

## Objectif :

Découvrir l'utilisation des images sous processing.

### 1. Formats d'image:

Trois formats de fichier d'image sont acceptés dans Processing : **PNG, JPEG, ou GIF.**

### 2. Importer une image:

#### a) Déposer/glisser :

#### b) Importer une image:

Syntaxe : PImage a;

a=loadImage("Nom du fichier");

image (a, X, Y, "Largeur", "hauteur");

I) Ecrire le programme ci-dessous et sauvegarder sous exemple1.pde  
size(500,400);

PImage ile;

ile = loadImage("Image1.JPG");

image(ile,50,10,400,380);

Qu'obtenez-vous?

Réponse :

#### c) Image importé du web

II) Ecrire le programme ci-dessous et sauvegarder sous exemple2.pde  
size(400,400);

PImage webcam;

webcam = loadImage("http://www.gutenberg.org/files/3913/3913-h/images/rousseau.jpg");

image(webcam,0,0,width,height);

Qu'obtenez-vous?

Réponse :

### 3. Changer la taille d'une image:

Pour changer la taille de l'image, il suffit de rajouter deux paramètres à votre image, {largeur, hauteur},

III) Ecrire le programme qui affiche l'image ci-contre et sauvegarder sous exemple3.pde

Taille de la fenêtre 500\*400 pixels<sup>2</sup>

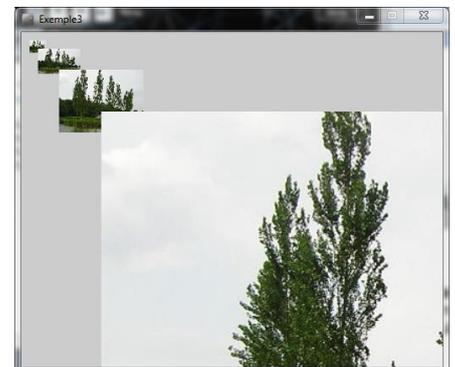
Taille image 1 : 20\*15 pixels<sup>2</sup>

Taille image 2 : 50\*30 pixels<sup>2</sup>

Taille image 3 : 100\*75 pixels<sup>2</sup>

Taille image 4 : 1000\*750 pixels<sup>2</sup>

Réponse :



## 4. Rendre transparent l'arrière-plan d'une image:

IV) Ecrire le programme qui affiche l'image ci-contre et sauvegarder sous exemple4.pde  
 Taille de la fenêtre 400\*300 pixels<sup>2</sup>  
 Taille image 1 : 400\*300 pixels<sup>2</sup>  
 Taille de Rousseau : celle d'origine.



Réponse :

## 5. Colorier les images:

Syntaxe : PImage a;  
 a=loadImage("Nom du fichier");  
 tint(R,G,B,Alpha);  
 image(a, X, Y, "Largeur", "hauteur");

V) Ecrire le programme qui affiche l'image ci-contre et sauvegarder sous exemple5.pde  
 size(500,150);  
 Taille de la fenêtre 500\*150 pixels<sup>2</sup>  
 Taille des images : 110\*110 pixels<sup>2</sup>



Réponse :

## 6. Création d'une image à partir de la valeur de chaque pixel:

```

VI) Ecrire le programme ci-dessous et sauvegarder sous exemple6.pde
size(200, 200);
// Conversion en pixels
loadPixels();
// Définit chaque pixel
for (int i = 0; i < pixels.length; i++) {
  float R = random(255);
  float G = random(255);
  float B = random(255);

  // Création de la couleur
  color c = color(R,G,B);
  //On met cette valeur dans le tableau de pixels
  pixels[i] = c;
}
// Affiche le résultat
updatePixels();
Qu'obtenez-vous?
    
```

Réponse :

VII) Ecrire le programme qui affiche l'image ci-contre et sauvegarder sous exemple7.pde

Taille de la fenêtre 256\*256 pixels<sup>2</sup>  
 Pour trouver les valeurs RGB, il faut deux boucles FOR:

```
for (int j = 0; j < 256; j++)  
{  
    for (int k = 0; k < 256; k++)  
    {  
        float R = 255-j;  
        float G = j;  
        float B = k;  
    }  
}
```

Réponse :

