

# Objects in Motion

Using Computers To Tell  
Objects How To Move

Garett Brown

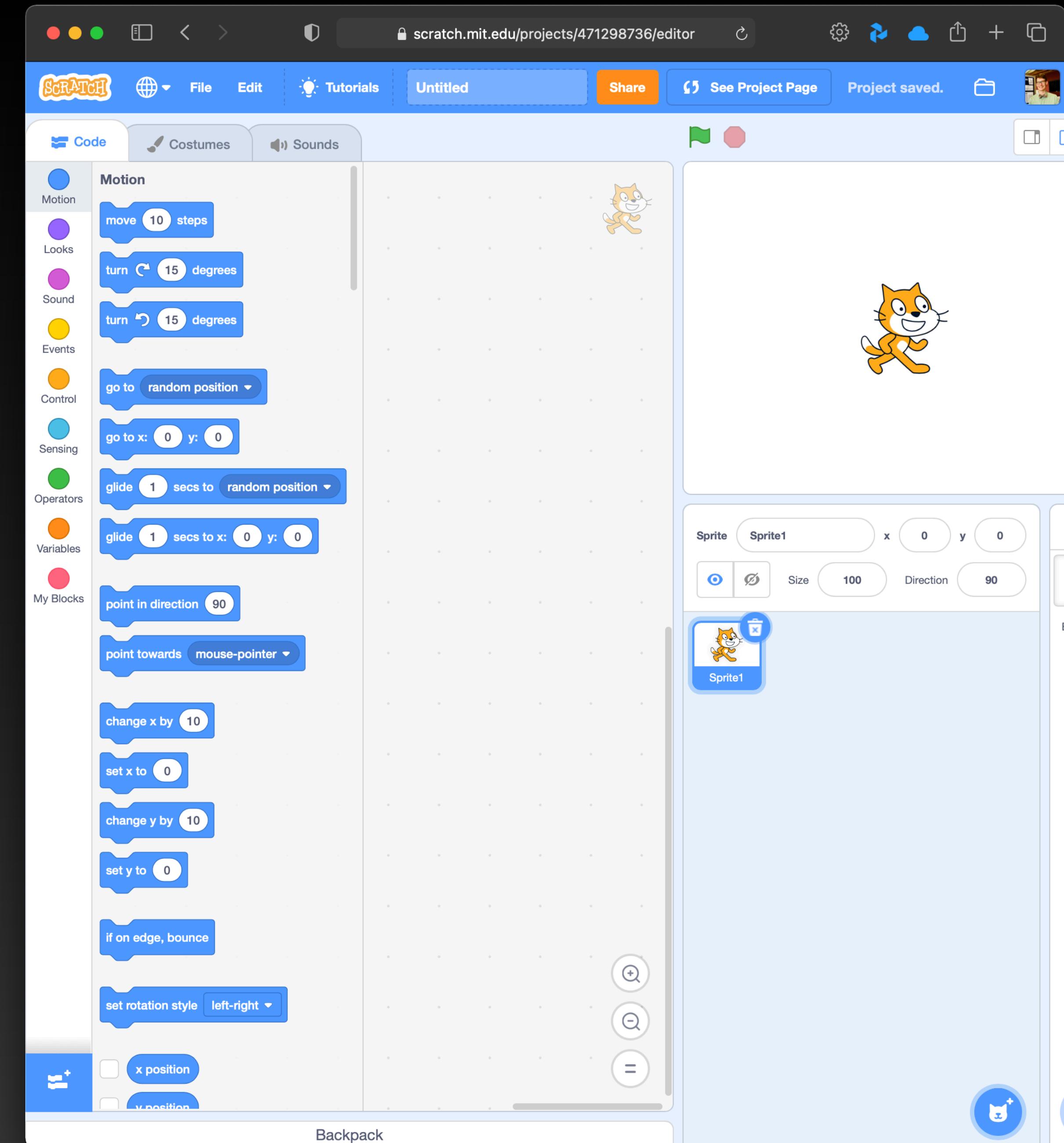
UNIVERSITY OF  
TORONTO



# Motion

## Making Objects Move

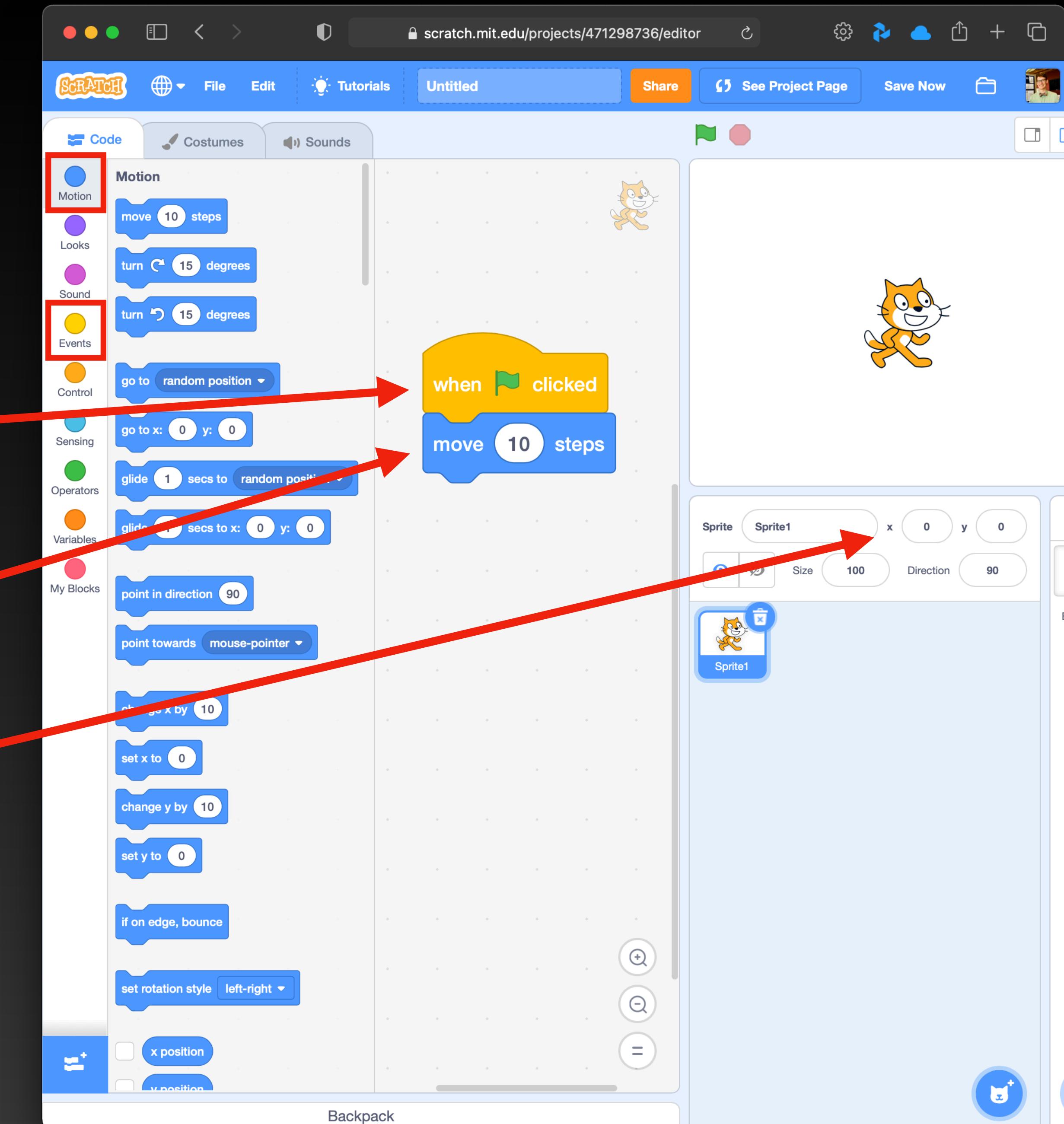
- We can write code to tell objects how to move.
- Let's try it!
- <https://scratch.mit.edu/projects/editor>
- Scratch is a program that can help us visualize the code more easily.



# Motion

## Making Objects Move

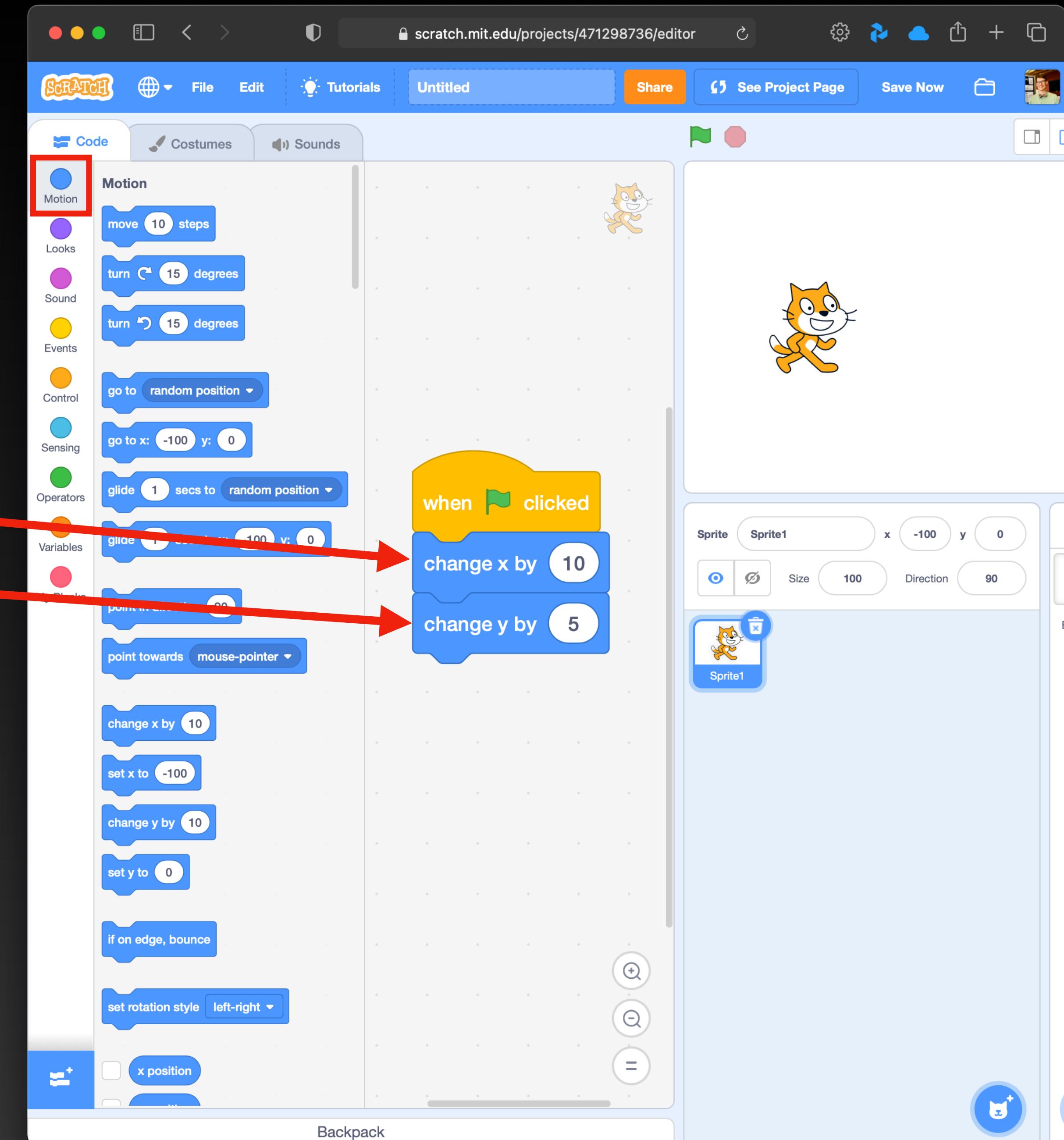
- Use “When  Clicked” to start the movement (from Events).
- Get the cat to move every time you click the green flag.
- Notice how the value for “x” changes as the cat moves.



# Motion

## Making Objects Move

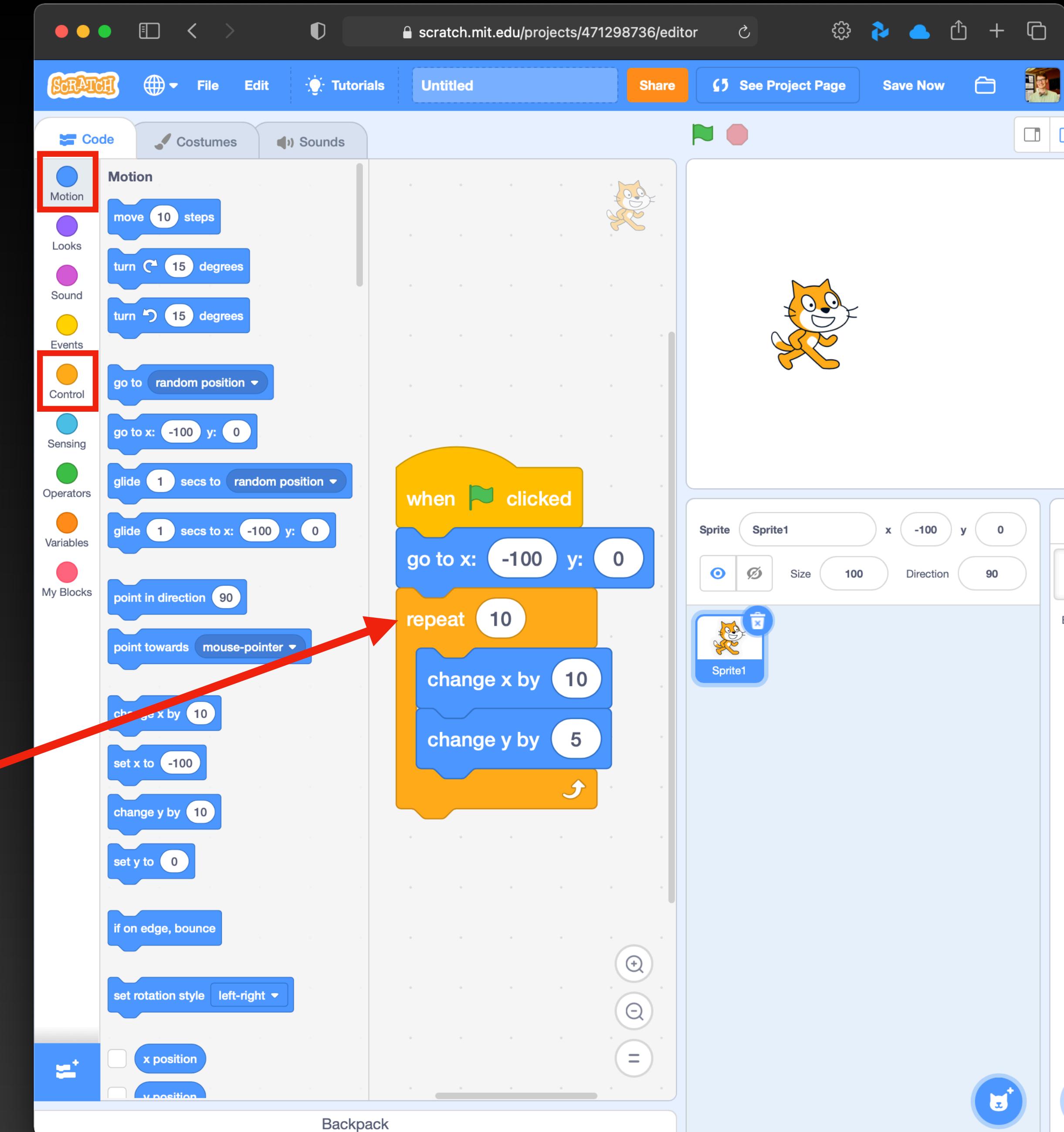
- Let's tell the cat to move 10 steps to the right and 5 steps up every time we click the flag. 
- You might need to click and drag the cat back to a good starting point first in order to have more space to move.



# Motion

## Making Objects Move

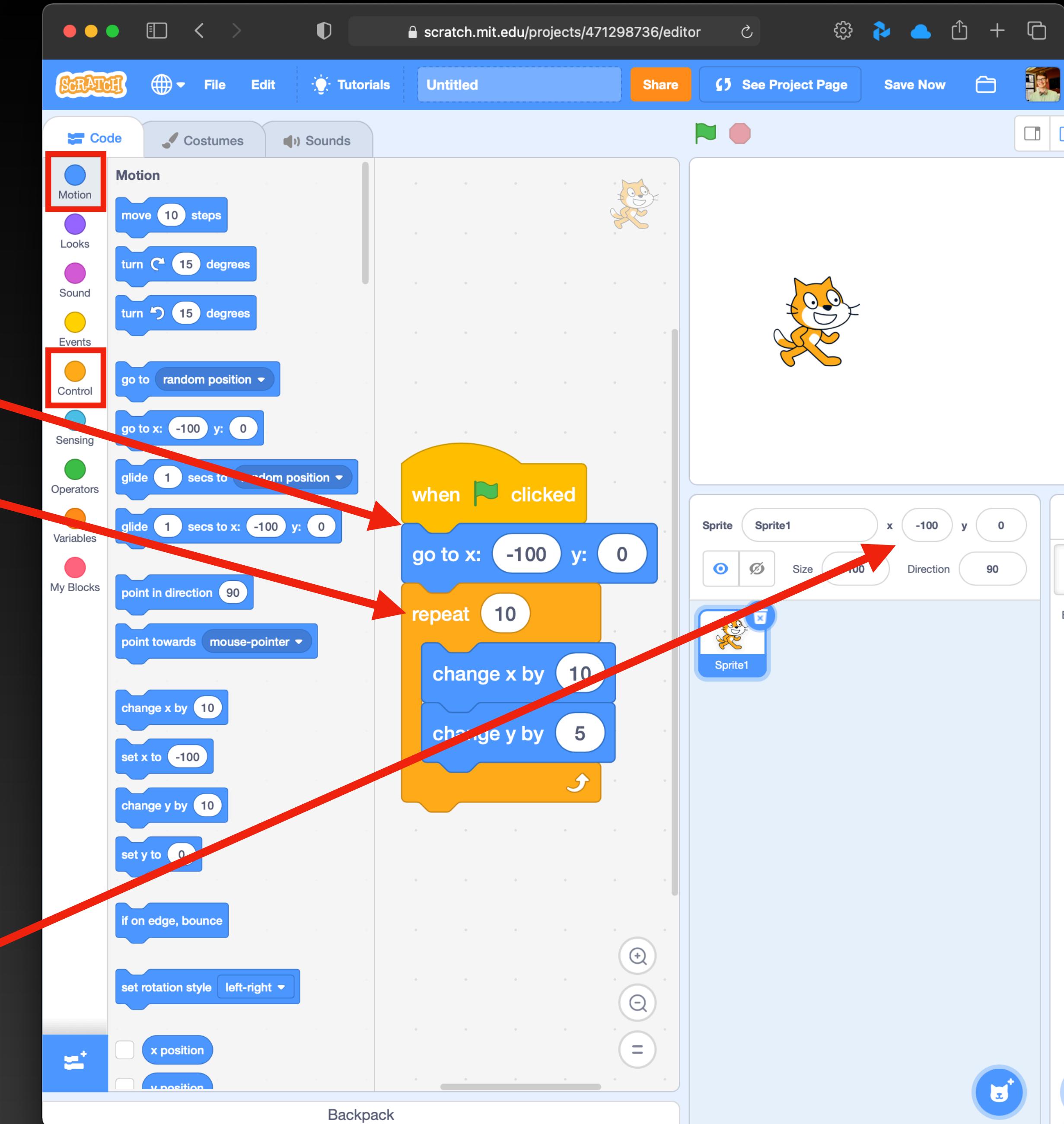
- Now, instead of dragging the cat back to the start and making the cat move by clicking over and over again, let's get the computer to do the work for us.
- For this we're going to use a "loop".
- In computer programming, a "loop" is a way for us to tell the computer to repeat a task over and over again.



# Motion

## Making Objects Move

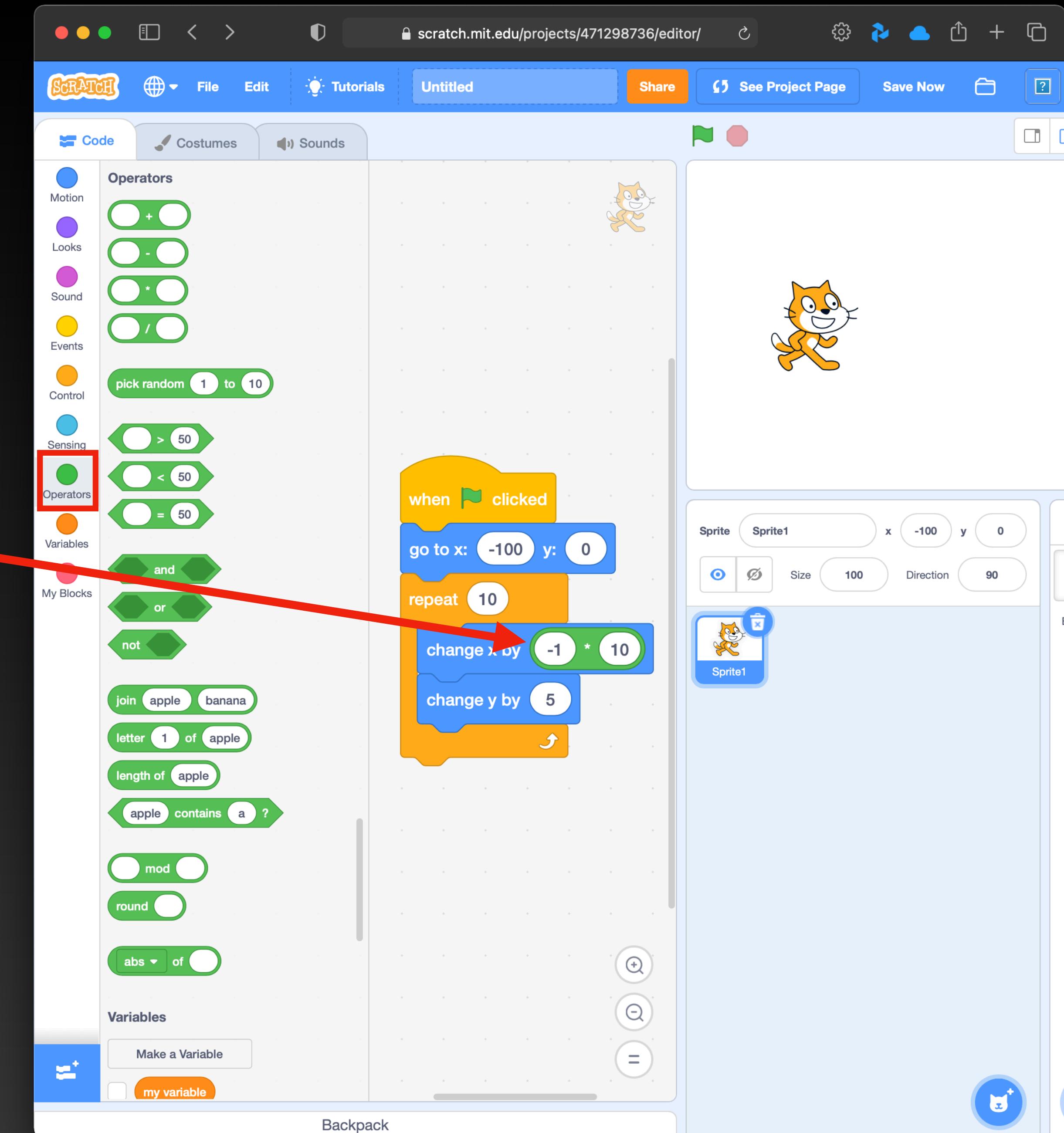
- Tell the computer where to start the cat.
- Then use a repeat loop to move the cat.
- If you repeat the movement for too long the cat will leave the screen. You can bring the cat back by typing in the starting point again here.



# Motion

## Making Objects Move

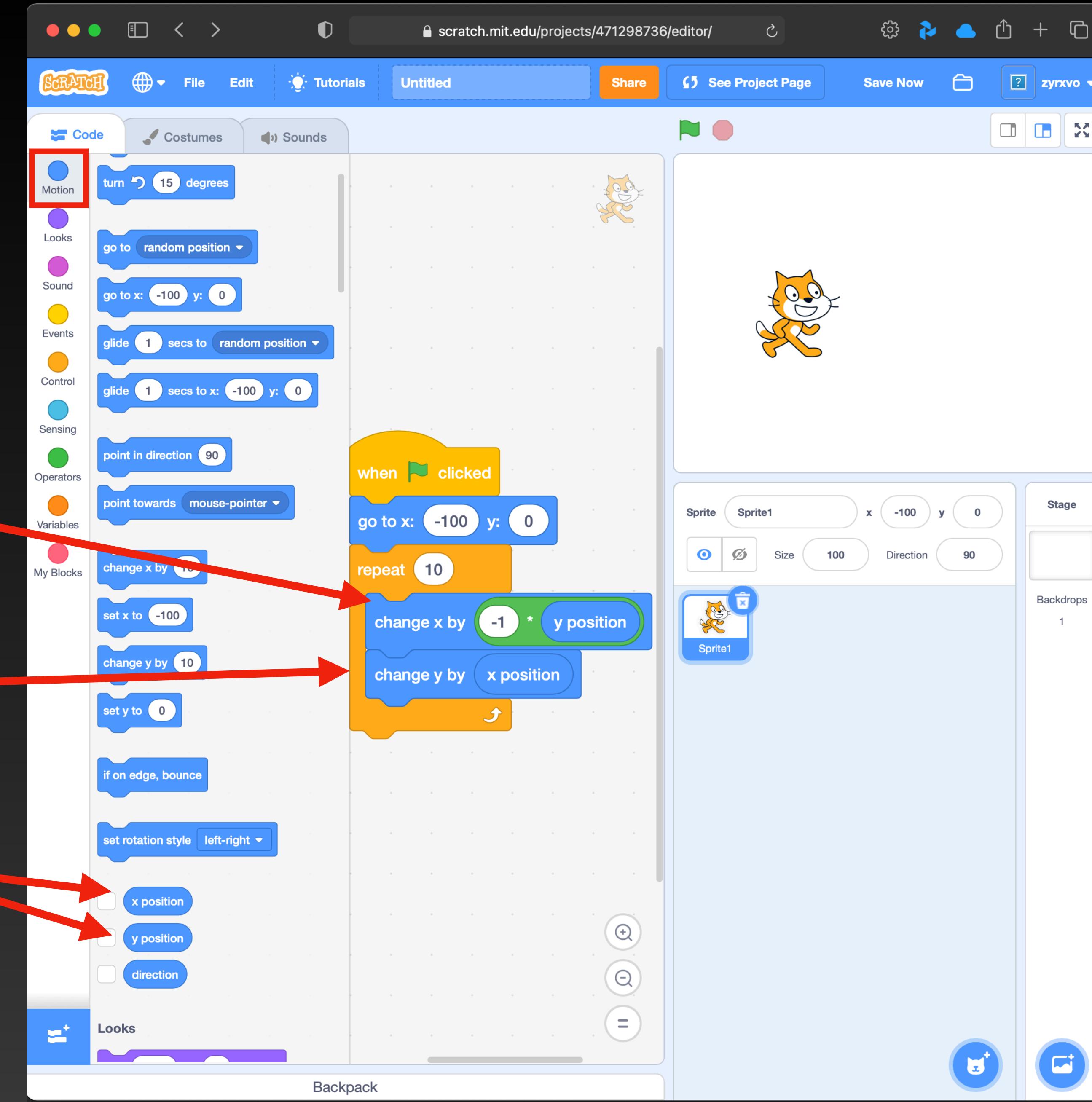
- Now let's add in some math to tell the cat to move backwards instead of forwards.
- We could type in  $-10$  instead of  $+10$ , but by using math we can let the computer do more work for us later.



# Motion

## Making Objects Move

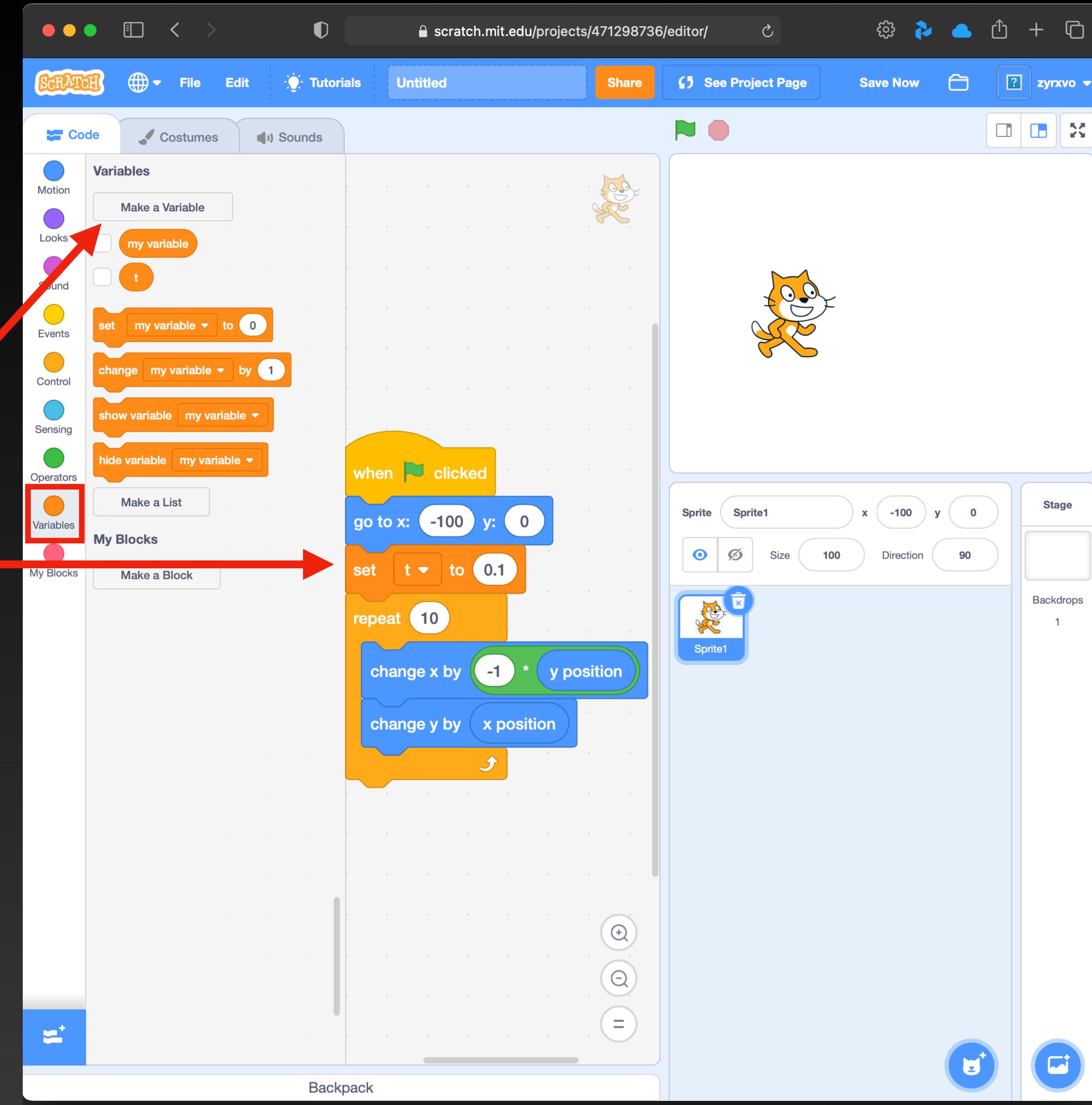
- Modify how the cat moves by changing “x” by the value of  $-1$  times the cat’s “y position”.
- And modify the “change y by” the cat’s “x position”.
- You can get the “x position” and “y position” from bottom of the “Motion” tab.
- What kind of motion do you see?



# Motion

## Making Objects Move

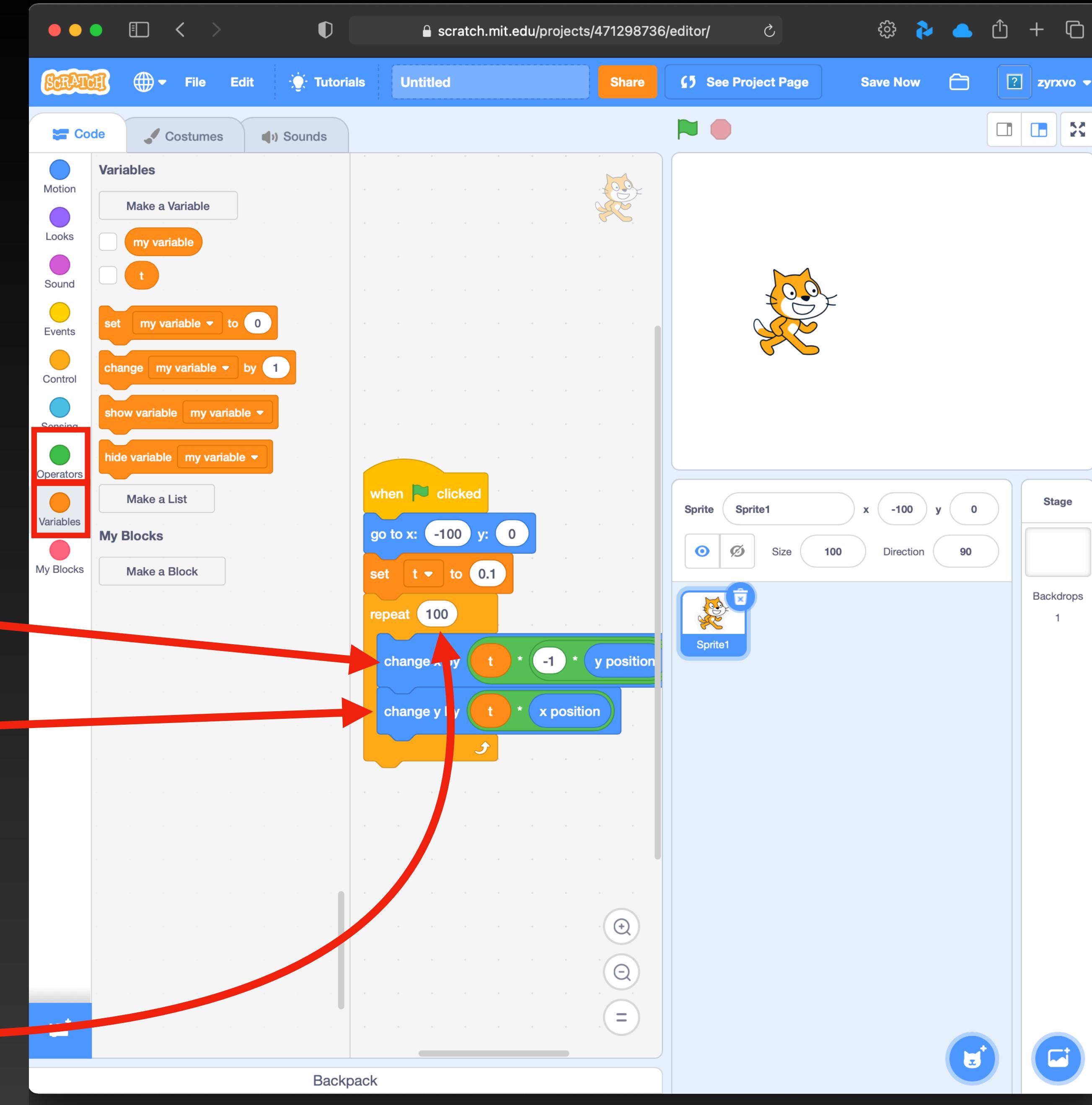
- To make the motion smoother, let's create a new variable called "t".
- Then set the variable t to 0.1
- We can use variables when coding to save information to use later in the program.
- We always need to set a variable before we can use it.



# Motion

## Making Objects Move

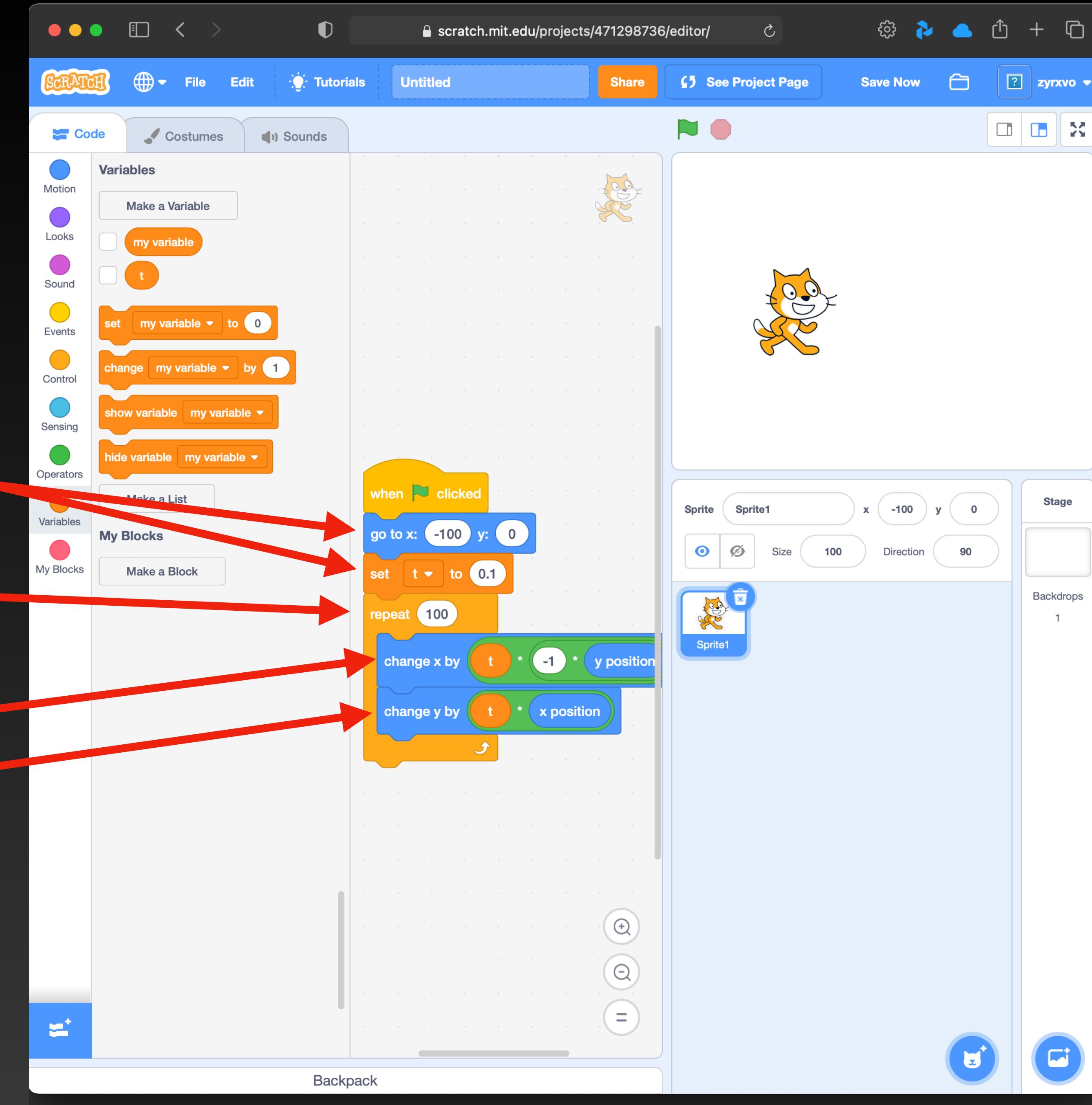
- Use the variable “t” with multiplication.
- Change x by “t” times -1 times y position
- Change y by “t” times x position.
- You may also need to increase the number of times we repeat the movement in order to watch it for longer.



# Motion

## Making Objects Move

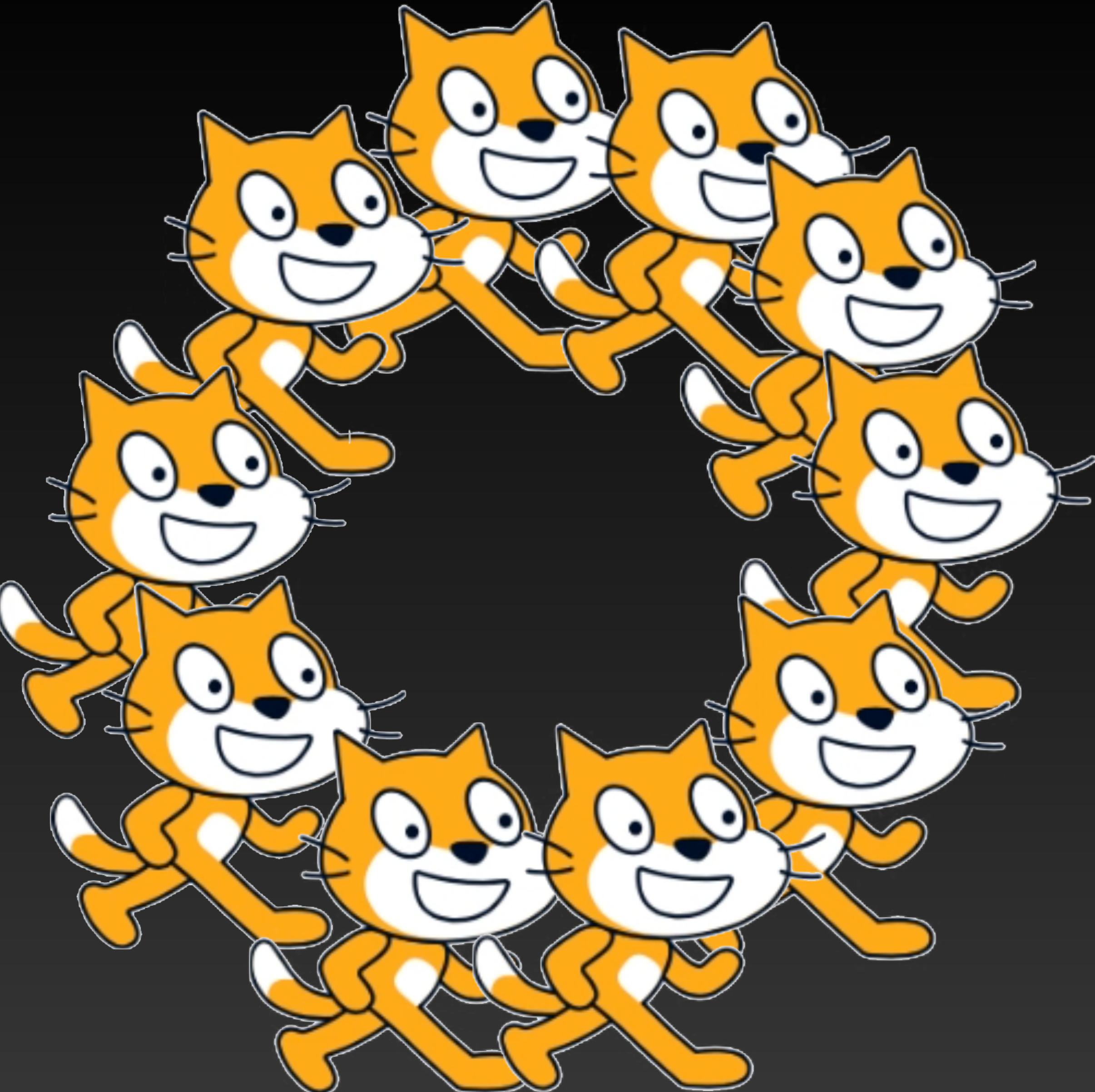
- Let's review our code:
  - We first set our variables x, y, and t,
  - Then we set up a loop,
  - Inside the loop we changed x by  $t * (-1) * y$   
changed y by  $t * x$
  - This is called an algorithm.



# Motion

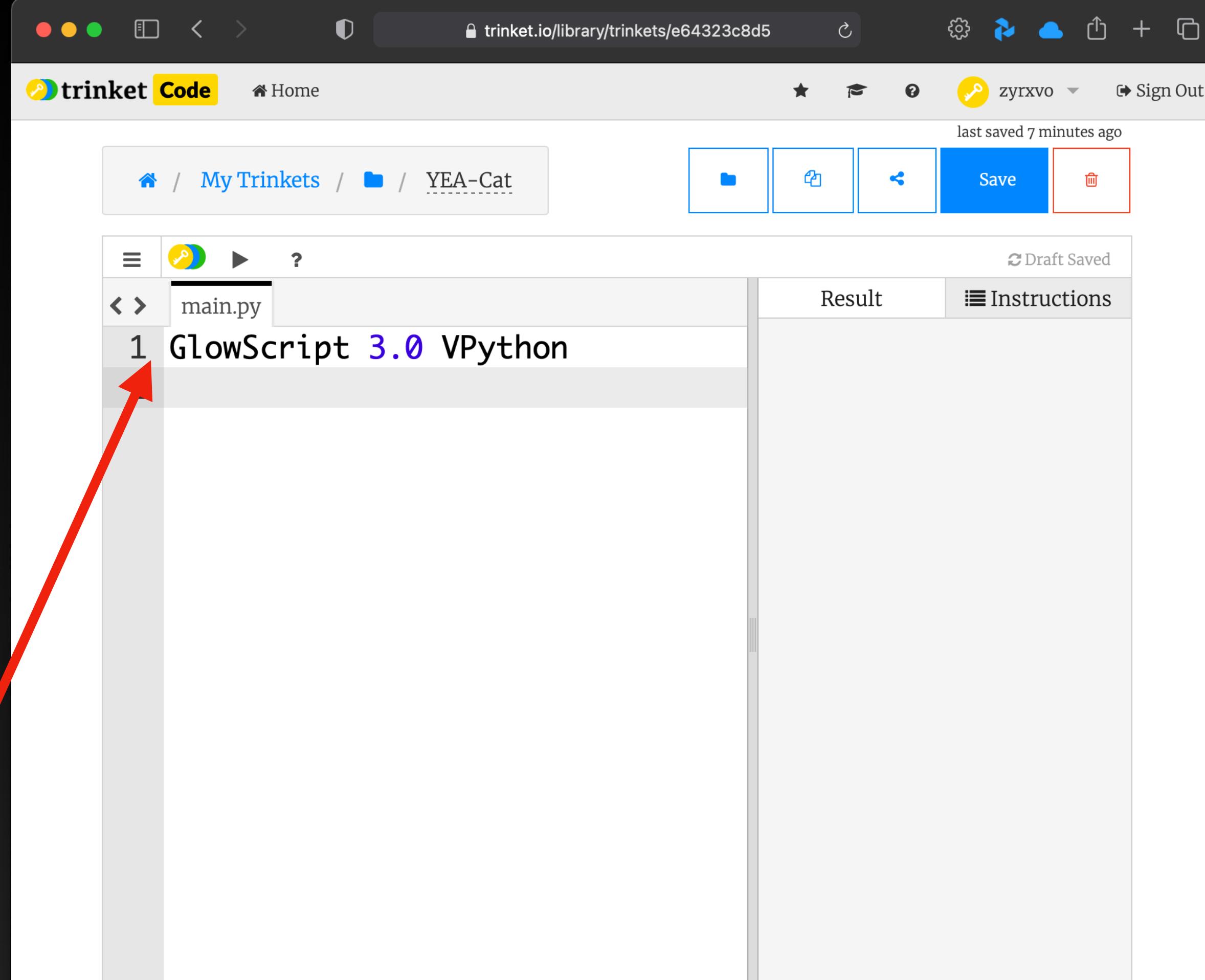
## Making Objects Move

- This algorithm describes the motion of an object as it moves in a circle.
- Now let's move this algorithm to the programming language Python.
- Let's try it!
- <https://trinket.io/library/trinkets/e64323c8d5>
- Trinket lets us run Python code from anywhere.



# Motion with Python

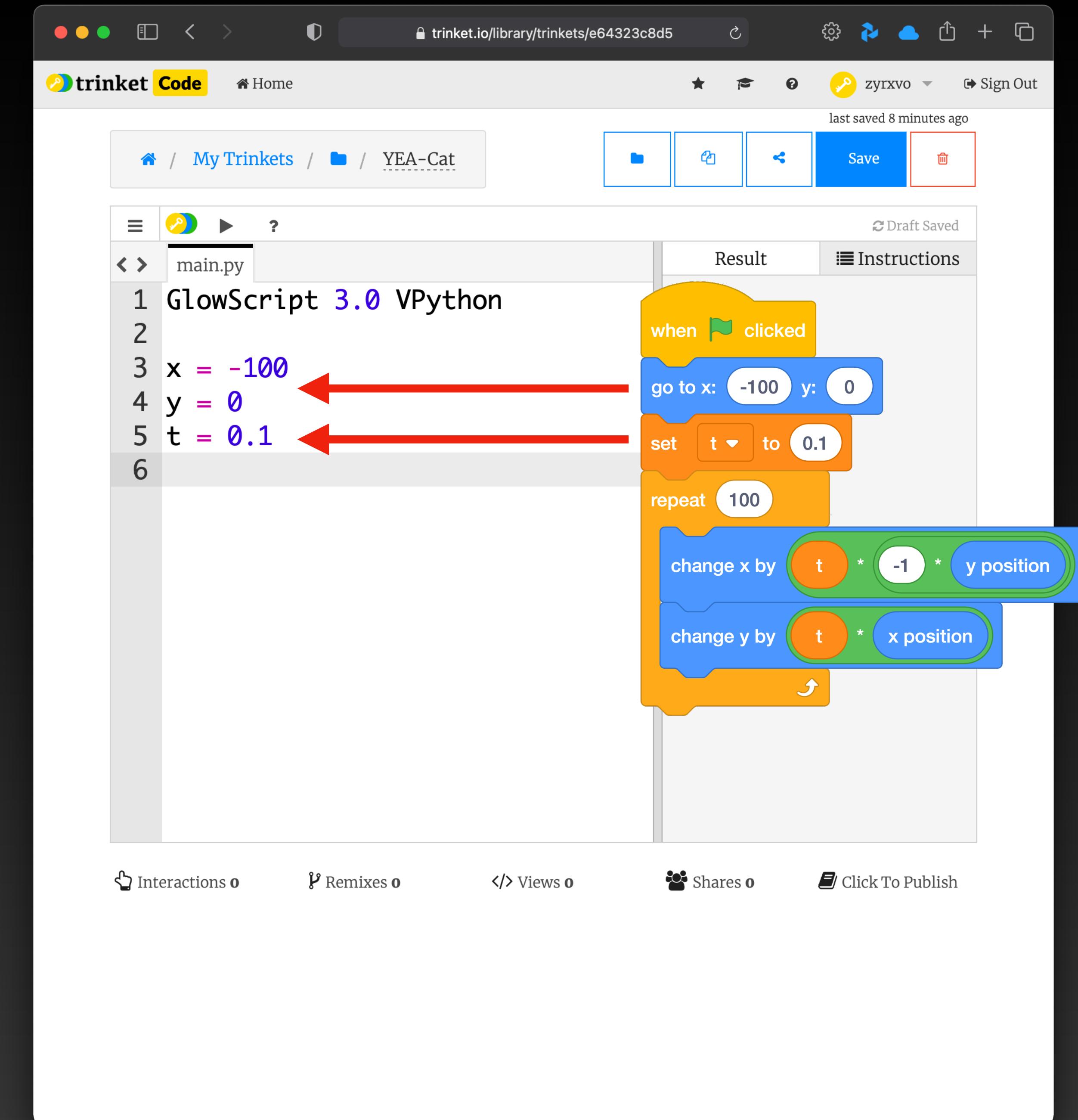
- This is what the Trinket Python editor looks like.
- In order to write Python code, we will need to type out the variable names and commands ourselves.
- Ignore the first line for now, but don't delete it!
- Let's do this step by step.



The screenshot shows the Trinket Code editor interface. The URL in the address bar is [trinket.io/library/trinkets/e64323c8d5](https://trinket.io/library/trinkets/e64323c8d5). The page title is "trinket Code". The file path is "My Trinkets / YEA-Cat / main.py". The code editor shows the first line: "1 GlowScript 3.0 VPython". A red arrow points to this line. The interface includes a toolbar with file, copy, paste, and save buttons, and tabs for "Result" and "Instructions". Below the editor are social sharing and publishing metrics: 0 interactions, 0 remixes, 0 views, 0 shares, and a "Click To Publish" button.

# Motion with Python

- We can create our variables  $x$ ,  $y$ , and  $t$  by simple typing their names.
- In order to set the variable to a specific value we use the “ $=$ ”.
- In Python, this is called “assigning”.  
Meaning we assign  $y$  to be equal to 0.

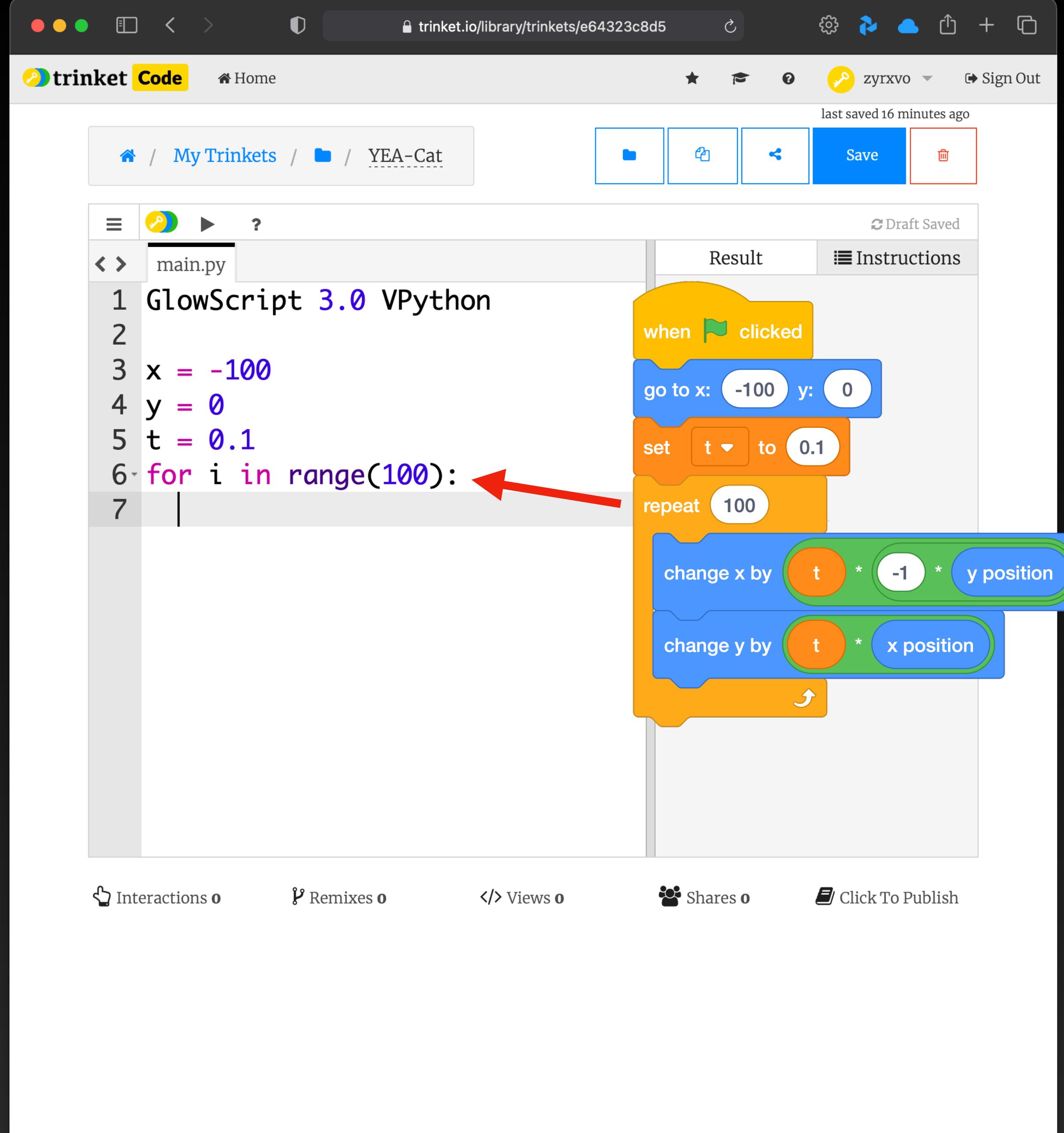


The screenshot shows the trinket.io Code editor with the following details:

- Code Editor:** The file is named `main.py` and is written in GlowScript 3.0 VPython. The code initializes variables  $x = -100$ ,  $y = 0$ , and  $t = 0.1$ .
- Scratch Script:** The Scratch script is triggered by a green flag click. It sets the stage background to "cat" and starts a repeat loop of 100 iterations. Inside the loop, the script moves the cat to  $x: -100$  and  $y: 0$ , then sets the variable  $t$  to 0.1, and finally changes the cat's x and y position by  $t * -1 * y$  and  $t * x$  respectively.
- Annotations:** Two red arrows point from the code lines `x = -100` and `t = 0.1` to the corresponding variable assignments in the Scratch script.

# Motion with Python

- In Python, if we want to repeat something a certain number of times, the easiest way to do it is by using a “FOR” loop.
- In English, line 6 says, “Create a list of 100 numbers and for each repetition of the loop assign  $i$  to the next number.”
- Because the range of numbers is 100, the loop will repeat 100 times.



The screenshot shows the trinket.io Code editor with a Python script named `main.py`. The code initializes variables `x`, `y`, and `t`, and then enters a `for` loop that repeats 100 times. Inside the loop, the script moves the cat to `x = -100` and `y = 0`, sets the step size to `t`, and then uses a `repeat 100` block to move the cat in a zigzag pattern. A red arrow points to the `for` loop line in the code editor.

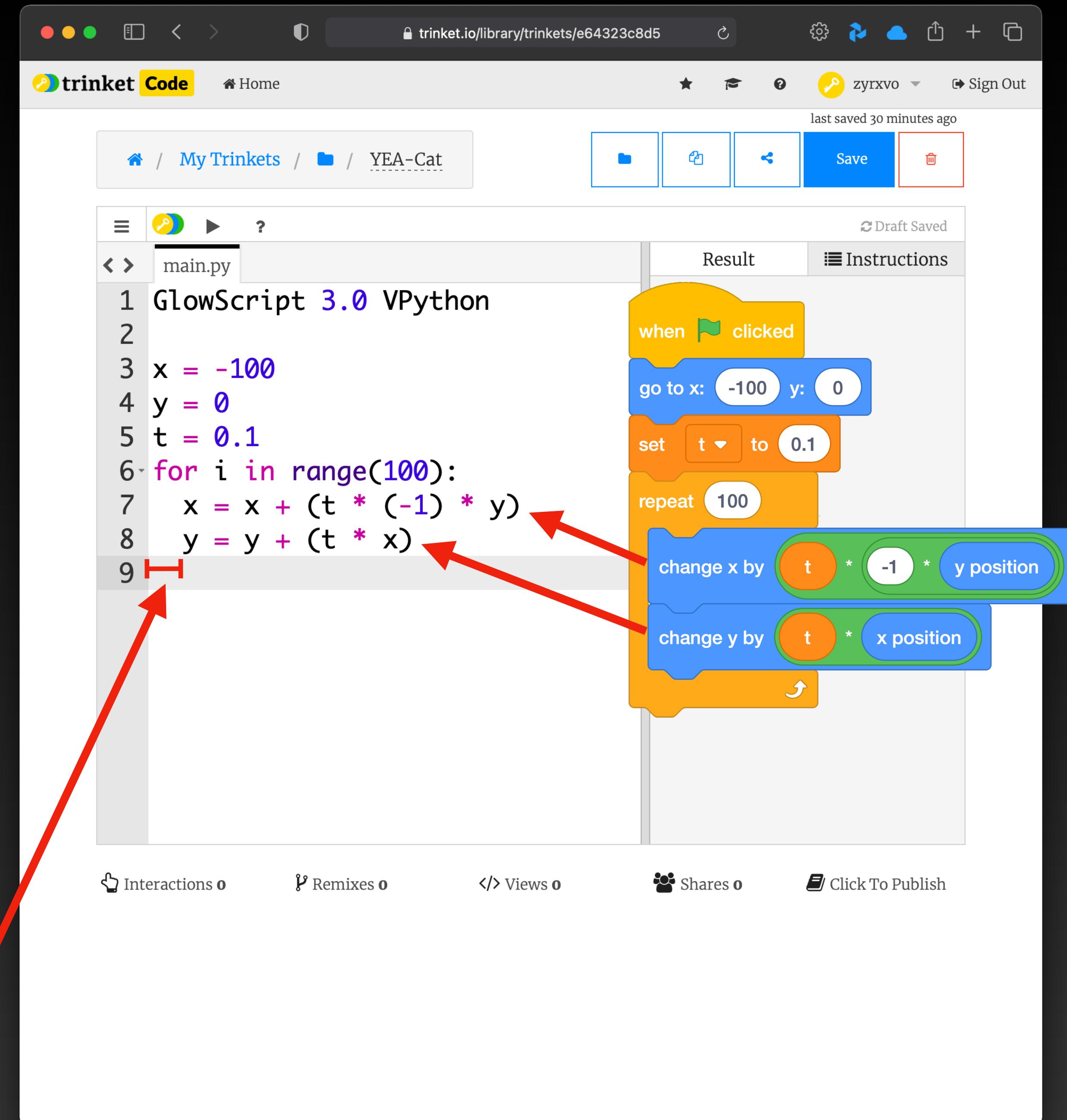
```
1 GlowScript 3.0 VPython
2
3 x = -100
4 y = 0
5 t = 0.1
6 for i in range(100):
7 |
```

The Scratch blocks on the right correspond to the Python code:

- A `when green flag clicked` hat block.
- An `go to x: -100 y: 0` movement block.
- An `set t to [0.1]` control block.
- An `repeat (100)` control block.
- Inside the repeat loop:
  - An `change x by (t * -1 * y position)` movement block.
  - An `change y by (t * x position)` movement block.

# Motion with Python

- In order to change  $x$  and  $y$  over and over again, we need to add a value to them and then assign that result to the variable again.
- So if  $y = 0$ , and  $t \cdot x = -10$ , then  $y + (t \cdot x) = -10$ .
- But if we don't assign the result  $-10$  to  $y$ , then  $y$  will never change.
- Also notice the indented space inside the loop, this is necessary for Python.

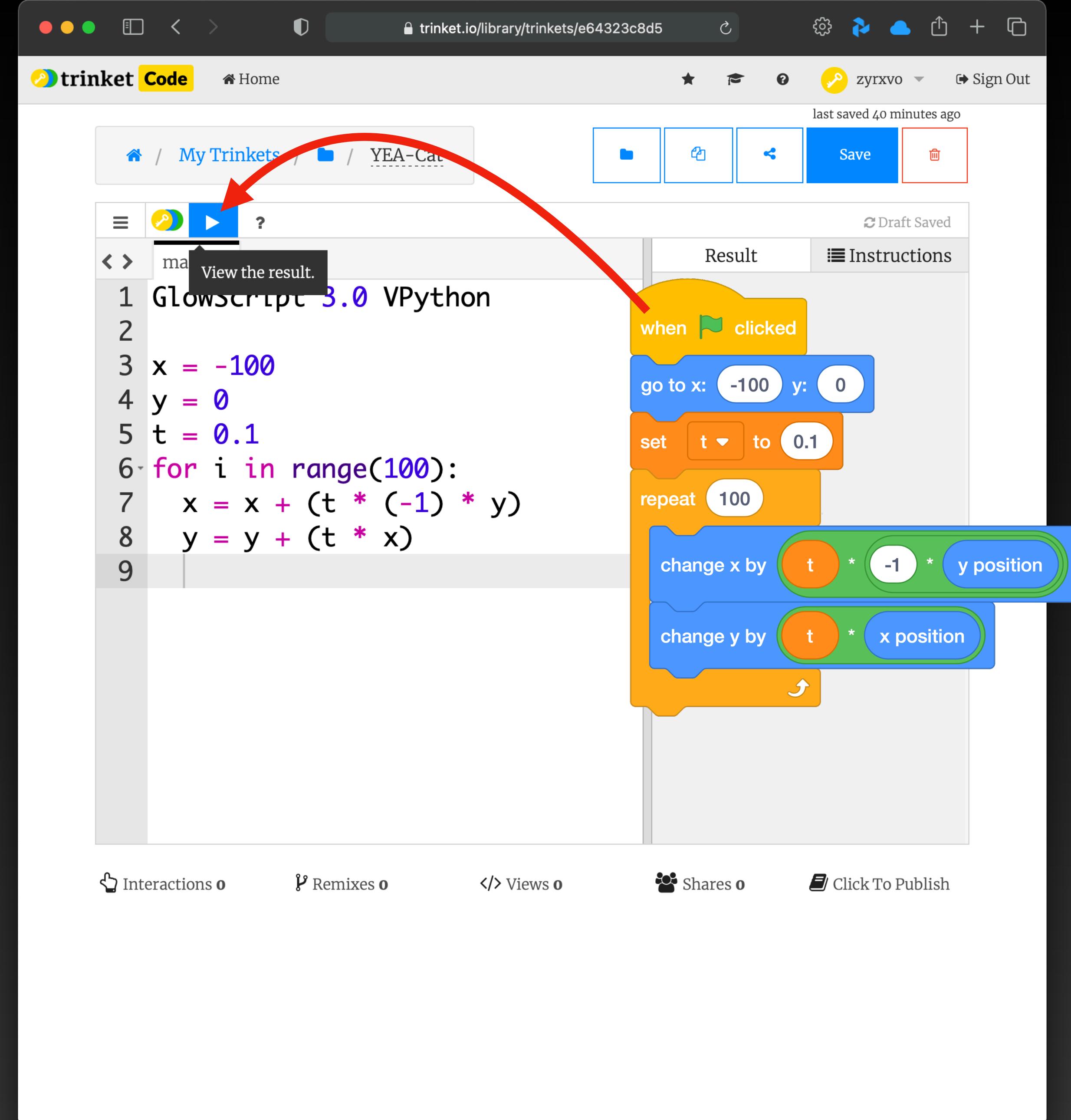


The screenshot shows the trinket.io Code editor with a Python script named `main.py`. The code uses the `GlowScript 3.0 VPython` module. It initializes variables `x = -100` and `y = 0`, sets a time step `t = 0.1`, and then enters a `for` loop that runs 100 times. Inside the loop, it calculates the next position for `x` and `y` using the formula  $x = x + (t * (-1) * y)$  and  $y = y + (t * x)$ . The Scratch3.0 blocks on the right side of the editor show the equivalent logic: a `when green flag clicked` hat, a `go to x: -100 y: 0` control, a `set t to 0.1` control, a `repeat (100)` control, and two nested `change y by` and `change x by` loops. Red arrows point from the code line `x = x + (t * (-1) * y)` to the Scratch `change y by` block and from the code line `y = y + (t * x)` to the Scratch `change x by` block.

```
1 GlowScript 3.0 VPython
2
3 x = -100
4 y = 0
5 t = 0.1
6 for i in range(100):
7     x = x + (t * (-1) * y)
8     y = y + (t * x)
9
```

# Motion with Python

- With our algorithm now written in Python we can run our code by using the ► button at the top.
- What do you see?



The screenshot shows the trinket.io Code interface. At the top, there's a navigation bar with icons for settings, refresh, cloud, and file operations. The URL is [trinket.io/library/trinkets/e64323c8d5](https://trinket.io/library/trinkets/e64323c8d5). On the right, there are buttons for saving, sharing, and deleting the trinket. The title bar says "trinket Code" and "Home". The page is titled "My Trinkets / YEA-Cat". A red arrow points from the "Run" button (a play icon) in the code editor to a tooltip that says "View the result." The code editor shows Python code for a "main" function:

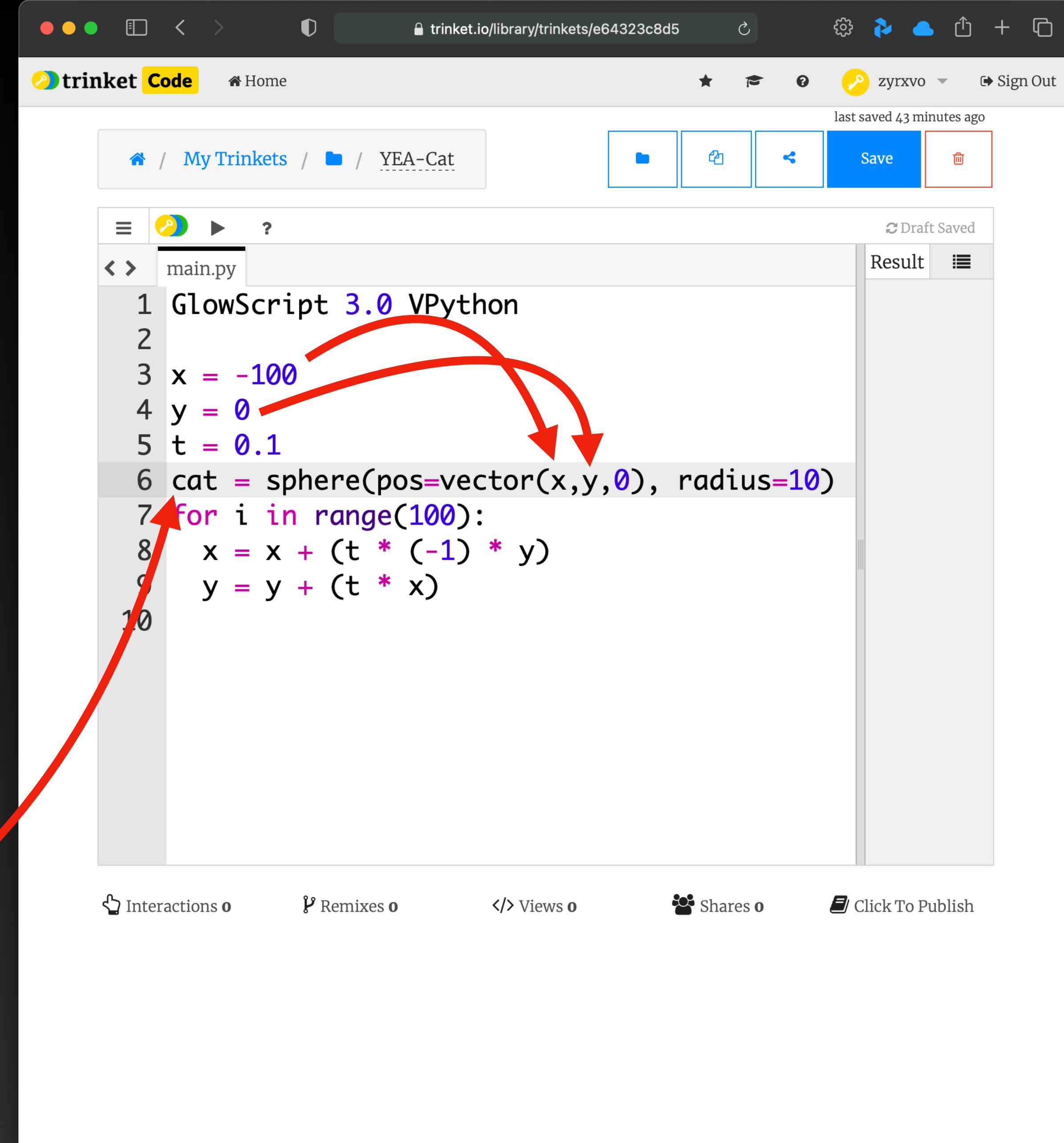
```
1 GlowScript 3.0 VPython
2
3 x = -100
4 y = 0
5 t = 0.1
6 for i in range(100):
7     x = x + (t * (-1) * y)
8     y = y + (t * x)
9
```

On the right, the Scratch script is displayed, corresponding to the Python code. It starts with a "when green flag clicked" hat block. Inside the script, there is a "repeat (100)" control block. Inside the repeat loop, there are two "change" blocks: "change x by (t \* (-1) \* y position)" and "change y by (t \* x position)". The "t" variable is set to 0.1. The Scratch script is color-coded: yellow for the hat, blue for the repeat, orange for the control, green for the change, and blue for the parameters.

Below the code editor, there are social sharing icons for Interactions, Remixes, Views, Shares, and a "Click To Publish" button.

# Motion with Python

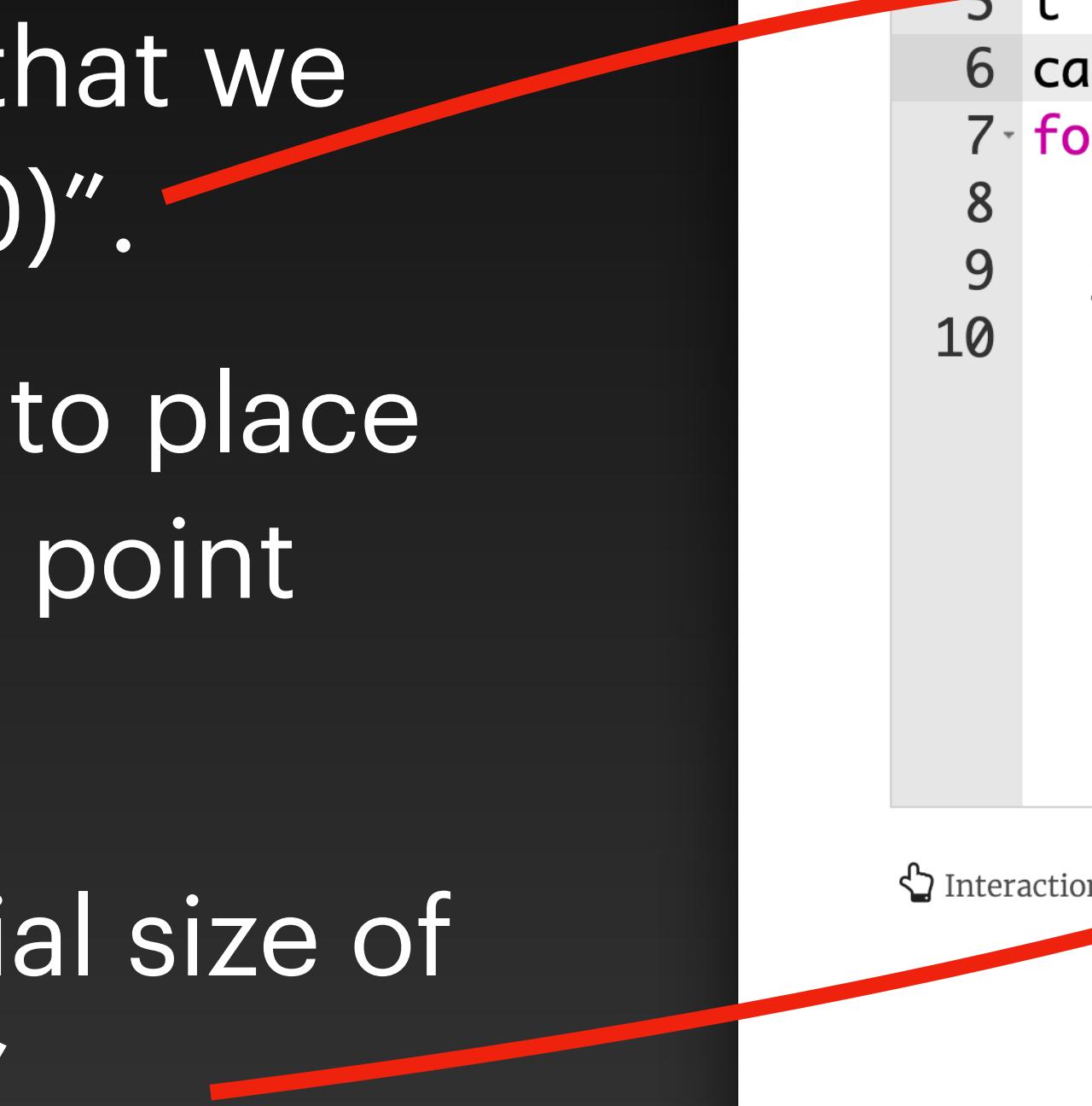
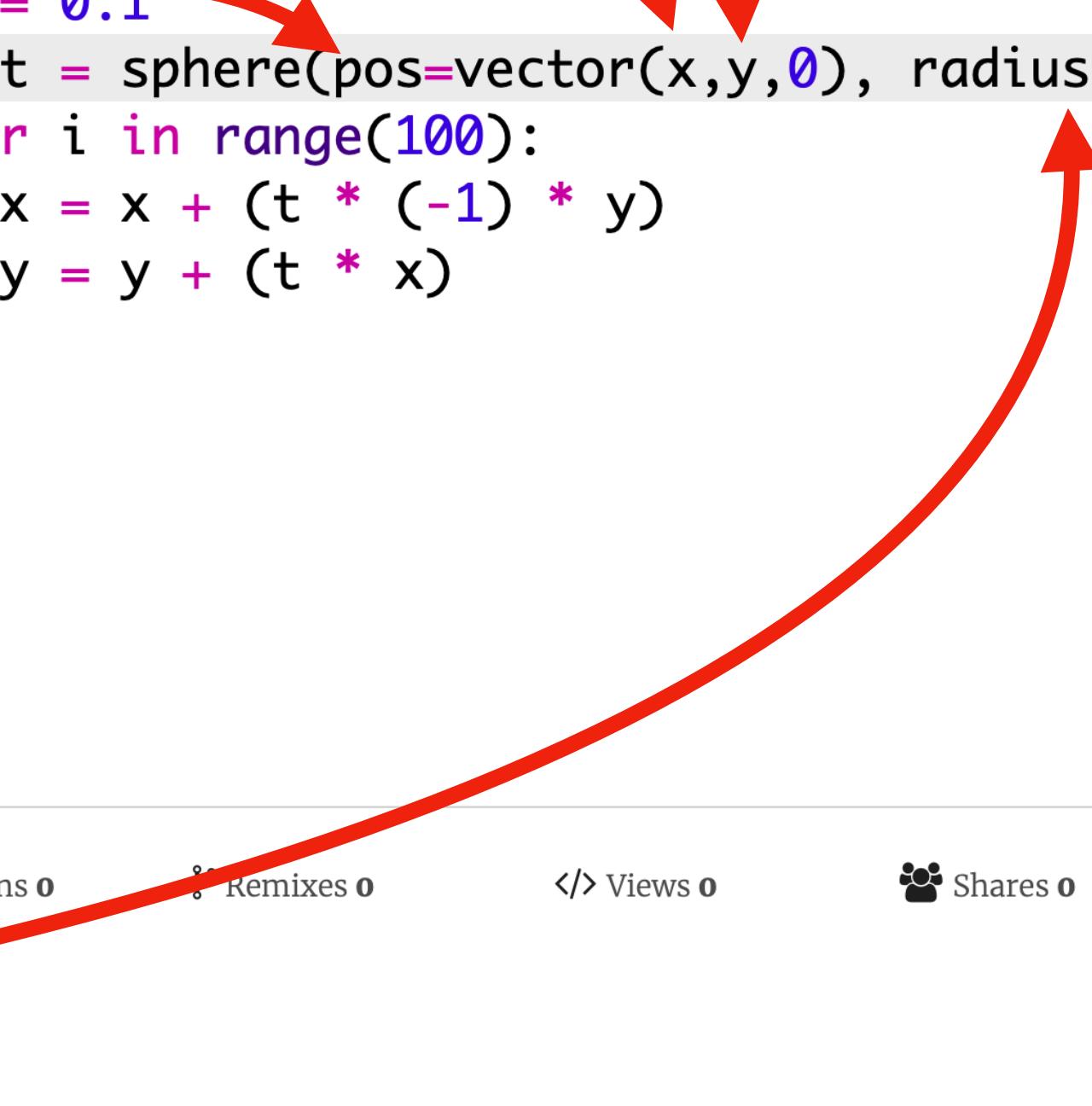
- We don't see anything because we haven't told Python what to draw.
- Scratch draws and moves the Cat for us, so now we need to tell the computer how to do it.
- Start by creating a new variable called "cat" and assigning a "sphere" object to it.

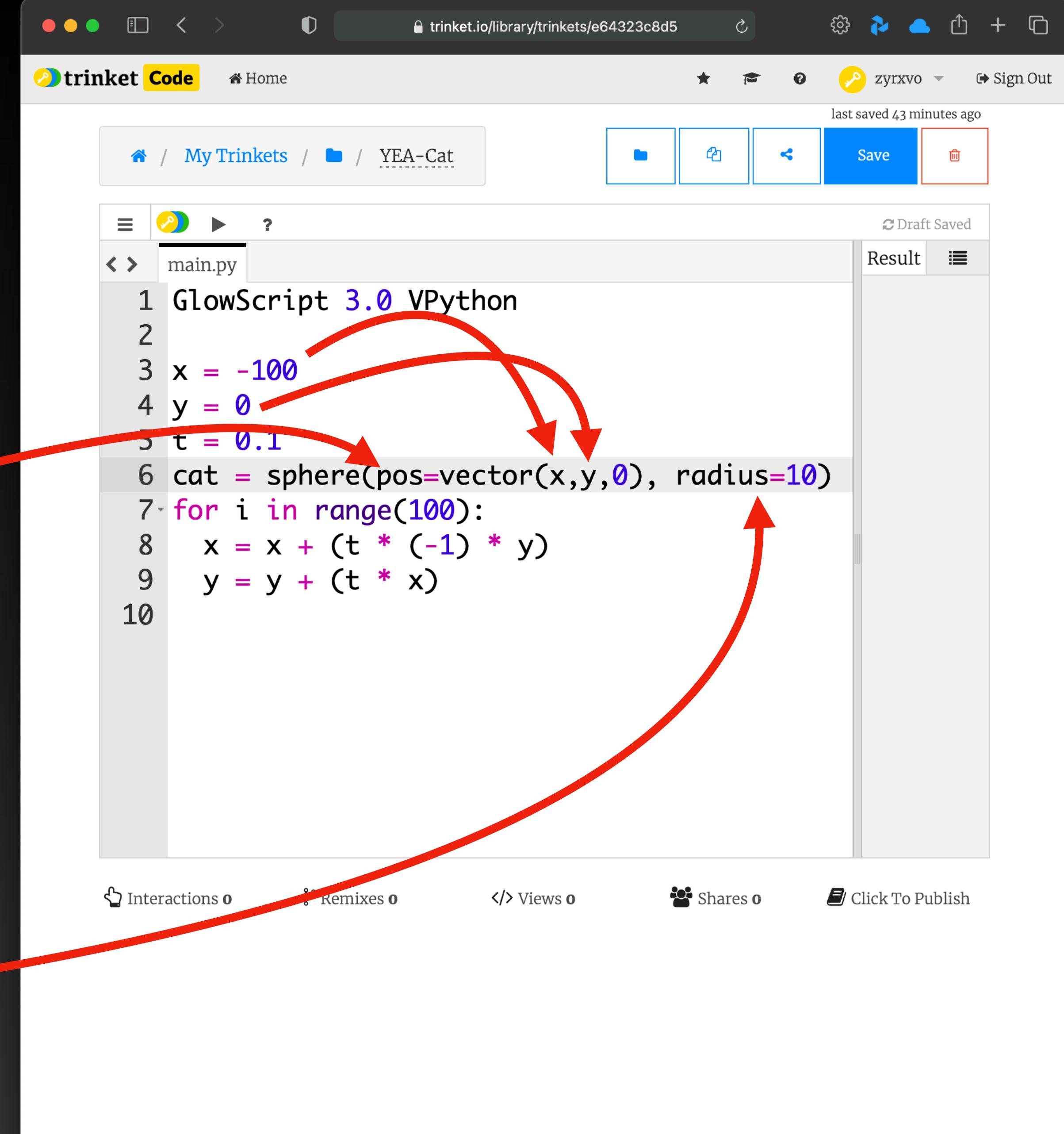


The screenshot shows a code editor on the trinket.io platform. The file is named 'main.py' and is written in GlowScript 3.0 VPython. The code initializes variables x, y, and t, and then creates a sphere at position (x, y, 0) with a radius of 10. A for loop iterates 100 times, updating the position of the sphere based on a complex motion equation involving x and y.

```
1 GlowScript 3.0 VPython
2
3 x = -100
4 y = 0
5 t = 0.1
6 cat = sphere(pos=vector(x,y,0), radius=10)
7 for i in range(100):
8     x = x + (t * (-1) * y)
9     y = y + (t * x)
```

# Motion with Python

- We need set the initial position “pos” of the cat. To do that we write “pos = vector(x,y,0)”. 
- This tells the computer to place our “cat” at our starting point defined by  $x$  and  $y$ .
- We can also set the initial size of the cat with “radius=10”. 



The screenshot shows a code editor on trinket.io with the following code:

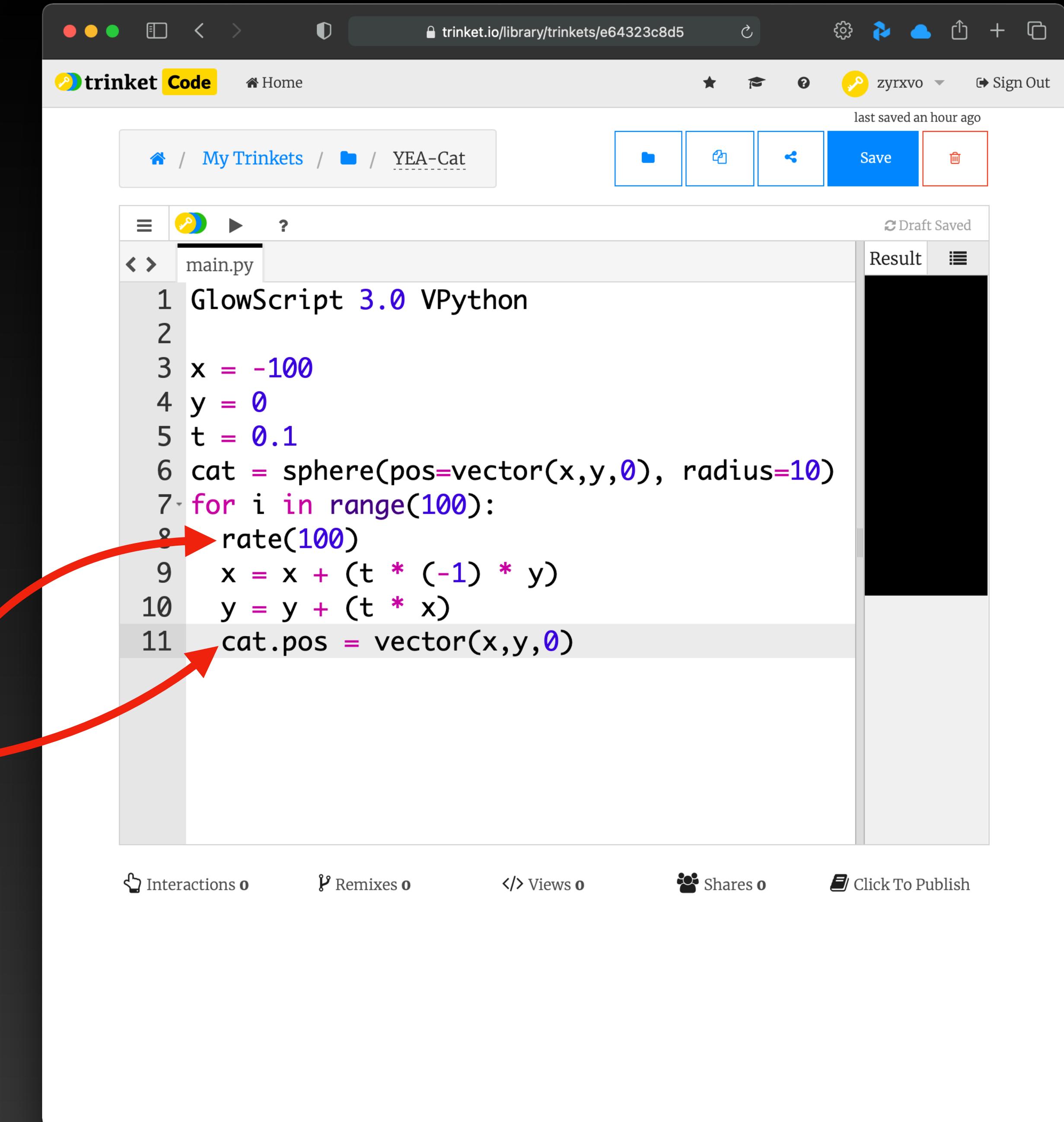
```
1 GlowScript 3.0 VPython
2
3 x = -100
4 y = 0
5 t = 0.1
6 cat = sphere(pos=vector(x,y,0), radius=10)
7 for i in range(100):
8     x = x + (t * (-1) * y)
9     y = y + (t * x)
10
```

Annotations with red arrows highlight the following parts of the code:

- An arrow points from the line `x = -100` to the `x` variable in the `pos=vector(x,y,0)` line.
- An arrow points from the `radius=10` value to the `radius=10` line.
- An arrow points from the `range(100)` in the `for` loop to the `i` variable in the `range` function.

# Motion with Python

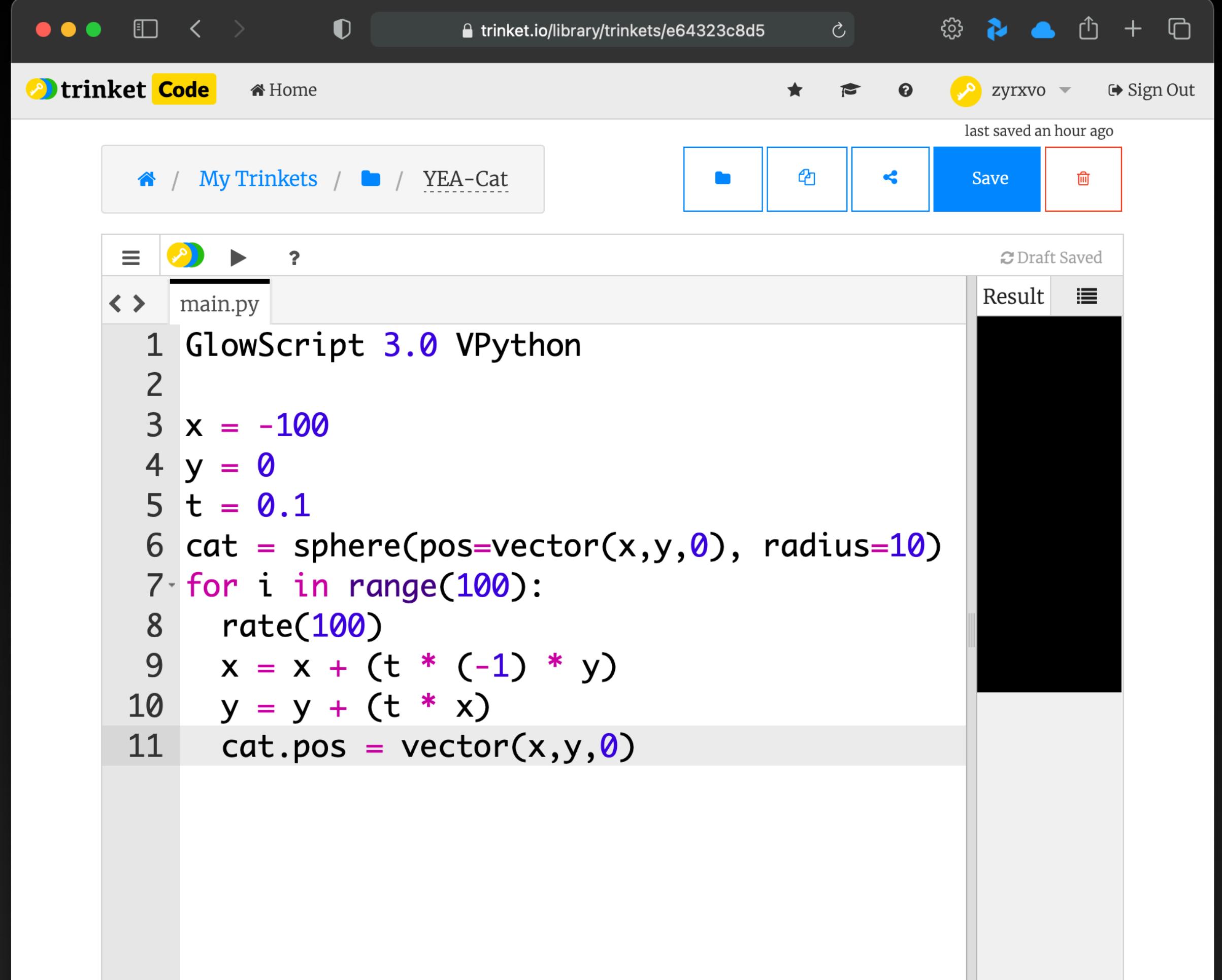
- Finally, we can update the  $x$  and  $y$  position of the “cat” every time we repeat the loop.
- We do this by writing “`cat.pos = vector(x,y,0)`”.
- Last of all, we need to add in a line that says “`rate(100)`” to tell the computer to run the code slow enough for us to watch.



```
1 GlowScript 3.0 VPython
2
3 x = -100
4 y = 0
5 t = 0.1
6 cat = sphere(pos=vector(x,y,0), radius=10)
7 for i in range(100):
8     rate(100)
9     x = x + (t * (-1) * y)
10    y = y + (t * x)
11    cat.pos = vector(x,y,0)
```

# Motion with Python

- The final code should look something like this 
- Be careful of any spelling mistakes.
- If your code is not working, you can go to the solution here:  
<https://trinket.io/library/trinkets/a2e31a1688>



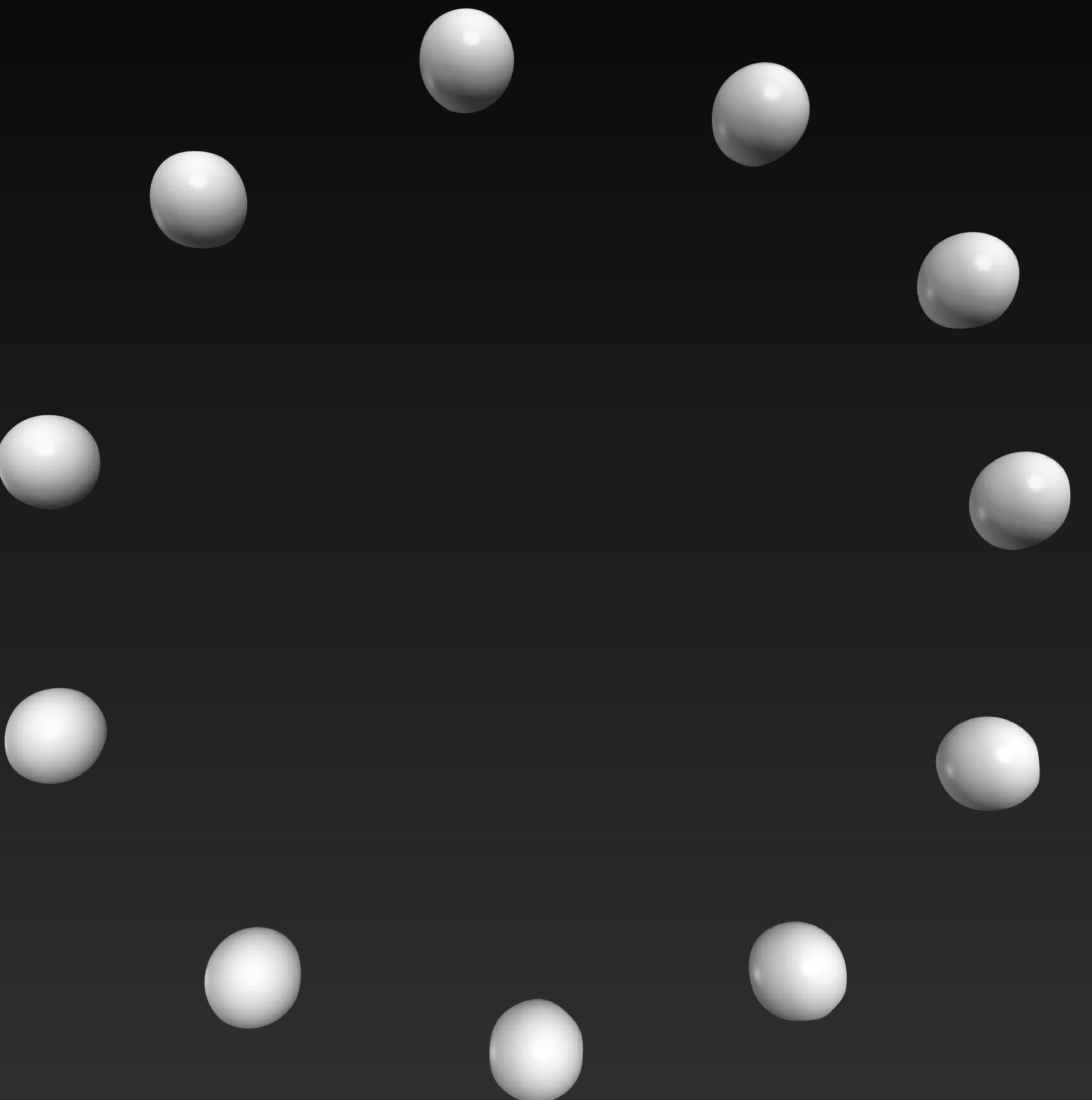
The screenshot shows a browser-based code editor for the trinket.io platform. The URL in the address bar is [trinket.io/library/trinkets/e64323c8d5](https://trinket.io/library/trinkets/e64323c8d5). The page title is "trinket Code". The code editor displays a file named "main.py" with the following content:

```
1 GlowScript 3.0 VPython
2
3 x = -100
4 y = 0
5 t = 0.1
6 cat = sphere(pos=vector(x,y,0), radius=10)
7 for i in range(100):
8     rate(100)
9     x = x + (t * (-1) * y)
10    y = y + (t * x)
11    cat.pos = vector(x,y,0)
```

The editor interface includes a toolbar with icons for file operations, a "Save" button, and a "Draft Saved" message. On the right, there is a "Result" panel which is currently blank. Below the code editor, there are social sharing and interaction statistics: 0 interactions, 0 remixes, 0 views, 0 shares, and a "Click To Publish" button.

# Motion with Python

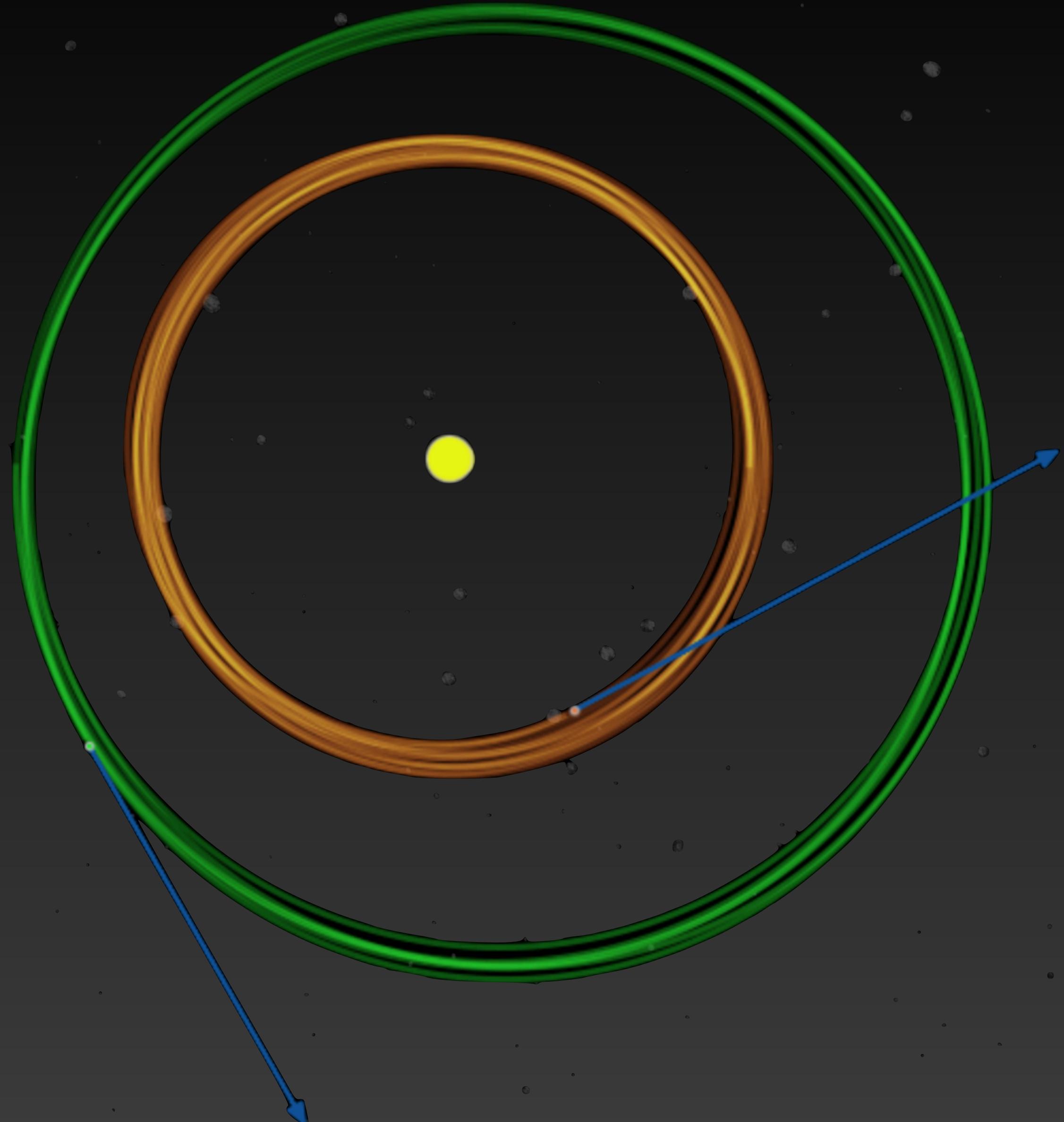
- You did it!
- It doesn't look like a cat, but you wrote your own code that described the motion of an object as it moved in a circle!
- Play around with your code by changing some things.



# Motion

## Planets

- Computer code can be used to describe the motion of many different things.
- Let's play around with code that other people have written and explore how planets and stars move because of gravity.
- <http://labs.minutelabs.io/Chaotic-Planets/>



Thank you  
for joining me today!