Let's Start with a Favorite Image of Yours

It can be in .jpg, .bmp, or .png format

Each pixel contains a red-green-blue, each in the range 0-255

The image has an aspect ratio, which is the ratio of the number of Y pixels : the number of X pixels (this image's aspect ratio is 1:1)

Loading and Drawing an Image

```
PImage MyImage;
void setup() {
  size(800, 800);
  //MyImage = loadImage("C:/MJB/Processing/ImageSketchBook/zelda.jpg");
  MyImage = loadImage("zelda.jpg");
}

void draw() {
  image(MyImage, 0, 0, 800, 800);
}
```

"PImage" is a variable type, just like int and float, but for images.

Declaring a variable up here, ahead of everything else, makes it so that it can be seen from anywhere in the program.

This loads the image from the file into the variable called MyImage.

This draws the image from the variable called MyImage.

What X-Y to draw its upper-left corner at.

How many pixels to use to draw the image.

What Happens if You Use Less Pixels than the Window Has?

```
void draw() {
  image(MyImage, 50, 50, 480, 480);
}
```

What Happens if You Use a Different Aspect Ratio?

```
void draw() {
  image(MyImage, 58, 59, 680, 380);
}
```
Translating an Image

```cpp
for( int i = 0; i < 6; i++ )
    pushMatrix();
    translate(100*i, 0, 0);
    Image MyImage, 0, 0, 200, 200;
popMatrix();
```

Rotating an Image

```cpp
for( int i = 0; i < 6; i++ )
    pushMatrix();
    translate(100*i, 0, 0);
    rotate( radians(45*i) );
    Image MyImage, 0, 0, 200, 200;
popMatrix();
```

Overwriting an Image

```cpp
int x = 100 * i;
int y = 200 * (6 - i);
setPixel( x, y, 255, 255, 255 );
```

Retrieving Image Colors

```cpp
int x = 100 * i;
int y = 200 * (6 - i);
int r, g, b;
getPixel( x, y, &r, &g, &b );
```