

# Processing : afficher du texte

## Présentation

Votre programme doit lire un fichier texte et l'afficher ligne par ligne.

L'instruction Processing pour afficher du texte est **text("texte à afficher",x,y)**

**A faire** : crée ce nouveau programme qui affiche le contenu du fichier **cigale.txt**. Ce fichier **cigale.txt** doit être dans le même dossier que ton projet.

```
# définir une variable globale qui contiendra le contenu du texte
contenu= 0

def setup():
    # utiliser la variable globale contenu
    global contenu
    # taille de la fenêtre
    size(400,400)
    frameRate(5)

    #lecture du contenu du fichier
    fichier=open('cigale.txt','r')
    contenu=fichier.read()
    fichier.close()

def draw():
    # utiliser la variable globale contenu
    global contenu
    background(255)
    fill(0) # équivalent à fill(0,0,0)
    # affiche tout le texte
    text(contenu,10,10)
```

Pour **personnaliser** l'affichage du texte vous pouvez :

- changer la **taille** de la police de caractère avec la fonction **textSize(x)**,
- changer la **justification** avec la fonction **textAlign(paramètre)** et les paramètres suivants :
  - LEFT valeur par défaut,
  - RIGHT,
  - CENTER

Modification de la fonction **draw()** pour afficher le texte, centré, avec une taille de 16 et une couleur verte :

```
# couleur du texte
fill(100, 255, 50)
# taille du texte
textSize(16)
# centrage du texte
textAlign(CENTER)
# affiche tout le texte
text(contenu,200,50)
```

Tout le texte n'est pas affiché car il est trop long.

## Affichage du texte ligne par ligne

La variable **contenu** contient l'ensemble du texte. En appliquant la méthode **split()**, il est possible de **découper** tout le texte en précisant quel caractère doit être utilisé pour cela :

- on pourrait faire un **découpage avec le point**, qui délimite la fin d'une phrase ,

- mais ici c'est le caractère **retour à la ligne** qui doit être utilisé. Ce caractère n'est pas affiché mais est utilisé pour passer à la ligne suivante. Le marque de fin de ligne se note `\n`.

Le résultat de la méthode **split()** retourne un **tableau** avec les différents éléments trouvés. IL ne reste alors qu'à afficher un nouvel élément du tableau à chaque appel de la fonction **draw()**.

**A faire :** modifie le programme précédent pour afficher le texte ligne par ligne :

```
# définir une variable globale qui contiendra le tableau des lignes du texte
tableau = []
# variable pour compter les lignes
ligne=0

def setup():
    # utiliser la variable globale tableau
    global tableau
    # taille de la fenêtre
    size(400,400)
    frameRate(1)

    #lecture du contenu du fichier
    fichier=open('cigale.txt','r')
    contenu=fichier.read()
    # découpage ligne par ligne
    tableau=contenu.split("\n")
    fichier.close()

def draw():
    # utiliser les variables globales tableau et ligne
    global tableau, ligne
    background(255)
    fill(0)
    # affiche une ligne du tableau
    text(tableau[ligne],10,50)
    # passer à la ligne suivante si on n'est pas à la dernière ligne
    if ligne < len(tableau) - 1 :
        ligne = ligne + 1
```

<html>

</html>

## Projet Processing ...

- [Projet Processing : du texte et des images](#)

From:

/ - **Les cours du BTS SIO**

Permanent link:

</doku.php/icn/facultatif/processing/texte1>

Last update: **2019/04/04 16:23**

