

TP 5 : Fonctions et tableaux

Préliminaires.

En suivant les mêmes instructions que les semaines précédentes, soumettez vos feuilles de travail pour la semaine 4, après les avoir récupérées du serveur JupyterHub si nécessaire. De même, téléchargez les feuilles de travail pour la semaine 5.

Exercice 1 (Premiers programmes compilés en C++, 20 minutes).

- (1) Ouvrir l'éditeur de texte `gedit`. Créez un nouveau fichier texte `bonjour.cpp` que vous enregistrerez dans le dossier `Info111/Semaine5`.
- (2) Écrire dans le fichier le programme suivant. **Attention**, le programme que vous écrivez doit respecter la même mise en page (retour à la ligne, espaces, majuscules).

```
#include <iostream>
using namespace std;

int main() {
    cout << "Bonjour !" << endl;
    return 1;
}
```

- (3) Ouvrir un terminal et utilisez la commande `cd` pour vous rendre dans le répertoire `Info111/Semaine5` où est enregistré votre fichier.
- (4) Vérifier avec `ls` que votre répertoire contient bien le fichier `bonjour.cpp`.
- (5) **Compiler le programme** `bonjour.cpp` en tapant la commande suivante sur votre terminal :

```
g++ bonjour.cpp -o bonjour
```

Si tout se passe bien, vous n'avez aucun message d'erreur.

Attention, cette commande ne peut fonctionner que si vous êtes dans le bon répertoire !

Si une erreur s'affiche, la lire et essayer d'identifier le problème en trouvant le numéro de ligne : avez-vous bien respecté le nom du fichier, les minuscules et majuscules, les retours à la ligne et espaces, les points-virgules à la fin des deux lignes d'instructions ? Après chaque correction, **enregistrer** le fichier et relancer la **compilation** en retapant la commande. **Rappel** : on peut parcourir l'historique des commandes tapées dans le terminal avec les touches \uparrow et \downarrow du clavier.

- (6) Lancer la commande `ls` ; quels fichiers se trouvent maintenant dans votre répertoire ?

```
g++ : La commande g++ fichierX -o fichierY lance le compilateur C++ gcc
sur fichierX et crée un fichier exécutable fichierY.
```

- (7) Votre répertoire doit contenir un fichier `bonjour` (sans extension). L'exécuter en tapant

```
./bonjour
```

Le texte "Bonjour !" doit s'afficher. Bravo, vous avez lancé votre premier programme C++ !

- (8) Modifier le fichier `bonjour.cpp` en remplaçant "Bonjour !" par le message de votre choix. L'enregistrer puis lancer la commande

```
./bonjour
```

Que remarquez-vous ?

- (9) Comment faire pour que l'exécution du programme reflète le changement effectué dans `bonjour.cpp` ? Le faire.

- (10) Ouvrir avec `gedit` le fichier `minmax.cpp`.

- (11) Compiler et exécuter ce programme comme précédemment :

```
g++ minmax.cpp -o minmax
```

```
./minmax
```

- (12) Modifier le programme pour qu'il affiche le *minimum* des deux nombres. Compiler et exécuter à nouveau.

Exercice 2 (Cœur du TP : les tableaux).

Comme les semaines précédentes, nous allons travailler dans l'application web **Jupyter**. Jupyter Notebook peut être lancé soit en local avec la commande suivante :

```
info-111 jupyter notebook
```

Les fichiers modifiés ne seront qu'accessibles sur votre session à l'université. **Laisser le terminal ouvert pendant tout le TP** (cela permet à votre travail sur les feuilles Jupyter d'être sauvegardé automatiquement).

Nous vous rappelons que l'application peut aussi être lancée sur le serveur **JupyterHub** en vous connectant à l'adresse suivante : <https://jupyterhub.lal.in2p3.fr/>

Dans ce cas, les fichiers modifiés seront accessibles chez vous en vous connectant à la même adresse, et si vous avez préalablement téléchargé vos fichiers locaux contenus dans votre répertoire **Info111/Semaine5** sur le serveur (en utilisant le bouton **Upload/Téléversement**).

Naviguer jusqu'à votre répertoire **Info111/Semaine5**.

Pour être plus efficaces en TP, apprendre à utiliser les fonctionnalités suivantes :

- Menu Cell -> Create Cell Above
- Menu Cell -> Create Cell Below
- Menu Cell -> Run All Above
- ...

Travailler successivement sur les feuilles :

- (1) `feuille1-tableaux-prise-en-main.ipynb`
- (2) `feuille2-tableaux-fonctions.ipynb`
- (3) `feuille3-fibonacci.ipynb`
- (4) `feuille4-avance.ipynb`

Exercice ♣ 3 (Python, Euler!).

Pour des explications, voir les deux derniers exercices de la feuille de TP 2.