

# CART 253 **Creative Computation 1**

Email: [l.wilkins@concordia.ca](mailto:l.wilkins@concordia.ca)

Office Hours: Tuesday 12-1

Course Github: <https://github.com/LeeRobot/CART253-F-22>

# What we'll be doing today

- Inspiration
- Learning about variables
- Practicing using variables
- Work session for Project 1 (due next week)

Inspiration from our TA, Enric!

A variable is a piece of code that allows us to keep track of a specific element of information. Its a placeholder so that we can reference something again.

The first steps of interactivity

## User defined variables

Variables that the person writing the code creates, with unique names to track the number of something, the state of something, or its position in space or time.

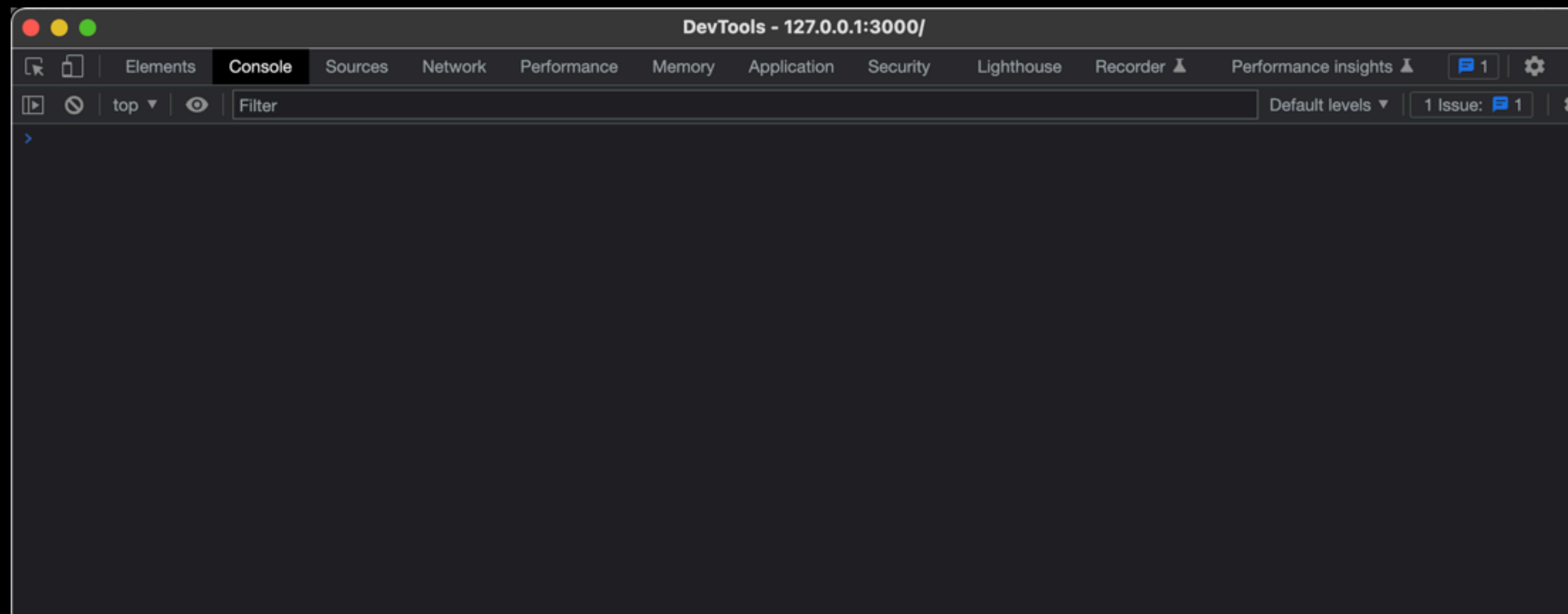
## Library/language defined variables

Variables that are pre-defined, to track specific things. These are the same for everyone, like tracking the mouse position, time, or window size.

# Types of Variables

# Viewing the variable

- You can view the contents of the variable by printing them using **console.log()** or **print()** (either works). You'll have to open the console to see the number.
- Chrome: View > Developer > Javascript Console
- Firefox: Tools > Web Developer > Browser Console



# Print debug

```
function setup() {  
  createCanvas(500, 500); // Create the canvas  
  background(255, 0, 0); // make the background red  
  print('This is the setup function');  
}  
  
function draw() {  
  fill(255, 204, 0); // fill in yellow  
  ellipse(200, 200, 200, 200); // draw circle in the middle  
  print('This is the draw function');  
}
```

# Print debug

Using print can help you understand:

- If you a part of your code is running
- If a variable contains the numbers you think it does!

This is a tool we'll use again and again throughout the class, you should get used to using the console!



# Track mouse position

```
function setup() {  
  createCanvas(500, 500); // Create the canvas  
  background(255, 0, 0); // make the background red  
}  
  
function draw() {  
  fill(255, 204, 0); // fill in yellow  
  
  ellipse(mouseX, mouseY, 200, 200); // draw circle in the middle  
}
```

These are existing variables p5 has defined for us to track the mouse check out the reference here <https://p5js.org/reference/#/p5/mouseX>

# Track mouse position

```
function setup() {  
  createCanvas(500, 500); // Create the canvas  
  background(255, 0, 0); // make the background red  
}  
  
function draw() {  
  fill(255, 204, 0); // fill in yellow  
  ellipse(mouseX, mouseY, 200, 200); // draw circle in the middle  
  print(mouseX);  
}
```

# Use the viable elsewhere

```
function setup() {  
  createCanvas(500, 500); // Create the canvas  
  background(255, 0, 0); // make the background red  
}  
  
function draw() {  
  fill(mouseY, 204, 0); // fill in yellow  
  ellipse(250, 250, mouseX, 200); // draw circle in the middle  
  print(mouseX);  
}
```

You can use the value from mouseX or mouseY anywhere that uses a similar number. We can manipulate numbers using the map() and constrain() functions, but we'll talk more about that later.

# Other Variables

keyCode

movedX

movedY

mouseX

mouseY

# When we get to mobile...

deviceOrientation

accelerationX

accelerationY

accelerationZ

pAccelerationX

pAccelerationY

pAccelerationZ

rotationX

rotationY

rotationZ

pRotationX

pRotationY

pRotationZ

turnAxis

setMoveThreshold()

setShakeThreshold()

deviceMoved()

deviceTurned()

deviceShaken()

# Using Random()

Random() is a function that generates a random number. You can set it to generate between 2 numbers, or between 0 and another number.

```
function setup() {  
  createCanvas(500, 500); // Create the canvas  
  background(255, 0, 0); // make the background red  
}  
  
function draw() {  
  fill(random(255), 204, 0); // fill in yellow  
  ellipse(250, 250, 200, 200); // draw circle in the middle  
}
```

# Draw & Setup

```
function setup() {  
  createCanvas(500, 500); // Create the canvas  
  background(255, 0, 0); // make the background red  
  fill(random(255), 204, 0); // fill in random  
}  
  
function draw() {  
  ellipse(250, 250, 200, 200); // draw circle in the middle  
}
```

# Draw & Setup

```
function setup() {  
  createCanvas(500, 500); // Create the canvas  
  background(255, 0, 0); // make the background red  
}  
  
function draw() {  
  fill(0, random(200, 255), 100); // fill in random  
  ellipse(250, 250, 200, 200); // draw circle in the middle  
}
```



# Making a variable

**Let** is a command that lets us define a variable. You can find it in the reference [here](#). We will add this line to create a variable

Let is the command we use to define a variable

let myVar = 2;

**2 is the value of myVar, which can change as our program runs.**

= and ; are syntax, it helps the program understand what we are trying to do.

myVar is the **unique** name I gave to my variable. It could be anything, but it helps to be descriptive.

# Scope

Scope refers to the access certain parts of a program have to each other.

```
Let myVar = 0;
```

```
function setup() {  
  
}
```

```
function draw() {  
  
}
```

Things outside of the setup, draw, or preload functions are “global” which means they can be accessed from anywhere

If I make a variable inside draw, I can only see it inside draw.

# Scope and timing

```
function setup() {  
  createCanvas(500, 500); // Create the canvas  
  background(255, 0, 0); // make the background red  
}  
function draw() {  
  Let myVar = 0;  
  fill(0, myVar, 100); // fill with myVar  
  ellipse(250, 250, 200, 200); // draw circle in the  
  middle  
  myVar++;  
}
```

Putting a variable in draw would reset it to 0 at the beginning of every draw function. Sometimes this is what you want, but not in our case.

```
function setup() {  
  Let myVar = 0;  
  createCanvas(500, 500); // Create the canvas  
  background(255, 0, 0); // make the background red  
}  
function draw() {  
  fill(0, myVar, 100); // fill with myVar  
  ellipse(250, 250, 200, 200); // draw circle in the middle  
  myVar++;  
}
```

Putting the variable in setup would mean it is out of scope from being used in draw

# Changing a variable

Let myVar = 0;



```
function setup() {
```

```
  createCanvas(500, 500); // Create the canvas
```

```
  background(255, 0, 0); // make the background red
```

```
}
```

```
function draw() {
```

```
  fill(0, myVar, 100); // fill with myVar
```

```
  ellipse(250, 250, 200, 200); // draw circle in the middle
```

```
  myVar++;
```

```
}
```

# Print our own variable

```
Let myVar = 0;
```

```
function setup() {  
  createCanvas(500, 500); // Create the canvas  
  background(255, 0, 0); // make the background red  
}
```

```
function draw() {  
  fill(0, myVar, 100); // fill with myVar  
  ellipse(250, 250, 200, 200); // draw circle in the middle  
  myVar++;  
  print(myVar);  
}
```

# Print our own variable

```
Let myVar = 0;
```

```
Let myVar2 = 255;
```

```
function setup() {
```

```
  createCanvas(500, 500); // Create the canvas
```

```
  background(255, 0, 0); // make the background red
```

```
}
```

```
function draw() {
```

```
  fill(0, 255, 100); // fill with myVar
```

```
  ellipse(250, 250, myVar, myVar2); // draw circle in the middle
```

```
  myVar++;
```

```
  myVar2—;
```

```
  print(myVar);
```

```
}
```

# Print our own variable

```
Let myVar = 0;
```

```
Let myVar2 = 255;
```

```
function setup() {
```

```
  createCanvas(500, 500); // Create the canvas
```

```
  background(255, 0, 0); // make the background red
```

```
}
```

```
function draw() {
```

```
  fill(0, 255, 100); // fill with myVar
```

```
  ellipse(250, 250, myVar, myVar2); // draw circle in the middle
```

```
  myVar++;
```

```
  myVar2 = random(200);
```

```
  print(myVar);
```

```
}
```

# Style & formatting

Full style guide [here](#).

- Get rid of extra white space
- use TAB to indent functions, everything inside the function should be aligned.
- Curly brackets should be at the first line of the function, and alone at the end of the function.

```
function setup() {  
  createCanvas(500,500);}
```



```
function setup() {  
    createCanvas(500,500);  
}
```





# First Project: Portrait, due week 3

- Make a portrait of yourself, your friend, your partner, your cat, your sibling! A real person (or living creature in your vicinity), no celebrities or cartoons.
- It doesn't have to be realistic. Use shapes and explore color and form!
- Come to class ready to share on week 3, we'll talk about code and also aesthetics. Upload code to Moodle by midnight after class. Critique in class is part of your grade, so you must attend!
- Full details on Git and Moodle.

Play with color, **shapes**, the order you draw shapes in, merging two shapes together, using alpha channels, try bezier curves, use **points**, experiment with stroke, try a different **style**, be abstract, try a new function, **explore the reference, explore the internet.**