Name:____________________________

Working with Variables

J. Wilcox

So far we have managed to create some images using a few different basic shapes and color. Our next goal is to start working towards being able to write more complicated programs. Many programs that we see only run once when something happens (for instance when we click on the play button), but there are also programs that will run over and over again.

Let's look at a simple program:

```
size (400,400);

background (0);

fill (150);
stroke (255);
ellipse(100,100,100,50);
```

If we run the program, it will create a window that is 400 pixels by 400 pixels with a black background that has a gray ellipse drawn on it. How can we get the program to run over and over again? The first thing we need to do is to start thinking about 'blocks of code'.

A block of code always starts with an open curly bracket '{' and ends with a closed curly bracket '}'. Our program is going to have just two blocks of code. One block is going to be called 'setup' and one block is going to be called 'draw'. The first block of code, 'setup', is going to happen just once. It will be used to create the canvas we draw on and set up initial conditions (for example the color of the canvas or where the figure might start on the screen). The second block of code, 'draw', will happen over and over again.

* reminder – *ellipse(x, y, width, height);* – the first two numbers tell the (x,y) coordinates of the ellipse's center.

```
void setup () {
        size (400,400);
}

void draw () {
        background (0);
        fill (150);
        stroke (255);
        ellipse(100,100,100,50);
}
```

The command 'size' is in the setup block of code. It will only happen once.

The lines of code for background, fill, stroke and ellipse are all in the draw block of code. They will be repeated over and over.

*notice that ALL of the lines of code are between the curly cue brackets

When we type our code in with our new functions, void setup() <u>and</u> void draw(), and run the program, what happens? ____ Why is the ellipse just sitting there? The same ellipse is being drawn over and over…not very exciting. There is nothing changing. How do we get the computer to draw a new ellipse in a different place? We need to use something called a variable. The word variable means _____________________________. Processing has a couple of variables that it understands already. Two of the variables are mouseX (the mouse's X position…in other words, how far to the left or right of the canvas your mouse's cursor is…) and mouseY (the mouse's Y position….in other words how far up or down on the canvas your mouse's cursor is…) *notice that 'x' and 'y' are capitalized! Let's change the first argument in the line of code for the ellipse to *mouseX*. Now the line of code is *ellipse(mouseX, 100, 100, 50);* Run the program….move the mouse around…what happens?

____

____

How does it change if the second argument is changed to mouseY…
ellipse(mouseX, mouseY, 100, 50);? ____

____

*here is something kind of fun…instead of the first argument being *mouseX* make it *width – mouseX* so now the code would look like this… *ellipse (width – mouseX, mouseY, 100, 50);*
you might also like….*ellipse (mouseX, height – mouseY, 100,50);*

So far our program shows just one ellipse. The draw function is continually adding a new background (covering up the last ellipse) and then a new ellipse. Let's put the background line in the setup block. Now the code looks like this:

```
void setup () {
        size (400,400);
        background (0);
}

void draw () {
        fill (150);
        stroke (255);
        ellipse(mouseX, mouseY, 100, 50);
}
```

How does this change the program? What are you seeing? ____________________________

____

____

What if you put mouseX or mouseY in for other numbers? Some other things to try…

What happens with the command *ellipse(mouseY, 100, 100, 50);?*

When you move the mouse down, the ellipse ____.

Or what about *ellipse(mouseY, mouseX, 100, 50);?*

____

____

____

Some challenges –

Instead of an ellipse, try using a rectangle, a triangle or more than one ellipse.

Create a shape that changes color as the mouse is moved around the screen. Remember that the command to color an object is *fill (____,____,____);*

Make a figure (any shape…ellipse, rectangle…) that starts off very large when the cursor is on the left hand side of the screen, but that figure shrinks (both height and width of the shape decreases) as the cursor is moved to the right side of the screen (or down the screen).

Tell me about something you tried out.
Here is my code (you do not have to use all of the lines) –

```
void setup () {
        size (400,400);
        ______________________________

        ______________________________

}


void draw () {
        fill (________________________);
        stroke (255);

        ______________________________

        ______________________________

        ______________________________

        ______________________________

        ______________________________

        ______________________________

        }
```

Here is what happened! ____

____

____

____

____

____

//notes for teacher

Could connect the way ‘movement’ is created with flipbooks. Here is a link to a video that shows a simple flipbook being created in the first 2 minutes.

https://www.youtube.com/watch?v=Un-BdBSOGKY