

Tutorat Javascript -- ImageJ

JUNG Frédéric : frederic.iung@etu.u-bordeaux.fr

16/10/2018 11:14 AM



Vous devez télécharger le dossier contenant les images [ici](#) :

A chaque exercice vous complèterez le code de l'exercice précédent.

Toutes les informations pour réaliser cet exercice sont disponible dans l'[API](#). **L'exercice n'a pas d'intérêt si vous allez chercher les réponse sur le net.** L'API est nécessaire et suffisante pour réaliser ces exercices.

Exercice 1

```
input_images = ["cremi.png", "fredo.jpg", "ketchup.jpg", "mayo.jpeg"];
```

Vous devez ouvrir les images une par une (cf `openImage`) et afficher les informations suivantes dans le terminal
ImagJ :

`openImage`

```
public static ImagePlus openImage(java.lang.String path)
```

Opens the specified file as a tiff, bmp, dicom, fits, pgm, gif, jpeg or text image and returns an ImagePlus object if successful. Calls HandleExtraFileTypes plugin if the file type is not recognised. Displays a file open dialog if 'path' is null or an empty string. Note that 'path' can also be a URL. Some reader plugins, including the Bio-Formats plugin, display the image and return null. Use IJ.open() to display a file open dialog box.

- afficher la **moyenne des largeurs** des images
- afficher la **moyenne des hauteurs** de ces images
- afficher le nombre d'images

Exercice 2

- passer les images en **8-bit**.
- **rogner** les images de moitié (cf `makeRectangle`)

`makeRectangle`

```
public static void makeRectangle(double x,  
                                double y,  
                                double width,  
                                double height)
```

Creates a subpixel resolution rectangular selection.

Aller plus loin

- permettre à l'utilisateur de choisir lui même le répertoire où sont stockés les images et remplir le tableau `input_images` avec les images contenus dans ce répertoire.