

---

**formation**

**Professeurs du lycée Diderot**

**janv. 11, 2019**



---

## Table des matières

---

<b>1</b>	<b>Quoi ? pour qui ?</b>	<b>1</b>
<b>2</b>	<b>1. Où se trouve le raccourci vers thonny</b>	<b>3</b>
<b>3</b>	<b>2. Prise en main :</b>	<b>5</b>
3.1	Quelques instructions à essayer dans le shell : . . . . .	5
3.2	Quelques exemples à enregistrer dans un fichier : . . . . .	5
<b>4</b>	<b>3. Quelques trucs sur python :</b>	<b>7</b>
4.1	3.a. Variable . . . . .	7
4.2	3.b. Typage et ce que provoque l'absence de typage . . . . .	7
4.3	3.c. Deux trucs : <- en faire une note ? Je suis pas sur que ca mérite un titre. . . . .	8
4.4	3.d. Deux fonctions d'entrée / sortie : . . . . .	8
4.5	3.e. Retour sur les problèmes de typage : formule magique : ...=float(input(« ... »)) . . . . .	8
<b>5</b>	<b>4. Librairies :</b>	<b>9</b>
5.1	4.a. Généralités : . . . . .	9
5.2	4.b. Quelques exemples avec les librairies « maison » . . . . .	9
<b>6</b>	<b>5. python avec les élèves : prise en main</b>	<b>11</b>
<b>7</b>	<b>6. gestion salle info</b>	<b>13</b>
<b>8</b>	<b>7. un peu plus loin :</b>	<b>15</b>
8.1	7.a . . . . .	15
<b>9</b>	<b>7.b. un ou deux autres exemples d'activités à faire avec les élèves.</b>	<b>17</b>



# CHAPITRE 1

---

## Quoi ? pour qui ?

---

Ce document est utilisé pour former les collègues du lycée à l'utilisation de l'environnement python du lycée. Il est à destination des professeurs n'ayant pas ou peu de connaissances sur le langage python.

L'objectif est d'introduire, à l'aide d'exemples et d'exercices :

- le langage de programmation
- l'environnement de développement thonny utilisé au lycée
- les bibliothèques propres au lycée Diderot
- quelques constats pratiques d'ordre pédagogique.



## CHAPITRE 2

---

### 1. Où se trouve le raccourci vers thonny

---

Logiciels > MATH > \_Maths > *RACCOURCIS MATH* > Thonny python





# CHAPITRE 3

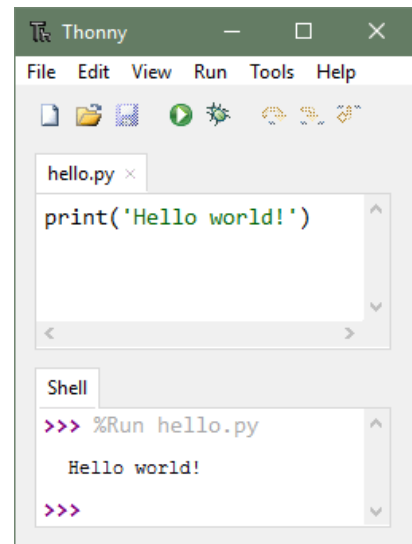
## 2. Prise en main :

Quand on lance Thonny, il y a deux zones :

- la zone du haut où on travaille dans un fichier, qui peut être exécuté quand on le demande, et qui peut être enregistré (sur cette image, le fichier s'appelle « Hello.py »)
- la zone du bas qui est un **shell** : comme un écran de calculatrice, mais qui exécute des commandes Python (on ne peut pas l'enregistrer)(c'est dans cette zone qu'apparaissent les résultats des commandes exécutées depuis un fichier de la zone du haut)


### 3.1 Quelques instructions à essayer dans le shell :

```
3 + 4
print("hello world")
"hello" + "world"
```



### 3.2 Quelques exemples à enregistrer dans un fichier :

```
3 + 4
print("hello world")
"hello" + "world"
```

Pour exécuter le fichier, il faut cliquer sur le bouton  ou appuyer sur **F5**. Il faut donner un nom au fichier. Le fichier sera sauvegardé par défaut dans : USER/python



## CHAPITRE 4

---

### 3. Quelques trucs sur python :

---

Python est un langage de programmation libre qui fonctionne sur toutes plate-formes (Windows, Mac, Linux, Android, ...). Il a été créé au début des années 90 et baptisé en l'honneur de la série Monty Python's Flying Circus.

Épuré et lisible, Python est utilisé par de nombreuses entreprises et organisations (Google, NASA, ...), dans la recherche scientifique et est enseigné au lycée, en prépa, et à l'université.

Les projets de **nouveaux programmes** précisent le choix de python comme langage de programmation utilisé à partir de la seconde.



#### 4.1 3.a. Variable

Une des difficultés principales pour les élèves au début (en seconde) est l'utilisation de variables.

---

**Note :** Proposer ici l'utilisation de « boîtes » comme aide didactique à la notion de variable ?

---

Quelques lignes à tester :

```
x = 4
y = x + 1
print(y)
```

#### 4.2 3.b. Typage et ce que provoque l'absence de typage

Les variables peuvent être de différents types : par exemple « entier » `int` ou « chaîne de caractères (texte) » `str` ou encore « nombre à virgule flottante » `float`.

Souvent, on peut suggérer facilement à Python le type utilisé. Par exemple, `a=2` est de type `int`, `a=2.0` est de type `float`, `a='2'` ou `a="2"` est de type `str`.

Si Python ne sait pas trop quel est le type d'une variable, il essaye de faire un choix (par exemple il considère que la variable est de type « texte » `str`) ou alors il renvoie un message d'erreur.

Les messages d'erreur apparaissent en rouge dans le shell. Même s'ils sont indigestes, il faut expliquer aux élèves que dans un message d'erreur il y a deux informations très utiles :

Essayez de faire exécuter ces instructions :

```
3 * '13'
'3' * '13'
'3' + '13'
3 + '13'
3 * 13
3 * 13.0
```

### 4.3 3.c. Deux trucs : <- en faire une note ? Je suis pas sur que ça mérite un titre.

- pour aller vite dans le shell, on peut rappeler les instructions précédemment tapées en appuyant sur la flèche du haut
- pour mettre un commentaire, il suffit de le précéder d'un #

### 4.4 3.d. Deux fonctions d'entrée / sortie :

Essayez de faire exécuter ces instructions :

```
input("tapez quelque chose")
a = input("tapez quelque chose")
a
print(a)
type(a)
```

### 4.5 3.e. Retour sur les problèmes de typage : formule magique : ...=float(input(« ... »))

Si vous avez essayé les lignes de code précédentes avec diverses entrées tapées au clavier, vous avez peut-être remarqué que quel que soit ce qu'on tape, Python considère que ce qui vient du `input()` est du type `str`.

Du coup, cela va sans cesse provoquer des erreurs dans les programmes des élèves. Un des moyens d'éviter cela est de leur donner la ligne de `input` sous la forme suivante :

```
a = float(input("tapez quelque chose"))
```

a sera alors automatiquement du type « nombre à virgule flottante », ce qui est en général ce qu'on veut.

Exercice : Demander deux nombres à l'utilisateur et afficher leur produit.

## 4. Librairies :

### 5.1 4.a. Généralités :

Python ne charge pas toutes les commandes disponibles lorsqu'on le lance. Si on a besoin d'une commande non chargée, il faut demander le chargement de la librairie.

```
print(sqrt(2))
```

solutions :

1. On importe la fonction `sqrt` de la librairie `math`. On peut ensuite l'utiliser directement.

```
from math import sqrt
print(sqrt(2))
```

2. Une autre solution qui importe tout le contenu de la librairie `math` :

```
from math import *
print(sqrt(2))
print(pi)
```

### 5.2 4.b. Quelques exemples avec les librairies « maison »

La librairie `entree_tk` permet de répondre au problème de typage évoqué précédemment :

```
from entree_tk import *
demander_texte()
a=demander_reel()
print(type(a))
print(a)
```

La librairie `repere` permet d'afficher facilement une fenêtre graphique munie d'un repère où on fera par exemple afficher la courbe d'une fonction.

```
from repere import *
creer_fenetre()
trace_segment(1,2,3,5)
trace_point(3,3)
```

Un autre exemple, plus élaboré :

```
from math import *
from graphique import *
creer_fenetre()
for i in range(100): #i va de 0 à 100
    x = i / 10 - 5 #x va de -5 à 5
    trace_point(x, cos(x))
```

La même chose avec un « while » au lieu du « for » (quel est le plus simple avec les élèves?) :

```
from math import *
from graphique import *
creer_fenetre()
i = 0
while i < 101: # i va de 0 à 100
    x = i / 10 - 5 # x va de -5 à 5
    trace_point(x, cos(x))
    i = i + 1
```

Je propose cette version plus simple :

```
from math import *
from graphique import *
creer_fenetre()
x = -5
while x < 5: # x va de -5 à 5
    trace_point(x, cos(x))
    x = x + 0.1
```

# CHAPITRE 6

## 5. python avec les élèves : prise en main

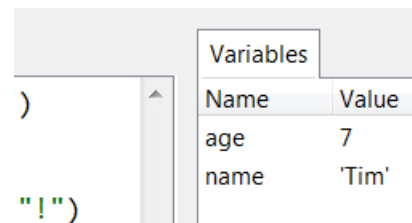
- Les élèves enregistrent leur travail dans le dossier USER/python/.

Lors de la vidéoprojection de thonny, les caractères sont trop petits. On peut utiliser ctrl +/- crl molette pour zoomer

- La config est enregistrée automatiquement quand on quitte thonny
- Thony propose un affichage dynamique du contenu des variables qui peut être particulièrement utile dans un cadre pédagogique. Son utilisation n'est **pas compatible avec l'import de librairies**. En effet, l'onglet affichera le contenu de toutes les fonctions et variables importées ce qui peut ralentir considérablement l'utilisation.
- Pour afficher les numéros des lignes : tools > options > show lines numbers
- debug et les différents types de « pas à pas »
- exemple de programme :

```
xA = float(input("Abscisse de A ? "))
yA = float(input("Ordonnée de A ? "))
xB = float(input("Abscisse de B ? "))
yB = float(input("Ordonnée de B ? "))
xM = (xA + xB) / 2
yM = (yA + yB) / 2
print("Coordonnées du milieu : (" + str(xM) + " ; " + str(yM) + ")")
```

→ exo : le refaire et l'améliorer en ajoutant de quoi afficher la distance AB → on peut demander exactement la même chose à une classe à condition d'écrire au tableau la « formule magique » ou d'utiliser la librairie `entree_tk`



Variables	
Name	Value
age	7
name	'Tim'





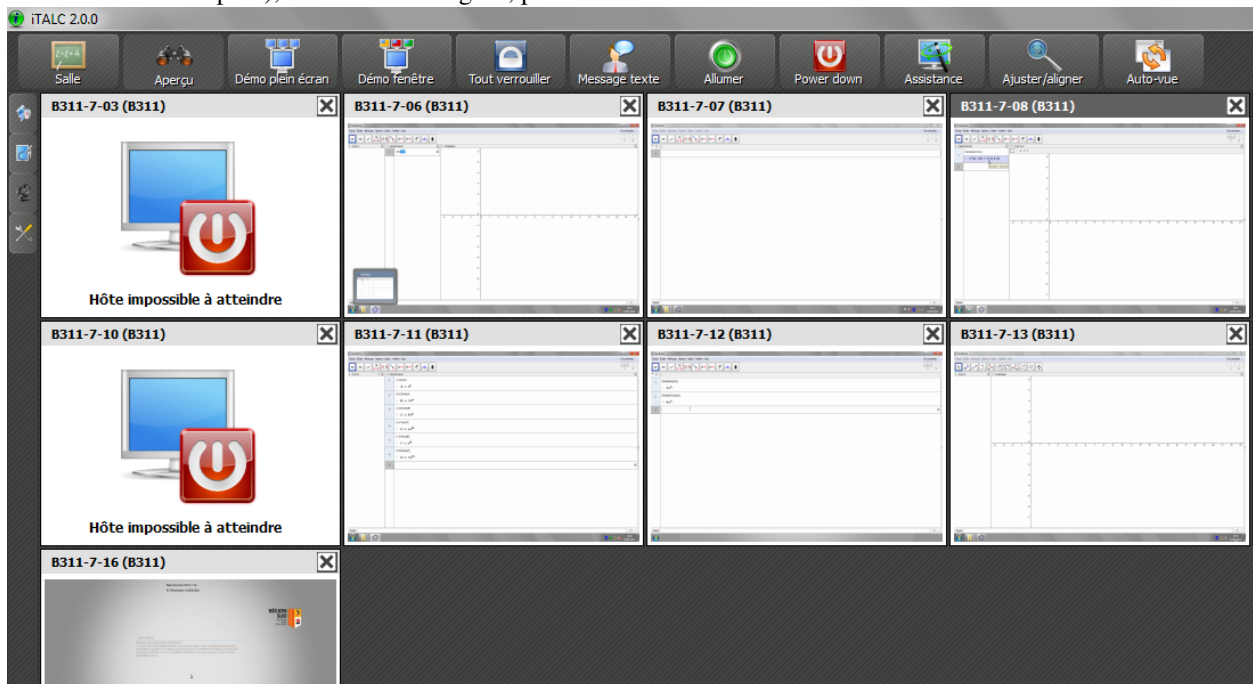
## CHAPITRE 7

### 6. gestion salle info

Ce qui ne marche pas : parler à un groupe d'élèves devant leur écran.

Ce qui peut marcher :

- faire se lever tout le groupe, venir devant le tableau et parler avec le vidéo-projecteur
- verrouiller tous les écrans (ça peut se faire avec le logiciel iTALC (il y a souvent un ou deux écrans qui ne se verrouillent pas !), donner les consignes, puis déverrouiller



- envoyer son écran dans les écrans de tous les élèves (avec la commande « démonstration » du logiciel iTALC), parler en montrant à l'écran en même temps, puis arrêter la démonstration (les élèves sont encore bluffés par cette manipulation pour le moment :-)
- annoncer au groupe « je vais vous envoyer l'écran de -nom d'élève- » (avec la commande clic droit > « laisser faire une démonstration » du logiciel iTALC), laisser l'élève faire sa démonstration (en l'incitant à parler de façon

compréhensible), puis arrêter la démo (je ne sais pas comment faire alors je verrouille / déverrouille !)

---

### 7. un peu plus loin :

---

#### 8.1 7.a

- comment envoyer du code aux élèves ? / comment récupérer du code qu'ils ont fait ?
- comment peut-on (les élèves ou nous) télécharger thonny pour chez nous ? Il suffit de se rendre sur <https://pydiderot.readthedocs.io>
- comment peut-on demander à ajouter une fonction dans les librairies ? Ouvrir une issue sur <https://github.com/cspaier/pydiderot>



## CHAPITRE 9

---

7.b. un ou deux autres exemples d'activités à faire avec les élèves.

---

traceur de courbes ? discriminant ?

un peu plus de python : listes, boucles Les listes sont-elles nécessaires ?