

Après avoir vu comment créer des dessins, nous allons voir comment créer des animations avec Processing.

## 1. Le principe de l'animation avec Processing

Par défaut, Processing démarre le programme, exécute le dessin, et s'arrête à la fin des instructions du sketch (programme processing), en bas de la fenêtre d'édition.

Pour créer une animation, il faut prolonger l'exécution du dessin afin que celui-ci puisse se modifier à l'aide d'étapes, selon différents facteurs et conditions. C'est le rôle de la fonction `draw()` qui est une boucle infinie. La fonction `setup()` est appelée une seule fois au début de l'exécution du programme, elle sert souvent à initialiser les différents paramètres.

**Ces 2 fonctions ne prennent aucun paramètre et ne retournent aucune valeur.**

☞ Exécuter le programme ci-dessous :

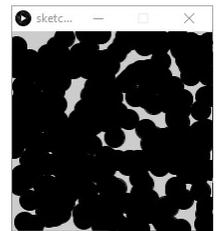
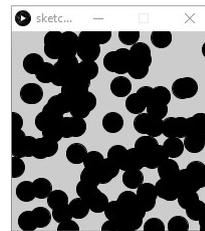
```
def draw():
    background(random(255))
```

À l'exécution du programme, l'ordinateur affiche plusieurs fois par seconde la fenêtre à l'écran. Normalement, la fenêtre clignote 60 fois par seconde (FPS : Frames per second) avec une couleur de gris aléatoire entre le noir et le blanc.

☞ Exécuter le programme ci-dessous :

```
def setup():
    size(200, 200)
    fill(0)

def draw():
    ellipse(random(200), random(200), 20, 20)
```



Cette animation finie par remplir l'écran.

☞ Exécuter le programme ci-dessous :

```
def setup():
    size(200, 200)
    fill(0)

def draw():
    background(255)
    ellipse(random(200), random(200), 20, 20)
```

Maintenant à chaque nouvelle exécution de la boucle `draw()`, un nouveau fond blanc est dessiné suivi par le dessin de l'ellipse.

Pour réduire le nombre d'affichage par secondes (60 par défaut), il est possible alors de le définir à l'aide de la fonction `frameRate()`.

☞ Exécuter le programme ci-dessous :

```
def setup():
    size(200, 200)
    fill(0)
    frameRate(5)

def draw():
    background(255)
    ellipse(random(200), random(200), 20, 20)
```

## 2. Balle dans la fenêtre

☞ Exécuter le programme ci-dessous :

`y = 10`

```
def setup():
    size(200, 200)
    fill(0)
    background(255, 255, 204)

def draw():
    global y #variable globale
    affiche_balle(y)
    y = depl_balle(y)

def affiche_balle(pos_y):
    """affiche la balle"""
    ellipse(100, pos_y, 20, 20)

def depl_balle(depl_y):
    """deplace la balle"""
    depl_y += 1
    return depl_y
```

☞ Modifier le programme afin de ne plus voir le tracé de la balle.

☞ Compléter la fonction `depl_balle()` afin de faire rebondir la balle (en bas et en haut de la fenêtre).

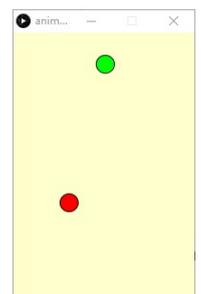
☞ Modifier la vitesse de déplacement de la balle.

☞ Modifier le programme afin que la balle se déplace en diagonale. La position de départ, ainsi que la vitesse, seront aléatoires.

## 3. Deux balles dans la fenêtre

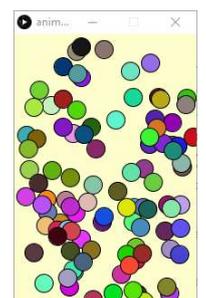
☞ En s'inspirant du programme précédent, réaliser un programme qui doit faire apparaître deux balles (une rouge et une verte) qui se déplace (position de départ et vitesse de départ aléatoire pour les deux balles).

En cas de collision entre les deux balles, ces deux balles doivent disparaître.



## 4. Cent balles dans la fenêtre

☞ Modifier le programme précédent afin d'afficher cent balles dans qui se déplace dans la fenêtre sans gérer les collisions.



Sources : [https://pixees.fr/informatiquelycee/python\\_proc\\_a5.html](https://pixees.fr/informatiquelycee/python_proc_a5.html)