

À faire individuellement ou par petits groupes de deux ou trois.

Remarques préliminaires et configuration initiale

Ici, vous allez configurer votre machine de travail pour faire les séries d'exercices du semestre. Les consignes diffèrent selon que vous travaillez sur les machines virtuelles de l'EPFL ou sur vos propres machines. Pour l'utilisation des machines virtuelles de l'EPFL, rendez-vous **juste après**. Pour l'installation sur votre propre machine, rendez-vous **ici**.

Si vous utilisez l'infrastructure des machines virtuelles de l'EPFL

Connexion

Connectez-vous sur l'infrastructure des postes de travail virtuels:

- en salle BC07-08: identifiez-vous directement sur les machines avec votre identifiant GASPAR;
- en salles CM 1 103, CM 1 110, et CM1112: ouvrez une session sous Windows, lancez VMWare Horizon Client, ajoutez le serveur vdi.epfl.ch, puis connectez-vous à ce serveur en utilisant votre identifiant GASPAR;
- si vous souhaitez profiter de l'infrastructure offerte par l'EPFL sur votre propre machine: en téléchargeant d'abord VMWare Horizon Client depuis <https://vdi.epfl.ch>, puis en procédant comme pour les salles CM 1 103, CM 1 110, et CM 1 112;

...puis choisissez la machine virtuelle IC-C0-IN-SC-INJ-2023-Fall (et pas une autre).

Notez bien qu'à chaque logout, vos données sur l'ordinateur sont **effacées**. Seulement le contenu de votre dossier **myfiles**, visible sur votre bureau après l'ouverture de la machine virtuelle, est sauvegardé et réapparaît au prochain login. **Faites donc attention à toujours travailler dans votre dossier réseau!**

Vous pouvez aussi accéder à **myfiles** en le montant comme dossier réseau sur votre propre machine. Si vous n'êtes pas sur le réseau de l'EPFL, vous devrez vous connecter au VPN d'abord. Plus d'info: <https://mynas.epfl.ch>; <https://studinfo.epfl.ch/core/index.asp?article=18>; <https://vpn.epfl.ch>.

Configuration

Voici la procédure pour configurer le tout sur un poste de travail virtuel. **Sauf indication contraire, ceci est à faire une seule fois lors de la première séance d'exercices:**

1. Loggez-vous sur une machine virtuelle (via une des machines de la salle d'exercices ou via votre propre machine par l'intermédiaire de VMware Horizon Client comme indiqué sur <https://vdi.epfl.ch>).
2. Lancez **Firefox** depuis la barre latérale, puis allez sur la page Moodle du cours et téléchargez (dans votre dossier **Downloads**, par défaut) le fichier de configuration **setup.sh**.
3. Ouvrez l'application **Terminal** (par exemple via le lanceur en bas de la barre latérale). Puis tapez exactement les commandes suivantes en faisant bien attention à respecter les espaces:

```
cd Downloads
chmod +x setup.sh
./setup.sh
```
4. Observez la machine travailler pour vous. Cela prend un certain temps. À la fin de l'opération, s'il n'y a pas eu d'erreur, fermez la fenêtre du terminal puis lancez **Visual Studio Code** (aussi depuis le lanceur en bas à gauche). Choisissez ensuite **File** → **Open Workspace...**, puis naviguez vers votre fichier workspace, qui est ce fichier:

```
/Desktop/myfiles/ICC_Programmation/icc_prog.code-workspace.
```

5. Dans la liste des fichiers de gauche, repérez et ouvrez **test.py**. Cliquez ensuite sur le petit bouton en forme de «Play» en haut à droite. Cela prend un peu de temps la première fois, mais devrait finalement vous afficher comme output:

Welcome to Python!

Si ce n'est pas le cas, appelez l'enseignant ou un-e assistant-e.

Rappel: cette procédure est à effectuer uniquement la première fois que vous accédez à votre machine virtuelle. Les fois suivantes, vous ouvrez directement VS Code, puis faites **Open Workspace...** si nécessaire pour retrouver l'ensemble de vos fichiers de travail.

Si vous utilisez votre propre machine

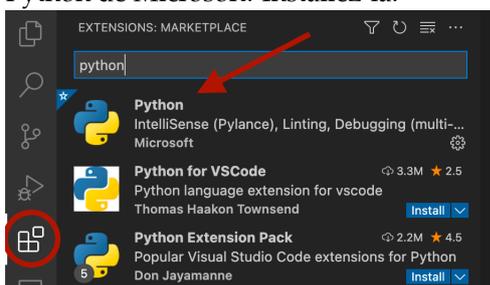
Disponibilité

Il est de votre propre responsabilité de faire en sorte que votre machine de travail soit opérationnelle, à jour, connectée au Wi-Fi, avec suffisamment de batterie, etc. En cas d'imprévu, Les machines virtuelles sont toujours une solution de secours possible, mais il est plus agréable de toujours travailler dans le même environnement de travail.

Configuration

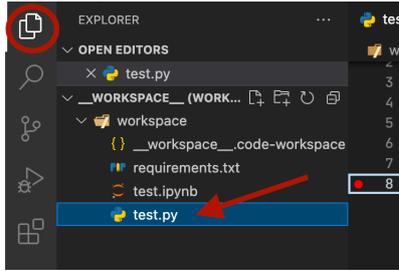
Attention, cette procédure a été testée avec plusieurs configurations d'ordinateurs, mais pas toutes celles que vous êtes susceptibles d'avoir. En cas de souci, contactez un-e assistant-e!

1. Ouvrez l'application Terminal (macOS, Linux) ou Command Prompt (Windows). Tapez-y **python3** (sans espace) suivi de la touche Retour. Si un interpréteur Python s'affiche, vous avez déjà la bonne version de Python et vous pouvez sauter l'étape suivante.
2. Téléchargez et installez Python 3.10 pour votre système d'exploitation.
macOS: <https://www.python.org/downloads/macos/>
(ou, si vous connaissez Homebrew, faites depuis le terminal: `brew install python@3.10`)
Linux: via votre package manager (demandez à un assistant-e si vous ne savez pas comment faire)
Windows: <https://www.python.org/downloads/windows/>
3. Téléchargez et installez Visual Studio Code: <https://code.visualstudio.com/download>.
4. Ouvrez Visual Studio Code et sélectionnez, dans la barre de gauche, l'icône qui affiche les extensions (voir la capture d'écran ci-après). Faites une recherche avec «python» pour trouver l'extension Python de Microsoft. Installez-la.



5. Téléchargez depuis la page Moodle du cours le dossier compressé de base dans lequel nous allons travailler. Décompressez-le où vous souhaitez le ranger. Supprimez si nécessaire le fichier zip une fois décompressé.
6. Dans Visual Studio Code, assurez-vous que l'installation de l'extension du point 4 est terminée. À partir du menu **File** et **Open Workspace**, choisissez le fichier **icc_prog.code-workspace** dans le dossier décompressé du point 5. Ouvrez ensuite le fichier **test.py** à partir de l'explorateur de

fichier (première icône de la barre de gauche).



7. En bas à droite de la fenêtre, cliquez sur **Select Python Interpreter**. Si un choix de workspaces vous est présenté, choisissez **Entire workspace**. Dans la liste suivante, choisissez l'entrée qui correspond à la version de Python 3.10 que vous venez d'installer ou qui était déjà présente sur votre machine.
8. Quittez Visual Studio Code et relancez-le. Si votre workspace n'est pas automatiquement rechargé, rouvrez-le comme à l'étape 6.
9. Si Visual Studio Code vous propose, via une alerte en bas à droite, d'installer **mypy** et/ou **pylint**, acceptez et effectuez l'installation.
10. Rouvrez le fichier **test.py**. Cliquez ensuite sur le petit bouton en forme de Play en haut à droite. Cela prend un peu de temps la première fois, mais devrait finalement vous afficher comme output **Welcome to Python!**
Si ce n'est pas le cas, appelez l'enseignant ou un-e assistant-e.

Exercice 1. Les types de données

Partie «papier-crayon»

Nous avons vu dans le cours qu'il existe plusieurs types de données. Cet exercice sera idéalement réalisé sur papier, puis vos réponses pourront être vérifiées avec Python.

Sans exécuter ou tester le programme sur ordinateur, indiquez le type des variables contenues dans le programme suivant. Indiquez «erreur» si vous pensez que le code proposé contient une erreur.

```
1 a = 1.4
2 b = 4 / 2
3 c = "1" + "2"
4 d = 1 + "2"
5 e = 1 + 2
6 f = "a" + "b"
7 g = a + b
8 h: str = 22
```

Partie sur Visual Studio Code

Vous allez maintenant vérifier vos réponses avec Python mais avant cela, il faut pouvoir créer sur Visual Studio Code un nouveau fichier et l'enregistrer dans votre **workspace**.

Au démarrage de VS Code, votre première action sera toujours d'ouvrir votre workspace, représenté par le fichier **icc_prog.code-workspace**. Vous réutiliserez le même workspace tout le semestre et conserverez ainsi votre code d'une semaine à l'autre.

Lorsque vous créez des fichiers Python pour les séries, habituez-vous à structurer votre code: par exemple, nommez vos fichiers en fonction de la série et du numéro de l'exercice (**s01e01.py**). Les fichiers Python doivent se terminer par l'extension **.py**. Une fois ce fichier créé, vous pouvez maintenant vérifier vos réponses en appelant la fonction **type()**, elle-même appelée à l'intérieur d'une fonction **print()**. Voici un exemple d'utilisation:

```
1 a = 1.4
2 print(type(a))
3 # Ceci devrait afficher: <class 'float'>, indiquant donc que la variable 'a' est de type 'float'
```

Faites de même pour toutes les autres variables. Si vous avez fait des erreurs, essayez de comprendre pourquoi. Si vous avez encore des doutes, appelez l'enseignant ou un-e assistant-e.

Exercice 2. Variables et opérations

- (a) Créez un nouveau fichier qui s'appelle **s01e02.py**. *Note: Prenez l'habitude de créer un nouveau fichier à chaque fois, même si cela n'est pas précisé dans la donnée de l'exercice.*
- (b) D'abord, stockez dans une nouvelle variable appelée **my_name** une chaîne de caractères qui indique votre prénom. Écrivez ensuite une ligne qui affiche **Hello, <votre nom>!**, où **<votre nom>** est déterminé par la variable **my_name**. Vous pouvez utiliser le modèle ci-dessous, où vous remplacez les points de suspension par du vrai code.

```
1 my_name: str = ...
2 print( ... )
```

- (c) Modifiez votre programme pour faire en sorte que la longueur de votre nom s'affiche en dessous de la ligne de salutations. Vous devez obtenir cette information en «demandant à Python» de déterminer la longueur de cette chaîne de caractères en question, et non de lui faire afficher un nombre que vous prédéfinissez. Pour ce faire, consultez le cours pour savoir comment utiliser la fonction **len()**.

Indice: si vous avez une variable qui s'appelle **my_variable** et qui contient une chaîne de caractères, l'expression **len(my_variable)** vous renvoie une valeur de type **int** qui est la longueur de la chaîne de caractères **my_variable**.

Testez ensuite votre code avec plusieurs valeurs différentes pour la variable **my_name** pour vous convaincre que votre code fonctionne dans des circonstances différentes.

Exercice 3. Manipulation de chaînes de caractères

En essayant le code, ou en faisant des recherches sur internet, trouvez ce que produisent comme valeurs les expressions **my_name[1:3]** et **my_name.upper()**. Modifiez ensuite votre programme de l'exercice 2 pour qu'il affiche votre nom:

- (a) tout en majuscules;
- (b) avec seulement la première lettre en majuscule;
- (c) avec un trait d'union inséré au milieu de la chaîne de caractères. Indiquez comment vous interprétez dans votre code «milieu de la chaîne de caractères» et vérifiez le bon fonctionnement de votre code pour des prénoms de longueurs paires et impaires.

Exercice 4. Arithmétique des nombres entiers & interpolation dans les strings

Considérez ce programme:

```
1 distance = 10
2 speed = 3
3 duration = distance // speed
4
5 print(f"Distance à parcourir: {distance} km")
6 print(f"Vitesse: {speed} km/h")
7 print(f"Durée du trajet: {duration} h")
```

- (a) Que se passe-t-il si vous enlevez le **f** qui précède les strings donnés comme arguments à **print**? Quelle est donc l'utilité de ce **f**?
- (b) Qu'est-ce que ce programme est censé calculer? Que fait-il à la place?
- (c) Insérez un breakpoint sur la première ligne et exécutez ce programme ligne par ligne. Quel endroit pose problème? Qu'en déduisez-vous sur les opérateurs de division? Corrigez la ou les lignes concernées.
- (d) Faites en sorte que ce programme affiche la durée du trajet à la minute près en affichant ceci:

Durée du trajet: 3 h 20 min.

Vous aurez probablement besoin d'effectuer un certain nombre d'opérations arithmétiques pour transformer vos valeurs en nombre d'heures et de minutes. N'hésitez pas à chercher sur Internet les fonctions qui pourraient vous aider. Testez avec d'autres valeurs initiales pour vous assurer que votre code est correct.

Exercice 5. Calendrier et introduction des concepts de la semaine prochaine

Considérez le programme suivant:

```
1 day: int = 22
2 month: str = "09"
3 year: int = 2023
```

- (a) Avez-vous remarqué que la variable **month** est de type **str**, tandis que **day** et **year** sont de type **int**. Comment expliquez-vous cela?
- (b) En utilisant les **f-string** vus dans l'exercice précédent, affichez le message suivant: "Aujourd'hui, nous sommes le 22.09.2023."
- (c) Toujours en utilisant les **f-string** et en incrémentant de sept la variable **day**, affichez le message suivant: "Le prochain cours d'ICC aura lieu le 29.09.2023".
- (d) Quel(s) problème(s) rencontrez-vous désormais s'il faut afficher le message suivant: "Le troisième cours d'ICC aura lieu le 06.10.2023".

*Teaser: Si on arrive à la fin du mois, alors **day** retourne à 1 et **month** augmente de 1. (Et encore, cela est valide que pour 11 mois sur 12).*