

1 Pyzo

- Ouvrez votre gestionnaire de fichier puis copier le fichier *Général* → *Maths* → *Meilland* → *python_seconde.ipynb* dans votre dossier personnel
- Ouvrez **Seven_Pyzo** (*S'il vous le demande cliquez sur "use this environnement"*).
Vous voir le *shell* python avec ses chevrons »>.
- Testez le shell en calculant par exemple 3+4
Toujours dans le shell python de **Pyzo** exécuter les séquences suivantes.
- Installer *jupyter*.
 - `pip install --upgrade pip`
 - `pip install jupyter` Cela prend un peu de temps.
- Lancer notebook
 - **notebook**
Un navigateur s'ouvre. Chrome pose problème dans certaine salle, il faudra alors copier l'adresse et la coller dans un autre navigateur (firefox ou internet explorer).
 - Cliquer sur *Upload* et sélectionner votre fichier *python_seconde.ipynb*
 - Il ne reste plus qu'à double cliquer sur le fichier *python_seconde.ipynb*.

2 Jupyter notebook

Mes cours sont au format **ipynb**. Ces fichiers sont lisibles avec le programme **jupyter-notebook** (son ancien nom était **ipython-notebook**).

L'intérêt de ce format est que le fichier se lit avec un navigateur (firefox ou chrome) et que l'on peut exécuter du code python en même temps en appuyant sur :

- **CTRL+Entrée** : pour exécuter le code et rester sur la même cellule.
- **Maj+Entrée** : pour exécuter le code et passer à la cellule suivante.
- Les flèches *UP* et *DOWN* pour naviguer dans la page sans exécuter les cellules.
- **Entrée** : pour éditer la cellule
- **Esc** : pour sortir du mode édition.

Il existe aussi une version de jupyter en ligne : <https://try.jupyter.org/>

3 Les bases de python3

Les types

Les différents types entiers, flottants, complexes, chaînes de caractères, booléens, listes, tuples et dictionnaires.

- *entier (integer)* : 3
- *flotant (float)* : 2.3
- *chaînes de caractères (string)* : 'ISN'
- *Boléen* : True
- On aura toujours à l'esprit que $0.2 + 0.7 = 0.8999999999999999 \neq 0.9$
- *int()* permet de convertir, un nombre ou une chaîne de caractère en un entier.
- *float()* permet de convertir, un nombre ou une chaîne de caractère en un flottant.
- On utilise le caractère # pour écrire un commentaire

Les opérateurs

	Opération	algorithmique	python
	Addition	2+3	2+3
	Soustraction	12-5	12-5
	Multiplication	3*6	3*6
	Division	7/2	7/2
	Quotient de la division euclidienne	7 div 2 ou div(7,2)	7//2
	Reste de la division euclidienne	7 mod 2 ou mod(7,2)	7%2
	puissance	7^2	7**2
	racine carrée	$\sqrt{2}$ ou sqrt(2)	sqrt(2)

Variables

- Python utilise le symbole = pour affecter une valeur à une variable.
- Attention à ne pas confondre $a = 12$ et $a == 12$.

'Python3'

```
A=3; B=5
C='toto'
```

Entrées

- *input("texte")* permet de saisir du texte pour un programme.
- Il faudra éventuellement convertir ce texte dans le type voulu avec *int()* ou *float()*

```
a=input("Saisir un texte a: ")
b=int(input("Saisir un entier b: "))
c=float(input("Saisir un réel c: "))
```

Affichage

- On utilise de préférence la méthode *format()* pour afficher du texte et des variables.

```
print("La valeur de la variable a est",str(a),".") # si a est un nombre
print("La valeur de la variable a est " + str(a) + ".") # autre forme
print("La valeur de la variable a est {}".format(a)) # je préfère
print("Le produit de {} par {} est {}".format(a,b,a*b))
print("{}*{1}={2} et {0}/{1}={3}".format(a,b,a*b,a/b))
```

Connecteurs logiques

algorithmique	python
$a = b$	$a == b$
$a \neq b$	$a != b$
A et B	A and B

algorithmique	python
A ou B	A or B
non A	not(A)

Condition SI

```
if x<0:
    print(']-inf;0[')
```

```
if x<0:
    print(']-inf;0[')
else:
    print('[0;+inf[')
```

```
if x<0:
    print(']-inf;0[')
elif 0<=x<=20:
    print('[0;20[')
else:
    print(']20;+inf[')
```

```
print("Saisissez deux valeurs numériques")
a=float(input("Saisir a: "))
b=float(input("Saisir b: "))
if a==b :
    print("Vous avez saisi deux fois la même valeur, à savoir {}".format(a))
else :
    print("Vous avez saisi deux valeurs différentes {} et {}".format(a,b))
```

Boucle pour

```
for i in range(7):      # pour i allant de de 0 à 6
    print(i)
for i in range(1,7):    # pour i allant de de 1 à 6
    print(i)
for i in range(1,6,2):  # pour i allant de de 1 à 6 par pas de 2 donc: 1 3 5
    print(i)
```

La syntaxe générale est *for i in range(m,n,p)* :

i prend alors toutes les valeurs de *m* à *n-1* par pas de *p*

Tant que

```
i=1
while i<=5:
    print(i)
    i=i+1 #où en plus concis i+=1
# Affichage: 1 2 3 4 5
# À la sortie de la boucle i=6
```