

Python 1 - Initiation à Python

1 - Thonny

Python est un langage de programmation en lignes de code dont la syntaxe est proche du « langage naturel ». On peut installer Python directement sur l'ordinateur ou passer par des logiciels (environnements de développement) permettant d'offrir des fonctionnalités supplémentaires.

Nous utiliserons **Thonny** : gratuit, téléchargeable sur tous les systèmes d'exploitation.

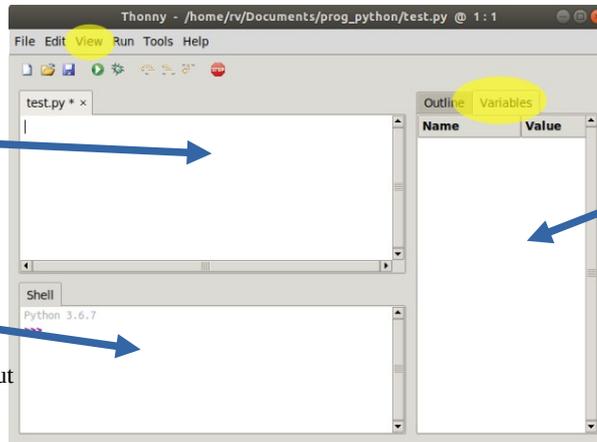
Zone de programmation

Permet de mettre du code enregistré en mémoire.

On active le programme en appuyant sur la flèche VERTE

Zone interactive (console ou shell)

On tape des instructions qui s'exécutent tout de suite mais une seule fois



Onglet des Variables

Permet d'observer directement le contenu en mémoire.

2 - Variables

Un langage de programmation permet de **stocker des données** dans la mémoire. Les **variables** permettent d'enregistrer des données puis de les lire plus tard.

Pour créer une variable, il suffit d'utiliser l'opérateur =.

On place **à gauche le nom** de la variable voulue.

On place **à droite l'expression** qu'on veut lui affecter.

01° Télécharger Thonny sur votre ordinateur.

02° Aller dans le menu **Affichage/Views** et sélectionner **Variables** si l'onglet Variables n'est pas ouvert.

03° Dans la zone Console, tapez ceci. Que constate-on ?

>>> **50 + 20** Effet : _____

>>> **a = 50** Effet : _____

>>> **b = 20** Effet : _____

04° Lorsqu'on affecte un contenu à une variable, Thonny affiche-t-il sa valeur dans l'onglet Variables ? _____

05° Lorsqu'on affecte un contenu à une variable, Thonny affiche-t-il sa valeur dans l'onglet Console ? _____

06° Dans la zone Console, tapez ceci. Que constate-on ?

>>> **a** Effet : _____

>>> **b** Effet : _____

>>> **a + b** Effet : _____

>>> **c = a + b** Effet : _____

07° Que fait-il taper pour **visualiser c** dans **la console** ? >>> _____

Cette année, vous allez rencontrer principalement :

* des entiers : on dit **integers** (int) en informatique.

* des nombres à virgules : on dit **floats** (float) en anglais.

* des chaînes de caractères : on dit **strings** (str) en anglais.

* des booléens : on dit **booleans** (bool) en anglais. Cela ne peut valoir que **True** ou **False**.

08° Pour chacun des exemples, donnez le nom du contenu obtenu en mémoire :

>>> a = 6	Type : _____	Que réalise = ? _____
>>> b = 3	Type : _____	
>>> a + b		
9	Type : _____	
>>> a * b		
18	Type : _____	
>>> a / b		
2.0	Type : _____	
>>> a // b		Que réalise // ? _____
2	Type : _____	
>>> t = "bon"	Type : _____	
>>> t		
'bon'	Type : _____	
>>> t * 2		
'bonbon'	Type : _____	
>>> t + t		
'bonbon'	Type : _____	
>>> a == b		Que réalise == ? _____
False	Type : _____	
>>> a != b		Que réalise != ? _____
True	Type : _____	

3 – Séquentialité lors d'une affectation

Lors d'une affectation, Python :

1. évalue **d'abord** l'expression de droite
2. affecte **ensuite** le résultat à la variable de gauche. En algorithmique, on note souvent **a ← 50**

Exemple : >>> note = note + 4 → si note contient 8, (note + 4) est évaluée à 12 PUIS on affecte 12 à a.

09° Fournir les réponses aux différentes instructions.

>>> a = 30	Que vaut a maintenant ? _____
>>> a = a + 1	Que vaut a maintenant ? _____
>>> a = a + 1	Que vaut a maintenant ? _____
>>> a + 1	Que vaut a maintenant ? _____

Lire le vocabulaire puis répondre à ces questions :

Vocabulaire

- + **Affectation** : associer une nouvelle valeur dans une variable (a = 50 par exemple)
- + **Incrémentation** : augmenter la valeur d'une variable de 1 (a = a + 1 par exemple)
- + **Evaluation** : calculer une expression (a + 1 par exemple)
- + **Concaténation** : additionner deux strings ("Bon"+"jour" évalué à "Bonjour" par exemple)

10° Comment se nomme cette opération ? Affiche-t-elle quelque chose ? >>> a = 50

11° Comment se nomme cette opération ? Affiche-t-elle quelque chose ? >>> a == 50

12° Comment se nomment ces opérations ? Affiche-t-elle quelque chose ? >>> a = "Cou"+"cou"

4 – Séquentialité sur un programme

Un programme est une suite d'instructions qui sont exécutées **une par une** et de **haut en bas**.
L'ordinateur ne passe à l'instruction suivante que lorsqu'il a fini celle en cours.

Dans un **programme** (contrairement à la Console), taper le nom d'une variable ne provoque pas son affichage:

```
Programme    a * 2                Evaluation de a * 2. Pas d'affichage ni stockage.  
                                         Une ligne ne contenant que cela est donc inutile.
```

13° Lancer un **programme** contenant uniquement ces 2 lignes. Obtient-on un affichage ?

```
a = 20  
a + 5
```

14° Fournir les effets progressifs et le contenu final des variables créées (sans lancer réellement le programme) :

```
age = 16                Effet : _____  
limite = 18            Effet : _____  
autorisation = age > limite  Effet : _____  
limite = 15            Effet : _____  
limite +1              Effet : _____  
age = age + 10        Effet : _____
```

15° Lancer le programme pour vérifier. Les variables s'affichent dans l'onglet VARIABLES (ou vous pouvez taper leurs noms dans la Console pour obtenir le contenu final).

En Python, on peut dire de faire certaines choses sous condition. La première condition est exprimée derrière le mot **if**. On doit placer derrière une expression qui sera évaluée comme un booléen (True ou False) puis le caractère : . Remarquez bien que les instructions à effectuer sont décalées de 4 espaces. On peut placer des conditions supplémentaires en utilisant **elif** qui veut dire sinon si.

16° A vous de deviner ce que va contenir **appreciation** au fur et à mesure du programme :

```
01 note = 12  
02 appreciation = "bof"  
03 if note > 13 :  
04     appreciation = "bien"  
05 elif note < 8 :  
06     appreciation = "pas bien"  
07 note = 7
```

On peut encore fait mieux : on peut placer un **else** généraliste qui veut dire : si aucune des possibilités précédentes n'est vraie, fait cela.

17° A vous de deviner ce que va contenir **autorisation** au fur et à mesure du programme :

```
01 recu = "1234"  
02 mdp = "12345"  
03 if recu == mdp:  
04     autorisation = True  
05 else:  
06     autorisation = False
```

18° On remarquera qu'on aurait pu faire plus simple en réalité. Expliquer ce que contient **autorisation** ci-dessous. La variable **recu** contient le mot de passe reçu. La variable **mdp** contient le vrai mot de passe attendu.

```
01 recu = "1234"  
02 mdp = "12345"  
03 autorisation = recu == mdp
```

19° Si $a = 10$ et $b = 20$, que va contenir $c = a > b$?

20° Si $\text{recu} = "1234"$ et $\text{mdp} = "5555"$, que va contenir $c = \text{recu} \neq \text{mdp}$?



www.infoforall.fr