

## Fiche de TP numéro 1 - Découverte de l'environnement et Fonctions

Pour réaliser les travaux pratiques, vous utiliserez le logiciel Thonny.

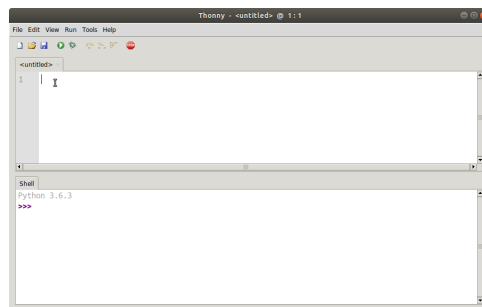
**Étape 1** - Dans votre répertoire personnel, créez un répertoire `algo1`, et un sous-répertoire `tp1`. Lancez maintenant Thonny.

**Ceux qui ont déjà programmé en Python et qui connaissent Thonny peuvent aller directement aux exercices.**

**Ceux qui ont déjà refait tous les exemples du cours sur Thonny peuvent aussi aller directement aux exercices.**

**Ceux qui ont déjà programmé en Python et qui ne connaissent pas Thonny doivent lire la partie suivante avant de démarrer les exercices.**

**Et tous les autres suivent le guide !**



Thonny est composé de deux fenêtres :

**Éditeur** : en haut. C'est là où vous saisissez vos programmes. L'éditeur vous permettra de les sauvegarder dans des fichiers ;


**Interprète** : en bas. Il exécutera soit le programme saisi dans l'éditeur, soit les expressions que vous saisissez directement dans l'interprète.

**Étape 2** - Testez les expressions Python suivantes en les saisissant dans l'interprète. Que se passe-t-il ?

`3/2`

`3//2`

`(5//2)+8`

**Étape 3** - Saisissez maintenant l'expression `(5+3)*2` dans l'éditeur. Pour exécuter ce qui est dans l'éditeur, il faut soit choisir l'option *Run / Run current script* du menu, soit cliquer sur le bouton , soit appuyer sur la touche `F5`. Thonny va alors vous demander d'enregistrer votre programme : enregistrez-le dans votre répertoire `algo1/tp1`, et nommez-le `intro.py`. Ça y est ! Que se passe-t-il ? Pourquoi ?

**Étape 4** - Vous allez transformer votre programme pour qu'il contienne maintenant une instruction :


`a = (5+3)*2`

Sauvegardez votre programme (`Ctrl+s` ou cliquez sur ). Exécutez-le. Que se passe-t-il ? Pourquoi ?



Saisissez maintenant dans le prompt l'expression :

```
>>> a
```

Que se passe-t-il ? Pourquoi ?

**Étape 5** - On va regarder en détail ce qu'il vient de se passer. Cliquez sur . Cela va réinitialiser l'environnement de votre interprète.

Puis allez dans le menu *View* et sélectionnez *Variables*. Un nouveau cadre apparaît dans Thonny, sur la droite.

**Étape 6** - Vous allez maintenant exécuter votre programme en mode débogage. Cliquez sur . Puis cliquez autant de fois que nécessaire sur  (*step into*). Assurez-vous de bien comprendre ce qui se passe à chaque clic. Observez ce qu'il se passe dans le cadre de droite.

---

## Exercices

Pour chaque fonction, vérifiez que vous avez bien utilisé des noms de variables et constantes qui suivent les recommandations données en cours. Écrivez la spécification de la fonction. N'oubliez pas de bien tester votre fonction avant de passer à la suivante.

**Exercice 1 :** Écrire et spécifier une fonction pour le calcul de l'aire d'un triangle équilatéral. La formule est :  $A = \frac{h^2 \sqrt{3}}{4}$

**Exercice 2 :** Écrire et spécifier une fonction `etoile_six_branches` qui retourne l'aire d'une étoile constituée de deux triangles équilatéraux. Vous penserez à réutiliser du travail déjà fait. . .

**Exercice 3 :** C'est l'été et vous avez décidé de repeindre votre chambre, les quatre murs et le plafond. Vous connaissez les dimensions de la pièce (largeur, longueur, hauteur), combien couvre 1 litre de peinture, et la contenance (en litres) des pots de peinture proposés. Écrire et spécifier une fonction qui retourne le nombre de pots de peinture à acheter. Proposez une décomposition en plusieurs petites fonctions.

**Exercice 4 :** Testez les expressions Python suivantes en les saisissant dans l'interprète. Que se passe-t-il ?

```
>>> a = 6
>>> type(a)
>>> b = a / 4
>>> type(b)
>>> nom = "Dupont "
>>> type(nom)
>>> cond1 = a < b
>>> type(cond1)
```

**Exercice 5 :** Spécifiez puis écrivez une fonction `annee_bissextile` qui prend le numéro d'une année et retourne `True` si cette année est bissextile, `False` sinon. Une année est bissextile si elle est un multiple de 4, sauf les siècles (1900 n'est pas bissextile), sauf tous les 400 ans (1600 est bissextile).

Exemples :

```
>>> annee_bissextile(2020)
True
>>> annee_bissextile(2021)
```

```
False
>>> annee_bissextile(2000)
True
>>> annee_bissextile(2100)
False
```

**Exercice 6 :** Spécifiez puis écrivez une fonction `nbre_jours_mois` qui prend en paramètre un numéro de mois et une année et retourne le nombre de jours de ce mois.

Exemples :

```
>>> nbre_jours_mois(2, 2020)
29
>>> nbre_jours_mois(2, 2021)
28
>>> nbre_jours_mois(7, 2021)
31
>>> nbre_jours_mois(9, 2021)
30
```

**Exercice 7 :** Spécifiez puis écrivez une fonction `numero_jour` qui prend en paramètre un numéro de jour, un numéro de mois et une année. Elle retourne le numéro du jour dans l'année.

Exemples :

```
>>> numero_jour(14, 9, 2020)
258
```

**Exercice 8 :** Spécifiez puis écrivez une fonction `nbre_jours_debut_ere` qui pour une année donnée en paramètre, retourne le nombre de jours depuis le début de l'ère chrétienne (en considérant le calendrier grégorien) jusqu'au 31/12 de l'année considérée.

Exemples :

```
>>> nbre_jours_debut_ere(1)
365
>>> nbre_jours_debut_ere(2020)
737790
```

**Exercice 9 :** Spécifiez puis écrivez une fonction `nbre_jours_debut_ere_jma` qui fait le même calcul que la fonction précédent mais pour une date donnée en jours/mois/année.

Exemples :

```
>>> nbre_jours_debut_ere_jma(5, 2, 2020)
737460
```

**Exercice 10 :** Spécifiez puis écrivez une fonction `nbre_jours_entre_deux_dates` qui retourne le nombre de jours entre deux dates.

Exemples :

```
>>> nbre_jours_entre_deux_dates(5, 2, 2020, 14, 9, 2020)
222
>>> nbre_jours_entre_deux_dates(5, 2, 2020, 14, 9, 2021)
587
>>> 587 - 222
365
```

**Exercice 11 :** Spécifiez puis écrivez une fonction `suiwant_syracuse` qui prend en paramètre un entier positif  $n$  et retourne la valeur suivante dans la suite de Syracuse. Si  $n$  est pair, la valeur suivante est la moitié de  $n$ , sinon, c'est  $3n + 1$ .

Exemples :

```
>>> suiwant_syracuse(5)
16
>>> suiwant_syracuse(2)
1
```

**Exercice 12 :** Spécifiez puis écrivez une fonction `nb_etapes_syracuse` qui prend en paramètre un entier positif  $n$  et retourne le nombre de valeurs suivantes dans la suite de Syracuse avant d'atteindre la valeur 1.

Exemples :

```
>>> nb_etapes_syracuse(5)
5
>>> nb_etapes_syracuse(11)
14
```

**Exercice 13 :** Spécifiez puis écrivez une fonction `altitude_syracuse` qui prend en paramètre un entier positif  $n$  et retourne la plus grande valeur sur le chemin de  $n$  à 1 dans la suite de Syracuse.

Exemples :

```
>>> altitude_syracuse(11)
52
>>> altitude_syracuse(5)
16
```