

# PROCESSING



## Comment dessiner des formes simples en codant ?



Le principe du logiciel est de dessiner en codant afin de pouvoir créer des animations graphiques, mais aussi des interfaces Hommes Machines pour piloter des objets. Commençons par les bases en découvrant les fonctions qui permettent de dessiner des formes géométriques élémentaires (carré, rectangle, cercle...) avec quelques attributs, la couleur, leur position....



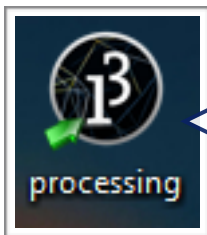
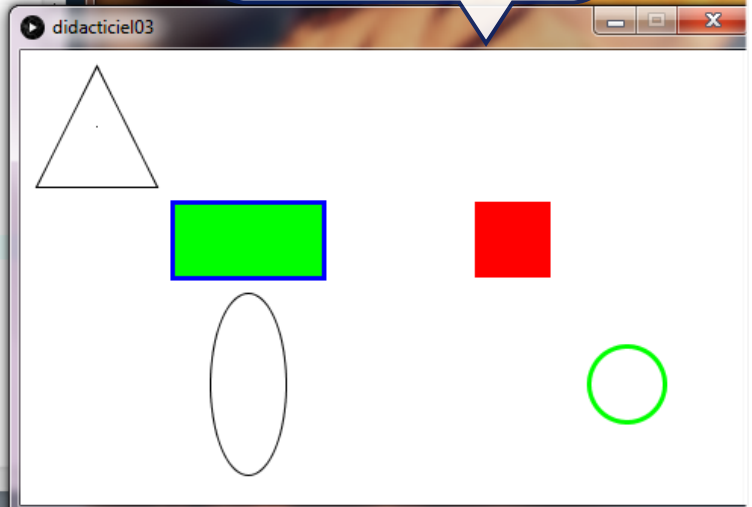
<https://processing.org/download/>

Interface de programmation

```
didacticiel03 | Processing 3.2.2
Fichier Modifier Sketch Dépendances Outils Aide

didacticiel03
6 //dessin d'un rectangle,d'un carré
7 stroke(0,0,255);
8 strokeWeight(3);
9 fill(0,255,0);
10 rect(100, 100, 100, 50);
11 noFill();
12 noStroke();
13 fill(255,0,0);
14 rect(300, 100, 50, 50);
15 //dessin d'une ellipse, d'un cercle
16 stroke(0,255,0);
17 noFill();
18 ellipse (400,220,50,50);
19 stroke(0);
20 strokeWeight(1);
21 noFill();
22 ellipse (150,220,50,120);
23 //dessin d'un triangle
24 stroke(0,0,0);
25 strokeWeight(1);
```

Dessin généré dans une nouvelle fenêtre



1

Lancer Processing



Chaque valeur donnée dans les instructions qui suivent est un exemple. Dans vos projets adaptez la méthode pour obtenir la meilleure solution correspondant au besoin.

# L'ESPACE DE DESSIN

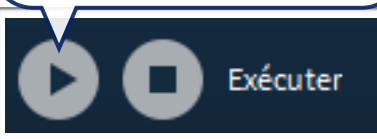
2

Définir les dimensions de la zone de dessin en utilisant la fonction `size` :

```
size(480,300);
```

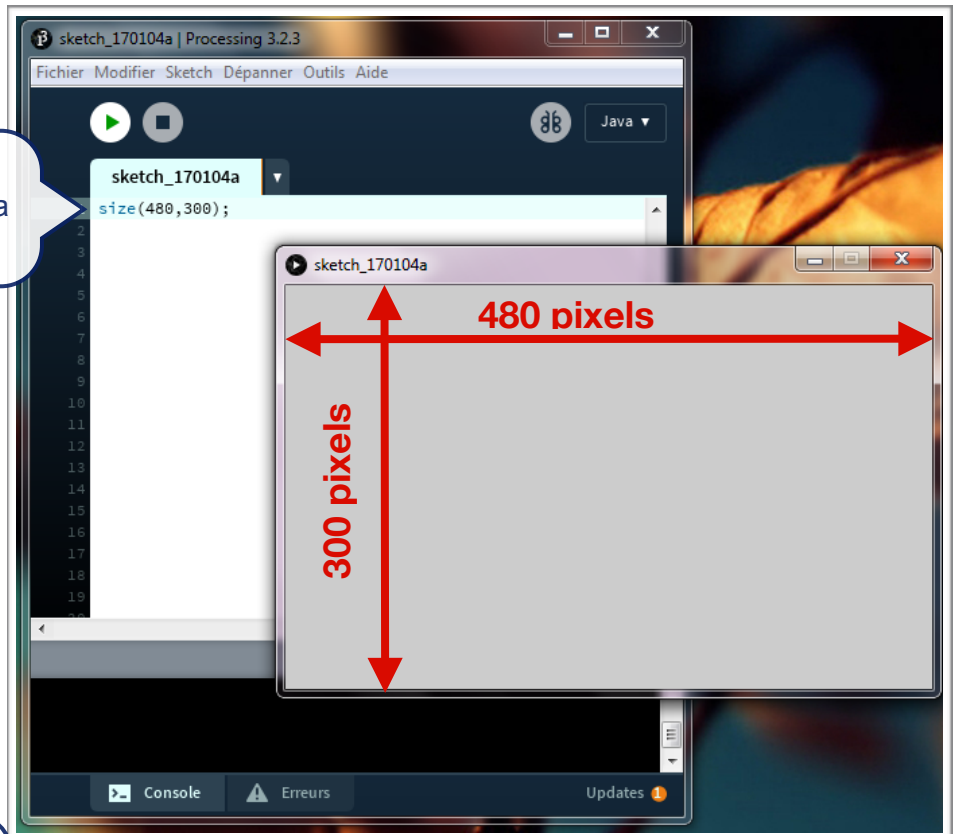
3

Executer pour tester



4

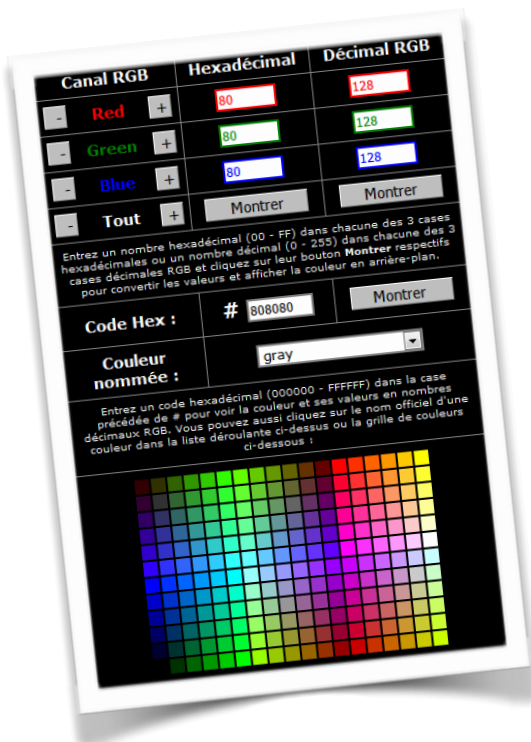
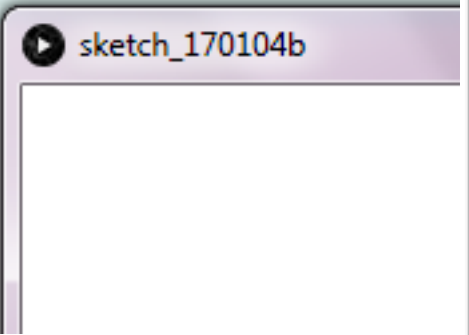
Arrêter pour revenir au mode d'édition du programme



5

par défaut la couleur d'arrière plan est grise pour la modifier il faut ajouter une fonction `background` avant celle de `size`

```
background(255,255,255);  
size(480,300);
```



la fonction `background` attend trois valeurs correspondant au codage RGB pour obtenir une couleur.

**background(R,G,B);**

<http://www.proftnj.com/RGB3.htm>



On utilise un repère orthonormé dans lequel l'unité est le pixel, dans notre cas :

- $X_o=0, Y_o=0$  pour le point O
- $X_a=480, Y_o=0$  pour le point A
- $X_b=0, Y_o=300$  pour le point B

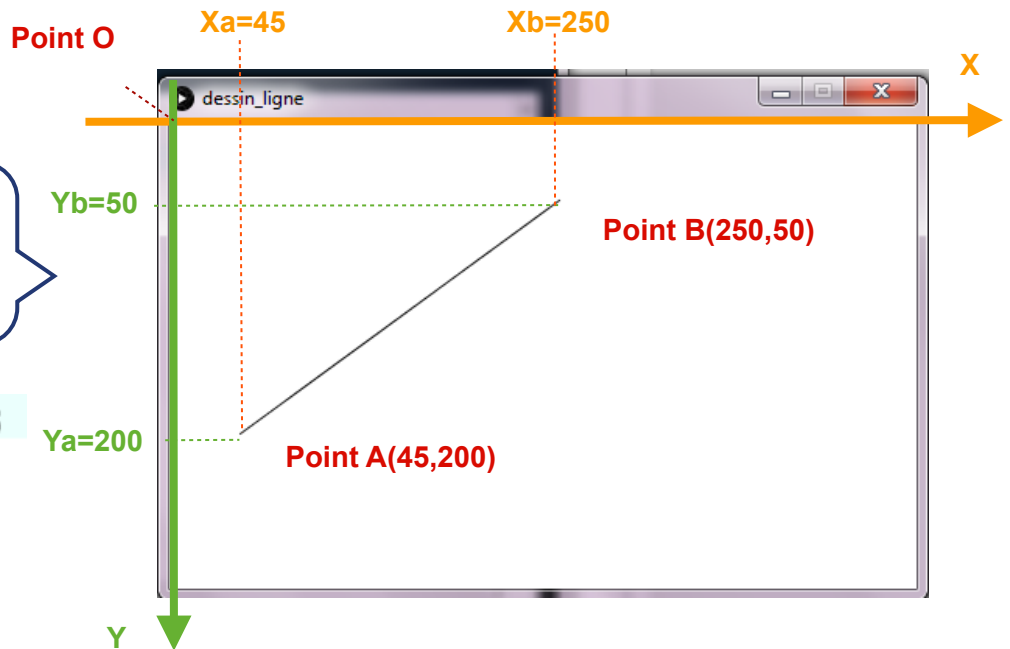


## DESSIN D'UNE LIGNE

6

Utiliser la fonction `line(Xa,Ya,Xb,Yb);`

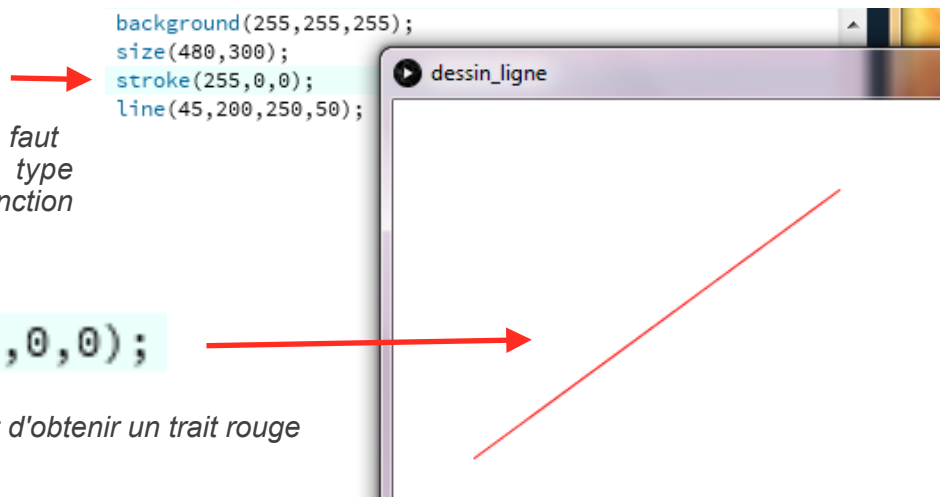
```
line(45,200,250,50);
```



Pour colorier la ligne, il faut placer une fonction de type `stroke(R,G,B);` avant la fonction de dessin de la ligne

```
stroke(255,0,0);
```

Permet d'obtenir un trait rouge



# DESSIN D'UN RECTANGLE, D'UN CARRE

8

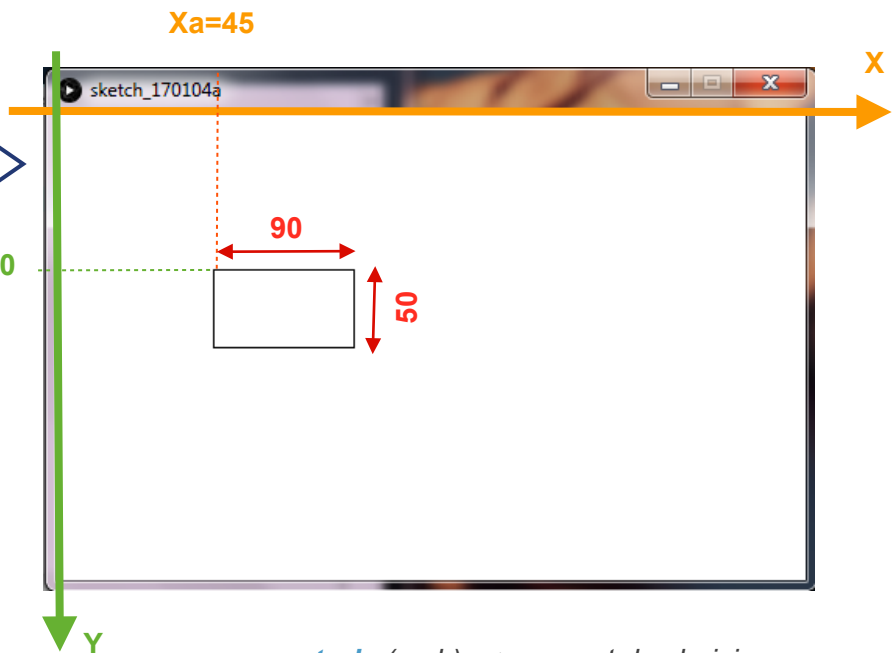
Utiliser la fonction  
`rect(Xa,Ya,longueur,Hauteur);`

```
rect(100, 100, 90, 50);
```

9

Executer pour tester

```
stroke(0,0,255);  
strokeWeight(3);  
fill(0,255,0);  
rect(100, 100, 90, 50);
```



*stroke(r,g,b); --> permet de choisir une couleur pour le contour.*

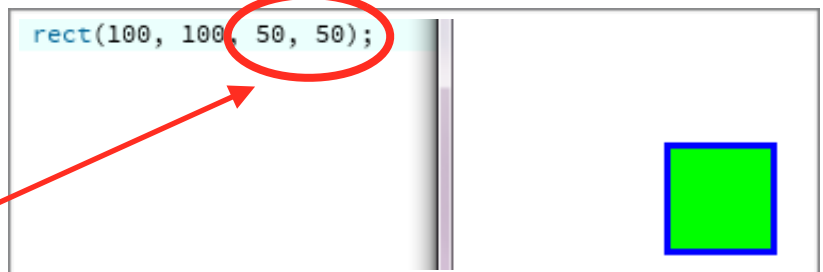
*strokeWeight(x); --> définit l'épaisseur du trait.*

*fill(r,g,b); --> remplit d'une couleur la forme*



i

*Pour obtenir un carré évidemment la valeur donnée à la longueur doit être égale à celle donnée pour la hauteur*

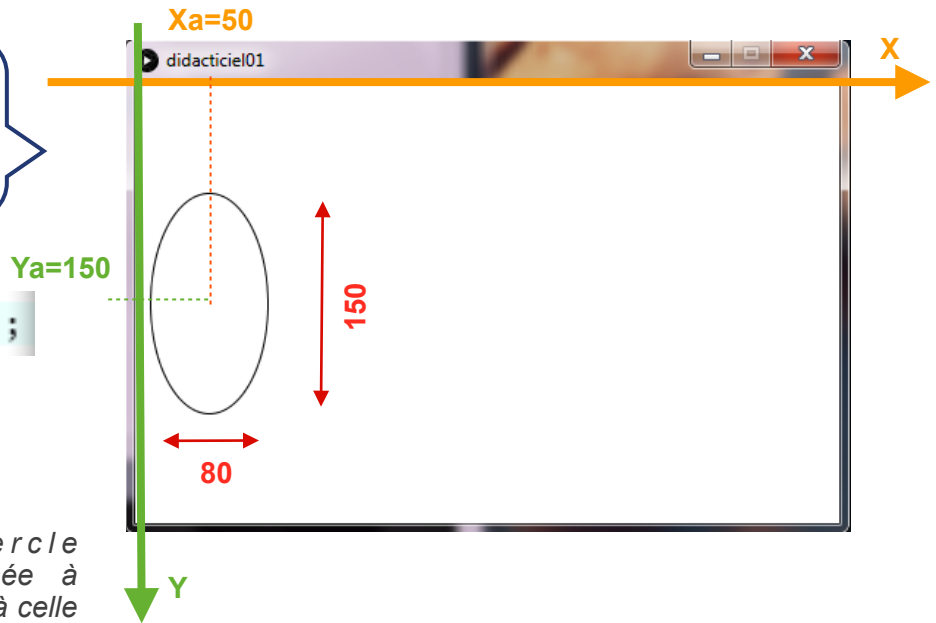


## DESSIN D'UNE ELLIPSE, D'UN CERCLE

10

Utiliser la fonction `ellipse(Xa,Ya,longueur,Hauteur);`

```
ellipse(50, 150, 80, 150);
```



**i**

Pour obtenir un cercle évidemment la valeur donnée à la longueur doit être égale à celle donnée pour la hauteur

```
noStroke();  
fill(255,0,0);  
ellipse(50, 150, 80, 80);
```

`noStroke();` --> supprime le contour



## DESSIN D'UN TRIANGLE

11

Utiliser la fonction `triangle(Xa,Ya,Xb,Yb,Xc,Yc);`

```
triangle(120, 200, 180, 60, 240, 200);
```

