

Algorithmes et Pensée Computationnelle

Introduction à Python

1 Objectifs du TP

Ce document constitue une introduction au langage de programmation Python. Les objectifs du TP sont les suivants :

- Comprendre comment créer, éditer et lancer un script Python.
- Découvrir l'environnement de travail utilisé pour Python tout au long du semestre.
- Se familiariser avec quelques notions de base pour commencer à programmer en Python.

2 Introduction

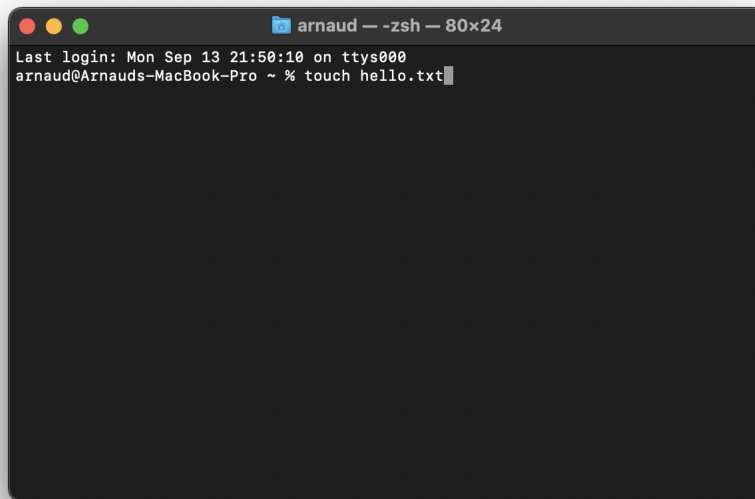
2.1 Création d'un script Python

Pour commencer à programmer en Python, il est nécessaire de créer un fichier appelé script Python. Pour se faire, il suffit de créer un simple fichier vide et de changer l'extension du fichier. Il existe plusieurs manières d'effectuer cela. La manière la plus simple est de passer par le terminal de l'OS.

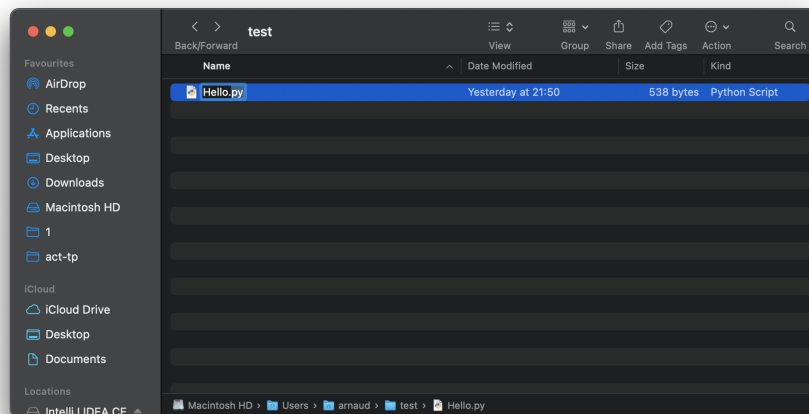
Question 1: (🕒 5 minutes)

Dans le terminal, entrer la commande `touch hello.txt` pour créer un fichier texte vide nommé `hello.txt` changer l'extension du fichier directement depuis l'explorateur de fichier de `.txt` à `.py`.

>_ Solution



```
arnaud — -zsh — 80x24
Last login: Mon Sep 13 21:50:10 on ttys000
arnaud@Arnauds-MacBook-Pro ~ % touch hello.txt
```



2.2 PyCharm

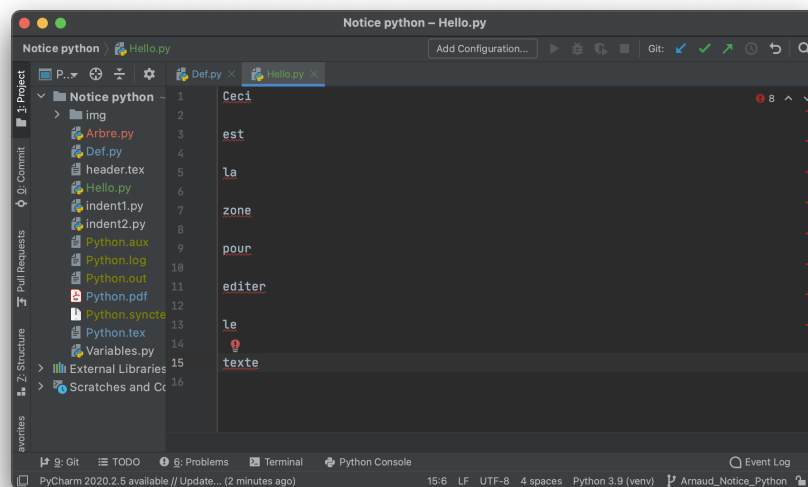
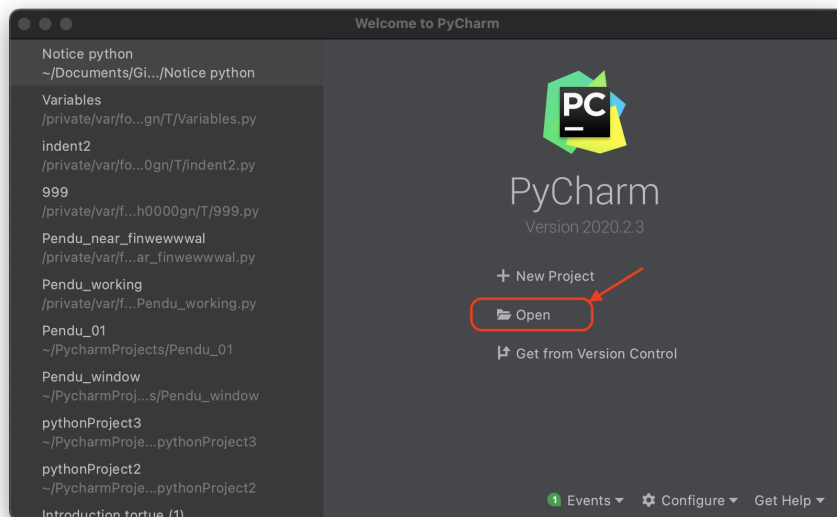
Lors du cours, il vous a été demandé d'installer PyCharm, l'IDE Python le plus populaire de ces dernières années. Un IDE (Integrated Development Environment) est un programme qui combine différents outils de développement et qui facilite grandement le travail d'un programmeur.

Question 2: (🕒 5 minutes)

Lancez le programme PyCharm et ouvrez le fichier **hello.py**.

>_ Solution

La fenêtre qui vient de s'ouvrir est similaire à un éditeur de texte basique. C'est ici que le code peut être entré et modifié. Notez qu'il est aussi possible créer directement un fichier depuis PyCharm.



2.3 Création de variable

Les variables dans les langages de programmation sont similaires à des noms données à une valeur précise. Pour assigner une valeur à une variable en Python, il suffit de respecter la forme suivante **variable=valeur**.

Conseil

- En Python, une variable est toujours dynamique (il n'existe pas de variable statique).
- On peut assigner n'importe quelle suite de caractères non-réservée en tant que variable
- Il est aussi possible d'assigner des chaînes de caractères (strings en anglais) ou encore des valeurs booléennes à des variables
- En réassignant une nouvelle valeur à une variable déjà définie, la valeur de la variable va être écrasée et remplacée par la nouvelle valeur
- Il est possible d'additionner les variables du même type
- Plusieurs variables peuvent avoir la même valeur

Question 3: (🕒 5 minutes)

Ouvrez et exécutez le script python **Variables.py** qui contient quelques exemples d'attribution de variables. Observez ce que le programme vous renvoie.

2.4 Indentation

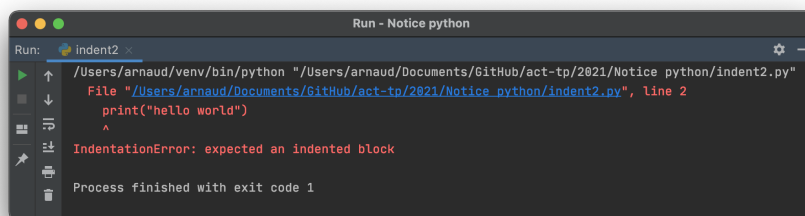
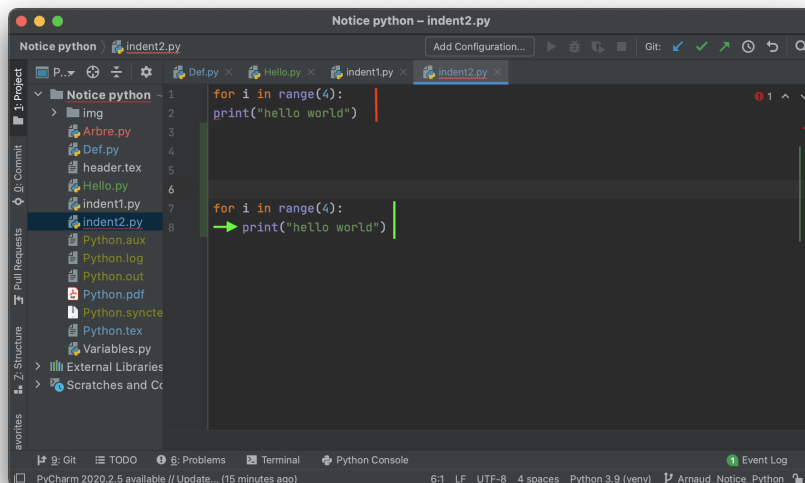
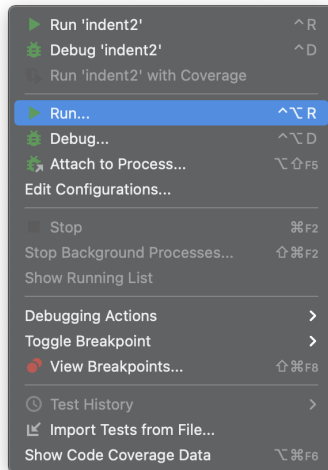
Python est un langage de programmation sensible aux erreurs d'indentation. Il sera donc important de bien comprendre comment celles-ci fonctionnent. Prenons par exemple une simple boucle **for** qui sera traitée un peu plus loin dans ce cours.

Question 4: (🕒 5 minutes)

Ouvrez et exécutez les deux scripts python **indent1.py** et **indent2.py** en allant sur le barre de menu (tout en haut) et en cliquant sur **Run > Run 'hello'..** Un des deux produit une erreur alors que l'autre fonctionne correctement. Expliquez en quoi consiste cette erreur.

>_ Solution

Dans un des scripts, la console produit une erreur d'indentation. Après une boucle `for`, il est nécessaire de bien respecter l'indentation pour définir son utilité. L'indentation est importante dans la syntaxe d'un script Python. Une erreur d'indentation peut changer le fonctionnement d'un script ou tout simplement empêcher celui-ci de s'exécuter.



2.5 Les fonctions

Dans les langages de programmation, on retrouve un très grand nombre de fonctions. Ces fonctions sont des blocs de code qui, lorsqu'ils sont invoqués avec certains paramètres, effectuent certaines actions. Une des

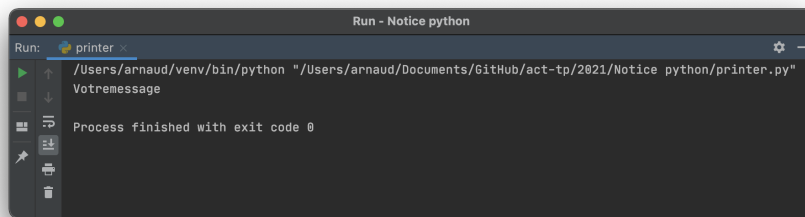
fonctions basique et plutôt importante en Python est la fonction **print()**. Cette fonction permet d'afficher à l'écran le contenu de la parenthèse.

Question 5: (🕒 5 minutes)

Entrez la ligne de code `print("Votremessage")`. Dans le bas de l'écran, dans la rubrique **Run** un message apparaît. Quel-est ce message ?

>_ Solution

Le message qui apparaît est celui que vous avez entré entre guillemets (à la place de "Votremessage"). C'est le but d'une fonction `print()`



En plus des fonctions incluses dans les librairies Python, il est possible de créer des fonctions complètement personnalisées (généralement plusieurs fonctions sont réunies en une seule) au moyen de la fonction `def nomdefonction()`: et de l'utiliser à n'importe quel moment en l'invokant avec `nomdefonction()`.

Question 6: (🕒 5 minutes)

Ouvrez et exécutez le script python `Arbre.py`. À quoi sert la fonction `def build_tree(t, branch_length, shorten_by, angle)` ?

>_ Solution

Cette fonction permet de dessiner un arbre à l'aide d'un algorithme. Nous verrons plus tard dans ce cours comment fonctionnent ces algorithmes.

