Lesson 16: Loading and Using Images

Lesson aim:

To learn how to load external files - images in this case - into our program.

Why:

Having the background that is a certain colour, or changing the colour of a shape, is great. But in modern games and animation, images are often used instead of blocks of colour. In games these are often called “textures” and there are hundreds in each scene. We will only be dealing with a small number of big images. But is is useful to use a custom image rather than a colour.

How to use images

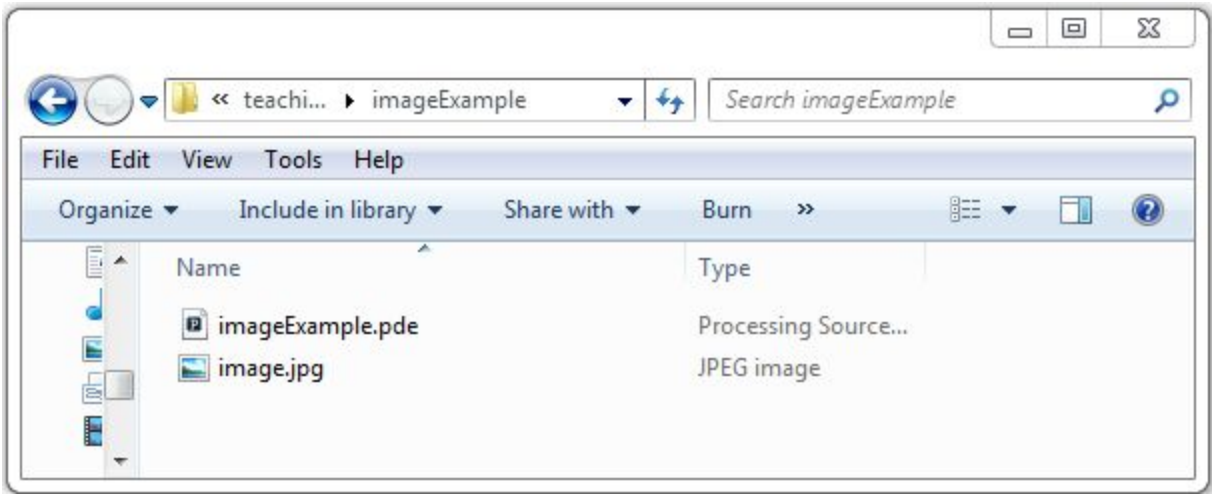
There are **four steps** to using an image in processing:

1. Finding an image file we want to use in a program
2. Make an image variable.
3. Load the image file into the program and store it in the image variable.
4. Display the image variable on the screen.

Step 1: Getting an image to use.

1. Make a new processing program, name starting with “**Lesson16**”
2. Save the program with a name, in this example I’ve used “imageExample”.
3. Find an image you want to show on the screen (online or in the pictures folder - hint when using google images: click on the image → view image → right click on image → save image as → save it in your program folder.)
4. Your image can be named whatever you want - name it after what it shows.
5. Move your image to the folder of the program you just created.

See below:



Step 1: Make an image variable

We need to store your image. Like we do with whole numbers (`int`) and true or false statements (`boolean`), images have a different variable type: `PImage`
Like all variables, we declare (make) them right at the top of the program.

My example is an image of a medal:

```
PImage myMedalImage;
```

Step 2: Load the image file into the image variable.

This needs to happen in the `setup()` ***{code block}***. As the image isn't going to change; we load it once, right at the start of the program.

```
void setup() {
  myMedalImage = loadImage("image.jpg");
}
```

The variable name is the same as we made is step 1 (I chose `myMedalImage`), and the name of the file is the same as the one we made in step 0 (`image.jpg` in this example). If that is spelt wrong, or in the wrong place, the program won't run.

Step 3: Display the image variable on the screen.

In the same way we can show text to the screen using the `text()` function, there is a function to show images to the screen. Helpfully its called `image()`. This function shows an image variable at a specific location.
I want my image to start it in the top left hand corner so I'm going to put it at 0,0.

```
image(myMedalImage, 0, 0);
```

Congrats! You should be able to see your custom image on the screen!

Tasks to get familiar to using images:

- Move your image around the screen using:
  - The mouse
  - The keyboard (using variables)
- Add a second image into the same program. You will need to repeat steps 0-3 using a different image (different file, different variable, load it in separately and show it on the screen separately.)
- **Project task:** now you know about images you can add the to your project. Ideas to try:
  - An image as background to the game
  - A different image for the end game screen.