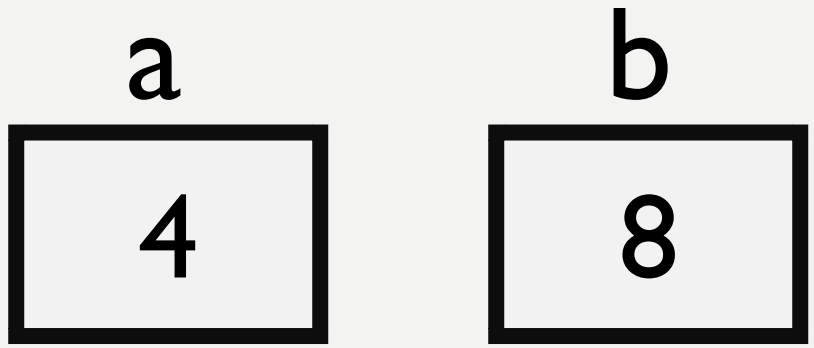
VARIABLES

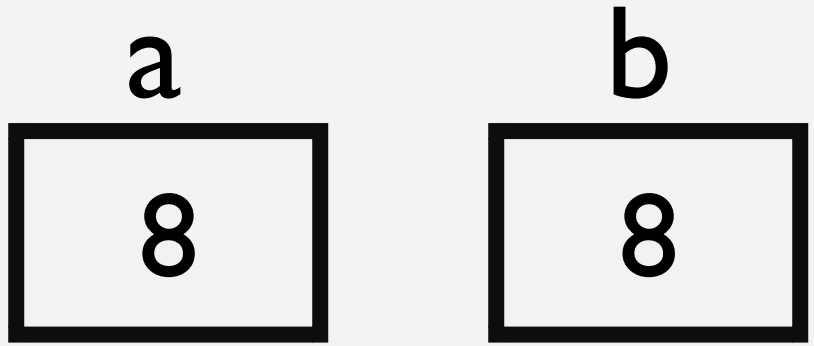
Exemples

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
```



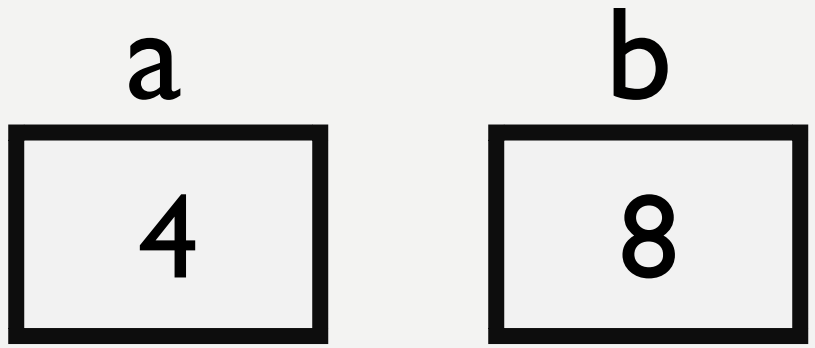


VARIABLES

Exemples

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)
```



VARIABLES

Exercices

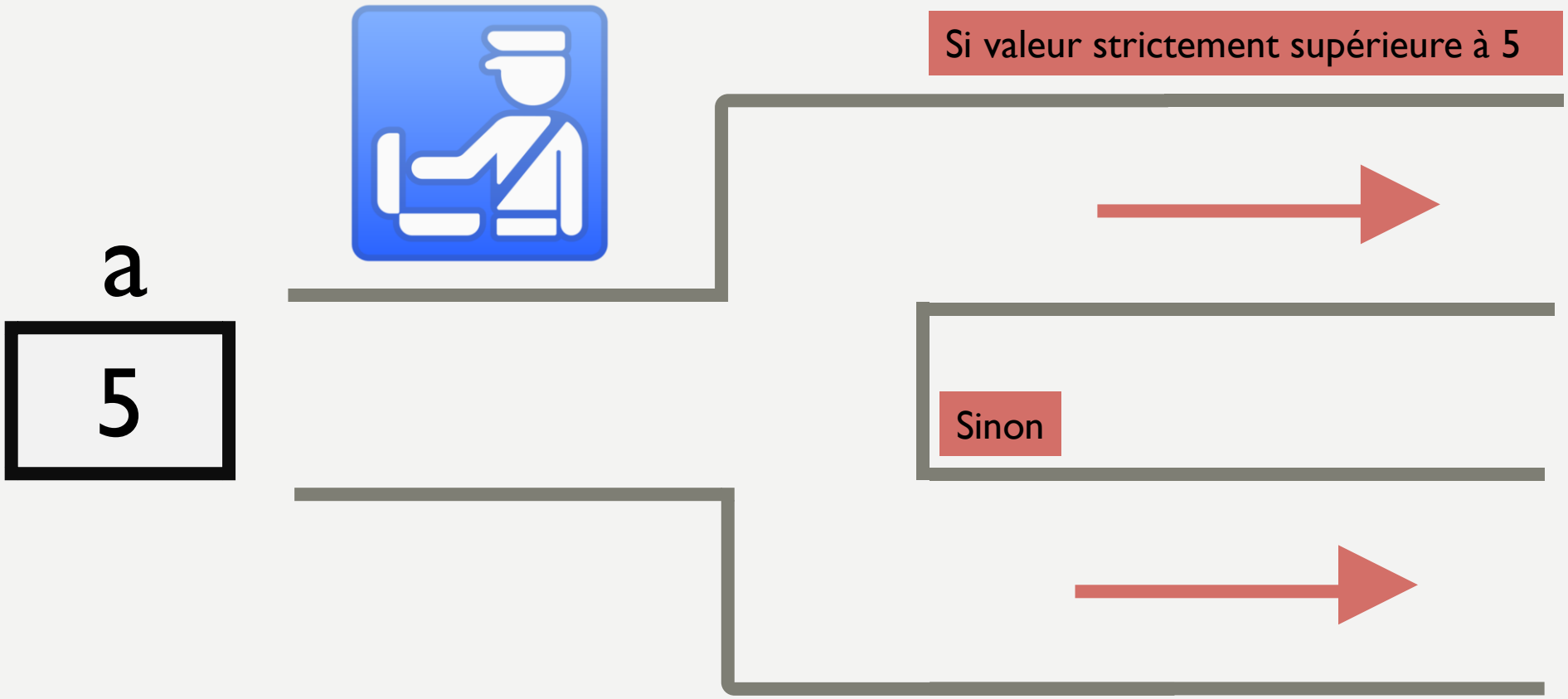
- Créer une nouvelle variable qui s'appelle « direction » et qui a comme valeur Vector2(12,-45)
- Créer une nouvelle variable qui s'appelle « point » et qui a comme valeur Vector2(250,250)
- Faire une addition de ces deux variables et stocker le résultat dans une variable appelée « destination »
- Afficher 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    if a == 5:
7        print("La valeur de a est bien égale à 5")
```

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

CONDITIONS

Comparaisons

==	<	<=
!=	>	>=

CONDITIONS

Exemple

GODOT

```gdscript
var a = 4
if a == 5:
    print("La valeur de a est bien égale à 5")
if a <= 4:
    print("Oui!")
```

==

<=

CONDITIONS

Vecteur

GODOT

```gdscript
var a = Vector2(5,8)
if a == Vector2(5,8):
    print("YES -> 1")
if a.x == 8:
    print("YES -> 2")
if a.y <= 8:
    print("YES -> 3")
if Vector2(5,8) == a:
    print("YES -> 4")
if Vector2(5,4) != a:
    print("YES -> 5")
if a < Vector2(5,9):
    print("YES -> 6")
if a == 9:
    print("YES -> 7")
```

CONDITIONS

Vecteur

GODOT

```gdscript
var a = Vector2(5,8)
if a == Vector2(5,8):
    print("YES -> 1")
if a.x == 8:
    print("YES -> 2")
if a.y <= 8:
    print("YES -> 3")
if Vector2(5,8) == a:
    print("YES -> 4")
if Vector2(5,4) != a:
    print("YES -> 5")
if a < Vector2(5,9):
    print("YES -> 6")
if a == 9:
    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 « vector0 » 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
- Assignation de variables
- Conditions
- Appel d'autres fonctions

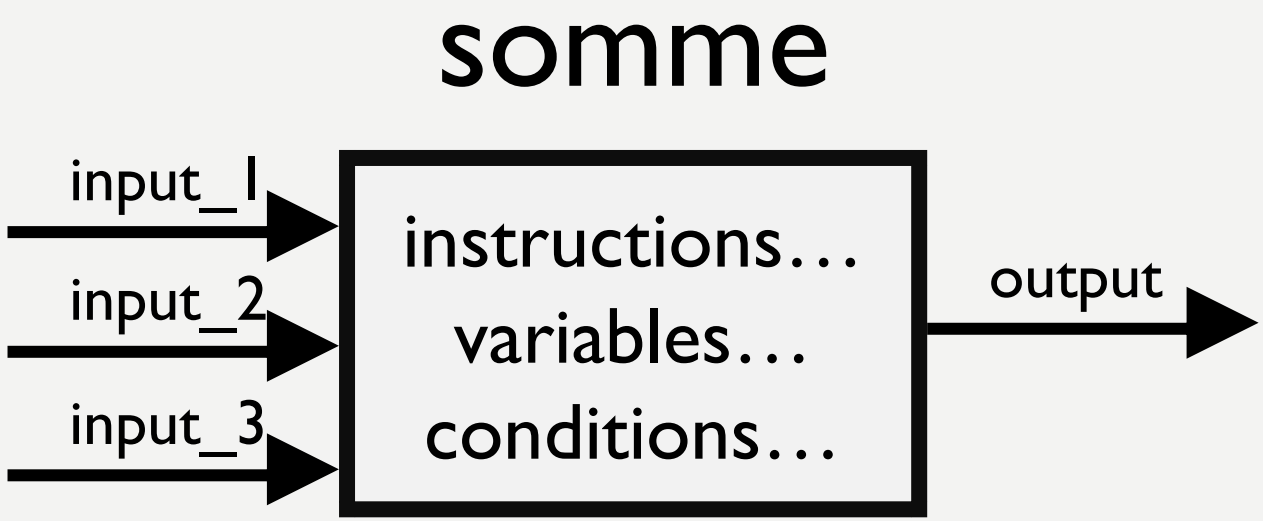
somme

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.



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 placées entre ( ) et séparées 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

```gdscript
func somme(var input_1, var input_2, var input_3):
    var result = input_1 + input_2 + input_3
    return result #output
```

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

```gdscript
func somme(var input_1, var input_2, var input_3):
    var result = input_1 + input_2 + input_3
    return result #output


func _ready():
    var result = somme(4, 5, 6)
    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.
Dans l'exemple, le bout de code dans la fonction somme va être exécutée avec
input_1 = 4, input_2 = 5 , input_3 = 6

FONCTIONS

Appel d'une fonction

GODOT

```gdscript
func somme(var input_1, var input_2, var input_3):
    var result = input_1 + input_2 + input_3
    return result #output


func _ready():
    var result = somme(4, 5, 6)
    print("La somme est égale à " , result)
```

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. L'appel de la fonction somme aura donc le même effet que de faire :
var result = 4 + 5 + 6
Sauf que l'on a pas besoin de réécrire cette opération à chaque fois, seulement d'appeler la fonction avec les valeurs que l'on a besoin.

FONCTIONS

Quezaco?

Les exemples montrées jusqu'à maintenant sont assez simples, mais il faut imaginer que parfois on souhaite exécuter des centaines d'instructions à plusieurs endroits du script. Pour éviter de dupliquer le code, on le factorise (on le rend unique) dans une fonction.
Voici un exemple pour la recherche du minimum entre 3 valeurs.

```
15  func _ready():
16      # je veux retrouver quelle est la valeur minimale entre value1, value2, value3
17      var value1 = 5
18      var value2 = 9
19      var value3 = 4
20      var result
21
22      #première comparaison entre value1 et value2
23      if value1 < value2 :
24          result = value1
25      elif value2 < value1 :
26          result =  value2
27      else:
28          result = value1
29
30      #seconde comparaison entre le min de value1 ou value2 avec value3
31      if result < value3 :
32          # cette ligne est inutile, mais c'est pour l'exemple
33          result = result
34      if value3 < result :
35          result =  value3
36      else:
37          # cette ligne est inutile, mais c'est pour l'exemple
38          result = result
```

FONCTIONS

Quezaco?

On voit ici que la recherche du minimum entre deux nombres est bien plus lisible et efficace car nous avons factorisé la recherche dans une fonction.

```gdscript
func _ready():
    # je veux retrouver quelle est la valeur minimale entre value1, value2, value3
    var value1 = 5
    var value2 = 9
    var value3 = 4

    # ici lorsque min_value va être executé, a = value1, b = value2
    var result = min_value(value1, value2) #result va donc être value1 ou value2

    #il me reste à comparer avec value3
    #result est donc le minimum est value1, value2 et value3
    result = min_value(result, value3)


func min_value(var a, var b):
    if a < b :
        # la fonction s'arrête au return s'il est appelé, le reste n'est pas executé
        return a
    if b < a :
        return b
    # si les deux conditions précédentes ne sont pas respectées
    # alors je retourne la valeur de a par défaut
    return a
```

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 disparaitre ou apparaitre la ballon (reprendre la scène avec le ballon)