

## Licence 1 Informatique – Semestre 2 2022/2023

### Algorithmique, structures de données et programmation en Python

#### TD n°1 : Environnement de programmation, variables, affectation, calculs, lecture/écriture, actions conditionnelles

#### Programme de la séance :

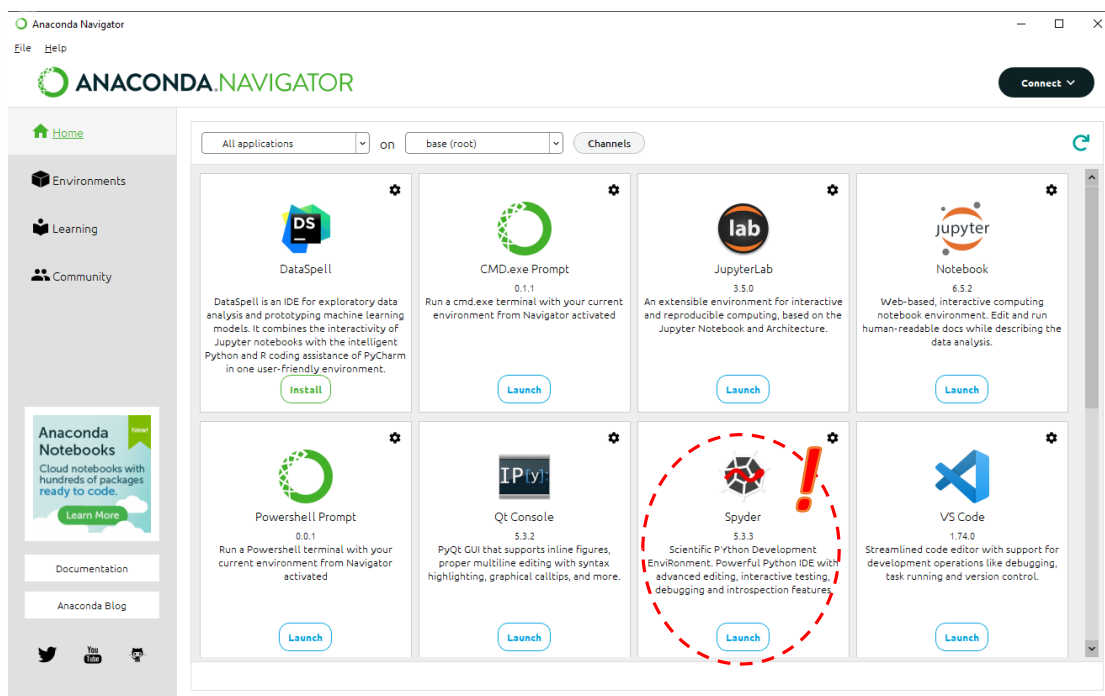
- 1°) Comment démarrer Python
- 2°) Présentation environnement de programmation
- 3°) Comment sauvegarder et récupérer son code
- 4°) Comment déposer son travail sur Moodle
- 5°) Réalisation d'un petit programme, à reproduire :
  - Saisie et affichage des contenus de variables, affectation, calcul simples
  - Affichage de messages, ...
- 6°) Exercices à faire

#### 1°) Python – Distribution Anaconda, Editeur de code Spyder

Nous utilisons la distribution Anaconda (<https://www.anaconda.com/products/distribution>) pour développer en Python. D'autres environnements sont possibles, mais il présente l'avantage d'intégrer une panoplie d'outils et de packages qui nous permettent de fonctionner sans installations additionnelles.

La distribution est déjà installée sur les ordinateurs de nos salles machines. Si vous souhaitez utiliser votre propre matériel, le plus simple est de récupérer le programme d'installation en ligne (voir la section « Anaconda Installers » sur le site web) et de lancer le setup correspondant à votre système (Windows, MacOS ou Linux). La procédure d'installation ne requiert pas de manipulations ou de configurations particulières.

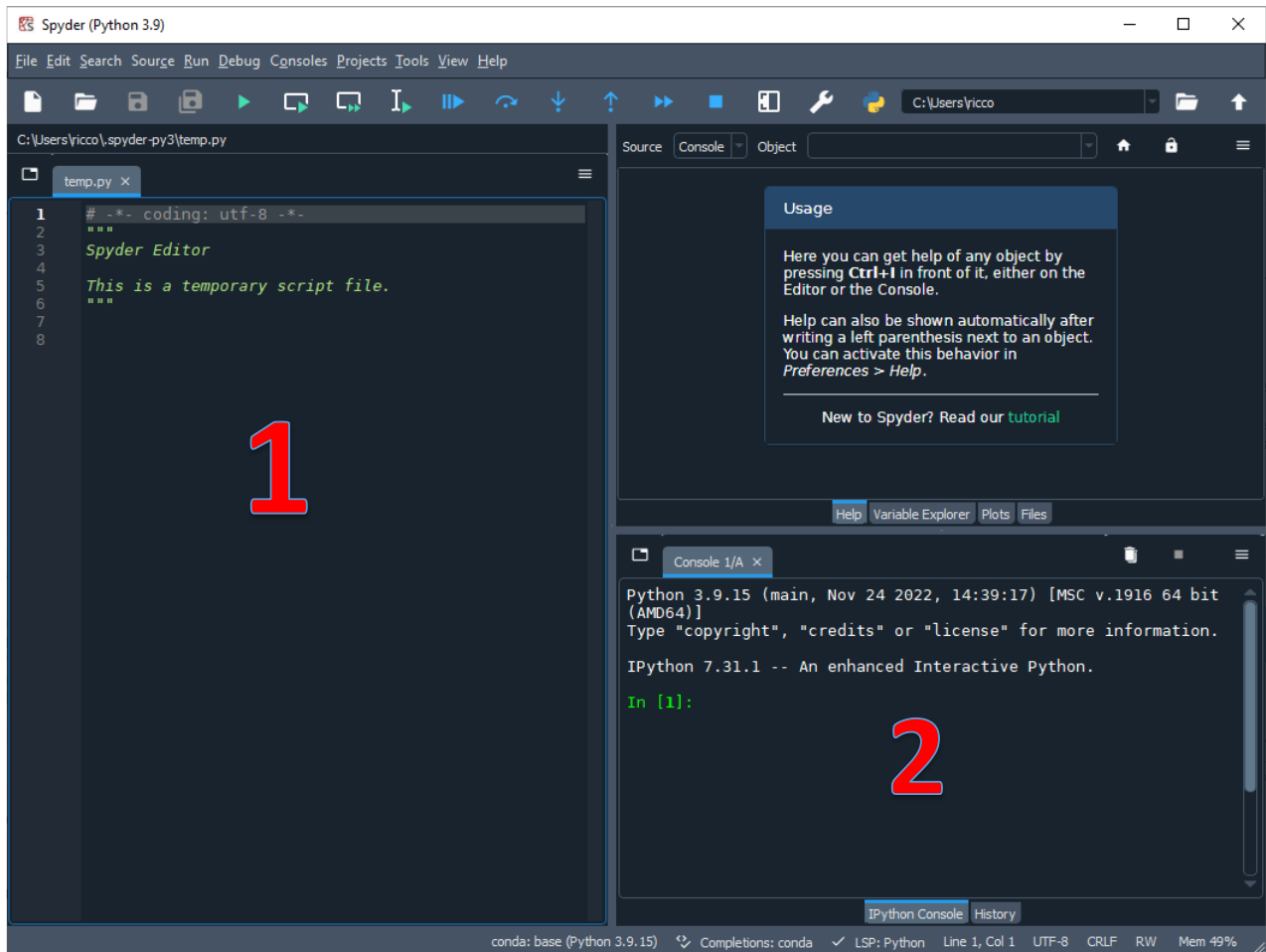
Pour démarrer l'environnement, sélectionnez « **Anaconda Navigator** » dans la liste des logiciels installés sur votre machine, une fenêtre énumérant la liste des outils disponibles apparaît.



## 2°) Environnement de développement Spyder

Nous utilisons Spyder pour rédiger nos programmes Python. Cet environnement de développement intégré (EDI) permet à la fois de coder et de lancer leur exécution dans la console sans avoir à jongler avec un terminal additionnel.

Dans « Anaconda Navigator », cliquez sur « Launch » de Spyder. L'application est démarrée, deux fenêtres nous importent particulièrement dans l'espace de travail.



- (1) L'éditeur qui vous permet de rédiger votre code. Au démarrage, un programme par défaut est automatiquement édité, il lui est attribué le nom de fichier « temp.py ». Il n'est pas enregistré sur votre ordinateur initialement.
- (2) La console qui permet d'interagir avec votre programme lorsqu'il est exécuté : pour effectuer les saisies (input), et pour visualiser les affichages (print).

## 3°) Gestion de vos programmes Python pour chaque séance

Pour une bonne organisation de votre travail, le plus simple est de créer un dossier spécifique pour chaque séance. Pour la première séance, vous pouvez créer le répertoire « Python\_Seance\_1 » sur votre bureau (ou, mieux, dans le dossier regroupant tous le travail que vous effectuerez pour Python tout au long du semestre).

Dans ce dossier, vous regrouperez les fichiers « .py » correspondant à chaque exercice demandé. J'insiste là-dessus : **un exercice = un fichier « .py »**. Par exemple, pour le premier exercice de la séance, vous le nommez « **exercice\_1.py** », etc.

A l'issue de la séance, il vous suffit d'archiver votre dossier de travail contenant les fichiers Python (« exercice\_1.py », « exercice\_2.py », etc.) et les éventuels fichiers additionnels demandés (word), et de l'archiver sur Moodle comme indiqué ci-dessous.

#### 4°) Comment déposer son travail sur Moodle

Sur la **plateforme Moodle**, dans l'espace consacré à l'UE du TD-Algorithmique structures des données et programmation et dans le devoir correspondant au TD n°1, vous déposerez le document Word contenant les algorithmes (textuel) de tous vos programmes ainsi que votre dossier compressé contenant vos programmes.

#### 5°) Un premier exemple – Exercice 1

### Pour chaque exercice vous devez rédiger d'abord l'algorithme

Nous rédigeons un premier programme qui effectue les tâches suivantes pour prendre en main notre outil de développement : (a) il doit prendre en entrée une valeur représentant un prix hors taxe saisie par l'utilisateur ; (b) il doit calculer le prix toutes taxes comprises (TTC) qui correspond en appliquant une TVA de 20% ; (c) il doit enfin afficher le prix TTC calculé. Voici le code correspondant dans l'éditeur Spyder.

The screenshot shows the Spyder Python IDE interface. On the left, the 'Source' editor displays a Python script named 'exercice\_1.py'. The script calculates the Total Tax Inclusive (TTC) price based on a user input for the price excluding taxes (pht). A large red number '1' is overlaid on the code editor. On the right, the 'Console' pane shows the execution of the script using the 'runfile' function. The output displays 'Prix HT : 100' and 'Prix TTC = 120.0'. A large red number '2' is overlaid on the console output. A large red number '3' is overlaid on the toolbar, specifically pointing to the 'Run' button (a green play icon).

```
# -*- coding: utf-8 -*-
#saisie du prix hors taxes
pht = input("Prix HT : ")
#conversion en numérique du pht
pht = float(pht)
#Remarque : on aurait pu faire directement
#pht = float(input("Prix HT : "))
#calcul du prix TTC
pttc = pht * 1.2
#affichage du résultat
print("Prix TTC = ", pttc)
```

```
In [2]: runfile('D:/_Travaux/university/
Cours_University/Supports_de_cours/Informatique/
Python/CM_TD_L1_Informatique/TD/TD1/exercice_1.py',
wdir='D:/_Travaux/university/Cours_University/
Supports_de_cours/Informatique/Python/
CM_TD_L1_Informatique/TD/TD1')
Prix HT : 100
Prix TTC = 120.0

In [3]: |
```

L'objectif maintenant pour vous est de reproduire l'exemple ci-dessus.

Rappelons les principales étapes :

- (a) Créez un dossier pour cette première séance.
- (b) Lancez l'éditeur Spyder.

- (c) Dans l'éditeur de code (1), supprimez les lignes générées automatiquement par l'outil (sauf la première ligne UTF-8), puis commencez à rédiger votre programme.
- (d) Après quelques lignes, enregistrez votre code Python (menu FILE / SAVE AS....) dans votre dossier en lui attribuant le nom « **exercice\_1.py** ».
- (e) Continuez à rédiger votre code en veillant à enregistrer à intervalles réguliers (menu FILE / SAVE ; ou encore le raccourci CTRL + S).
- (f) **Veillez à insérer un maximum de commentaires dans votre code** pour que l'on puisse suivre votre raisonnement. Une ligne de commentaire commence par le caractère « # ». Comme vous le constatez dans la copie d'écran ci-dessus, Spyder met la ligne automatiquement en grisé.
- (g) Une fois le code rédigé, vous pouvez lancer l'exécution de votre programme en cliquant sur la flèche verte (3 ; ou en actionnant le menu RUN / RUN ; ou encore en utilisant le raccourci clavier F5).
- (h) Vous interagissez alors avec la console (2) pour saisir le prix hors taxes, puis pour voir afficher le prix tous taxes comprises.

Ça y est, vous venez d'écrire et exécuter votre premier programme Python.

## 5°) Exercices

### Exercice 2 - Location (version 1)

Ecrire un programme qui doit calculer le prix de la location d'une voiture. Il fait saisir à l'utilisateur un nombre de jours (entier) et un kilométrage (réel). Il doit produire en sortie le montant de la location en s'appuyant sur le calcul suivant :

Prix location =  $0.5 * \text{kilométrage} + 60 * \text{nombre de jours}$

### Exercice 3 – Présentation

Ecrire un programme qui prend en entrée : le prénom d'une personne, son nom, son année de naissance. Il doit afficher une phrase du type « Bonjour **Nom Prénom**, vous avez **xx** ans ».

Exemple : Bonjour Kylian Mbappé, vous avez 24 ans.

### Exercice 4 – Division entière

Ecrire un programme qui prend en entrée 2 entiers A et B. Il doit afficher le résultat de division de A par B (le quotient), et le reste de cette même division.

Exemple : 13 divisé par 4 est égal à 3, et le reste est égal à 1

### Exercice 5 – Conversion horaire

Ecrire un programme qui prend en entrée une valeur représentant une durée exprimée en secondes. Il doit afficher sa correspondance en heures – minutes - secondes.

Exemple : 3800 secondes correspondent à 1 heure, 3 minutes et 20 secondes

Rappel : 1 h = 3600 secondes, 1 minute = 60 secondes

### Exercice 6 - Opérateurs booléens

Ecrire un programme qui prend en entrée : un premier prénom d'une personne (A), sa taille en cm, son poids en kg ; les mêmes informations pour une seconde personne (B). On considère que la première personne est plus corpulente que la seconde si sa taille ET (simultanément donc) son poids sont strictement supérieurs.

Votre programme doit afficher une phrase du type « 'Prénom de A' est plus corpulent que 'prénom de B' = True » (ou False selon que les conditions sont remplies ou non).