LearnBlox

# Aide-mémoire Luau

Les 6 notions pour écrire tes premiers scripts Roblox, et réussir le défi Facile. Garde-le à côté de Roblox Studio.



## 1 Afficher un message

`print` écrit dans la fenêtre **Output**. Pratique pour vérifier que ton script tourne.

```
print("Salut, je suis Bloxi !")
```

## 2 Les variables

Une variable garde une valeur sous un nom. `local` la crée, `=` la change.

```
local vies = 3
local pseudo = "Bloxi"
vies = vies - 1   -- il en reste 2
```

## 3 Changer une Part

Un script peut régler une Part comme dans Properties, mais pendant le jeu.

```
local dalle = workspace.MaPlateforme
dalle.Transparency = 0   -- 1 = invisible
dalle.CanCollide = true  -- solide
dalle.Anchored = true    -- ne tombe pas
dalle.Color = Color3.fromRGB(255, 210, 63)
```

## 4 Les conditions

`if` fait quelque chose seulement si c'est vrai. `else`, sinon.

```
if vies > 0 then
    print("Encore en vie !")
else
    print("Perdu...")
end
```

## 5 Répéter avec while

`while true do` répète pour toujours. **Toujours** un `task.wait()` dedans, sinon Studio se fige.

```
while true do
    dalle.Transparency = 1
    task.wait(1)   -- pause d'1 seconde
    dalle.Transparency = 0
    task.wait(1)
end
```

## 6 Réagir quand on touche

`Touched` lance ta fonction quand quelque chose touche la Part.

```
local lave = script.Parent
local function toucher(hit)
    local perso = hit.Parent
    local h = perso:FindFirstChild("Humanoid")
    if h then h.Health = 0 end
end
lave.Touched:Connect(toucher)
```

**Un script rangé dans une Part ? Elle s'appelle `script.Parent`.**

Une erreur ? Ouvre Output (menu Window) : la ligne rouge donne le numéro de la ligne fautive. Majuscules comprises : `Transparency`, pas `transparency`.

**Les solutions du défi arrivent le 11 novembre à 14 h.**

Et la suite des Bases, gratuite, avec Bloxi pour t'aider.

learnblox.fr

LearnBlox

# Aide-mémoire niveau 2

Les 6 notions en plus pour les défis Moyen et Difficile. Quand il n'y a pas de ligne rouge, c'est ici qu'il faut chercher.



## 7 Répéter un nombre de fois

`for` compte pour toi. Le 3e nombre, s'il y en a un, est le pas de chaque tour.

```lua
for i = 1, 5 do
    print(i)          -- 1, 2, 3, 4, 5
end
for i = 5, 1, -1 do
    print(i)          -- 5, 4, 3, 2, 1
end
```

## 8 Compter

Un compteur est une variable qu'on augmente. Pour la changer, il faut toujours un `=`.

```lua
local points = 0
points = points + 1    -- points vaut 1
if points == 3 then    -- == pour comparer
    print("Gagné !")
end
```

## 9 Placer une Part

`Vector3.new(x, y, z)` : une position. `y` est la hauteur. Celle d'une Part est dans Properties.

```lua
local bloc = workspace.MonBloc
bloc.Position = Vector3.new(0, 20, 0)
-- monter de 1 :
local haut = Vector3.new(0, 1, 0)
bloc.Position = bloc.Position + haut
```

## 10 Fonctions et listes

Une fonction ne fait rien tant qu'on ne l'appelle pas. `GetChildren()` liste ce qu'un objet contient.

```lua
local function saluer()
    print("Salut !")
end
saluer()                -- là, elle s'exécute
for _, objet in modele:GetChildren() do
    print(objet.Name)
end
```

## 11 Un seul toucher à la fois

`Touched` se déclenche plusieurs fois à chaque pas. Une variable « attente » bloque les doublons.

```lua
local attente = false
dalle.Touched:Connect(function(hit)
    if attente then return end
    attente = true
    print("Touché une fois !")
    task.wait(1)
    attente = false
end)
```

## 12 Cliquer sur une Part

Un `ClickDetector` rangé dans une Part permet de cliquer dessus avec la souris.

```lua
local clic = script.Parent.ClickDetector
clic.MouseClick:Connect(function(joueur)
    print(joueur.Name .. " a cliqué")
end)
```

## ? Pas de ligne rouge, mais ça ne marche pas ?

Le script fait exactement ce qui est écrit. Ajoute des `print()` pour voir où il passe et ce que valent tes variables, puis compare avec ce que tu voulais.

**Les solutions des 3 défis arrivent le 11 novembre à 14 h.**

Et la suite des Bases, gratuite, avec Bloxi pour t'aider.

learnblox.fr