Avant toute chose, organisez votre ordinateur en choisissant un emplacement dans lequel vous stockerez vos fichiers liés à Python. Créez un dossier permettant de contenir tous ces documents, nommé par exemple "Programmes Python". Créez dans ce dossier un dossier nommé "TP1".

Exercice I.1 *(Hello World !)*
Lancez le logiciel Pycharm. Créez un nouveau projet dont l'emplacement sera votre dossier TP1. Créez ensuite dans ce projet un nouveau fichier Python nommé `hello.py` et contenant le code correspondant. Observez aussi la coloration automatique ainsi que la complétion automatique et les informations qui s'affichent automatiquement.

1. Le fichier ainsi créé se situe bien dans le dossier TP1 créé précédemment. Vérifiez-le à l'aide du gestionnaire de fichiers.
2. Vous pouvez l'exécuter (commande Run). Trouvez le raccourci permettant d'exécuter un programme.
3. Ajoutez de nouvelles déclarations de variables et de nouveaux affichages pour vérifier que vous avez compris la syntaxe.

Exercice I.2 *(Structure if)*
Créez un nouveau fichier Python nommé `conditions.py` et contenant le code correspondant.

1. Exécutez ce programme.
2. Modifiez le programme en prenant age=15 et en enlevant le "else". Exécutez à nouveau le programme, que se passe-t-il ?
3. Modifiez le programme pour ajouter un troisième cas testant si age est supérieur ou égal à 17, et afficher dans ce cas le message "Mineur, mais bientôt majeur". Exécutez à nouveau le programme en prenant différentes valeurs pour age (15, 17, 19 par exemple).

Exercice I.3 *(Boucles for)*
Créez un fichier nommé `boucles.py` et contenant le code correspondant.

1. Exécutez ce programme.
2. Modifiez le programme pour afficher les puissances de 2 de 0 à 10.
3. Modifiez votre programme pour qu'il affiche des messages comme :
   `2 puissance 3 vaut 8`
   `2 puissance 4 vaut 16`
4. Que vaut 2^{10} ?
5. Écrivez une boucle for permettant de calculer la somme des entiers de 1 à 10.

Exercice I.4 *(En-tête de fichier)*
En utilisant des commentaires, on peut ajouter un en-tête au fichier afin d'indiquer les informations suivantes :

— nom du fichier
— date de dernière modification
— nom du propriétaire du fichier
— description

Par exemple, pour le fichier `hello.py`, on pourrait écrire ce qui est indiqué ci-contre. Vous introduirez désormais **systématiquement** un tel en-tête dans les fichiers que vous écrirez.

```
##########################
#  EN TETE STANDARD      #
#  Fichier hello.py      #
#  Cree le 09/09/16      #
#  Pierre-Alain Reynier  #
#  Hello World           #
##########################
```

Exercice I.5 *(Fonction range)*
Créez un fichier nommé `testrange.py` et contenant le code ci-contre. N'oubliez pas l'en-tête !

Testez les différentes boucles proposées. Cette fonction prend donc entre 1 et 3 arguments :

— l'argument obligatoire correspond à la condition d'arrêt
— le premier argument (facultatif) correspond à la valeur initiale
— le troisième argument (facultatif) correspond au pas (mise à jour de la variable qui réalise la boucle)

Exercice I.6 *(Messages d'erreurs)*
Reprenez le programme `testrange.py`. Effectuez l'une après l'autre chacune des modifications suivantes, observez ce qu'il se passe (message d'erreur retourné) et expliquez.
— enlever le : après le range
— enlever l'indentation devant un print
— enlever les guillemets autour d'un message d'un print

Exercice I.7 *(Modules)*
Créez un nouveau fichier contenant le programme `module.py`.

1. Utilisez le programme.
2. Mettez en commentaire la commande permettant d'importer le module, et testez à nouveau le programme.
3. Modifiez la commande d'import en utilisant : `from math import *`
   Testez le nouveau programme.

Exercice I.8 *(input)*
Créez un nouveau fichier contenant le programme `saisie.py`.

1. Utilisez le programme, essayez différentes saisies.
2. Ajoutez maintenant une partie demandant de rentrer un nombre réel, et convertissez la saisie en une variable de type float. Affichez le résultat.
3. Essayez de ne pas saisir un âge sous la forme d'un entier pour observer l'erreur qui se produit. Que se passe-t-il avec un âge négatif ?
   On peut en réalité tester si ce qui a été saisi par l'utilisateur est bien un entier, pour simplifier, nous supposerons la plupart du temps que l'utilisateur respecte les consignes.

Exercice I.9 *(Somme)*
Créez un nouveau fichier contenant le programme `somme.py`.

1. Exécutez le programme.
2. Modifiez le programme pour ne pas utilier de boucle. Testez votre nouveau programme.