

Scalable Snowmen

Hack Club

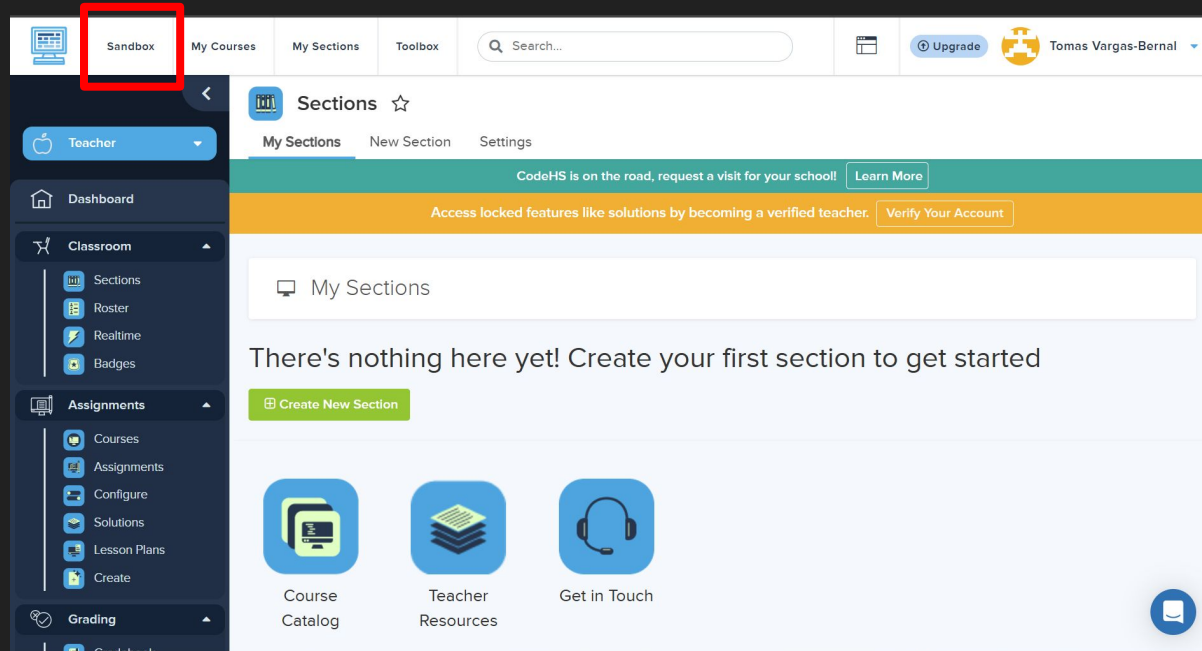
<https://codehs.com/sandbox/id/new-sandbox-program-ByCbmo>

Scalable Snowmen

Getting Started:

Open up CodeHS & Go to Sandbox

codehs.com

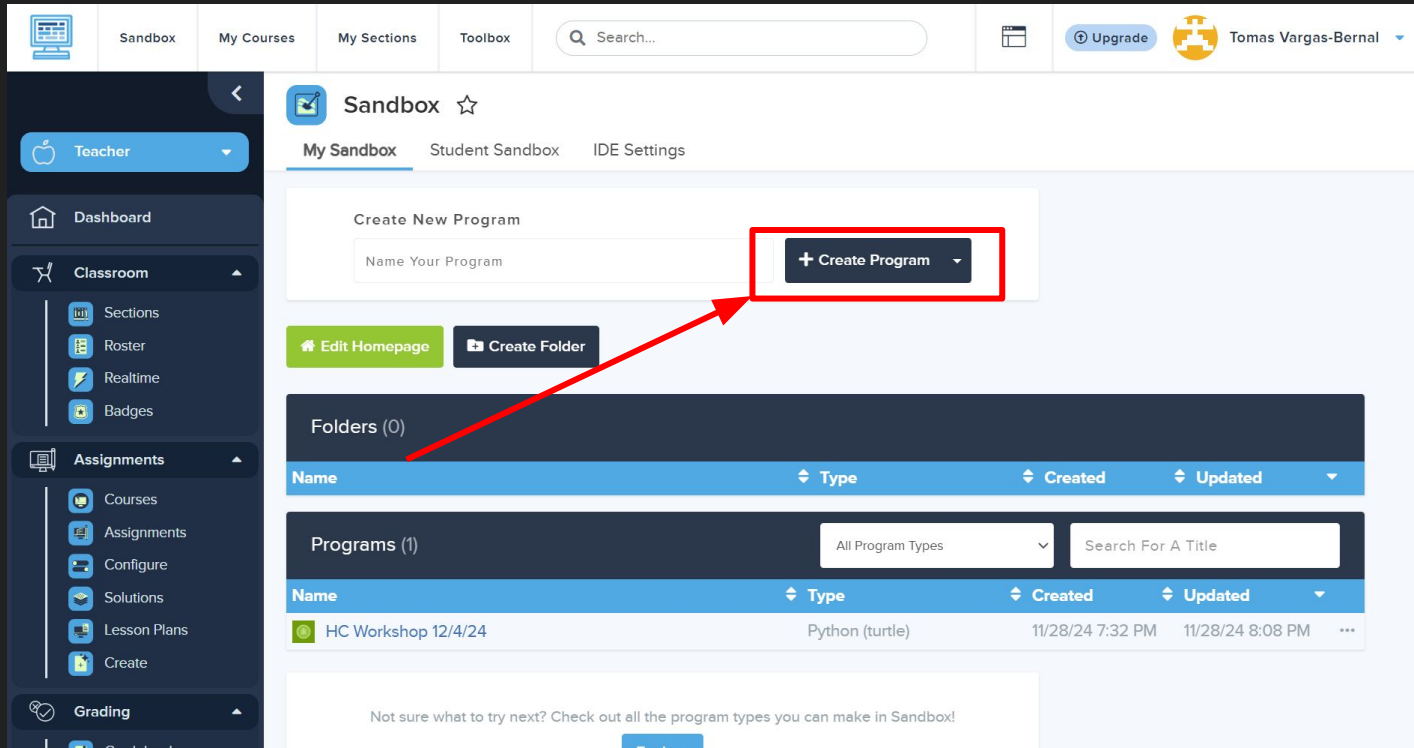


The screenshot displays the CodeHS user interface. At the top, a navigation bar includes a 'Sandbox' menu item (highlighted with a red box), 'My Courses', 'My Sections', and 'Toolbox'. A search bar and an 'Upgrade' button are also visible. The user's name, 'Tomas Vargas-Bernal', is shown in the top right. The main content area is titled 'Sections' and features a 'My Sections' list, a 'Create New Section' button, and three cards: 'Course Catalog', 'Teacher Resources', and 'Get in Touch'. A sidebar on the left provides navigation for 'Teacher', 'Dashboard', 'Classroom', 'Assignments', and 'Grading'.

Scalable Snowmen

Getting Started:

Create a new program



The screenshot shows the Scalable Snowmen interface. The top navigation bar includes 'Sandbox', 'My Courses', 'My Sections', and 'Toolbox'. A search bar is present, and the user's name 'Tomas Vargas-Bernal' is displayed. The left sidebar contains navigation options: Teacher, Dashboard, Classroom (Sections, Roster, Realtime, Badges), Assignments (Courses, Assignments, Configure, Solutions, Lesson Plans, Create), and Grading. The main content area is titled 'Sandbox' and has tabs for 'My Sandbox', 'Student Sandbox', and 'IDE Settings'. Under 'My Sandbox', there is a 'Create New Program' section with a text input field 'Name Your Program' and a '+ Create Program' button. A red box highlights the button, and a red arrow points to it from the 'Folders (0)' section below. Below the 'Create New Program' section are 'Edit Homepage' and 'Create Folder' buttons. The 'Folders (0)' section is empty. The 'Programs (1)' section shows a table with one program:

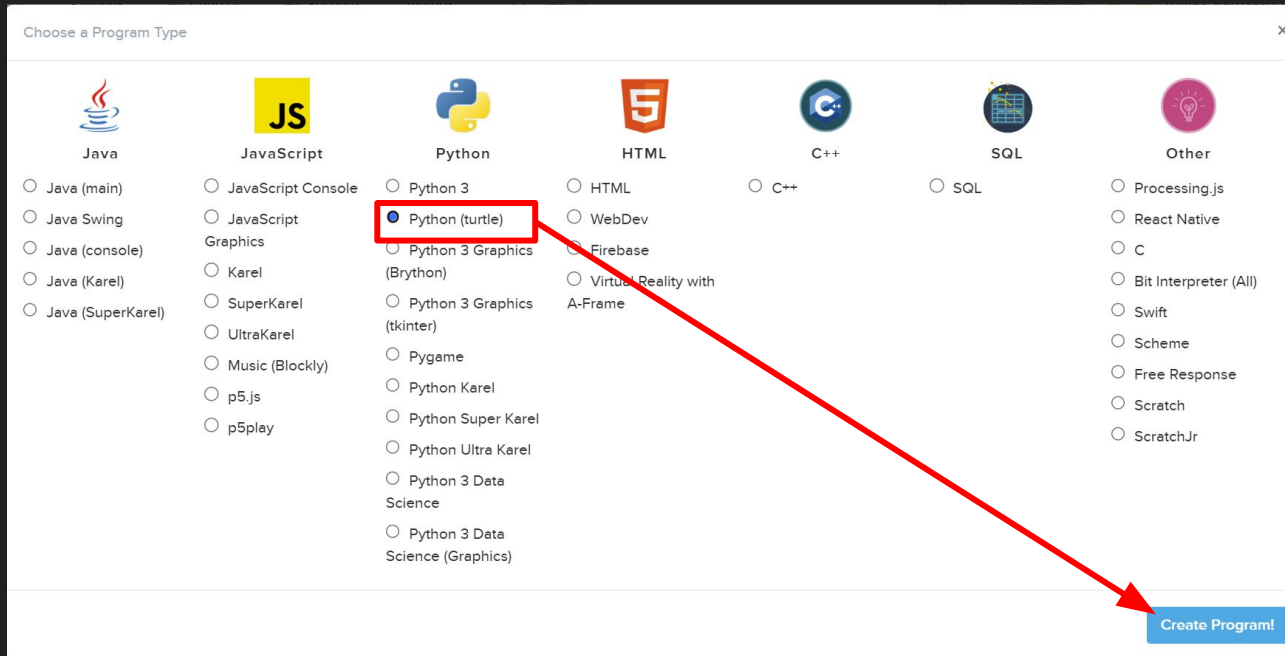
Name	Type	Created	Updated
HC Workshop 12/4/24	Python (turtle)	11/28/24 7:32 PM	11/28/24 8:08 PM

At the bottom, there is a message: 'Not sure what to try next? Check out all the program types you can make in Sandbox!' with a 'Explore' button.

Scalable Snowmen

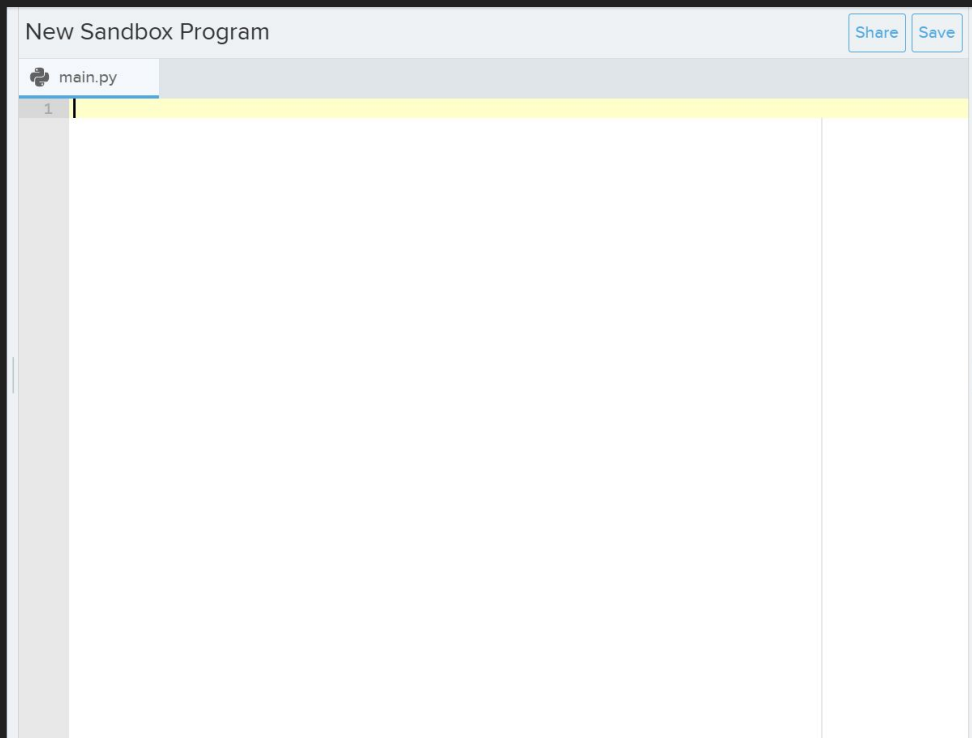
Getting Started:

Create a Python (turtle) program

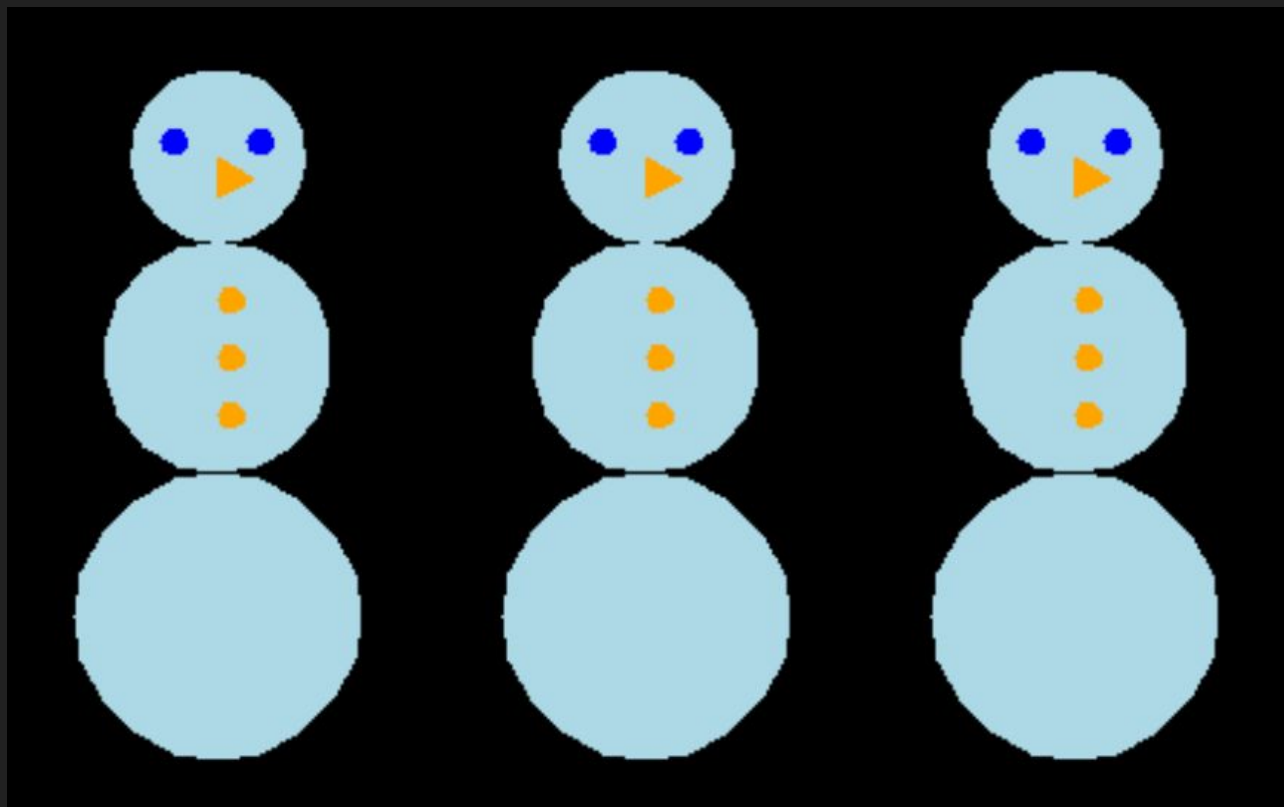


Setting Up

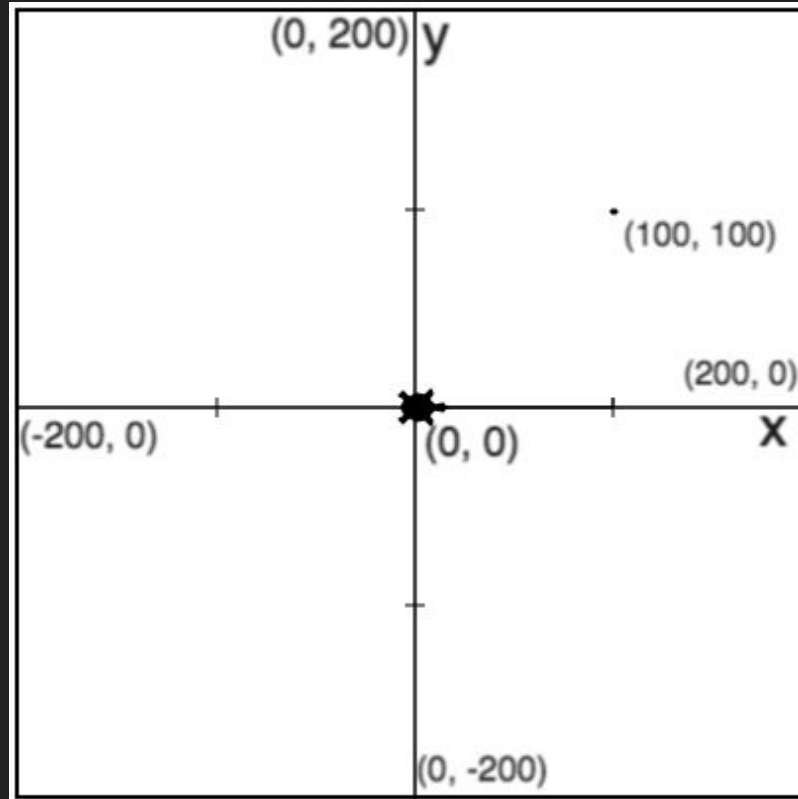
- Delete all existing code



Final Product



Tracy the Turtle Positioning



Taking User Input

- `input()` function is used to collect user input.
- `float()` converts the input to a decimal number, suitable for dimensions.

```
main.py
1 radius1 = float(input("What radius for the first circle?: "))
2 radius2 = float(input("What radius for the second circle?: "))
3 radius3 = float(input("What radius for the third circle?: "))
4
```


Setting Up the Turtle

- `Turtle()` creates a new turtle object.
- `speed(0)` sets the turtle's speed to the fastest.
- `hideturtle()` hides the turtle cursor for a clean look.
- `bgcolor("black")` changes the background color.

```
5 turtle = turtle.Turtle()
6
7 speed(0)
8 hideturtle()
9 bgcolor("black")
10
```

Snowman function

- reusable functions to simplify the code.

```
11 ▾ def Snowman(x, y, radius1, radius2, radius3):  
12     turtle.penup()  
13     turtle.goto(x, y)  
14     turtle.color("lightblue")  
15  
16     # Head  
17     turtle.begin_fill()  
18     turtle.circle(radius1)  
19     turtle.end_fill()
```

Snowman function

- Remember to check indentation!

```
21     # Body piece 1
22     turtle.penup()
23     turtle.setheading(270) # Face downward
24     turtle.forward(radius2 * 2)
25     turtle.setheading(0) # Reset heading to the right
26     turtle.begin_fill()
27     turtle.circle(radius2)
28     turtle.end_fill()
29
30     # Body piece 2
31     turtle.penup()
32     turtle.setheading(270) # Face downward
33     turtle.forward(radius3 * 2)
34     turtle.setheading(0) # Reset heading to the right
35     turtle.begin_fill()
36     turtle.circle(radius3)
37     turtle.end_fill()
```

Snowman function

- Remember to check indentation!

```
39     # Eyes
40     turtle.color("blue")
41     turtle.penup()
42     turtle.goto(x, y + radius1) # Position for eyes
43     turtle.setheading(0)
44     turtle.forward(radius1 / 2)
45     turtle.begin_fill()
46     turtle.circle(radius1 / 6)
47     turtle.end_fill()
48
49     turtle.backward(radius1) # Move to the other eye
50     turtle.begin_fill()
51     turtle.circle(radius1 / 6)
52     turtle.end_fill()
```

Snowman function

- Remember to check indentation!

```
67     # Buttons
68     turtle.penup()
69     turtle.goto(x, y) # Position for buttons
70     turtle.setheading(270) # Face downward
71     for i in range(3):
72         turtle.forward(radius2 / 2)
73         turtle.begin_fill()
74         turtle.circle(radius1 / 6)
75         turtle.end_fill()
76
77     # Reset turtle heading and position
78     turtle.penup()
79     turtle.setheading(0)
80
```

Draw function

- `tracer(0, 0)` disables real-time drawing for faster rendering.
- `update()` updates the screen with the drawn objects.

```
82 ▾ def draw():
83     turtle.tracer(0, 0)
84     turtle.hideturtle()
85     Snowman(x, y, radius1, radius2, radius3)
86     Snowman(x+120, y, radius1, radius2, radius3)
87     Snowman(x-120, y, radius1, radius2, radius3)
88     turtle.update()
```

Finishing Touches

- Last lines of code

```
89 x = 0
90 y = 0
91 draw()
```

FIND
OUT



Mess

AROUND