

Python en s'amusant !

I - Dire bonjour

Dans Python, la commande **print()** est une fonction, c'est-à-dire une sorte de mini programme préétabli. Pour que cette fonction entre en application, il te suffit d'écrire le mot correspondant. Grâce à la fonction **print()**, l'ordinateur fait apparaître à l'écran tout ce qui se trouve entre les parenthèses.

Fais dire à l'ordinateur : Bonjour, je m'appelle...

II – Jouons avec les chiffres

Avec Python, c'est facile de faire des calculs. Il te suffit de taper un calcul et d'appuyer sur entrée pour obtenir la réponse.

Additionne : 2+5

Soustrais : 89-123

Multiplie : 45 X 678

Divise : 456 / 34

Amuse-toi à calculer ce qui te passe par la tête

III – Les variables

Une variable est un peu comme une boîte de rangement étiquetée dans laquelle on stocke des informations. Tu peux changer les informations mais l'étiquette ne change pas.

Pour indiquer à l'ordinateur ce que doit être ta variable, utilise le signe =. On appelle cela assigner une valeur à la variable.

On compte les bananes !

- *Assigne à la variable bananes, la valeur 15.*
- *En utilisant les variables bananes et bananesmangées, calcule combien de bananes le singe a laissées s'il a mangé 7 bananes*

Note : dans python, on appelle chaîne, un mot ou n'importe quelle séquence de lettres et de symboles. Les chaînes sont écrites entre guillemets.

- *Crée une variable et affecte-lui une chaîne*
- *Fais apparaître à l'écran le texte*

Laisser l'utilisateur décider

Si tu veux que l'utilisateur d'un programme décide quelle valeur donner à la variable, tu peux utiliser la fonction **input()**

input() est une fonction qui indique à l'ordinateur d'arrêter le programme tant que l'utilisateur n'a pas tapé quelque chose et appuyé sur la touche entrée.

*Utilise la fonction **input()** pour demander le nom à quelqu'un puis affiche à l'écran le nom de la personne.*

Créer un personnage

- *Crée un titre pour introduire ton programme que tu intituleras « Création de ton personnage »*
- *Utilise la fonction **input()** pour demander à l'utilisateur de choisir le nom de son personnage*
- *Répète ensuite cette opération pour l'âge, les forces et les faiblesses du personnage, sur une nouvelle ligne à chaque fois*
- *Demande à l'ordinateur de créer le personnage en utilisant la fonction **print()**. Place chaque instruction sur une nouvelle ligne.*

Calcul l'âge que tu auras dans un an...

Lorsqu'un utilisateur fournit un input, Python le stocke en tant que chaîne. Si l'input s'avère être un nombre, tu dois indiquer à l'ordinateur de transformer cette chaîne en nombre en utilisant `int()`. Ceci indique à l'ordinateur de traiter la variable comme un nombre entier.

- *Affecte à la variable `âge` la chaîne « Quel âge as-tu ? »*
- *Crée une nouvelle variable correspondant à l'âge que tu auras à ton prochain anniversaire et affiche-la à l'écran sous la forme : Tu auras ... ans à ton prochain anniversaire.*

III – Prendre des décisions

La notion de condition (ne fonctionne pas sur repl.it)

Une condition est une portion de code qui compare deux informations. Pour établir ces comparaisons, les conditions utilisent des opérateurs (`==`, `>`, `<`....)

- *Crée une variable `âge` et affectes-y ton âge*
- *Demande à l'ordinateur si tu as 50 ans*
- *Que répond-il ?*
- *Demande-lui maintenant si tu as ton âge*
- *Que répond-il ?*

Une instruction conditionnelle

Pour introduire une condition dans ton script, tu as besoin d'une instruction conditionnelle.

Exemple de la vie de tous les jours :

S'il pleut : tu prends ton parapluie

Sinon : tu ne le prends pas

En langage python, le si est un **if** et le sinon un **else**

Aimes-tu les moutons ?

- *Crée une variable reponseutilisateur*
- *Demande à l'utilisateur s'il aime les moutons. Lui dire qu'il doit répondre par oui ou non.*
- *Ecris un script qui indique "Beh Beh" si la réponse de l'utilisateur est oui et "les moutons ne t'aiment pas non plus" si la réponse de l'utilisateur est non.*

Nous pouvons également rajouter une condition avec un elif (sinon si) :

- *Crée une variable reponseutilisateur*
- *Demande à l'utilisateur s'il aime les moutons. Lui dire qu'il doit répondre par oui, non, peut-être ou ceux aux yeux bleus.*
- *Ecris un script qui indique "Beh Beh" si la réponse de l'utilisateur est oui, "les moutons ne t'aiment pas non plus" si la réponse de l'utilisateur est non, « hummmmm » si la réponse est ceux aux yeux bleus, « décide-toi, humain ! » si la réponse est peut-être et « Tu n'as rien compris ! » si la réponse est autre.*

IV – Planifier des programmes

1 – La fonction randint

Pour introduire un élément au hasard, nous pouvons utiliser la fonction `randin()`. Il faut d'abord importer le module (ou bibliothèque) **random** en faisant un : `from random import *`

randim s'utilise de cette manière :

Tape ce script dans repl.it et observe ce qu'il se passe :

```
from random import *  
x = randint(1,3)  
print(x)
```

Fais tourner le programme plusieurs fois. Explique les résultats obtenus

2- Le château du dragon affamé !!!!!

Je vais exécuter un programme. *Tu devras reconstituer le script qui permet de le faire fonctionner... bon courage !*

V – Les boucles

Une boucle désigne simplement une portion de code qui se répète. Une boucle évite au programmeur de réécrire sans cesse la même chose.

1 – La boucle While (tant que)

a- *Un peu de théorie*

Exemple d'écriture d'une **boucle while**

Ecris le programme suivant, regarde ce qu'il se passe et explique le code :

```
x = 1
while x < 100 :
    print("la valeur de x est", x)
    x = x*2
```

A toi de jouer en créant un code qui exécute le programme suivant : tant que je réponds « non » j'écris « mauvaise réponse » et le programme continue et quand je réponds oui j'imprime « bonne réponse »

Note : **les boucles while** ne s'arrêtent que lorsqu'une chose change. Si cette chose ne change pas, elles peuvent se répéter indéfiniment. (tant qu'on répond « non », dans l'exemple précédent, le programme se répète !)

b- *On complique un peu !*

Jouons ensemble à la devinette.

Ecris le script de ce jeu. Voici quelques aides : il faut utiliser la fonction `randint`, une boucle `while` et des instructions conditionnelles...

A toi de jouer !

2- la boucle for

a- Un peu de théorie

Avec les **boucles for**, nous pouvons dire quand les boucles peuvent s'arrêter. On doit donc indiquer combien de fois l'action doit se répéter. Pour cela, nous avons besoin d'utiliser une fonction appelée **range()** (fourchette).

Exemple d'écriture d'une **boucle for**

Ecris le programme suivant, regarde ce qu'il se passe et explique le code :

```
for i in range(0, 11):  
    print("La valeur de i est :", i)
```

b- Un exercice

Ecris un script, qui te calcule les tables de multiplication.

Aides : commence par définir une variable table à laquelle tu affecteras la table que l'utilisateur veut calculer. Tu calculeras pour chaque table de 0 à 12. Exemple pour la table de 2 tu iras de 0x2 à 12x2.

A toi de jouer !

c- Aller le dernier exercice du cours de SNT sur python !

Ecris le retour écran de ce programme et explique-le :

```
for x in range(1,5) :  
    print("houhou")  
    for y in range(1,5) :  
        print("hihi")  
    print("huhu")
```

FIN