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

```java
PImage MyImage;

void setup( )
{
  size( 800, 800 );
  MyImage = loadImage( "zelda.jpg" );
}

text("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.
How many pixels to use to draw the image.
What X-Y to draw its upper-left corner at.

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.
How many pixels to use to draw the image.
What X-Y to draw its upper-left corner at.
What Happens if You Use Less Pixels than the Window Has?

```c
void
draw()
{
    image(MyImage, 50, 50, 400, 400);
}
```

What Happens if You Use a Different Aspect Ratio?

```c
void
draw()
{
    image(MyImage, 50, 50, 600, 300);
}
```
Translating an Image

```cpp
void draw() {
  for (int i = 0; i < 6; i++) {
    pushMatrix();
    translate( i*100, i*100);
    image( MyImage, 8, 0, 200, 200);
    popMatrix();
  }
}
```

Rotating an Image

```cpp
void draw() {
  for (int i = 0; i < 5; i++) {
    pushMatrix();
    translate( 100, 100);
    rotate( radians(1*60));
    image( MyImage, 0, 0, 200, 200);
    popMatrix();
  }
}
```
**Overwriting an Image**

```java
void draw() {
    Image(myImage, 0, 0, 888, 888);
    for (int x = 0; x < myImage.width; x++) {
        for (int y = 0; y < myImage.height; y++) {
            Color diff = new Color(255, 0, 0);
            myImage.set(x, y, diff);
        }
    }
}
```

**Retrieving Image Colors**

```java
void draw() {
    Image(myImage, 0, 0, 888, 888);
    for (int x = 0; x < myImage.width; x++) {
        for (int y = 0; y < myImage.height; y++) {
            Color c = myImage.get(x, y);
            myImage.set(myImage.width - x, y, c);
        }
    }
}
```