

Variables, branches, loops

1) Read: chapter 2 and 3. You may skip the “functions” section.

2) Try out the following and observe what happens when move mouse in the stage:

```
void setup() {  
  size(200,200);  
  background(255);  
}  
  
void draw() {  
  //background(255);  
  stroke(0);  
  fill(175);  
  rectMode(CENTER);  
  rect(mouseX,mouseY,50,50);  
}
```

Now try to remove “background(255)” from setup(), uncomment “background(255)” in draw() – you are moving the sentence from setup() to draw(). Run it again. Observe. Think about why it is now different.

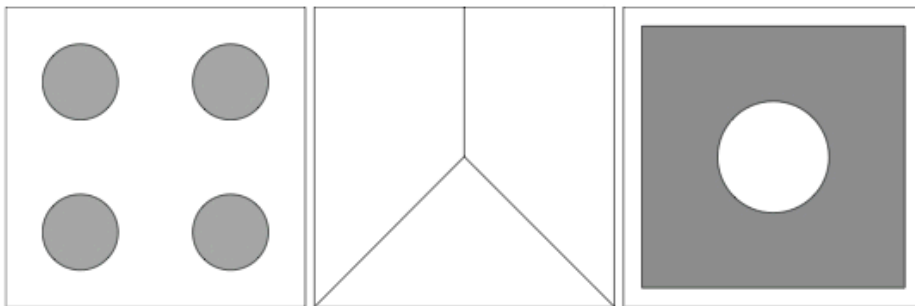
3)



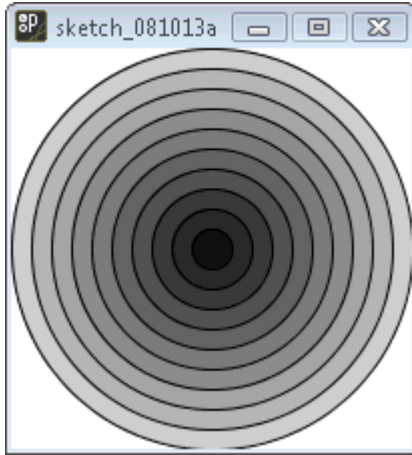
Step 1: Write code that draws the following screenshots with hard-coded values. (Feel free to use colors instead of grayscale.)

Step 2: Replace all of the hard-coded numbers with variables.

Step 3: Write assignment operations in **draw()** that change the value of the variables. For example, “variable1 = variable1 + 2;”. Try different expressions and see what happens!



4) Use a “For” loop to draw a picture as shows below:



5) The same task as described in 4), but this time using a “while” loop.

6) try out the following code into Processing, and move your mouse over the squares. Observe what happens. Your task is to modify the following code that shows the same behavior, but with 4x4 squares in the stage.

```
void setup() {  
    size(200,200);  
}  
  
void draw() {  
    background(255);  
    stroke(0);  
    line(100,0,100,200);  
    line(0,100,200,100);  
  
    noStroke();  
    fill(0);  
  
    if (mouseX < 100 && mouseY < 100) {  
        rect(0,0,100,100);  
    } else if (mouseX > 100 && mouseY < 100) {  
        rect(100,0,100,100);  
    } else if (mouseX < 100 && mouseY > 100) {  
        rect(0,100,100,100);  
    } else if (mouseX > 100 && mouseY > 100) {  
        rect(100,100,100,100);  
    }  
}
```

7) try to use nested loops to draw a picture as shows below. Hint: you might need random() function. Look it up in the reference.

