

## 1. Référence

Le guide de référence sur Processing en Python est accessible à l'adresse suivante : <https://py.processing.org/>.

Toutes les instructions utilisables avec le logiciel Processing en Python et leurs détails sont décrites à l'adresse : <https://py.processing.org/reference/>.

Une référence existe en français, **pour le langage Java** : <https://fr.flossmanuals.net/processing/introduction/>.

## 2. L'espace de dessin

Par défaut, l'espace de dessin (fenêtre de visualisation) est une fenêtre de dimensions 100 pixels par 100 pixels. L'instruction `size()` permet de dimensionner cet espace. Il prend deux arguments `size(largeur, hauteur)`.

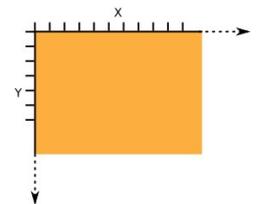
✂ Exécuter le programme ci-dessous :

```
size(200, 500)
print(width)
print(height)
```

La taille de l'espace de dessin utilisé peut-être connu grâce aux mots clés **width** et **height**.

### 2.1. Coordonnées dans l'espace

Quand on travaille en 2 dimensions (2D), on utilise deux axes de coordonnées x et y correspondant respectivement à la largeur (axe horizontal) et à la hauteur (axe vertical) d'une situation. Par convention de la mesure de l'espace, le coin en haut à gauche correspond aux valeurs  $x=0$  et  $y=0$ . Les valeurs x sont croissantes vers la droite et les valeurs y sont croissantes vers le bas.



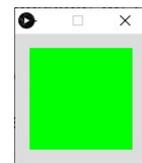
## 3. Formes et couleurs

### 3.1. Couleur de fond

La fonction `background()` permet de modifier la couleur de fond de la fenêtre.

✂ Tester la ligne ci-dessous :

```
background(0, 255, 0)
```

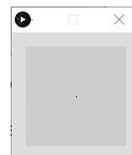


Cette fonction prend paramètres : la composante rouge, la composante verte et la composante bleue.

### 3.2. Le point

✂ Tester la ligne ci-dessous :

```
point(50, 50)
```

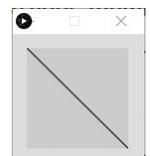


### 3.3. La ligne

La fonction `line(x1, y1, x2, y2)` permet de dessiner une ligne de point de départ  $x_1, y_1$  et d'arrivée  $x_2, y_2$ .

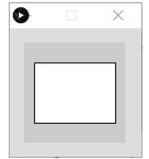
✂ Tester la ligne ci-dessous :

```
line(0, 0, 100, 100)
```



## 3.4. Le rectangle

Un rectangle se dessine par quatre valeurs en faisant l'appel de `rect(x, y, largeur, hauteur)`. La première paire de valeurs `x` et `y`, par défaut (mode **CORNER**) correspond au coin supérieur gauche du rectangle. La deuxième paire de valeurs définit la dimension du rectangle.

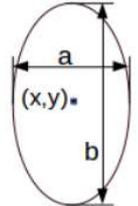


✍ Tester la ligne ci-dessous :  
`rect(10, 20, 80, 60)`

## 3.5. L'ellipse

La fonction `ellipse(x, y, a, b)` prend quatre paramètres.

- `x` et `y` les coordonnées du centre, `a` la largeur horizontale et `b` la largeur verticale.

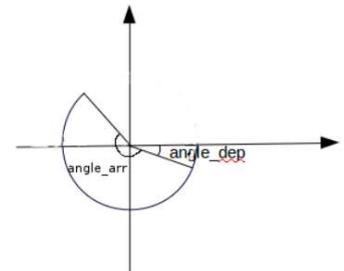


✍ Dessiner un cercle au centre de la fenêtre.

## 3.6. L'arc

La fonction `arc(x, y, a, b, angle_dep, angle_arr)` permet de dessiner un morceau d'ellipse. La fonction `arc()` prend 6 paramètres.

- les quatre premiers paramètres sont les mêmes que pour la fonction `ellipse()` ;
- `angle_dep` correspond à l'angle de départ et `angle_arr` correspond à l'angle d'arrivée. Les angles sont donnés en radian ;
- pour utiliser des angles en degré, il faut utiliser la fonction `radians()`.



✍ Tester la ligne ci-dessous :  
`arc(200, 200, 100, 50, radians(45), radians(270))`

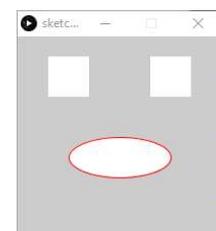
## 3.7. Autres figures

Processing propose d'autres fonctions pour les figures comme le triangle, le quadrilatère, les courbes et les formes libres.

## 3.8. Contours

Le contours peut être rendu invisible avec la fonction `noStroke()`.  
La fonction `stroke()` permet aussi de déterminer la couleur du contour.

✍ Tester les lignes ci-dessous :  
`size(200, 200)`  
`noStroke()`  
`rect(30, 20, 40, 40)`  
`rect(130, 20, 40, 40)`  
`stroke(255, 0, 0)`  
`ellipse(100, 120, 100, 40)`



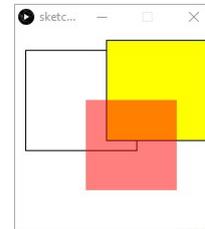
## 3.9. Remplissage

La fonction `fill()` permet de colorier l'intérieur d'une forme.

La fonction `noFill()` permet d'attribuer aucune couleur à l'intérieur d'une forme.

✍ Tester les lignes ci-dessous :

```
size(200, 200)
background(255)
stroke(0)
rect(10, 20, 110, 100)
fill(255, 255, 0)
rect(90, 10, 100, 100)
noStroke()
fill(255, 0, 0, 128)
rect(70, 70, 90, 90)
```



## 3.10. Transparence

Dans l'exemple ci-dessus, la fonction `fill(255, 0, 0, 128)` est définie par quatre composantes.

- Pour rappel, les trois premières définissent les composantes rouge, verte et bleue.
- La quatrième, la composante alpha définit la transparence.

## 4. Dessiner

✍ Réaliser les deux figures ci-dessous :

- Pour le Pac-Man, il peut être judicieux d'utiliser les fonctions de transformation `translate()` et `rotate()` : <https://fr.flossmanuals.net/processing/les-transformations/>.
- Pour le fantôme, il est possible d'utiliser la fonction `vertex()` pour dessiner des formes libres : <https://fr.flossmanuals.net/processing/les-formes/>

