

Mémo Processing

1/ Environnement de l'espace de travail :

`size(largeur, hauteur);`

Méthode permettant de déterminer les **dimensions** en pixels de la fenêtre de visualisation (largeur, hauteur).

```
// les dimensions de la fenêtre de visualisation
// est de 500px de large et 300px de haut
size(500,300);
```

`height`

Propriété permettant de connaître la **hauteur en pixels** de la fenêtre de visualisation.

```
// dessiner un point au milieu de la fenêtre de visualisation
point(width/2, height/2);
```

`width`

Propriété permettant de connaître la **largeur en pixels** de la fenêtre de visualisation.

```
// dessiner un point au milieu de la fenêtre de visualisation
point(width/2, height/2);
```

2/ Les couleurs

`code couleur RVB et alpha`

Le **code couleur RVB** se présente sous la forme de 3 codes de chiffres compris entre **0** et **255**.

Chaque code représente le dosage nécessaire de chacune des couleurs primaires pour obtenir la couleur désirée (0= pas de couleur, 255=max de couleur).

Les trois codes représentent respectivement le dosage du rouge, du vert et du bleu.

Exemple :

le blanc : (255,255,255)	- le noir : (0,0,0)	- le rouge : (255,0,0) -
le vert : (0,255,0)	- le bleu : (0,0,255)	- le jaune : (255,255,0) -
le cyan : (0,255,255)	- le gris : (125,125,125)	- le rose : (253,108,158) -
l'orangé : (244,102,27)	- le violet : (102,0,153)	- le caramel : (126,51,0) ...

Il est possible d'ajouter un 4e code : la transparence ou alpha, lui aussi compris entre 0 et 255 (0 = transparent et 255 = opaque).

`background(valeur);`

Méthode permettant de déterminer la **couleur de fond** de la fenêtre de visualisation.

Les valeurs vont de 0 à 255.

> une seule valeur = niveaux de gris.

> trois valeurs = couleur RVB.

```
// le fond de la fenêtre de visualisation est noir
background(0);
// le fond de la fenêtre de visualisation est blanc
background(255);
// le fond de la fenêtre de visualisation est bleu R=0,V=0 et B=255
background(0,0,255);
```

`fill(valeur);`

Méthode permettant de définir la couleur du fond d'une forme que l'on souhaite dessiner :

> une seule valeur = niveaux de gris ou 2 valeurs = niveau de gris + alpha (la transparence)

> trois valeurs = couleur RVB ou 4 valeurs = couleurs RVB + alpha (la transparence)

```
// le fond de la forme est bleu R=0, V=0, B=255 avec une transparence de moitié
fill(0,0,255,125);
// dessine un carré de sommet haut gauche situé aux coordonnées (20,20) et de côté de 200
rect(20,20,200,200);
```

stroke(valeur);

Méthode permettant de définir la couleur du contour d'une forme que l'on souhaite dessiner :

> une seule valeur = niveaux de gris ou 2 valeurs = niveau de gris + alpha (la transparence)

> trois valeurs = couleur RVB ou 4 valeurs = couleurs RVB + alpha (la transparence)

```
// le contour de la forme est rouge R=255, V=0, B=0 avec une transparence de 20
```

```
stroke(255,0,0,20);
```

```
// dessine un carré de sommet haut gauche situé aux coordonnées (20,20) et de côté de 200
```

```
rect(20,20,200,200);
```

noFill() et noStroke(valeur);

La couleur de remplissage de la forme et son trait de contour peuvent être désactivés avec ces deux méthodes.

3/ Les outils de dessin

point(x,y);

Dessine un point aux coordonnées (x,y).

line(x1,y1,x2,y2);

Dessine un segment de droite entre deux points aux coordonnées (x1,y1) et (x2,y2).

triangle(x1,y1,x2,y2,x3,y3);

Dessine un triangle.

quad(x1,y1,x2,y2,x3,y3,x4,y4);

Dessine un quadrilatère irrégulier ou régulier.

rect(x1,y1,largeur,hauteur);

Dessine un rectangle ou un carré.

ellipse(x1,y1,diamètre horizontal,diamètre vertical);

Dessine une ellipse ou un cercle.

strokeWeight(valeur)

L'épaisseur d'un trait exprimée en pixel.

```
// détermine l'épaisseur d'une ligne à 10 px
```

```
strokeWeight(10);
```

```
// dessine une ligne d'épaisseur 10 px
```

```
line(10,10,50,50)
```

4/ Le texte

text("monTexte", positionX, positionY);

Méthode permettant de **dessiner un texte** aux coordonnées (x,y) dans la fenêtre de visualisation.

```
// Dessine la lettre "A" aux coordonnées (10,10)
```

```
text("A",10,10);
```

```
textSize(valeur);
```

Méthode permettant de **déterminer la taille d'un texte** en pixels.

```
// Les dimensions du caractère seront de 50px de haut
textSize(50);
// Dessine la lettre "A" aux coordonnées (10,10)
text("A",10,10);
```

5/ transformations

```
translate(x,y);
```

Permet de déplacer par translation, l'axe des repères de notre fenêtre de dessin de x sur l'axe horizontal et de y sur l'axe vertical.

```
rotate(angle);
```

Permet de déplacer par rotation, l'axe des repères de notre fenêtre de dessin d'un angle en radians (sens des aiguilles d'une montre).

```
45° = PI/4, 90° = PI/2, 180° = PI et 360° = 2*PI
```

5/ Autres

```
random(valeur);
```

Méthode permettant de **générer aléatoirement un nombre** compris entre 0 et 'valeur'.

```
// Donne une taille comprise entre 0 et 10 : 1, 2... ou 10 ?
textSize(random(10));
```

```
for(valeur de départ; condition; actualisation){ instructions};
```

Une boucle for permet d'**effectuer automatiquement une tâche répétitive**.

```
// Répète 100 fois les instructions
for(int i=0; i<100 ; i=i+1){ instructions};
```

```
save("monImage.png");
```

Méthode permettant d'**enregistrer** dans le dossier de votre programme ('sketch1.pde') votre image.

Les variables

Une variable est une sorte de boîte repérée par le programme (l'ordinateur) à l'aide d'une adresse (nom). Pour avoir accès au contenu de cette variable (boîte), il suffit de la désigner par son adresse (nom).

Il existe différents types de variables :

```
int des nombres entiers,
float des nombres décimaux,
String du texte,
boolean des valeurs vrai/faux,
color une couleur...
```

Toutes les variables possèdent un nom, un type et une valeur.

```
// Création d'une variable 'monTexte'
String monTexte="Max";
```