


# TP 5 VARI 1

Informations techniques PC Suse :

- (a) Pour démarrer une session : utilisateur **licencep** et mot de passe **7002n\***. Vous trouverez :
  - un gestionnaire de fichiers en haut à gauche placé dans le dossier personnel HOME
  - une icône lézard  en haut à droite pour accéder au menu.
- (b) Pour démarrer *Processing* : clic sur l'icône lézard en haut à droite → Développement → Processing.
- (c) Pour démarrer une *console* : clic sur l'icône lézard en haut à droite → Terminal → Konsole.
- (d) Pour ouvrir un gestionnaire/navigateur de fichiers : clic sur l'icône lézard → Utilitaires → Dolphin.
- (e) Pour lancer une commande : Alt + Space
- (f) Pour modifier un fichier, clic droit sur le fichier → Ouvrir avec → Kate (ou autre éditeur de votre choix).

## 1 Commandes Linux et Réseaux

**Démarrer une console/terminal** en suivant les instructions au point (c) ci-dessus.

**Exercice 1** Taper la commande suivante dans la console. Elle télécharge le fichier `html` associé à la page web utilisée pour NFP135 (VARI1).

```
wget cedric.cnam.fr/~porumbed/vari1/
```

Par la suite taper `cat index.html` pour visualiser ce fichier `html`. Observer que les balises (en : *tags*) `html` ont la forme `<tr>` ou `<td>` ou `<html>`, etc.

**Rappel :** `man xyz` affiche le manuel d'une commande `xyz` (mais remplacer `xyz` par votre vraie commande). Taper `/abc` et *Entrée* pour chercher `abc` dans le manuel. Pour quitter le manuel, taper "q".

**Exercice 2** Taper `man wget` pour trouver le manuel de la commande `wget`.

- Quel est l'objectif de `wget` ?
- Quel est l'objectif de l'option `--dns-timeout` ?

## 2 Processing

**Exercice 1** Écrire une fonction `doubler(int)` qui renvoie le double d'un nombre entier. Remplir le code ci-après pour le faire fonctionner.

```
... doubler (int x){
    ....
}
void setup(){
    int salaire = 1000;
    int nouveauSalaire = doubler(salaire);
    println("J'ai doublé le salaire et mon nouveau salaire est "+nouveauSalaire);
}
```

**Exercice 2** Écrire une fonction qui renvoie le cube d'un float. Ce programme devrait afficher 1030.301.

```
... cube (...) {
    ....
}
void setup() {
    println(cube(10.1));
}
```

**Exercice 3** Écrire une fonction `ajouterIntérêts(float capital, float taux)` qui permet d'ajouter au montant `capital` les intérêts gagnés au taux `taux` et renvoie le nouveau montant. Par exemple, `ajouterIntérêts(1000,0.05)` devrait renvoyer 1050. Remplir le programme ci-après.

```
... ajouterIntérêts (...) {
    ....
}
void setup() {
    println(ajouterIntérêts(1000,0.05));
}
```

**Note** : si le capital est déposé sur 2 ans et les intérêts sont capitalisés, on peut calculer le nouveau capital via :

```
ajouterIntérêts(ajouterIntérêts(1000,0.05),0.05).
```

**Exercice 6** Le programme ci-après permet de tracer des lignes à des positions aléatoires. **Rappel** : La fonction `draw()` est appelée de manière répétitive en continu. L'appel de fonction `random(600)` renvoie un float aléatoire entre 0 et 600. Modifier le programme pour le faire afficher des ellipses de couleurs différentes (utiliser `fill(random(255),...,...)`). Les tailles des ellipses doivent être inférieures à 300.

```
void setup() {
    size(600,600);
}
void draw() {
    //tracer une ligne du pixel x au pixel y,
    line(random(600),random(600), //pixel x = coordonnées aléatoires <600
        random(600),random(600)); //pixel y = coordonnées aléatoires <600
}
```

**Exercice 7** Écrire une méthode (c.-à.-d. une fonction qui renvoie `void`) pour afficher un nom de 4 lettres. L'argument de la méthode est un tableau avec 4 cases de type `char` (lettre). Corriger une erreur dans le programme ci-après. Remplir le reste du code pour le faire afficher "Toto". **Note** : Utiliser la fonction `print(...)` pour afficher un caractère sans sauter une ligne.

```
.... afficherNomDe4Lettres(char [] nomDe4Lettres) {
    ....
}
void setup() {
    char [] nom4Lettres = new char [4]; //tableau de 4 lettres
    nom4Lettres[0] = 'T';
    afficherNomDe4Lettres(nomDe4Lettres); //corriger une erreur ici
}
```

**Exercice 4** Soit le programme ci-après. Dire ce qu'il afficher sans le faire tourner. Dans une deuxième étape, taper le code et vérifier votre réponse.

```
int detMax(int a, int b){
    if(a>b)
        return a;
    return b;
}
void setup() {
    int a=9, b=8;
    println(detMax(b,5));
    println(detMax(5,a));
}
```

**Exercice 5** Écrire une fonction `detMaxTab(int [])` qui appelle 2 fois la fonction `detMax(...)` ci-dessus (la copier dans le nouveau programme) pour trouver le maximum d'un tableau de 3 cases. Utiliser cette fonction et une fonction `detMinTab(int [])` pour écrire une fonction `notesValides(int [])` qui renvoie `true` si les notes d'un tableau de 3 cases sont comprises entre 1 et 20 et `false` sinon. Commencer avec :

```
boolean notesValides(int [] notes) {
    if((detMinTab(notes)>0)&& ...)
        return true;
    ... ..
```