

TP 11 VARI 1 Révisions

1 Commandes terminal

Démarrer une console/terminal en utilisant une des méthodes indiquée sur le site ci-dessous, comme au TP précédent. Le mieux est peut-être d'utiliser la machine virtuelle Antix.

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

Exercice 1 Taper la commande suivante dans un terminal

```
wget cedric.cnam.fr/~porumbed/1giga
```

Elle télécharge un fichier d'1GB et affiche le temps utilisé pour cela. **Calculer la vitesse (débit) du réseau** $\frac{1000\text{MB}}{\text{temps (sec)}}$, ex., si cela prend 9 secondes on a $\frac{1000}{9}=111\text{MBs}$. Vous pouvez faire le calcul avec une calculette.

Exercice 2 Exécuter la commande suivante pour faire une copie du fichier. Remarquer la vitesse du disque dur!

```
time cp 1giga 1gigacopie
```

Exercice 3 Taper la commande ci-après pour télécharger le fichier source Toto2.java.

```
wget cedric.cnam.fr/~porumbed/Toto2.java
```

Consulter le contenu du fichier Toto2.java à l'aide de la commande `cat`, ou avec un éditeur comme `geany`. Pour **compiler** le programme Java, taper :

```
javac Toto2.java
```

Et pour l'exécuter :

```
java Toto2
```

Exercice 4 Modifier le programme pour le faire afficher correctement l'indice de masse corporelle (IMC). L'IMC est le poids divisé par la taille *en mètres*.

2 Processing : fonctions et boucles

Exercice 0 Soit le code à droite. Avant de l'exécuter, essayer de comprendre chaque instruction; n'hésitez pas à consulter les programmes sur les animations 3D du dernier cours. Réaliser les tâches suivantes :

1. Diminuer la taille de la boîte dans l'appel `box(...)`
2. Transformer la sphère (avant-dernière ligne) en une boîte de taille similaire.
3. Augmenter la vitesse de rotation sur l'axe Z pour la faire égale à celle sur l'axe Y
4. Ajouter une rotation sur l'axe X.
5. Ajouter une troisième boîte, en utilisant encore un appel à `translate(...)` et `box(...)`.

```
float angle = 0; //variable globale
void setup() {
  size(600,600,P3D);
  noStroke();
  fill(50,50,200);
}
void draw() {
  background(200,200,200);
  lights();
  angle = angle - 2*PI/300;
  translate(300,300);
  rotateY(angle);
  rotateZ(angle/10);
  box(120);
  translate(120,30,0);
  sphere(20);
}
```

Exercice 1 Écrire une fonction `boolean revenuImposable(float revenuFiscalRef)` qui renvoie `true` si le revenu fiscal de référence `revenuFiscalRef` est supérieur à 9710 euros ou `false` sinon. Il faut donc renvoyer une valeur de type `boolean` qui indique si le revenu est imposable ou pas. Remplir le code ci-après pour le faire fonctionner. Lancer le programme plusieurs fois pour tester plusieurs valeurs aléatoires de revenu, ligne 4.

```
1 boolean revenuImposable (... ..)
```

```

2    ...
3 void setup() {
4     float revenuNet = random(90000); //le revenu fiscal de référence
5     boolean imposable;
6     imposable = revenuImposable(revenuNet);
7     if(imposable){
8         println(" Avec_un_revenu_de_ "+revenuNet+" _vous_payez_des_impôts");
9     }else{
10        println(" Avec_un_revenu_de_ "+revenuNet+" _vous_ne_payez_pas_d'impôt");
11    }
12 }

```

Exercice 2

Écrire un fonction `float calculerImpots(float revenuFiscalRef)` qui renvoie le montant des impôts à payer pour un revenu fiscal de référence `revenuFiscalRef`. Il s'agit d'un impôt progressif, défini sur 2 tranches :

- Pas d'impôt pour la tranche 0 – 9710 euros.
- 14% pour tout gain au delà de 9710 euros.

Par exemple, pour un revenu de 10000 euros, l'impôt vaut $0.14 \times (10000 - 9710)$.

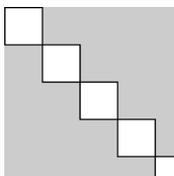
Faire fonctionner le code ci-après.

```

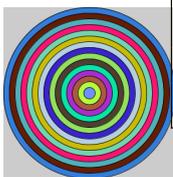
.. calculerImpots(.. revenuFiscalRef){
    ...
}
void setup() {
    float revenuNet = random(90000);
    float impots =
        calculerImpots(revenuNet);
    println(" Votre_revenu=" +revenuNet);
    println(" Vos_impôts=" +impots);
}

```

Exercice 3 Tracer 20 rectangles de taille 30×30 le long de la diagonale d'une toile de taille 600×600 . La figure à droite montre le début de la diagonale, il faut faire 20 rectangles.



Exercice 4 Tracer 15 cercles de même centre et de rayons 20, 40, 60, ..., 300. Chaque cercle aura une couleur aléatoire. Vous aller avoir besoin de commencer avec $i = 15$ et finir avec $i = 1$; utiliser $i--$ au lieu de $i++$.



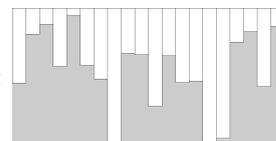
Exercice 5 Écrire une méthode `tracerBarre(int colonne, float hauteur)` qui permet de tracer un rectangle de largeur 50 et hauteur `hauteur` à la position $(50 \times \text{colonne}, 0)$. Plus précisément, le coin en haut à gauche est situé aux coordonnées $(50 \times \text{colonne}, 0)$ et la taille du rectangle est $50 \times \text{hauteur}$. Faire fonctionner le code suivant.

```

void setup() {
    size(1500,500);
    for(int i=0;i<30;i++)
        tracerBarre(i,random(500));
}

```

Résultats final attendu :



Exercice 6 Écrire une méthode `histogramme(float[] vals)` pour tracer un histogramme associé aux valeurs `vals`. On considère que `vals` est un tableau de 30 éléments `float`. La méthode devrait faire 30 fois appel à `tracerBarre(...)`. Faire fonctionner le code ci-après.

```

void setup() {
    size(1500,500);
    float [] valeurs = new float [30];
    for(int i=0;i<30;i++)
        valeurs[i] = i*10; //ou random
    histogramme(valeurs);
}

```

Affichage attendu :

