

# Driver Overview

Ya boi

May 30, 2016

## 1 Now this is a story all about how

I had a hard time fitting this code on the first page so I'll say what I can about the driver class here and then the code will be included on the next page. The formatting is a bit weird because of the whole "transporting code to L<sup>A</sup>T<sub>E</sub>X" thing but I'll try to keep it readable.

The driver is the entry point to the rest of the program and contains multiple variables which are passed as parameters to instantiate an ImageToText object (which is poorly named but it is the hub by which everything else is accessed) and then to call the appropriate ImageToText methods. This table explains the variables in the driver class which require user input. There are some bugs and caveats which govern what options you can choose and when but those will be mentioned within the code as comments. The list of all user dependent variables has three sections, which I'll make into an itemized list for easy reference.

- Top - Variables at the top of the page outside of any control loops.
- Middle - Variables within the ascii creation control loop.
- Bottom - Variables within the color conversion and resizing control loop.

You would think that the variables specific to each function (ascii and color) would be within the same loops but they aren't necessarily. I never claimed to be good at this.

General	Ascii	Color
fileName	fontsize	dithering
extension	transparency	colortype
simpleswitch	background	dithertype
reduceBy	loop	blockSize
		expandSize

<b>String fileName</b> <b>String extension</b> <b>int simpleswitch</b> <b>int reduceBy</b> <b>boolean dithering</b> <b>ColorTag colortype</b> <b>DitherTag dithertype</b>	Actual file name of graphic without extension and without path File extension type ".type", used for finding file and for determining method calls. Cheap, temporary way for choosing between ascii and color conversion routines. Chooses by how much an image is compressed before further processing. Designates whether image will be dithered. Enumerated type which specifies color scheme image should be converted to. Enumerated type which specifies which dithering method to use.
<b>int fontsize</b> <b>boolean transparency</b> <b>Color background</b> <b>int loop</b>	Determines font size to use when constructing ascii image. Sets background pixels of the resultant image to be transparent. Chooses background color of resultant ascii image. Used to process multiple files with consecutive numbers in the name.
<b>int blockSize</b> <b>int expandSize</b>	Functionally identical to reduceBy. Averages pixel values by blockSize. Expands pixels to create larger resulting image.

---

```

1      public class Driver {
2          public static void main(String[] args){
3              int simpleswitch = 0;
4              String fileName = "dragonball1";
5              String extension = ".gif";
6              int reduceBy = 4;
7              ColorTag colortype = ColorTag.SNES;
8              DitherTag dithertype = DitherTag.FS;
9              boolean dithering = true;
10             FType filetype;
11
12             if (extension.compareTo(".gif") == 0)
13                 filetype = FType.GIF_TYPE;
14             else
15                 filetype = FType.PNG_TYPE;
16
17             ImageToText itt = new ImageToText(fileName, extension, filetype, colortype, dithertype);
18
19             if(simpleswitch == 0) {
20                 int fontsize = 20;
21                 boolean transparency = false;
22                 Color background = Color.BLACK;
23
24                 int loop = 1;
25                 if (loop == 1) {
26                     itt.colorText(background, reduceBy, fontsize, transparency);
27                 }
28                 if (loop > 1) {
29                     while (loop < 4) {
30                         itt.colorText(background, reduceBy, fontsize, transparency);
31                         loop++;
32                     }
33                 }
34             }
35             else {
36                 int blockSize = 1;
37                 int expandSize = 4;
38                 itt.blockImageCreate(reduceBy, blockSize, expandSize, dithering);
39             }
40         }
41     }

```

---

Figure 1: Driver class with weird formatting cuz latex