


# TP 4 VARI 1

Web : [cedric.cnam.fr/~porumbed/vari1/](http://cedric.cnam.fr/~porumbed/vari1/)

Informations techniques PC Suse :

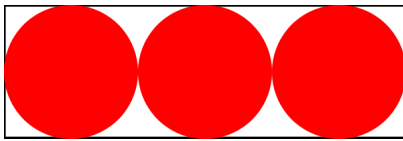
(a) Pour démarrer une session : utilisateur **licencep** et mot de passe **7002n\*\***. Remarquez

1. une tête de caméléon  en haut à droite pour accéder au menu.

(b) Pour démarrer *Processing* : clic sur la tête de coméléon en haut à droite → Développement → Processing.

(c) La page VARI1 : [cedric.cnam.fr/~porumbed/vari1/](http://cedric.cnam.fr/~porumbed/vari1/)

**Exercice 1** Réaliser un dessin qui ressemble à l'image ci-dessous.



**Exercice 2** Calculer et afficher l'aire de la surface blanche, à l'intérieur du rectangle et à l'extérieur du cercles. Rappels : l'aire du cercle est  $\pi \cdot \text{rayon}^2$  et l'aire d'un rectangle est le produit *largeur* · *hauteur*.

**Exercice 3** Soit le programme suivant. Déterminer ce qu'il affiche **sans le faire tourner**. Ensuite, taper le programme dans le logiciel **processing** et vérifier s'il affiche ce que vous avez prévu.

```
1 int a = 7;
2 int b = 9;
3 a = a + b;
4 b = a - b;
5 a = a - b;
6 println("a="+a);
7 println("b="+b);
```

**Exercice 4** Exécuter le code ci-après et essayer de comprendre chaque instruction. Remarquer que **Utopia** est une fonte de caractères. L'instruction **textFont(...)** indique la fonte et la police utilisé par la prochaine instruction **text(...)**. Modifier le code pour le faire afficher le texte «**Salut**» (en gras) sur tout l'écran. Vous allez avoir besoin d'appeler **size(1200,900)** pour avoir une très grande toile.

```
background(255,255,255); //le fond
fill(0,0,0); //couleur d'écriture
textFont(createFont("Utopia",18));
text("Toto_est_riche", 10, 30);
textFont(createFont("Utopia_Bold",18));
text("Toto_est_riche", 10, 50);
```

**Exercice 5** Réaliser le logo ci-après. Pour le texte, vous allez avoir besoin des instructions présentées à l'exercice précédent, car il faut écrire une partie du texte (**croix-rouge**) en gras. Pour la croix, il suffit de tracer deux rectangles rouges sans contour.



**Exercice 6** Soit le code ci-après qui permet de dessiner **une belle étoile**. Essayez de comprendre chaque nouvelle instruction :

- **beginShape()** indique qu'on commence à placer les sommets d'une figure géométrique ;
- les appels **vertex(...)** indiquent ces sommets ;
- **endShape()** termine la figure.

Modifier ce code pour le faire afficher une étoile bleue d'une taille **dix fois plus grande** !

```
int x=50,y=50;
beginShape(); //début figure
vertex(x + 0 , y -50);
vertex(x + 14, y -20);
vertex(x + 47, y -15);
vertex(x + 23, y + 7);
vertex(x + 29, y + 40);
vertex(x + 0 , y + 25);
vertex(x - 29, y + 40);
vertex(x - 23, y + 7);
vertex(x - 47, y -15);
vertex(x - 14, y -20);
endShape(); //fin figure
```

**Exercice 7** Réaliser le dessin ci-après. Utilisez toutes les connaissances que vous avez acquises. Tous les coups

sont permis, n'oubliez pas le dessin similaire du TP précédent.



**Exercice 8** Exécuter le code à droite. Vous allez voir que le triangle ne touche pas le point (0,0) comme on pourrait le penser à cause de la ligne 13. Cela vient de l'appel `translate(50,50)` à la ligne 4. Enlevez le commentaire à la ligne 7 et vous allez voir le triangle tourner. Modifier le code pour faire tourner le triangle en sens inverse.

```
1 float angle = 0;
2 void draw() {
3     background(200,200,200);
4     translate(50,50);
5     angle = angle + PI/100;
6
7     //rotate(angle);
8     noStroke();
9     fill(255,0,0);
10    text("toto",10,40);
11    beginShape();
12    vertex(0,0);
13    vertex(50,25);
14    vertex(50,50);
15    endShape();
16 }
```

**Exercice 9** Soit le code ci-après. Déterminer ce qu'il fait **sans le faire tourner**. Modifier le programme pour le faire tracer des pentagones bleus au lieu de triangles rouges. Les coordonnées de tous les sommets doivent être aléatoires.

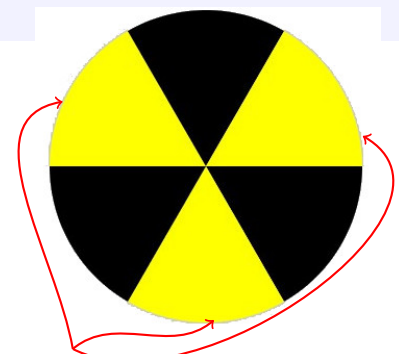
```
size(700,700);
noStroke();
fill(255,0,0);
for(int i=0;i<1000;i=i+10){ //100 valeurs entre 0 et mille
    //parce que on avance de 10 en 10
    beginShape();
    vertex(350,250);
    vertex(random(700),random(700));
    vertex(random(700),random(700));
    endShape();
}
```

**Exercice 10** Utiliser un appel à la fonction

`ellipse(centreX,centreY,largeur,hauteur)`

pour tracer un cercle rempli de couleur jaune. Appeler la fonction `arc` (voir ci-dessous) trois fois pour réaliser le dessin à droite.

`arc(centreX,centreY,largeur,hauteur,angleStart, angleEnd)`



couleur jaune

**Exercice 11** Vous avez vu à l'exercice 8 qu'on a tout tracé dans un bloc de code nommé `draw()`. On dit qu'on utilise une méthode `draw()`. Ajouter une méthode nommée `setup()` dans laquelle vous ne faites qu'appeler `size()`. Inspirez vous des exercices 8 et 10 pour faire tourner le dessin de l'exercice 10. Vous pouvez faire appel à `rotate(angle)` comme à la ligne 7 de l'exercice 8. Par contre, cette méthode fait tourner le dessin autour du pixel (0,0). Supposons que le centre de la toile est le pixel (350,350). Pour tourner le dessin autour du pixel (350,350), vous avez besoin (dans le bon ordre) de :

- `translate(350,350)`
- `rotate(...)`
- `translate(-350,-350)`