D/Variable Scope

The rule of variable scope is defined simply: a variable created inside a block (code enclosed within braces: { and }) exists only inside that block. This means that a variable created inside `setup()` can be used only within the `setup()` block, and likewise, a variable declared inside `draw()` can be used only inside the `draw()` block. The exception to this rule is a variable declared outside of `setup()` and `draw()`. These variables can be used in both `setup()` and `draw()` (or inside any other function that you create). Think of the area outside of `setup()` and `draw()` as an implied code block. We call these variables *global variables*, because they can be used anywhere within the program. We call a variable that is used only within a single block a *local variable*. Following are a couple of code examples that further explain the concept. First:

```
int i = 12;   // Declare global variable i and assign 12

void setup() {
  size(480, 320);
  int i = 24; // Declare local variable i and assign 24
  println(i); // Prints 24 to the Console
}

void draw() {
  println(i); // Prints 12 to the Console
}
```

And second:

```
void setup() {
  size(480, 320);
  int i = 24; // Declare local variable i and assign 24
}

void draw() {
  println(i); // ERROR! The variable i is local to setup()
}
```