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");
}

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?

```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

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

Rotating an Image

```c
void draw()
{
    for( int i = 0; i < 6; i++ )
    {
        pushMatrix();
        translate( 300, 300 );
        rotate( radians(1*360) );
        image( MyImage, 0, 0, 200, 200 );
        popMatrix();
    }
}
```
Overwriting an Image

```java
word draw()
{
    Image( MyImage, 0, 0, 884, 884 );
    For( int x = 100; x < 200; x = x + 1 )
    {
        For( int y = 0; y < MyImage.Height; y = y + 1 )
        {
            uint8_t old = MyImage.get( x, y, GET );
            MyImage.set( x, y, GET, 0 );
        }
    }
}

Retrieving Image Colors

```