


# L Codent L Créent - Mémo Processing



## Liens et astuces

- Comment accéder aux séances ? C'est simple ! Recopier l'url : [www.tinyurl.com/LCLC2019](http://www.tinyurl.com/LCLC2019)
- Vous venez d'effacer votre code ? Le raccourci `ctrl + Z` permet d'annuler la dernière action faite (écriture, effacement...)
- Vous cherchez une couleur en particulier ? Recopier l'url : <https://htmlcolorcodes.com/>
-  Pensez à **enregistrer** vos travaux sur un éditeur (bloc-note, libre office) avant de **fermer l'onglet de l'activité**, sinon vous perdez le code :) (idem actualisation, etc)

## Structure de base, coordonnées, formes.



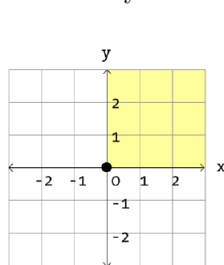
### Structure

- `setup()` :  
Instruction à exécuter.  
`run()`
- `# Ceci est un commentaire sur une ligne.`  
Il ne sera pas lu/exécuté par l'ordinateur ;) C'est votre mémo.

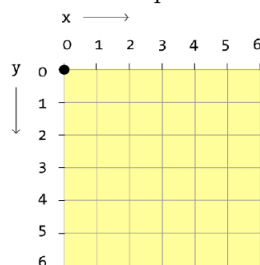


### Système de coordonnées

- Les axes x et y ne sont pas les mêmes que d'habitude :



Eighth Grade



Computer

- Le point du coin en haut à gauche de chaque fenêtre est à la coordonnées (0,0). La fonction `translation()` change cet axe.
- Unité de mesure : **pixel**.
- `width` : Longueur (L) de l'objet, • `height` : Hauteur (H) de l'objet.
- `size(width, height)` : Règle la taille (L,H) de la fenêtre principale.



### Bases de géométrie

Les commandes suivantes dessinent...


- `ellipse(x, y, l, h)` : une ellipse en (x,y), de longueur l et de hauteur h.
- `rect(x, y, l, h)` : un rectangle, dont le coin en haut à gauche est en (x,y), de longueur l et de hauteur h.
- `line(x1, y1, x2, y2)` : un segment, d'extrémité (x1, y1) à (x2, y2).
- `point(x, y)` : un point à la coordonnée (x, y)
- `triangle(x1, y1, x2, y2, x3, y3)` : un triangle dont les coins sont en (x1, y1), (x2, y2), (x3, y3).
- `quad(x1, y1, x2, y2, x3, y3, x4, y4)` un quadrilatère dont les coins sont en (x1, y1), (x2, y2), (x3, y3), (x4, y4).

## Bordures, remplissage, et contour des formes

- Où trouver les codes R-V-B des couleurs? Code déc. du tableau [https://fr.wikipedia.org/wiki/Couleur\\_du\\_Web#Noms\\_de\\_couleurs\\_SVG\\_1.0](https://fr.wikipedia.org/wiki/Couleur_du_Web#Noms_de_couleurs_SVG_1.0)
- `background(color)` : Règle la couleur de fond de la fenêtre principale.
- `fill(R,V,B)` : Règle la couleur de fond des prochaines formes à (R,V,B sont des nombres entre 0 – 255 et règlent les taux de Rouge, de Vert et de Bleu).
- `nofill()` : enlève le fond pour les prochaines formes.
- `stroke(R,V,B)` : Règle la couleur de la bordure des prochaines formes (même format que `fill(R,V,B)`).
- `noStroke()` : Enlève les bordures pour des prochaines formes.

## Des boucles et des tests

### Structure des tests et des boucles

- **Si...alors...**  
Ce test permet d'exécuter différentes actions selon des conditions. Voici la structure d'un test :  
If (une condition):  
  
    Execution d'un code  
  
Elif (une condition):  
  
    Execution d'un code  
  
Else (pas besoin de condition):  
  
    Execution d'un code
- **Boucle for : Pour... faire**  
La boucle permet de répéter une action sans la ré-écrire :  
For x in range(TRUC):  
  
    Exécution d'un code
- L'exécution du code peut dépendre de x.
- Dans TRUC on peut mettre :
  - (1, 12, 2) pour exécuter la commande pour  $x = 1, 3, 5, 7, 9$  (le pas est de 2).  
*Exemple : `print(range(1, 10, 4))` renvoie [1, 5, 9]*
  - (1, 7) pour exécuter la commande pour  $x = 1, 2, 3, 4, 5, 6$   
*Exemple : `print(range(1, 10))` renvoie [1, 2, 3, 4, 5, 6, 7, 8, 9]*
-  **Dans tous les cas** : L'indentation (=les deux espaces) sont importants.

### Opérateurs

- `=` : défini comme. Exemple : `Taille=6` affecte 6 à la variable appelée Taille.
- `==` : test d'égalité
- `>` : supérieur à. Et bien sûr : `>=` supérieur ou égal à
- `!=` : test d'inégalité

## Un peu d'animation...

### Affichage et animations

- `frameRate(fps)` : Règle les FPS (Nombres d'images/sec) de l'application.
- `print(string)` : Écrit une chaîne de caractère sur la console.
- `println(string)` : La même chose, avec un saut de ligne.
- `delay(millisecons)` ; Place une pause/un délai (en milisecondes).

### Variables globales

- `mouseX`, `mouseY` : Renvoie les coordonnées (X,Y) de la position de la souris.
- `pmouseX`, `pmouseY` : Renvoie les coordonnées (X,Y) de la position précédente de la souris.
- `draw()` : Se place après `setup()`, on y met les éléments **animés**
- `frameRate` : Règle le nombre d'images par secondes.
- `frameCount` : Renvoie le nombre d'images qui ont été affichées par `draw()`

Pour le jour  $J$ , du mois  $m$ , à l'heure  $H$ , la minute  $M$  et la seconde  $S$

### Heure & Date

- `month()` : Renvoie le mois actuel (format 4 pour avril par ex.)
- `day()` : Renvoie le jour  $J$  du mois  $m$  actuel.
- `hour()` : Renvoie l'heure  $H$  actuelle.
- `minute()` : Renvoie la minute  $M$  actuelle.
- `second()` : Renvoie la seconde  $S$  actuelle.