



# Introduction au développement de jeu vidéo

GDE-101 - 25.03.2025



# La Programmation

Ne fuyez pas, pauvres fous.





# SPOILER

Soit en ligne :

<https://onecompiler.com/lua>

- Facile et rapide.
- On vous recommande ceci pour cette semaine de cours.

Soit en téléchargeant :

<https://studio.zerobrane.com/>

- Demande un peu plus de maîtrise.
- Permet de faire un peu plus de choses.





# Plan



Introduction

Structures de contrôle

Le Lua

Programmation en direct

Exercice



# Introduction

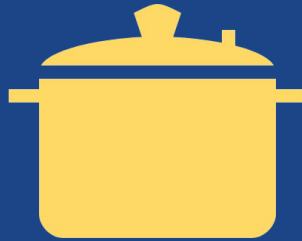


# Principe



**Input(s) (entrées)**

- Nombres
- Tableau
- Phrases
- ...



**Programme**



**Output(s) (sortie(s))**

Résultat



# Définitions

- **Langage de programmation** : notation conventionnelle pour décrire un programme
- **Variable** : Un nom associé à une valeur
- **Constante** : La même chose, mais qui est fixe
- **Type de donnée (“data type”)** :
  - Booléen (“bool”) : “Vrai” ou “Faux”
  - Nombre (“int”) : Un nombre entier
  - Nombre (“float”) : Un nombre réel
  - Chaîne de caractères (“string”) : Du texte



La programmation,  
c'est appliquer des  
éléments les uns  
après les autres.





# Structures de contrôle

Trois pour les gouverner toutes



Les explications  
suivantes  
s'appliquent à tous  
les langages de  
programmation.





## Trois structures principales



### La séquence

“D’abord je bois la potion, ensuite je gagne de la vie, ensuite je n’ai plus la potion”



### La condition

“Si j’ai une potion, alors je peux me soigner, sinon, je ne peux pas me soigner”



### La boucle

“Tant que je n’ai pas regagné ma vie, je bois la potion”

# Le Lua



On dirait de l'elfique





# C'est quoi ?

- Langage de script.
- Langage embarqué.
  - Particulièrement apprécié pour l'embarqué, le développement réseau et les jeux vidéo.
- Utilisé dans de nombreux jeux !
  - Transformice, World of Warcraft, Roblox, Garry's Mod, ...
- Utilisable pour PICO-8, mais aussi CryENGINE et LÖVE.
  - Futurs cours ?
- Facile à prendre en main, facile à comprendre et à utiliser.



Pour PICO-8, le Lua  
utilisé est “allégé”.  
Pour apprendre le  
Lua, on va essayer de  
s’y tenir.





# Programmation en direct

Alors en quoi consiste cette mission... Quête...  
Chose ?

<https://repl.it/languages/lua>



Essayez de faire des  
essais en même  
temps que les slides !  
Ouvrez le site Internet  
et suivez !



ZeroBrane a un  
chouette tutoriel  
intégré si vous  
voulez le suivre !





# La tradition

- Tapez `print("Hello, world!")`.

The screenshot shows a web-based Lua editor interface. At the top left, there is a dropdown menu labeled 'Lua'. To its right is a green 'Run' button with a play icon. Further right are three small icons representing different devices (a smartphone, a tablet, and a laptop) and a 'Share' button with a plus icon. The main area is split into two panels. The left panel is a code editor with a dark background, containing a single line of code: `1 print("Hello, world!")`. The right panel is a terminal window with a dark background, displaying the output `Hello, world!` in white text. A small red cursor icon is visible below the output text. In the top right corner of the terminal window, there are search and close icons (magnifying glass and 'x').

- Si ça vous écrit quelque chose dans la partie de droite, bravo !



# Les variables



```
Lua
```

```
1 nom_de_variable = 21
2 print(nom_de_variable)
```

Run ▶

21

Share



# Quelques opérations...

- **Négation** :  $-a$
- **Addition** :  $a + b$
- **Soustraction** :  $a - b$
- **Multiplication** :  $a * b$
- **Division** :  $a / b$
- **Division euclidienne (entière)** :  $a // b$ 
  - Exemple :  $5 // 2 == 2$
- **Reste de la division entière (modulo)** :  $a \% b$ 
  - Exemple :  $5 \% 2 == 1$
- **Exponentiation** :  $a ^ b$



# Exercice !

Créer un programme, avec une variable qui contient un nombre entre 2 et 10.

Multiplier dans une nouvelle variable ce nombre par 2, et ajouter le chiffre 5.

Enfin, affichez le résultat avec `print`



# Condition

## Simple “Si”

```
if condition then
  something
end
```

## “Si” / “Sinon”

```
if condition then
  something
else
  something_else
end
```

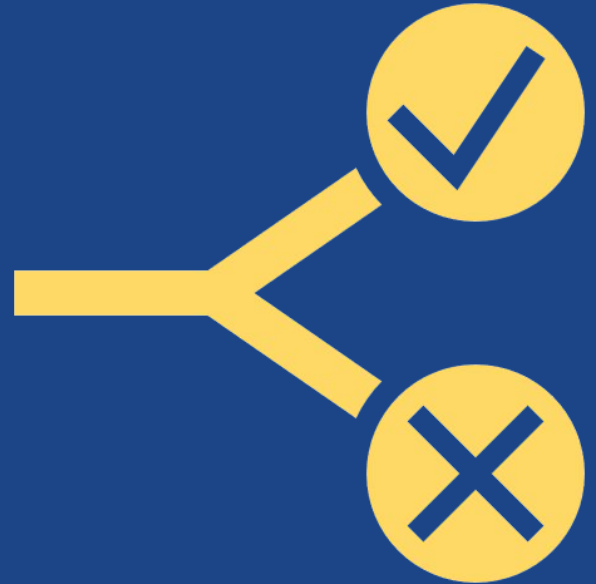
## “Si” / “Sinon si”

```
if cond1 then
  something
elseif cond2 then
  something_else
end
```



# Les opérateurs de relation

- En gros, les trucs pour comparer.
  - a strictement plus petit que b :  $a < b$
  - a strictement plus grand que b :  $a > b$
  - a plus petit ou égal à b :  $a \leq b$
  - L'autre sens :  $a \geq b$
  - a strictement égal à b :  $a == b$
  - a strictement pas égal à b :  $a \neq b$





# Les opérateurs logiques !

## - Opérateurs

- **Et** : a and b
- **Ou** : a or b
- **Non (le contraire de)** : not a

## - Calcul booléen

- pour un “et”, le premier “faux” pousse le résultat à “faux”
- pour un “ou”, le premier “vrai” pousse le résultat à “vrai”

Table de la loi **ET**

b\a	0	1
0	0	0
1	0	1

Table de la loi **OU**

b\a	0	1
0	0	1
1	1	1



# Exercice !

Petit programme, qui prend en entrée  
votre âge :

```
variable = tonumber(io.read())
```

Puis qui indique si vous êtes majeur·e à  
Madagascar (21 ans), sinon en Algérie  
(19 ans), sinon en Suisse (18 ans), sinon  
répond que vous êtes mineur·e.

Vous pouvez aussi utiliser `elseif` !



# Boucles

## While

```
x = 0
while x < 5 do
  print(x)
  x += 1
end
```

## Repeat

```
x = 0
repeat
  print(x)
  x += 1
until x > 4
```

## For

```
for i = 0, 4, 1 do
  print(i)
end
```



# Exercice !

Une boucle, qui demande à l'utilisateur de taper le chiffre 81, jusqu'à ce que le chiffre entré soit effectivement 81.



# Fonctions

```
function multiplier(x1, y1)  
    return x1 * y1  
end
```



# Les chaînes de caractères

- Également appelées string en anglais.
- Comme une variable normale, avec des guillemets :  
texte = "Des trucs là-dedans."
- Pour fusionner du texte, il faut utiliser deux points : ..
  - Exemple : print("Voici ".. un ".." texte")
- On peut calculer la longueur d'un texte avec #.

```
1 texte = "Ceci est une dédicace !"  
2  
3 print("Le texte est : "..texte)  
4 print("La longueur du texte est "..#texte..")
```

Run ▶

Le texte est : Ceci est une dédicace !  
La longueur du texte est 24.

Share



# Des tableaux pour s'organiser

- Un tableau (table), c'est une variable qui contient d'autres variables (ou constantes).
  - Utilisation : `tableau = {1, 2, 3}`.
- Pourquoi c'est utile ? Car on peut nommer les variables !
  - Exemple :  
`joueur = {}`  
`joueur.vies = 5`  
`print(joueur.vies)`
  - On peut aussi utiliser `joueur["vie"]`.
- On utilise `#` pour avoir la taille du tableau.





# Parcourir les tableaux

- Un tableau peut être parcouru avec... Une boucle !
  - Exemple :

```
for k,v in pairs(x) do
  print(k.." = "..v)
end
```
- Le tableau peut également être utilisé pour des séquences !
  - Très pratique pour parcourir un inventaire, par exemple.

```
Lua Run ▶ Share
```

```
1 tableau = {"a", "b", "c", "d", "e"}
2 for i=1,#tableau do
3   print("tableau["..i.."] = "..tableau[i])
4 end
5
```

```
tableau[1] = a
tableau[2] = b
tableau[3] = c
tableau[4] = d
tableau[5] = e
```



# Des questions ?

[discord.gg/8tVCZJG](https://discord.gg/8tVCZJG)

Par écrit : canal #en-direct

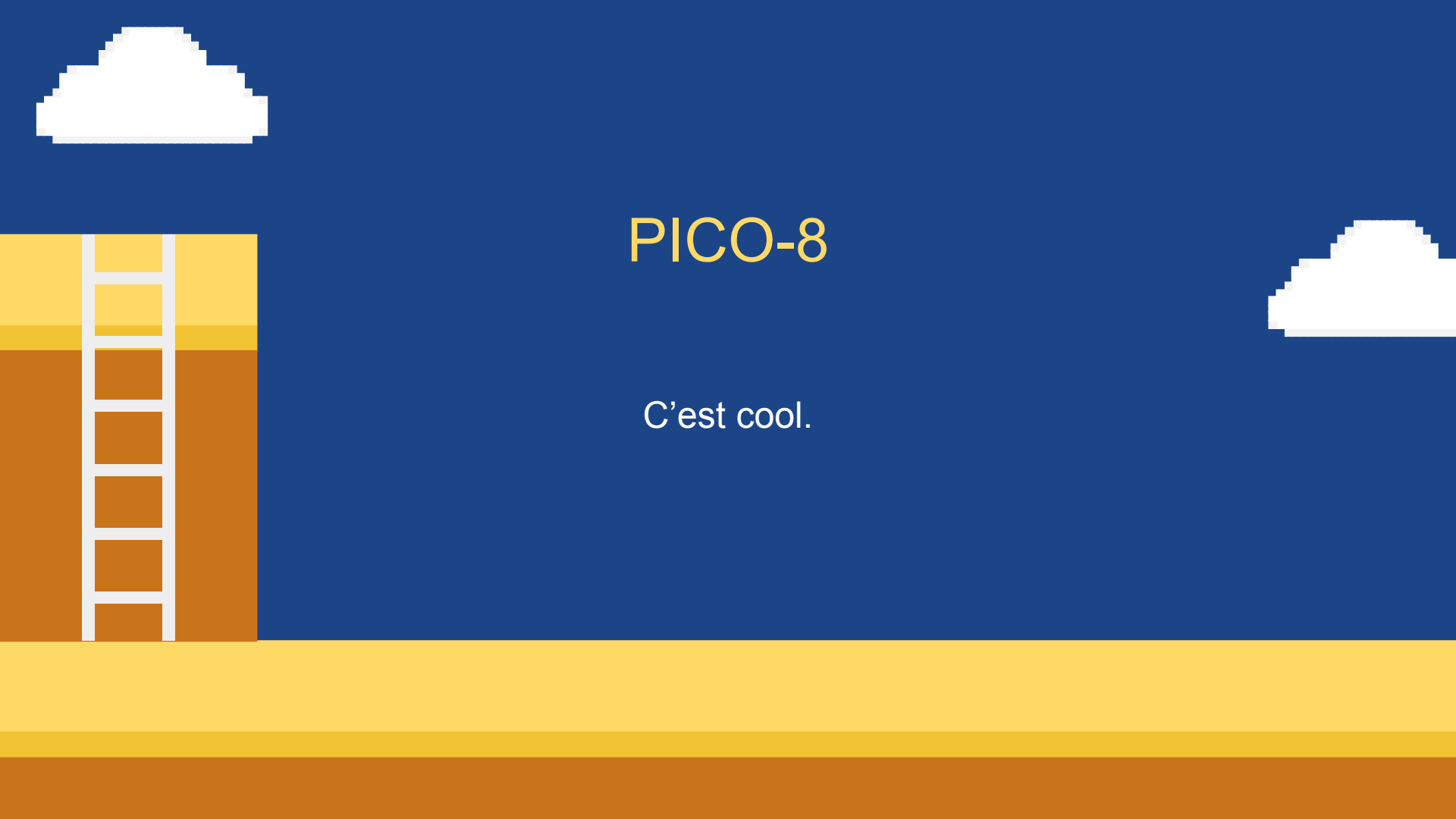
Par oral : lever la main (:



# Programmation - Exercice

Les bonnes bases





PICO-8

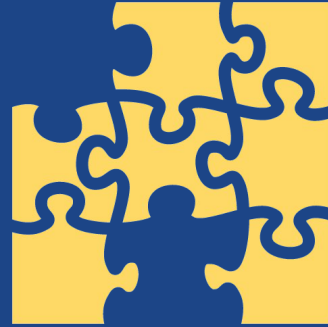
C'est cool.



## Les deux



Remplir  
les  
fonctions  
à trous



Étendre  
le jeu  
existant



# Memory

Histoire de ne rien oublier





# Les règles de base

- Le jeu commence avec seize cartes, donc huit paires de cartes, que l'on dispose, face cachée, sous forme de carré, sur le plateau.
- Le·a joueur·euse va retourner une première carte, puis une deuxième carte. Si les deux cartes ont la même face, le·la joueur·euse gagne un point. Si les deux cartes ont des faces différentes, il faut de nouveau les retourner face cachée.
- La condition de victoire est de trouver toutes les paires le plus vite possible.
- Il n'y a pas de condition de défaite.



# Fonctions à trous

- Fonction “Flux” : Il faut retourner au state wait, puis retourner les cartes cote verso.
- Fonction “Loop” : Il faut afficher les cartes qui sont dans le tableau avec le bon côté (recto ou verso).
- Fonction “If” : Il faut ajouter les trois autres boutons pour déplacer le curseur.



# Modifications possibles

- Changer les constantes (e.g., taille de la grille)
- Trouver des trios plutôt que des paires
- Tu ne peux retourner toutes les cartes qu'une seule fois
- ...



# Des questions ?

[discord.gg/8tVCZJG](https://discord.gg/8tVCZJG)

Par écrit : canal #en-direct

Par oral : lever la main (:



# Merci de votre attention !

[discord.gg/8tVCZJG](https://discord.gg/8tVCZJG)

Par écrit : canal #en-direct

Par oral : lever la main (:

CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.