



PYTHON Workshop I

Python Basics

“Ce qui ne se partage pas, se perd”



Hello!

I'm Khouas Aymen Rayane, Current
President of OpenMindsClub, and
Artificial Intelligence (SII) student.
Contact : aymenkhouas@gmail.com



01

**Start with
Python !**

02

**Conditions and
Loops**

03

Functions



01

Start with Python !

(operations / variables /
instructions de bases...etc)

Commençons avec l'affectation

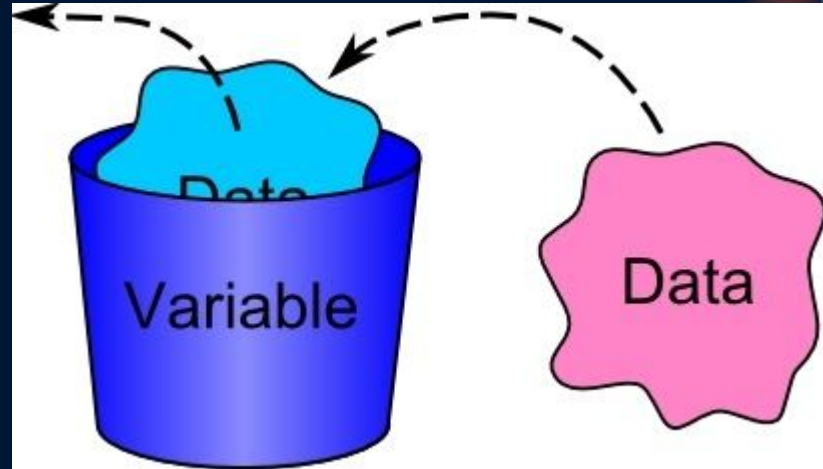
- L'affectation consiste à affecter une valeur à une variable, contrairement a plusieurs langages de programmation, en python

```
variable_1 = 20  
variable_2 = 1.5  
variable_3 = "c"  
variable_4 = True
```



Une variable ? C'est quoi ?

ID	VALUE
age	22
surname	"Chang"
searching	true
weight	9.35



Inputs and outputs

input

On utilise input pour faire entrer des données.

```
my_var= input("")
```

print

On utilise print pour afficher un message sur la console.

```
print("Hello world!")
```

Les Principaux Types que nous allons utiliser

Types simples

Types	exemple
int	2
float	1.5
bool	True\False

Conteneurs

Types	exemple
string	"Hello world"
list	[1,25,5]
tuple	(1,25,5)
dict	{"cle" : valeur}

Les opérations arithmétiques en python

Symbole opérateur	Nom opérateur	exemple
+	Addition	$x + y$
-	Soustraction	$x - y$
*	Multiplication	$x * y$
/	Division	x / y
//	Division Entière	$x // y$
%	Reste de Division	$x \% y$

```
>>> 20 + 13
33
>>> 5 - 1
4
>>> 4 * 5
20
>>> 45 / 5
9.0
>>> 45 // 5
9
>>> 46 % 5
1
>>> |
```

Average “division
avec /” fan



Average “division
avec //” Enjoyer



Incrementations

Symbole opérateur	exemple	
	faire	Au lieu de
<code>+=</code>	<code>x += 1</code>	<code>x = x + 5</code>
<code>-=</code>	<code>x -= 1</code>	<code>x = x - 5</code>
<code>*=</code>	<code>x * y</code>	<code>x = x * 5</code>
<code>/=</code>	<code>x / y</code>	<code>x = x / 5</code>
<code>//=</code>	<code>x // y</code>	<code>x = x // 5</code>
<code>%=</code>	<code>x % y</code>	<code>x = x % 5</code>



Les opérations arithmétiques en python

Symbole opérateur	exemple
+	$x == y$
-	$x < y$
*	$x < y$
/	$x > y$
//	$x != y$

```
>>> 5 == 5
True
>>> 5 == 6
False
>>> 5 != 5
False
>>> 5 <= 3
False
>>> |
```

Remarque

Le Signe '='

Affecte la valeur à droite du signe vers la variable à gauche du signe

Le Signe '=='

Compare les valeurs à droite et gauche du signe et retourne 'True' s'ils sont identiques et 'False' dans le cas contraire.

Les opérations booléennes en python

opérateur	exemple
not	not x
and	x and y
or	x or y
in	x in liste_elem
is	x is y

```
>>> True and True
True
>>> True and False
False
>>> True or True
True
>>> True or False
True
>>> False or False
False
>>> not True
False
>>> not False
True
>>> |
```

Un peu de logique

P	Q	not P	P or Q	P and Q
True	True	False	True	True
True	False	False	True	False
False	True	True	True	False
False	False	True	False	False



Python est un langage

Dynamique

Concrètement on peut modifier le type de variable pendant l'exécution du script.

Fortement typer

Garantit que les types de données sont respecté durant l'exécution
Pa exemple : "2" + 2 renvoie une erreur

Let's practice





02

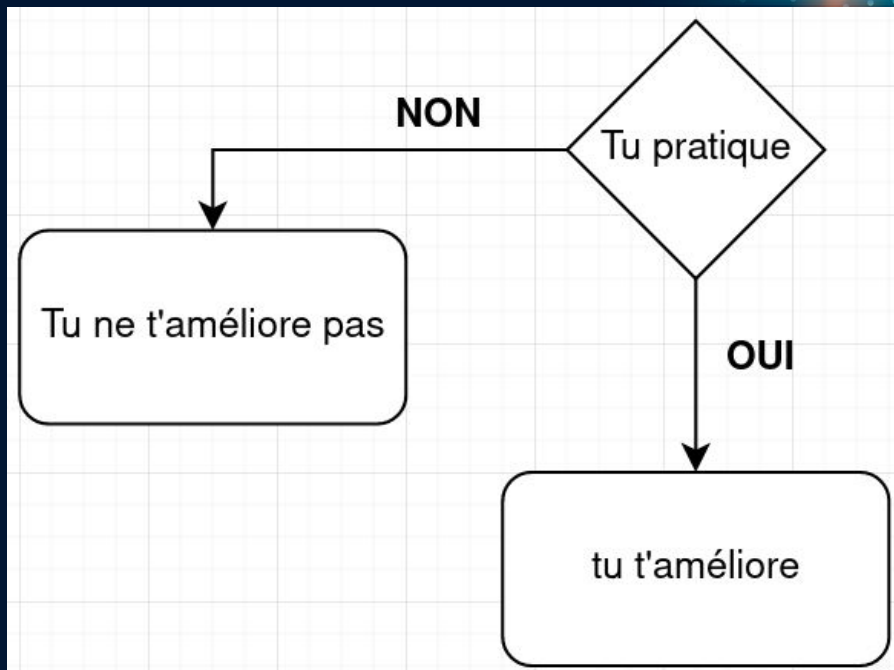
Conditions & Loops

Do it again ! But write it once !

Qu'est-ce qu'une condition ?

Si tu pratiques tu t'amélioreras, sinon tu ne t'amélioreras pas

```
if condition:  
    #code a executer si la  
    # condition est vrai.  
else:  
    #code a executer si la  
    #condition est fausse.
```



Do it again with “while” ?

```
while(condition_is_true):  
    # the code you want to run  
    # while the condition is  
    # true
```

Example

```
>>> i = 0  
>>> while (i<5):  
...     print(i)  
...     i+=1  
...  
0  
1  
2  
3  
4
```

Do it again with “for” ?

En général, la boucle ‘for’
est utilisé sur les
conteneurs, par exemple :

```
>>> for i in [1,2,True,5.4,8,"hello"]:  
...     print(i)  
...  
1  
2  
True  
5.4  
8  
hello
```

Mais il est possible de l'utiliser
de manière classique en
utilisant ‘range’ :

```
>>> for i in range(0,5):  
...     print(i)  
...  
0  
1  
2  
3  
4
```

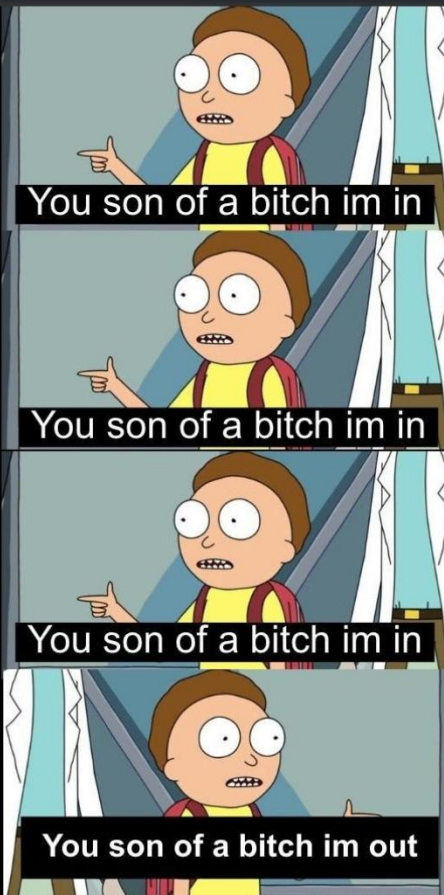
```
for i in range(3):
```

$i = 0$

$i = 1$

$i = 2$

$i = 3$



Let's practice



Petite Exo

Écrivez un script ou vous allez demander à un utilisateur de rentrer son nom, prénom et âge.

Testez l'âge de la personne, si son âge est inférieur à 25 ans affichez qu'il s'agit d'un gen Z, si son âge est entre 25 et 45, affichez qu'il s'agit d'un millénal et enfin si son âge est supérieur à 45, affichez qu'il s'agit d'un boomer





03

Functions

DRY code is the best code.
Don't repeat yourself, write
functions!

Qu'est-ce qu'une Fonction ?

Un bout de code qu'on appelle pour exécuter une certaine tâche

- Cette dernière peut recevoir des arguments
- Et peut retourner des valeurs

```
type help / copyright /  
>>> def addition(a, b):  
...     return a + b  
...  
>>> addition(5, 10)  
15  
>>> |
```

Pourquoi utiliser une Fonction ?

- Devise un programme large en plusieurs petites parties
- Facilite l'organisation du code
- Facilite la lisibilité du code
- Rend le code plus maintenable et facilite la détection d'erreur et de bug
- Permet d'éviter la répétition dans le code
- Facilite la coopération entre différentes personnes dans un projet

Les valeurs par défaut des fonctions

```
def table(nb, max=10):  
    """Fonction qui affiche la table de multiplication par nb"""  
    i = 0  
    while i < max:  
        print(i + 1, "*", nb, "=", (i + 1) * nb)  
        i += 1
```

```
>>> table(5)  
1 * 5 = 5  
2 * 5 = 10  
3 * 5 = 15  
4 * 5 = 20  
5 * 5 = 25  
6 * 5 = 30  
7 * 5 = 35  
8 * 5 = 40  
9 * 5 = 45  
10 * 5 = 50  
>>> table(5, max=5)  
1 * 5 = 5  
2 * 5 = 10  
3 * 5 = 15  
4 * 5 = 20  
5 * 5 = 25  
>>> table(5, max=1)  
1 * 5 = 5
```

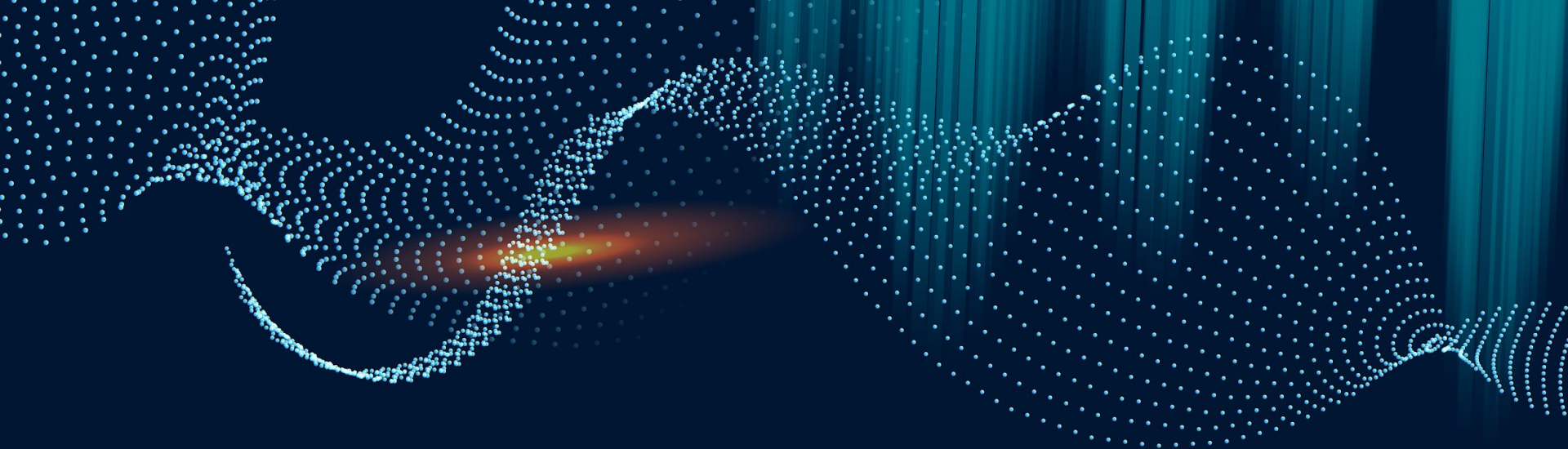
Petite Exo

Écrivez le script créé précédemment et mettez le Teste de l'âge de la personne dans une fonction qui prend comme paramètre l'âge de la personne (avec une valeur par défaut de 10).



Let's practice





OX

Bonus



Comments

Always comment your code !

More infos about comments

Comments

Les commentaires après le symbole '#' sur une ligne et servent typiquement à expliquer un bout de code

Docstrings

Les Docstrings commencent et finissent par ''' ou """ et sont typiquement utilisés pour décrire une fonction, module ou classe

The background features a dark teal color with several lighter teal, wavy, vertical lines that create a sense of motion. A bright orange and yellow glow emanates from the center-left, casting a soft light across the scene. The overall aesthetic is modern and digital.

Run your first script



**Organize your project into
modules**



Use the Doc

The link will be shared in the chat
and the server

DO You Have Any

Question?





A small project

The project will also be shared in
the server

Jeu du plus et du moins

Développer un Mini-jeu sur console en python, dont le but est de deviner un nombre piocher au hasard (entre 0 et 100).

Le joueur disposera de 10 tours, a chaque tour il donnera un nombre, si le nombre est le même que celui tiré au sort il gagne, sinon on affichera dans la console “plus” ou “moins” en fonction de si le nombre choisi est inférieur ou supérieur à celui pioché aléatoirement.

Si après 10 essais le joueur n’a pas trouvé le nombre il a perdu



Exemple

```
aymen@DESKTOP-TJMNVI0:/mnt/c/Users/aymen/documer  
essayer de deviner le nombre mystere...50  
c'est moins  
essayer de deviner le nombre mystere...25  
c'est plus  
essayer de deviner le nombre mystere...35  
c'est moins  
essayer de deviner le nombre mystere...30  
c'est plus  
essayer de deviner le nombre mystere...33  
Felicitation, solution trouver en 5 essais
```

Jeu de la roulette

Développer un Mini-jeu sur console en python, qui simule un jeu de roulette dans un casino.

Le joueur disposera au début du jeu de 1000 \$, a chaque tour le joueur choisit un nombre entre (0 et 49) plus un montant à miser qui sera déduit de la somme totale du joueur.

La roulette (le programme) vas ensuite choisir un numéro gagnant entre 0 et 49, et si le numéro gagnant est le numéro misé le joueur gagne 3 fois le montant miser sinon le joueur reprend la moitié du montant misé si le nombre est de même parité et rien dans le cas contraire.

La partie s'arrête une fois le montant du joueur atteint 0.



Exemple

```
aymen@DESKTOP-TJMNVI0:/mnt/c/Users/aymen/documents/projects/python/roulette_ga
.py
montant initial 1000 $.
nombre a miser
Tapez un nombre (entre 0 et 49 )... 5
mise
Tapez un nombre (entre 0 et 1000 )... 500
La roulette tourne... ... et s'arrête sur le numéro 45
Vous avez misé sur la bonne couleur. Vous obtenez 250 $
Vous avez à présent 1250 $
nombre a miser
Tapez un nombre (entre 0 et 49 )... 50
Ce nombre est supérieur à49
Tapez un nombre (entre 0 et 49 )... 12
mise
Tapez un nombre (entre 0 et 1250 )... 500
La roulette tourne... ... et s'arrête sur le numéro 25
Désolé l'ami, c'est pas pour cette fois. Vous perdez votre mise.
Vous avez à présent 750 $
nombre a miser
Tapez un nombre (entre 0 et 49 )... 5
mise
Tapez un nombre (entre 0 et 750 )... 750
La roulette tourne... ... et s'arrête sur le numéro 30
Désolé l'ami, c'est pas pour cette fois. Vous perdez votre mise.
Vous êtes ruiné ! C'est la fin de la partie.
```



Random numbers

THANKS!

Do you have any questions?
You can ask it in the discord server

By OpenMindsClub

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

Please keep this slide for attribution.

