

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

Exercice 1. Remarques préliminaires et configuration initiale

Ces remarques sont valables pour chaque séance d'exercices. Relisez-les si nécessaire!

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

- en salle BC07-08: identifiez-vous directement sur les machines avec votre identifiant GASPARG;
- en salles CM1103 et CM1112: ouvrez une session sous Windows, lancez VMWare Horizon Client, ajoutez le serveur vdi.epfl.ch en et utilisez votre identifiant GASPARG;
- sur votre propre machine: en téléchargeant d'abord VMWare Horizon Client depuis <https://vdi.epfl.ch>, puis en procédant comme pour les salles CM1103/CM1112;

... puis choisissez la machine virtuelle IC-C0-IN-SC (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!

Plus d'info: <http://mynas.epfl.ch>; <http://studinfo.epfl.ch/core/index.asp?article=18>

Visual Studio Code

Visual Studio Code est l'application que nous utiliserons pour développer en Python. Tout ce que vous créez comme fichier dans VS Code sera enregistré dans le **workspace**, l'espace de travail — autrement dit, le dossier qui contient tous les fichiers Python que vous écrirez ainsi que les informations propre à VS Code sur votre workspace. Au démarrage de VS Code, votre première action sera toujours d'aller ouvrir votre workspace, représenté par le fichier `__workspace__.code-workspace` dans le dossier dans lequel vous travaillerez. Vous réutiliserez le même workspace tout le semestre et, ainsi, conserverez votre code d'une semaine à l'autre.

VS Code est déjà installé sur les postes de travail virtuels. Alternativement, vous pouvez installer VS Code sur votre laptop personnel. **Notez toutefois que cette solution n'est pas recommandée car vous passerez de toute façon l'examen sur l'infrastructure EPFL.** C'est plus simple de directement prendre l'habitude d'évoluer dans cet environnement.

Si vous souhaitez quand même installer VS Code sur votre machine, la bonne installation et configuration relève de votre responsabilité. Vous pouvez le télécharger depuis <https://code.visualstudio.com>. Vous aurez également besoin de télécharger et installer Python 3.7.

Voici la procédure pour configurer le tout sur un poste de travail virtuel (donc aussi l'environnement que vous utiliserez pour l'examen intermédiaire). **Ceci, sauf indication contraire, 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'exercice 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 ceci, ligne par ligne et en faisant bien attention aux 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 le fichier `/Desktop/myfiles/Programmation/icc/__workspace__.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.
6. Testez l'interpréteur: allez dans le menu Terminal, puis Run Task, et sélectionnez bpython. Après un certain temps d'initialisation, l'interpréteur démarre dans le bas de la fenêtre. Vous pouvez essayer d'y taper des lignes de codes, qui seront exécutées immédiatement et qui vous montreront le résultat de l'évaluation de chaque ligne.

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.

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`.

Exercice 2. Hello World

- (a) Créez un nouveau fichier qui s'appelle `s01e01.py`. Écrivez un programme qui affiche **Hello**, `<votre nom>!`, où `<votre nom>` est déterminé par une variable. Vous pouvez utiliser le modèle ci-dessous. Comment avez-vous créé le nouveau string qui a été affiché par la fonction `print()`?

```
1 my_name = # remplacez ce commentaire par un String qui indique votre nom
2 print( ... ) # remplacez les ... par une expression appropriée
```

- (b) 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 du string 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 `s` de type `str`, le code `len(s)` vous renvoie une valeur de type `int` qui est la longueur du string `s`. (Vous pouvez naturellement déclarer une nouvelle variable dont vous pouvez choisir le nom, qui bien sûr ne doit pas forcément être `s`.)

Testez avec plusieurs noms différents 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 `s[1:3]` et `s.upper()` si `s` est une variable de type `str`, puis modifiez 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 du string. Indiquez comment vous interprétez dans votre code «milieu du string» 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) Modifiez la ligne 5 pour afficher exactement la même valeur finale dans le **print()**, mais sans les facilités offertes par le **f**. Comment faut-il faire pour que ça marche? Quelle version du code préférez-vous?
- (c) Qu'est-ce que ce programme est censé calculer? Que fait-il à la place?
- (d) Insérez des **print()** entre les calculs pour vérifier les valeurs intermédiaires. Quel endroit pose problème? Corrigez la ou les lignes concernées.
- (e) 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 fort probablement besoin d'arrondir, de soustraire et de multiplier des valeurs. Si nécessaire, cherchez sur internet les fonctions qui peuvent vous aider. Testez avec d'autres valeurs initiales pour vous assurer que votre code est correct.