

# Boucles avancées et animations

Valeur d'accueil et de reconversion en informatique (VARI1)

Daniel Porumbel (dp.cnam@gmail.com)

<http://cedric.cnam.fr/~porumbed/var1/>

## Déplacer la fenêtre

```
void setup() {  
    size(100, 100);  
    // displayWidth = largeur écran  
    for (int x=0; x<displayWidth-100; x++) {  
        surface.setLocation(x, 400);  
        delay(5);  
    }  
}
```

- `surface` fait référence à la toile/surface/fenêtre de dessin
- `delay` permet de faire une pause 5 milli-secondes

- 1 Comment faire la fenêtre re-partir vers la gauche ?
- 2 Comment répéter à l'infini ? Boucle `while(true)` ?
- 3 Tout faire avec une boucle `while(true)` ?

## Déplacer la fenêtre

```
void setup() {  
    size(100, 100);  
    // displayWidth = largeur écran  
    for (int x=0; x<displayWidth-100; x++) {  
        surface.setLocation(x, 400);  
        delay(5);  
    }  
}
```

- `surface` fait référence à la toile/surface/fenêtre de dessin
- `delay` permet de faire une pause 5 milli-secondes

- 1 Comment faire la fenêtre re-partir vers la gauche ?
- 2 Comment répéter à l'infini ? Boucle `while(true)` ?
- 3 Tout faire avec une boucle `while(true)` ?

## Déplacer la fenêtre

```
void setup() {  
    size(100, 100);  
    // displayWidth = largeur écran  
    for (int x=0; x<displayWidth-100; x++){  
        surface.setLocation(x, 400);  
        delay(5);  
    }  
}
```

- `surface` fait référence à la toile/surface/fenêtre de dessin
- `delay` permet de faire une pause 5 milli-secondes

- 1 Comment faire la fenêtre re-partir vers la gauche ?
- 2 Comment répéter à l'infini ? Boucle `while(true)` ?
- 3 Tout faire avec une boucle `while(true)` ?

## Déplacer la fenêtre

```
void setup() {  
    size(100, 100);  
    // displayWidth = largeur écran  
    for (int x=0; x<displayWidth-100; x++){  
        surface.setLocation(x, 400);  
        delay(5);  
    }  
}
```

- `surface` fait référence à la toile/surface/fenêtre de dessin
- `delay` permet de faire une pause 5 milli-secondes

- 1 Comment faire la fenêtre re-partir vers la gauche ?
- 2 Comment répéter à l'infini ? Boucle `while(true)` ?
- 3 Tout faire avec une boucle `while(true)` ?

```
void seDeplacer(float x, float y){
    surface.setLocation((int)x,(int)y);
}
void setup(){
    float rayon=200;
    float centreX = 400; float centreY = 300;
    float angle;
    for (angle=0; angle<10*PI; angle=angle+0.1){
        seDeplacer(centreX+rayon*cos(angle),
                   centreY+rayon*sin(angle));
        delay(50);
    }
}
```

- 1 Quel est le résultat du code ?
- 2 Comment faire tourner la fenêtre en sens inverse ?
  - Comment l'arrêter après avoir fait un demi-cercle ?
- 3 Comment la faire revenir à la position de départ ?

```
void seDeplacer(float x, float y){
    surface.setLocation((int)x,(int)y);
}
void setup(){
    float rayon=200;
    float centreX = 400; float centreY = 300;
    float angle;
    for (angle=0; angle<10*PI; angle=angle+0.1){
        seDeplacer(centreX+rayon*cos(angle),
                   centreY+rayon*sin(angle));
        delay(50);
    }
}
```

- 1 Quel est le résultat du code ?
- 2 Comment faire tourner la fenêtre en sens inverse ?
  - Comment l'arrêter après avoir fait un demi-cercle ?
- 3 Comment la faire revenir à la position de départ ?

```
void seDeplacer(float x, float y){
    surface.setLocation((int)x,(int)y);
}
void setup(){
    float rayon=200;
    float centreX = 400; float centreY = 300;
    float angle;
    for (angle=0; angle<10*PI; angle=angle+0.1){
        seDeplacer(centreX+rayon*cos(angle),
                   centreY+rayon*sin(angle));
        delay(50);
    }
}
```

- 1 Quel est le résultat du code ?
- 2 Comment faire tourner la fenêtre en sens inverse ?
  - Comment l'arrêter après avoir fait un demi-cercle ?
- 3 Comment la faire revenir à la position de départ ?



- `set (x, y, color (r, g, b) )` : colorer le pixel au coordonnées  $(x, y)$  avec la couleur générée pas le code RGB indiqué par  $(r, g, b)$
- Quel est le résultat du code suivant ?

```
void setup () {  
  size(900,400);  
  for (int x=0;x<900;x++)  
    for (int y=0;y<400;y++)  
      set (x, y, color (random(255) ,  
                        random(255) ,  
                        random(255)) ) ;  
}
```

**1** Faire le dessin de manière répétitive via `draw ()` !

# Quel est le résultat du code ?

```
void setup () {  
  size(600,600);  
  textSize(30);  
  fill(255,0,0); //rouge  
  for(int i=1; i<=9; i++)  
    for(int j=1; j<=9; j++){  
      text(i*j, 60*j, 60*i+30);  
    }  
}
```

- 1 Mettre chaque nombre dans un carré
- 2 Comment afficher une grille ?
- 3 Comment afficher la table de multiplication dans la console ?

# Quel est le résultat du code ?

```
void setup () {  
  size(600,600);  
  textSize(30);  
  fill(255,0,0); //rouge  
  for (int i=1; i<=9; i++)  
    for (int j=1; j<=9; j++) {  
      text(i*j, 60*j, 60*i+30);  
    }  
}
```

- 1 Mettre chaque nombre dans un carré
- 2 Comment afficher une grille ?
- 3 Comment afficher la table de multiplication dans la console ?

# Quel est le résultat du code ?

```
void setup () {  
  size(600,600);  
  textSize(30);  
  fill(255,0,0); //rouge  
  for(int i=1; i<=9; i++)  
    for(int j=1; j<=9; j++){  
      text(i*j, 60*j, 60*i+30);  
    }  
}
```

- 1 Mettre chaque nombre dans un carré
- 2 Comment afficher une grille ?
- 3 Comment afficher la table de multiplication dans la console ?

# Quel est le résultat du code ?

```
void setup() {  
  size(600,600);  
  textSize(30);  
  fill(255,0,0); //rouge  
  for(int i=1; i<=9; i++)  
    for(int j=1; j<=9; j++){  
      text(i*j, 60*j, 60*i+30);  
    }  
}
```

- 1 Mettre chaque nombre dans un carré
- 2 Comment afficher une grille ?
- 3 Comment afficher la table de multiplication dans la console ?

# Quel est le résultat du code ?

```
void setup() {  
  size(400,200);  
  PImage monImage = loadImage("pommes.png");  
                      //une image 200 X 200  
  image(monImage, 0, 0);  
  color c = get(40,40);  
}
```

- 1 L'appel `get(40,40)` récupère la couleur du pixel (40,40).  
Copier l'image à droite !
- 2 Faire une opération "Flip image" !
- 3 Faire un symétrie par rapport à la diagonale

# Quel est le résultat du code ?

```
void setup() {  
  size(400,200);  
  PImage monImage = loadImage("pommes.png");  
                      //une image 200 X 200  
  image(monImage, 0, 0);  
  color c = get(40,40);  
}
```

- 1 L'appel `get(40,40)` récupère la couleur du pixel (40,40). Copier l'image à droit !
- 2 Faire une opération "Flip image" !
- 3 Faire un symétrie par rapport à la diagonale

# Quel est le résultat du code ?

```
void setup() {  
  size(400,200);  
  PImage monImage = loadImage("pommes.png");  
                      //une image 200 X 200  
  image(monImage, 0, 0);  
  color c = get(40,40);  
}
```

- 1 L'appel `get(40,40)` récupère la couleur du pixel (40,40). Copier l'image à droit !
- 2 Faire une opération "Flip image" !
- 3 Faire un symétrie par rapport à la diagonale



# Quel est le résultat du code ?

```
void setup() {  
  size(400,200);  
  PImage monImage = loadImage("pommes.png");  
                      //une image 200 X 200  
  image(monImage, 0, 0);  
  color c = get(40,40);  
}
```

- 1 L'appel `get(40,40)` récupère la couleur du pixel (40,40). Copier l'image à droit !
- 2 Faire une opération "Flip image" !
- 3 Faire un symétrie par rapport à la diagonale

# Quel est le résultat du code ?

```
void setup() {  
    size(400,200);  
    PImage monImage = loadImage("pommes.png");  
                        //une image 200 X 200  
    image(monImage, 0, 0);  
    color c = get(40,40);  
}
```

- Le code à droite permet de récupérer la couleur RGB d'un pixel.
  - Utile pour le projet.
- Faire une pomme bleu, changer le rouge en bleu

```
color c = get(100,100);  
float rouge = red(c);  
float vert = green(c);  
float bleu = blue(c);  
println(rouge);  
println(vert);  
println(bleu);
```

# Quel est le résultat du code ?

```
void setup () {  
    size(800,600);  
    noFill();  
}  
int x = 0;  
int y = 400;  
void draw () {  
    background(200,200,200);  
    x = x + 5;  
    stroke(255,0,0);  
    ellipse(x,y,50,50);  
    delay(40);  
}
```

- 1 Comment faire la balle rebondir lorsqu'elle touche le bord à droite ?
- 2 Même question pour le bord à gauche.
- 3 Même question pour tous les bords.
- 4 Utiliser une balle de foot !

# Quel est le résultat du code ?

```
void setup () {  
    size(800,600);  
    noFill();  
}  
int x = 0;  
int y = 400;  
void draw () {  
    background(200,200,200);  
    x = x + 5;  
    stroke(255,0,0);  
    ellipse(x,y,50,50);  
    delay(40);  
}
```

- 1 Comment faire la balle rebondir lorsqu'elle touche le bord à droite ?
- 2 Même question pour le bord à gauche.
- 3 Même question pour tous les bords.
- 4 Utiliser une balle de foot !

# Quel est le résultat du code ?

```
void setup () {  
    size (800,600);  
    noFill ();  
}  
int x = 0;  
int y = 400;  
void draw () {  
    background (200,200,200);  
    x = x + 5;  
    stroke (255,0,0);  
    ellipse (x,y,50,50);  
    delay (40);  
}
```

- 1 Comment faire la balle rebondir lorsqu'elle touche le bord à droite ?
- 2 Même question pour le bord à gauche.
- 3 Même question pour tous les bords.
- 4 Utiliser une balle de foot !

# Quel est le résultat du code ?

```
void setup () {  
    size (800,600);  
    noFill ();  
}  
int x = 0;  
int y = 400;  
void draw () {  
    background (200,200,200);  
    x = x + 5;  
    stroke (255,0,0);  
    ellipse (x,y,50,50);  
    delay (40);  
}
```

- 1 Comment faire la balle rebondir lorsqu'elle touche le bord à droite ?
- 2 Même question pour le bord à gauche.
- 3 Même question pour tous les bords.
- 4 Utiliser une balle de foot !