Week 7:

This week, I have learned about different kinds of loops and how to format code more “properly”. Formatting is more to personal preference although some ways of doing it are more commonly used. Let’s start off with loops.  
  
  
First off, loops are just parts of code that are repeating. There are two types of loops (there are more but these are the most used). These loops are the “For” and “While” Loops.   
  
There isn’t a huge difference between the two types, although each have a more common application (not specifically mentioned in the website but what I found from experience with programming). While loops are more commonly used with a Boolean value or serve as a way to repeat something infinitely. In other words, they repeat things indefinitely unless you provide a parameter at which it stops when it is complete (although most used to repeat indefinitely).  
  
Here is the two cases demonstrated.  
  
While(true){

println(“Repeating!”);

}

and…

While(mouseIsPressed === false){

println(“Not pressed”);

}  
  
Although not being the only cases to use a while loop, it is the most common purpose of one. On the other hand, you have a for loop. In the lesson, the website is showing a while loop this way (which is a way that works but less commonly used).

Var y = 40  
While(y < 400){

text(message, 30, y);

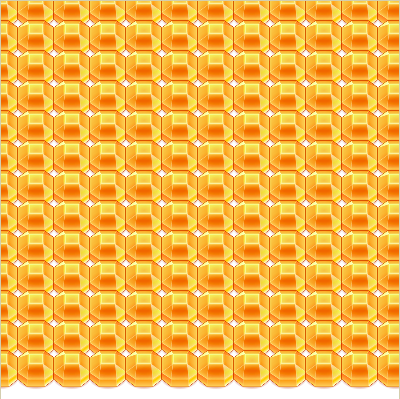
y += 20;

}

This is more the format you would use for a “For loop”. Here is a for loop.  
  
For(var y = 40; y < 400; y += 20){

text(message, 30, y);

}

Both of these loops perform the exact same thing but the for loop is what would be considered as cleaner code in this case. The for-loop repeats something until the value of a variable has reached the specified amount. These are not required uses for them but are the most common. Essentially, both types of loops do the exact same thing but each one is cleaner in a different case scenario. while loops are more common for indefinite loops; For loops are more common for specific amounts of repeating actions.   
  
This is from my point of view. Although, the lesson does say that the for loop is just a smaller way of writing a while loop.  
  
  
Anyways, enough with my grudge against how Khan academy used the while loop. We have also learned about something called nested loops. Nested loops are simply loops inside of loops. This could be used to fill rows and columns of a grid. What I mean by that is that it can be used to make a row of squares(for example) repeated for every column. Here is an screenshot taken from the lesson to show what I mean.  
  
  
  
Here is how to do this.

imageMode(CENTER);

for(var y = 0; y < 13; y++){

for(var x = 0; x < 12; x++){

image(getImage(“cute/GemOrange”), x\*36, y\*30, 40, 60);

}

}

This makes a row of gems for every section of the column. We have also learned about image, getImage and imageMode. Image displays the specified image at the specified co-ordinates and with a specific size. getImage gets an image from the library of images from khan academy.  
  
getImage(“cute/GemOrange”);

The “cute” is essentially the folder/category the image resides in, while GemOrange is the name of the image you are searching for.  
  
The last thing we have learned is imageMode(). This specifies the anchor points of the images. What I mean by that is that it changes what point is the default position. The default is CORNER which means the top left corner is at the coordinates specified. The second option is CENTER, this makes it so the center of the image is at the points specified.

Next thing we have learned is formatting. Formatting is a question of personal preference although here is the apparent most common way things are formatted.  
  
  
fill(0, 0, 0);

For each parameter, you have a space right after the comma.

var myVar = 3;

There is a space between on both sides of the equal sign along with a capital at the beginning of new words in the variable name(since there is no space, you can use capitals to differentiate the words. You can also use underscores but capital is most common).  
  
if(true) {

println(“Indent!”);

} else {  
 println(“Indent!”);

}  
  
When you have curly braces, you want to indent whatever is inside. You also want to give a space on each side of an else/else if statement. These are the most common “formatting rules” used for programming in javascript.  
  
  
In recap, this week we have learned about loops, I have gotten the chance to rant about the while loop not being correct for my standards and learned how to format properly(the most common way).