SnapPy: User Manual

Authors: Luigi Di Paolo, Andy Vodopi
Supervisor: Dr Stephen Blott

Contents

1 About the Application

SnapPy is a novel block-based visual programming environment designed to make learning Python more accessible and engaging for beginners. It bridges the gap between visual block-based programming and text-based programming by allowing users to create Python code by manipulating drag-and-drop blocks. As users build programs visually, SnapPy simultaneously generates the equivalent Python code, enabling a smooth transition from block-based thinking to text-based coding. The system features workspace management, real-time code generation and an integrated Python interpreter, all within a modern single page app built using Next.js.

2 Main functionalities

The main functionalities of the system are:

1. Logging in through Google authentication
2. Creating, managing, and deleting projects
3. Building Python programs using a visual block-based interface
4. Real-time automatic generation of equivalent Python code
5. Execution of Python code directly in the browser
6. Persistent storage of projects and workspaces
7. Support for essential programming constructs (variables, loops, conditionals)

3 Installation Guide

3.1 Installation

1. Install Node.js
2. Install Bun
3. Install Docker
4. Setup Google OAuth

3.2 Running the Project

1. Install dependencies:

```
bun install
```

2. Add your `GOOGLE_CLIENT_ID` and `GOOGLE_CLIENT_ID` in .env file
3. Start postgres db:

```
bun start-db
```

4. Synchronise prisma schema with db:

```
bun prisma-push
```

5. Start the development server:

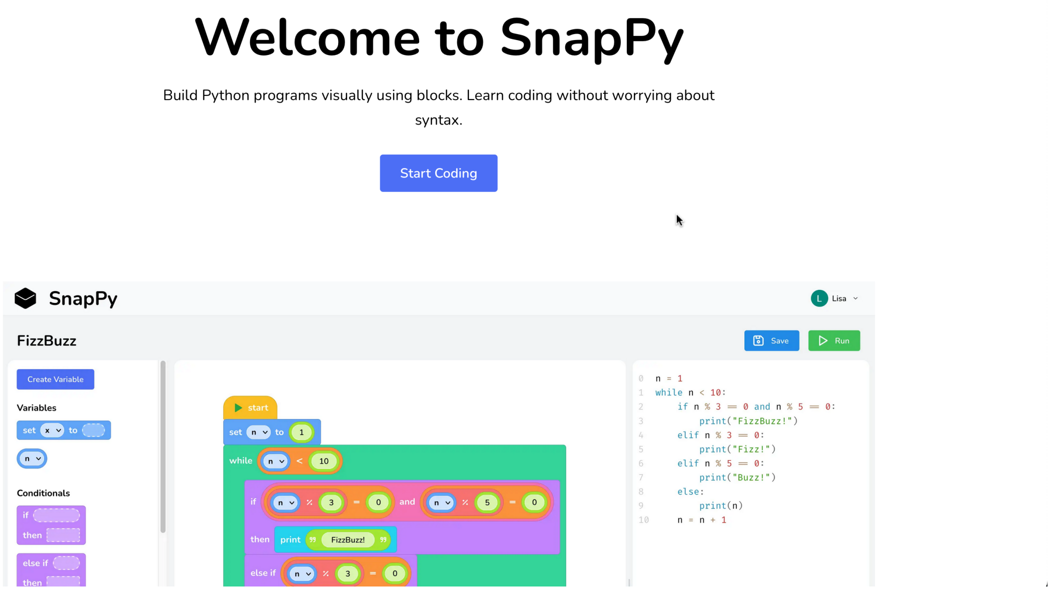
```
bun run dev
```

Visit the app at `http://localhost:3000`.

4 User Guide

4.1 Getting Started

When you go to the homepage, you will see this screen:



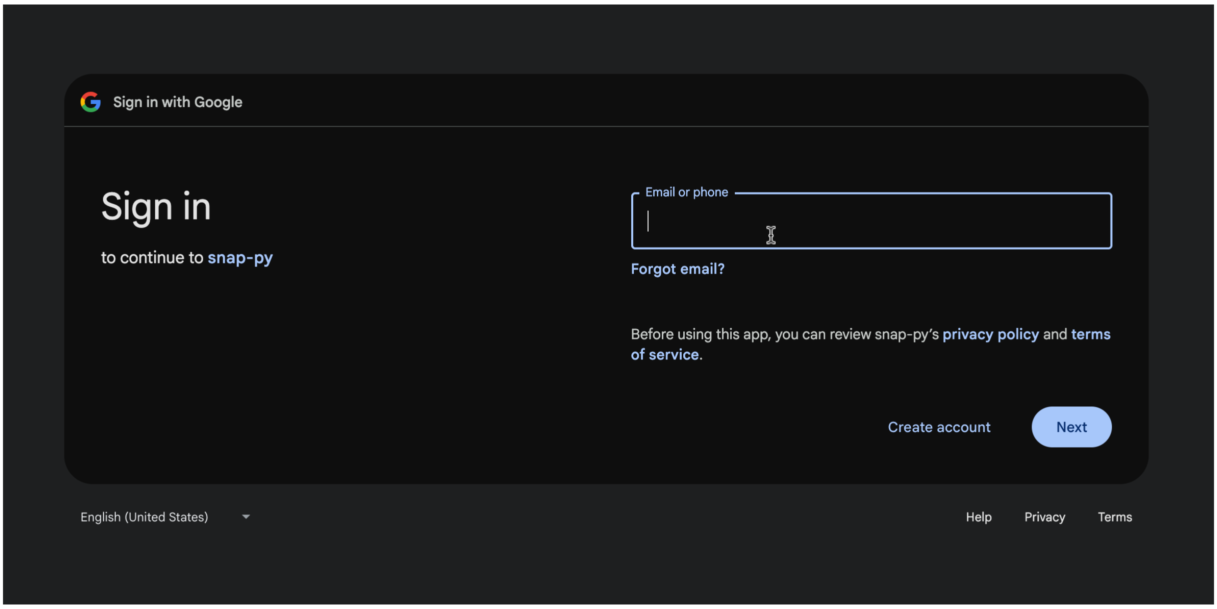
Welcome to SnapPy

Build Python programs visually using blocks. Learn coding without worrying about syntax.

```
n = 1
while n < 10:
    if n % 3 == 0 and n % 5 == 0:
        print("FizzBuzz!")
    elif n % 3 == 0:
        print("Fizz!")
    elif n % 5 == 0:
        print("Buzz!")
    else:
        print(n)
    n = n + 1
```

4.2 Signing In

To sign in, click the “Start Coding” button. A google authentication window will open.

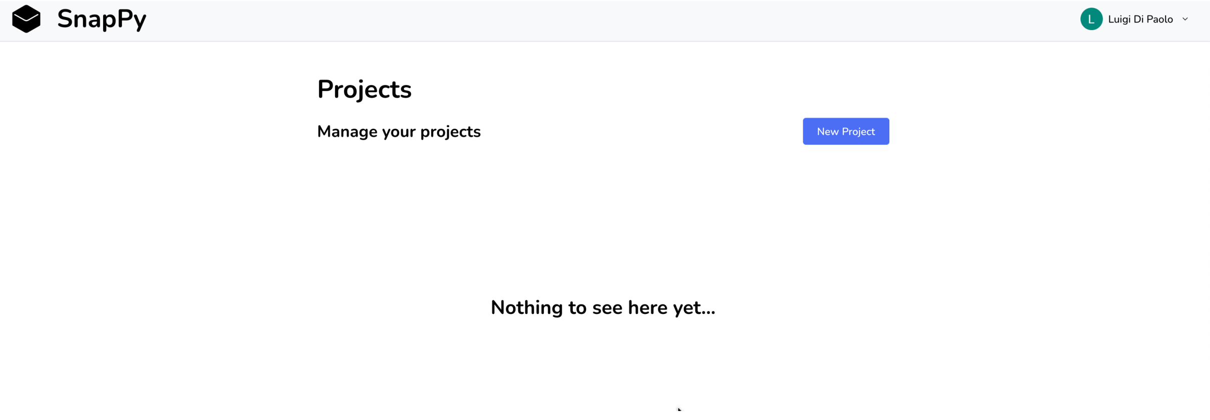


to continue to snap-py

Before using this app, you can review snap-py’s privacy policy and terms of service.

4.3 Projects Page

Once signed in successfully, you will be redirected on the projects page:

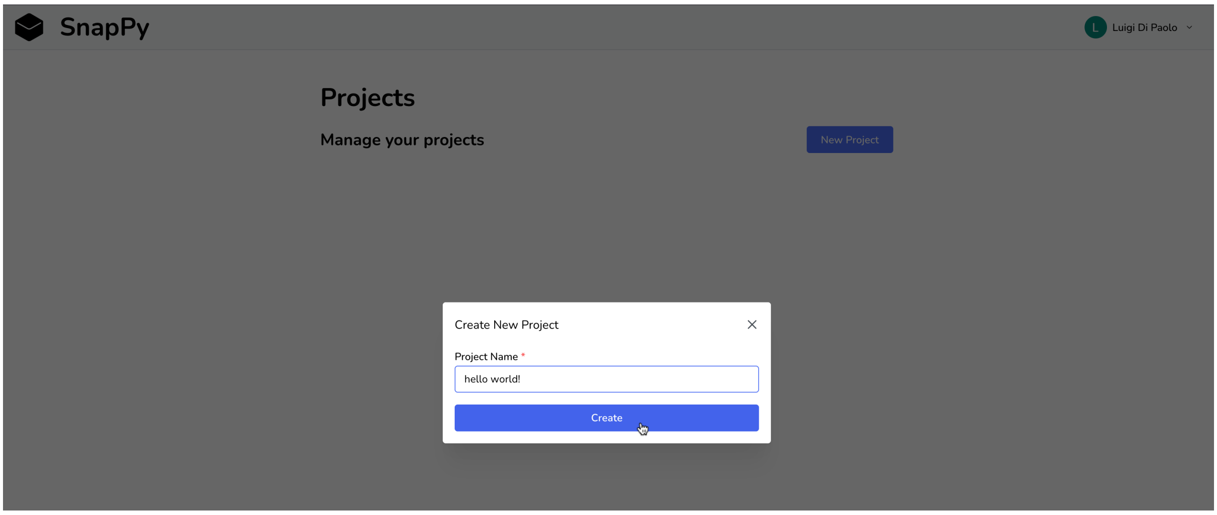


Manage your projects

Nothing to see here yet...

4.4 Creating Projects

To create a project, click New Project. This will open a modal:



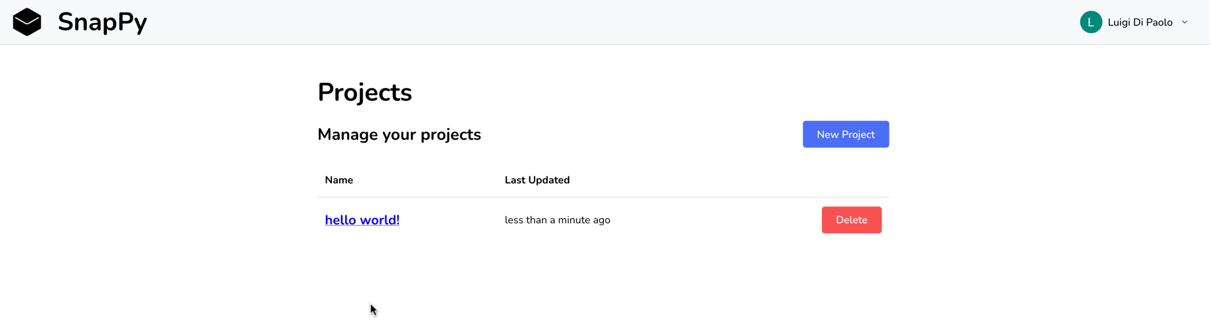
Projects

Manage your projects

Create New Project

Project Name *: hello world!

Name your project, then press create. Your project will appear in the dashboard.



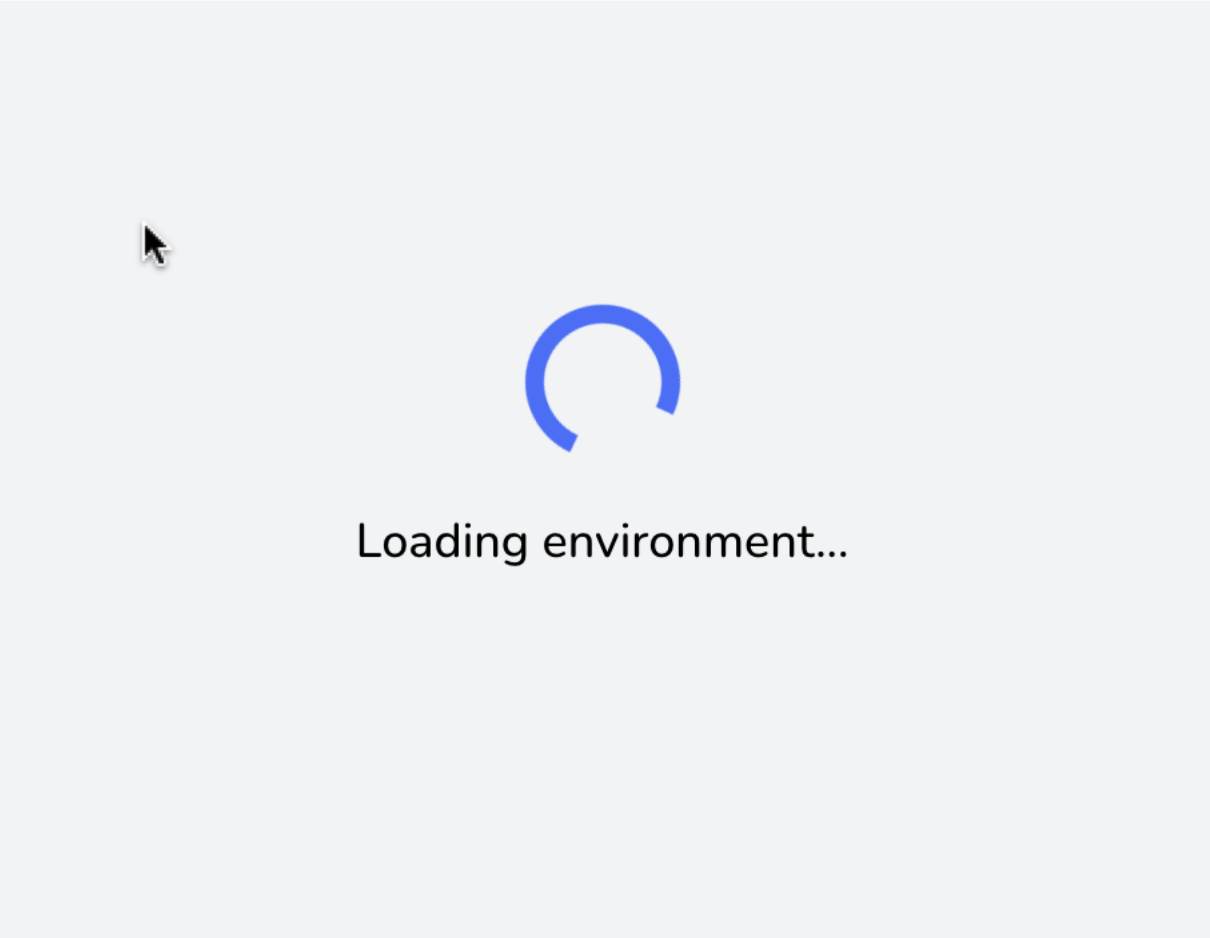
Projects

Manage your projects

Name	Last Updated	
hello world!	less than a minute ago	Delete

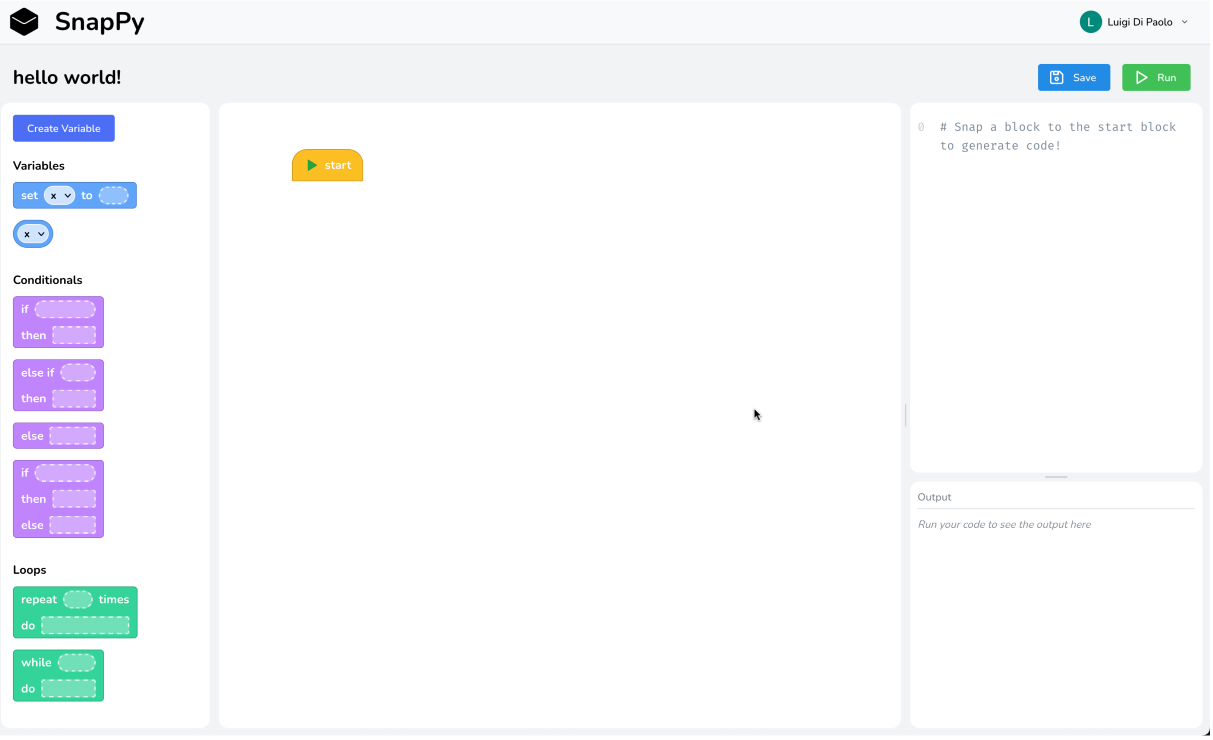
4.5 Loading Projects

Clicking on the project’s name will load the project and redirect you to the editor:



Loading environment...

Once the editor is loaded, you will see this page:



```
# Snap a block to the start block
to generate code!
```

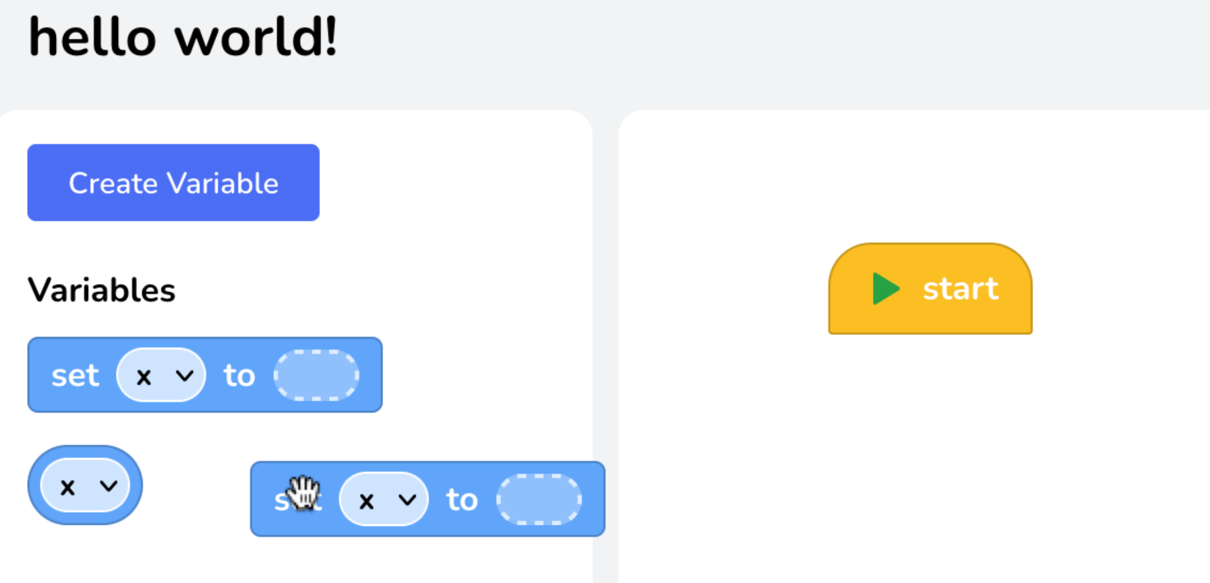
Run your code to see the output here

The page has three sections:

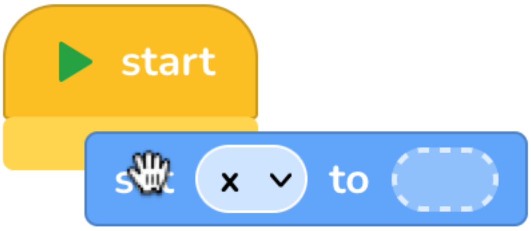
1. Workbench (left)
2. Canvas (center)
3. Code editor and output box (right)

4.6 Creating Programs

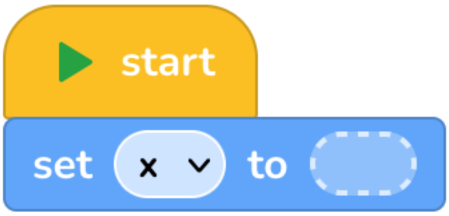
To create a program, drag a block from the workbench to the canvas and position it under the Start Block. The Start Block serves as the entry point of your program.



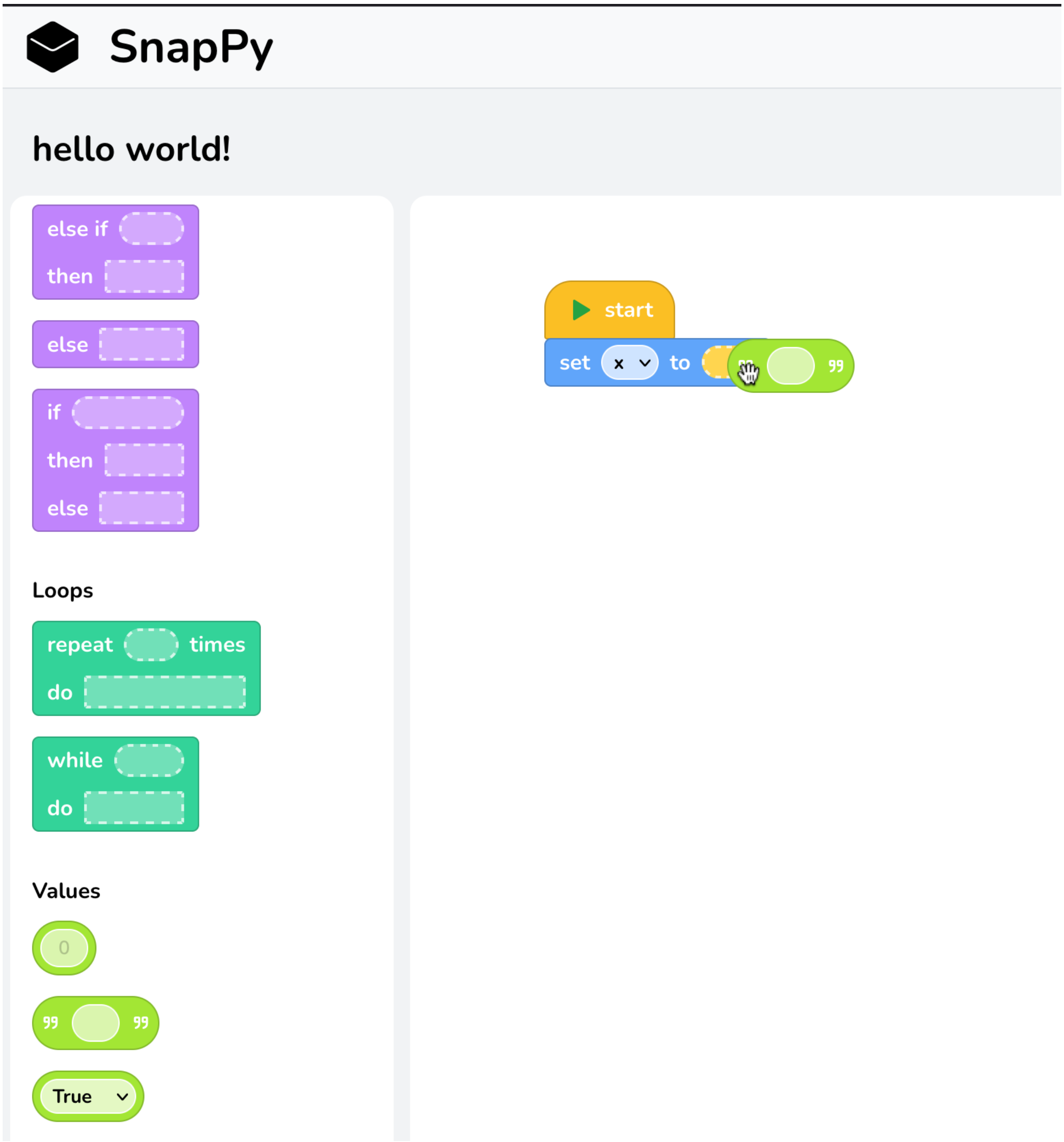
If close to a snapping area, the area will highlight in yellow:



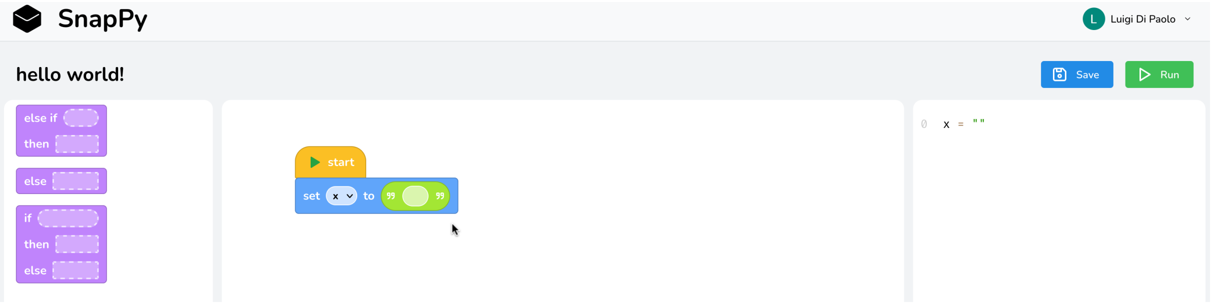
Release the mouse to snap the blocks:



Now let’s snap a string block in the print block:

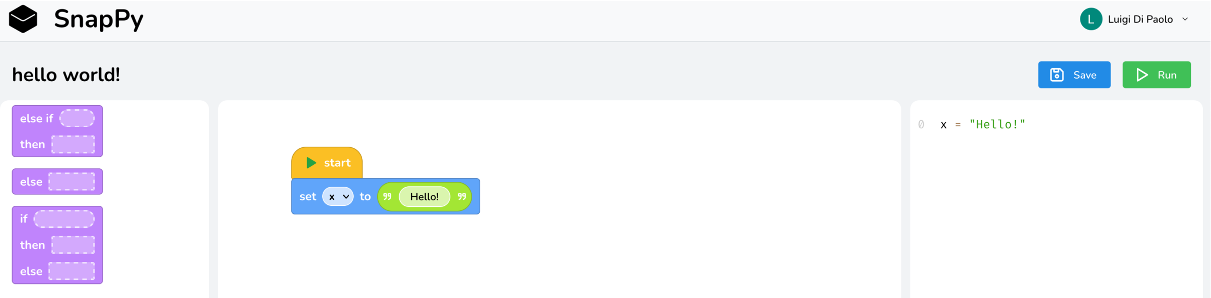


When the drop zone is highlighted, release to snap:



```
x = ""
```

Now click on the input box to add some text

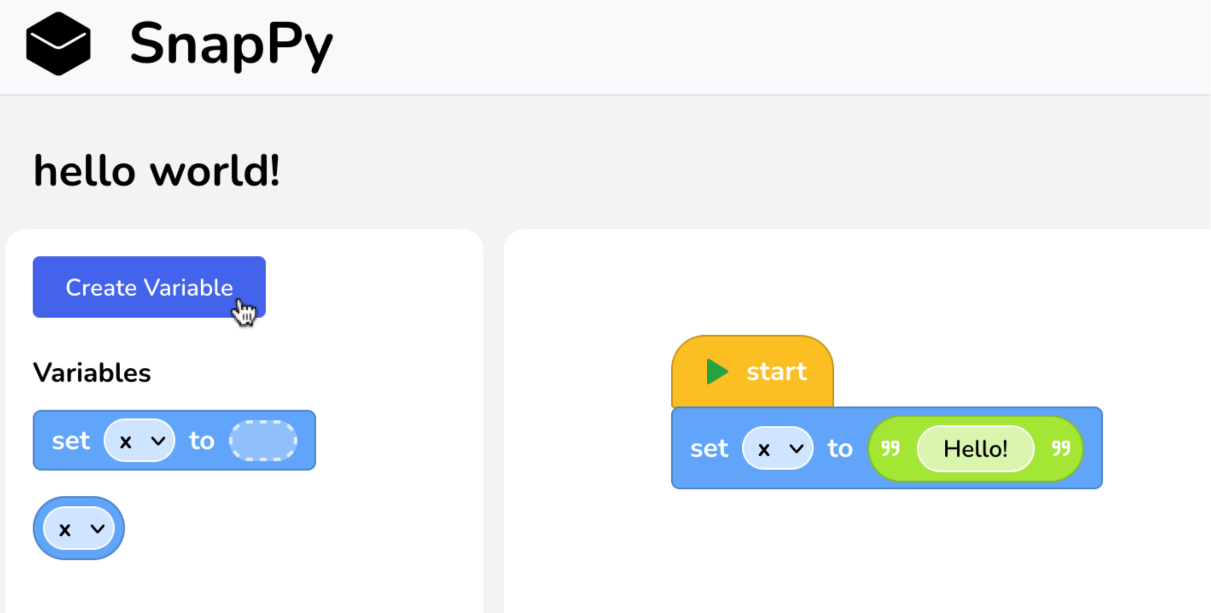


```
x = "Hello!"
```

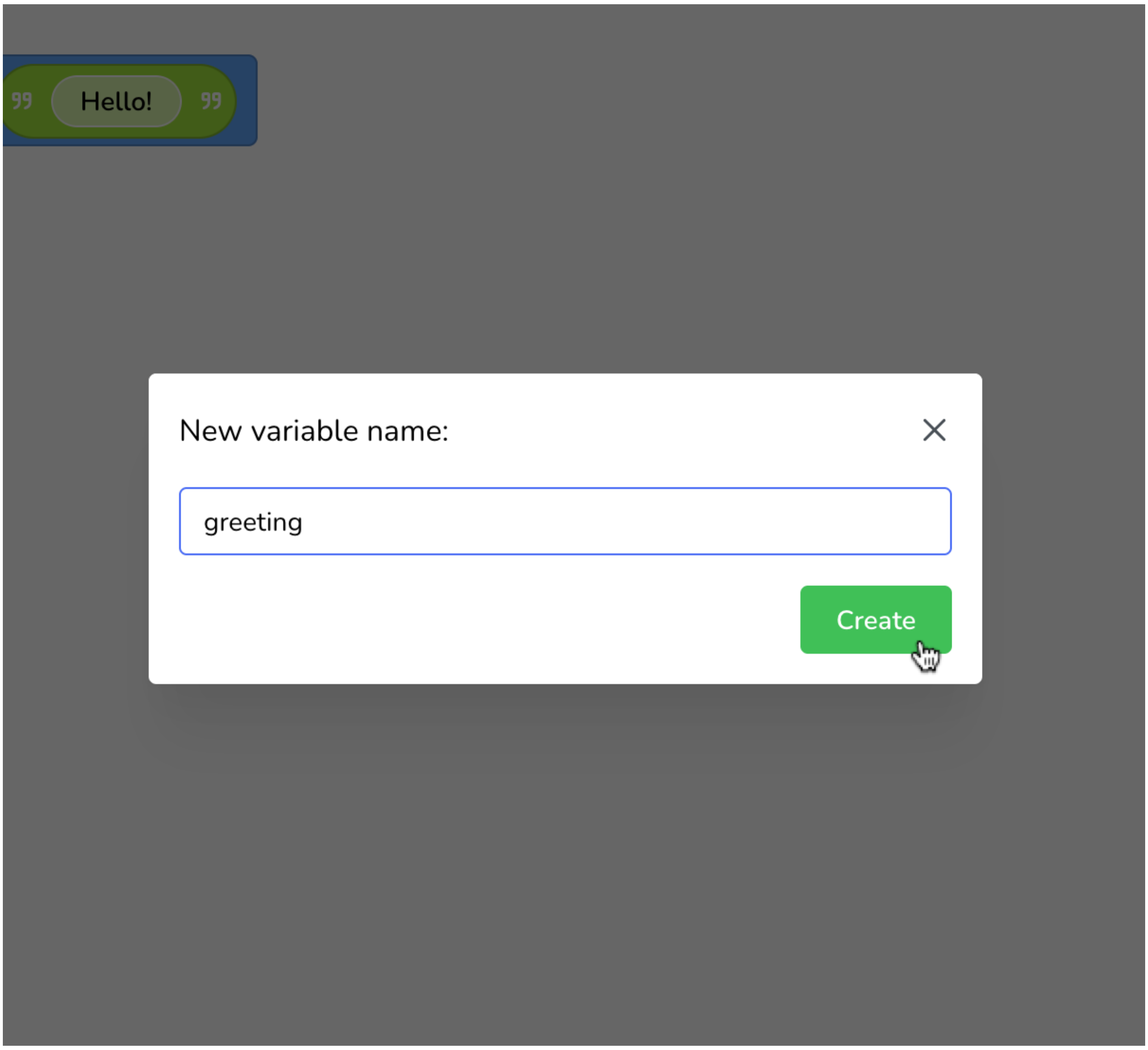
You will see on your right that the code has been generated automatically.

4.7 Creating New Variables

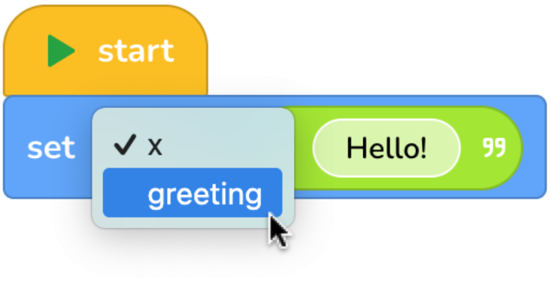
x is not a very descriptive name for this variable, so let’s create a more meaningful one. Click Create Variable above the workbench:



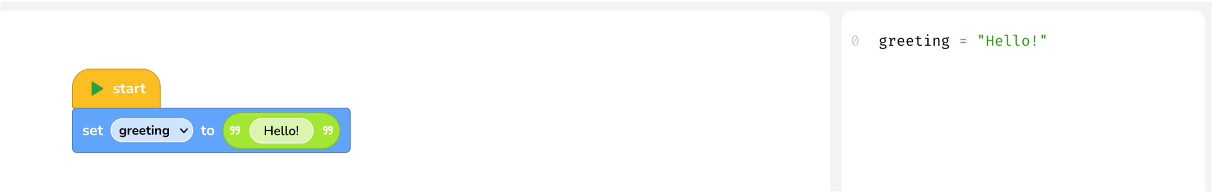
This will open a modal. Enter your new name in it and press Create:



Now select the new variable name from the dropdown:



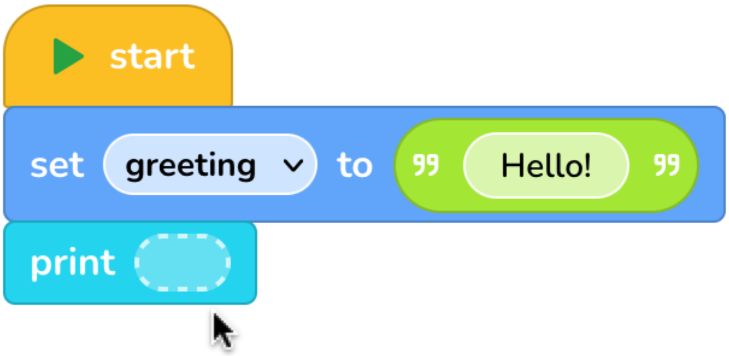
Notice how the code has been updated automatically:



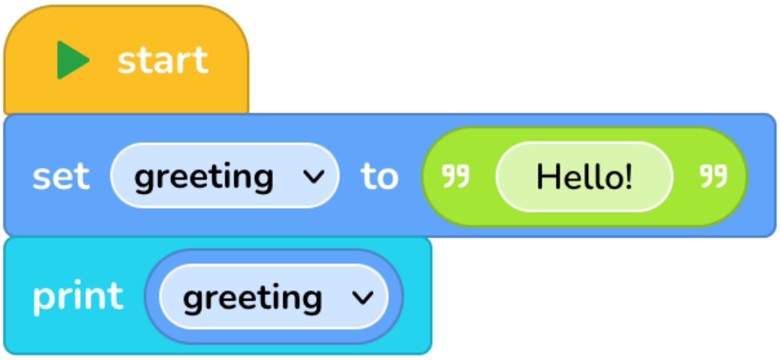
```
greeting = "Hello!"
```

4.8 Printing

Next, let's see some output. Add a print block below:

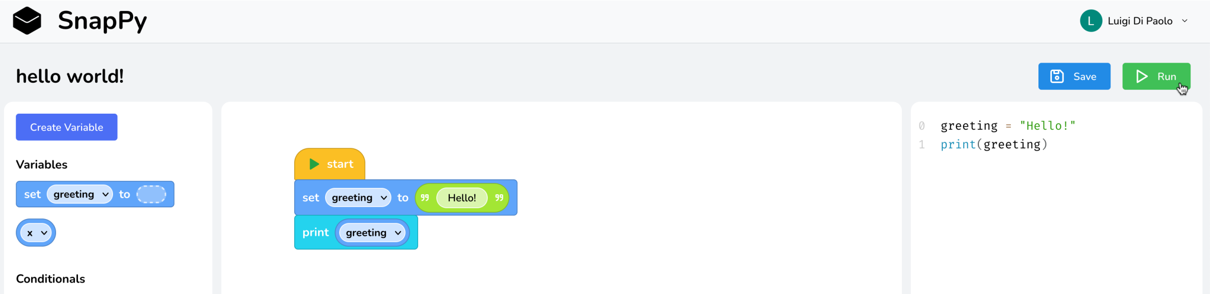


Next, add the variable in it:



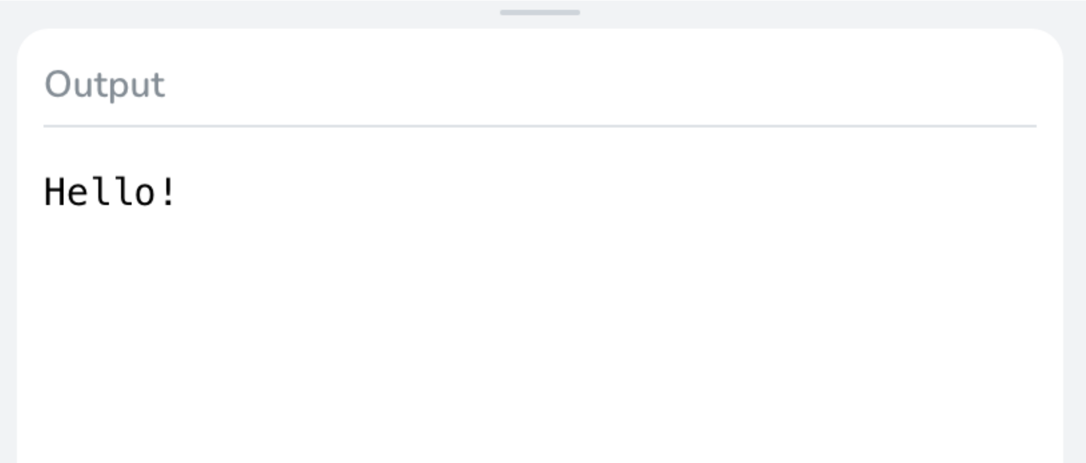
4.9 Executing the code

To run the program, click the run button:



```
greeting = "Hello!"
print(greeting)
```

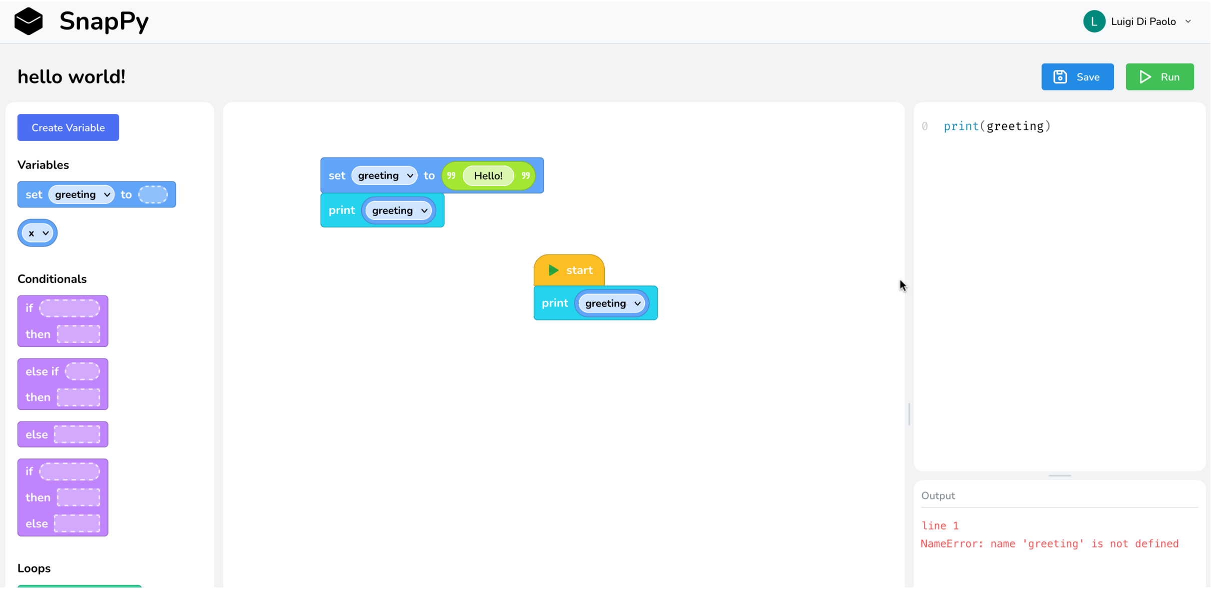
The output will appear below:



```
Hello!
```

4.10 Errors

You can have multiple program blocks on the canvas, but only the group connected below the Start Block generates executable code. In this example, 'greeting' is not defined within this connected sequence, which results in an error when running the program:

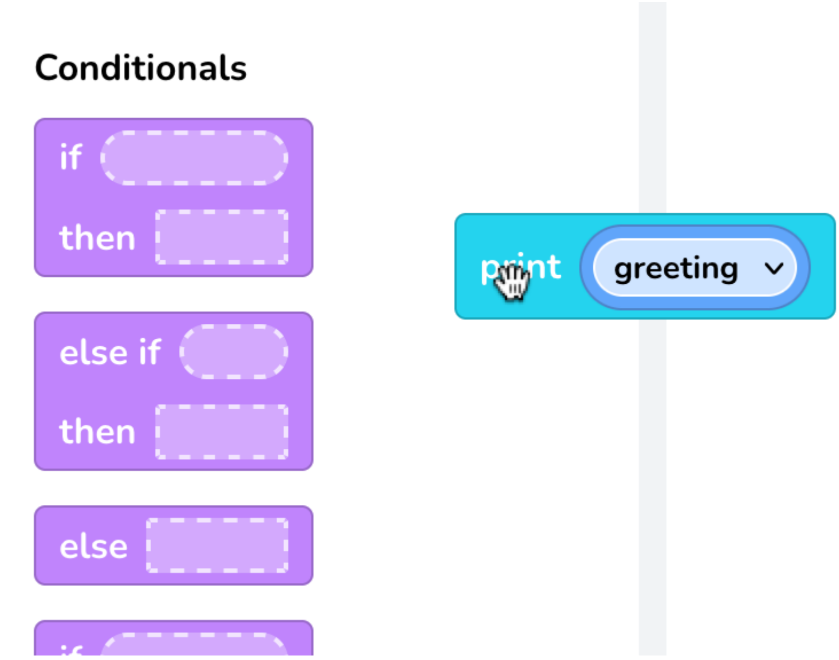


```
print(greeting)
```

```
line 1
NameError: name 'greeting' is not defined
```

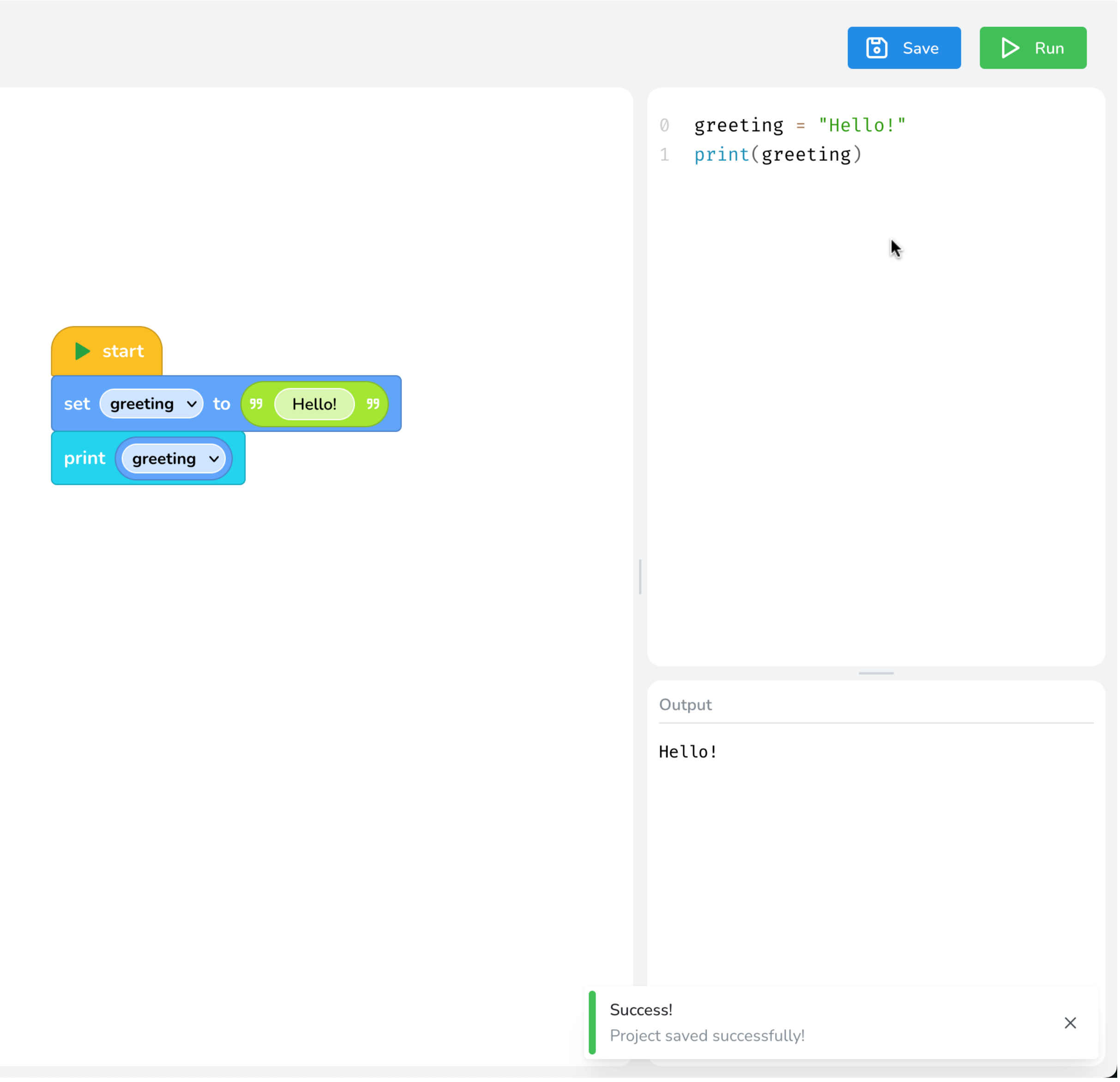
4.11 Deleting blocks

To delete a block, simply drag it back on the workbench:



4.12 Saving

To save the program and store it in the database, click Save on the top-right.
This will show a confirmation pop-up below:



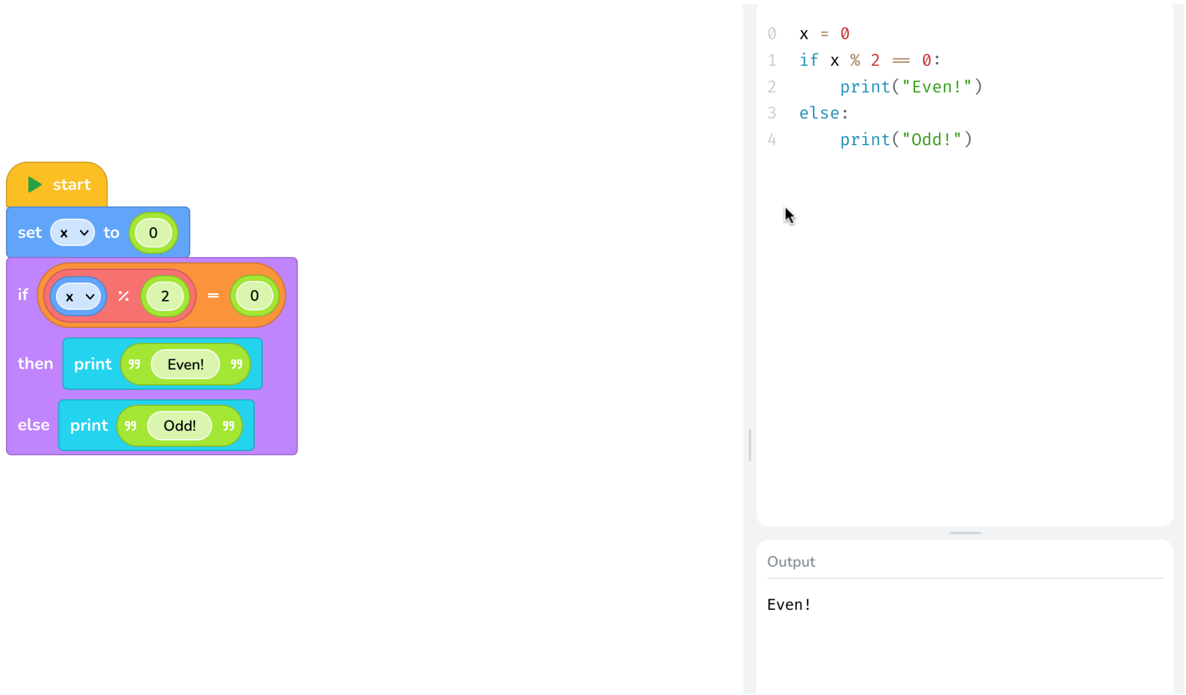
```
greeting = "Hello!"
print(greeting)
```

```
Hello!
```

Success!
Project saved successfully!

4.13 Complex nesting

We can create more articulate programs. This program takes a number and prints whether it is even or odd:



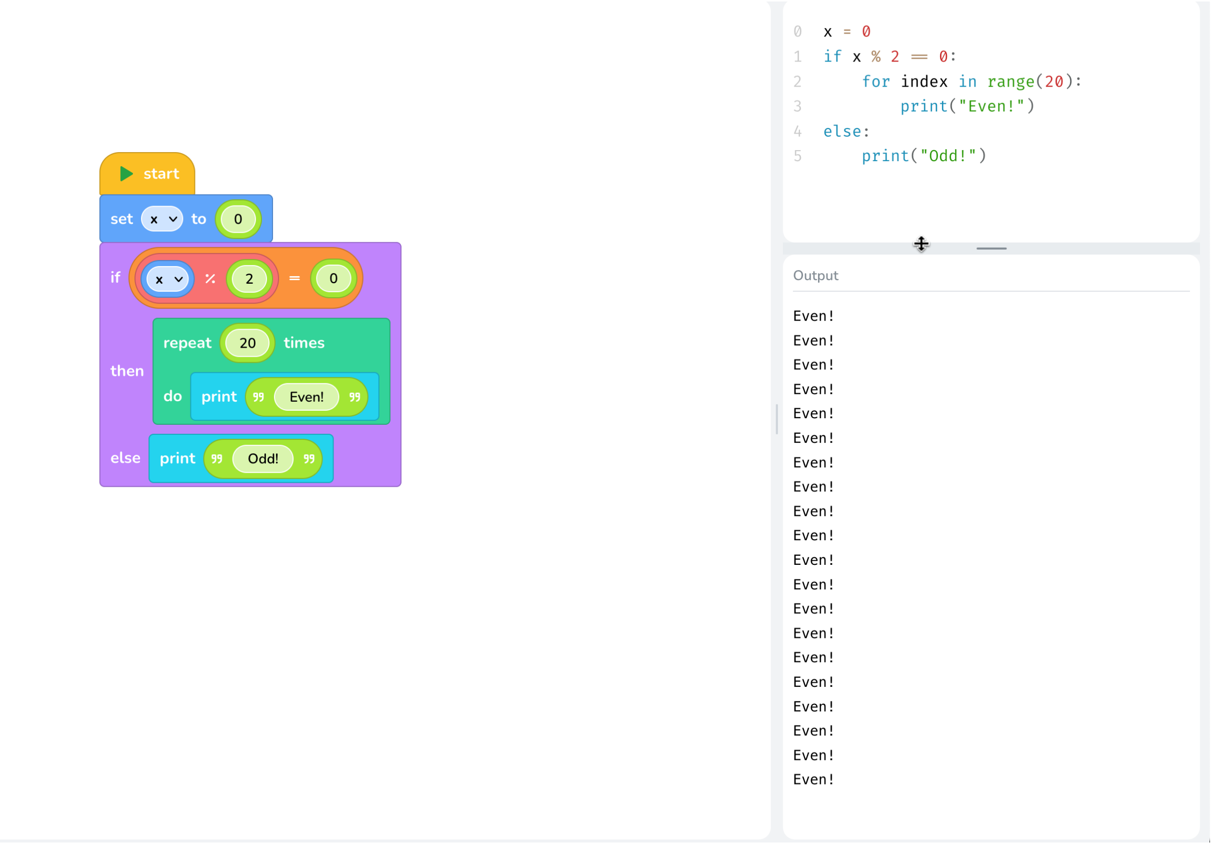
```
x = 0
if x % 2 == 0:
    print("Even!")
else:
    print("Odd!")
```

Output

```
Even!
```

4.14 Resizing The Editor

We can freely resize the editor. For example, this program has a lot of output and we can’t fit it all in the box:

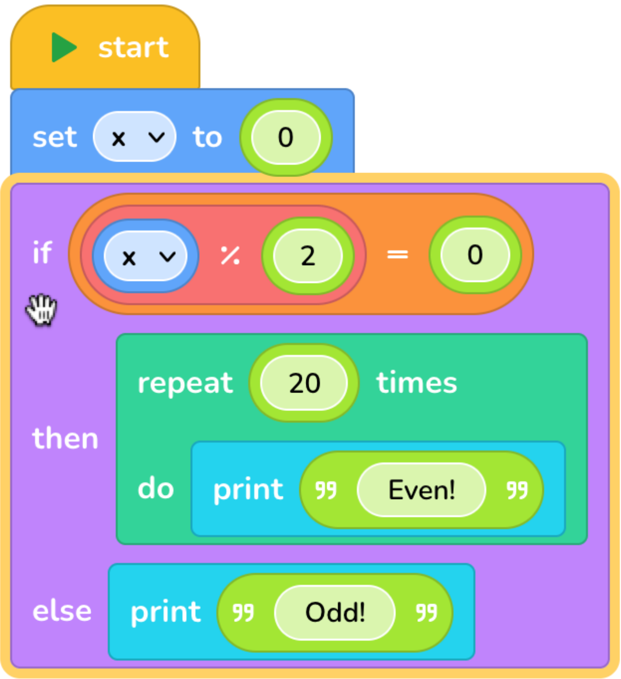


```
x = 0
if x % 2 == 0:
    for index in range(20):
        print("Even!")
else:
    print("Odd!")
```

```
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
Even!
```

4.15 Selecting Blocks

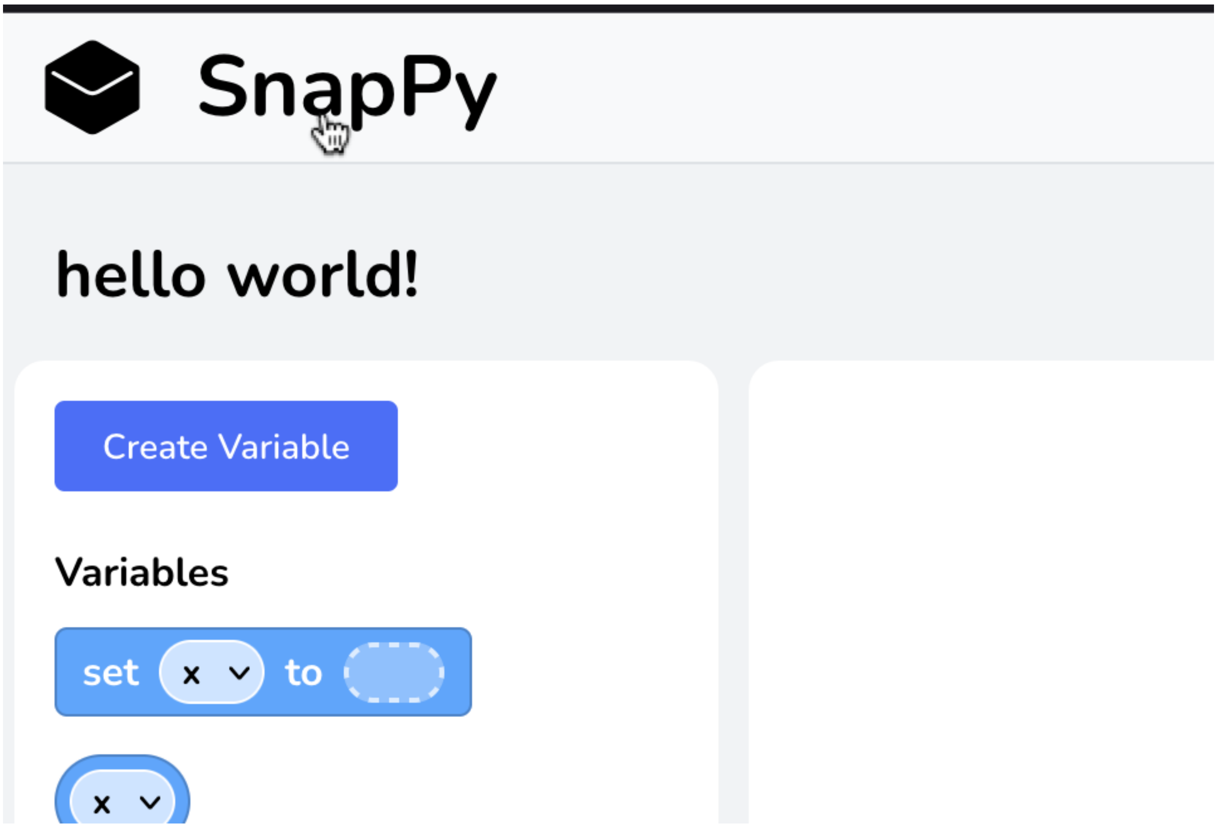
Simply click on a box to select it and highlight it:



```
start
set x to 0
if x % 2 = 0
then
    repeat 20 times
    do print "Even!"
else
    print "Odd!"
```

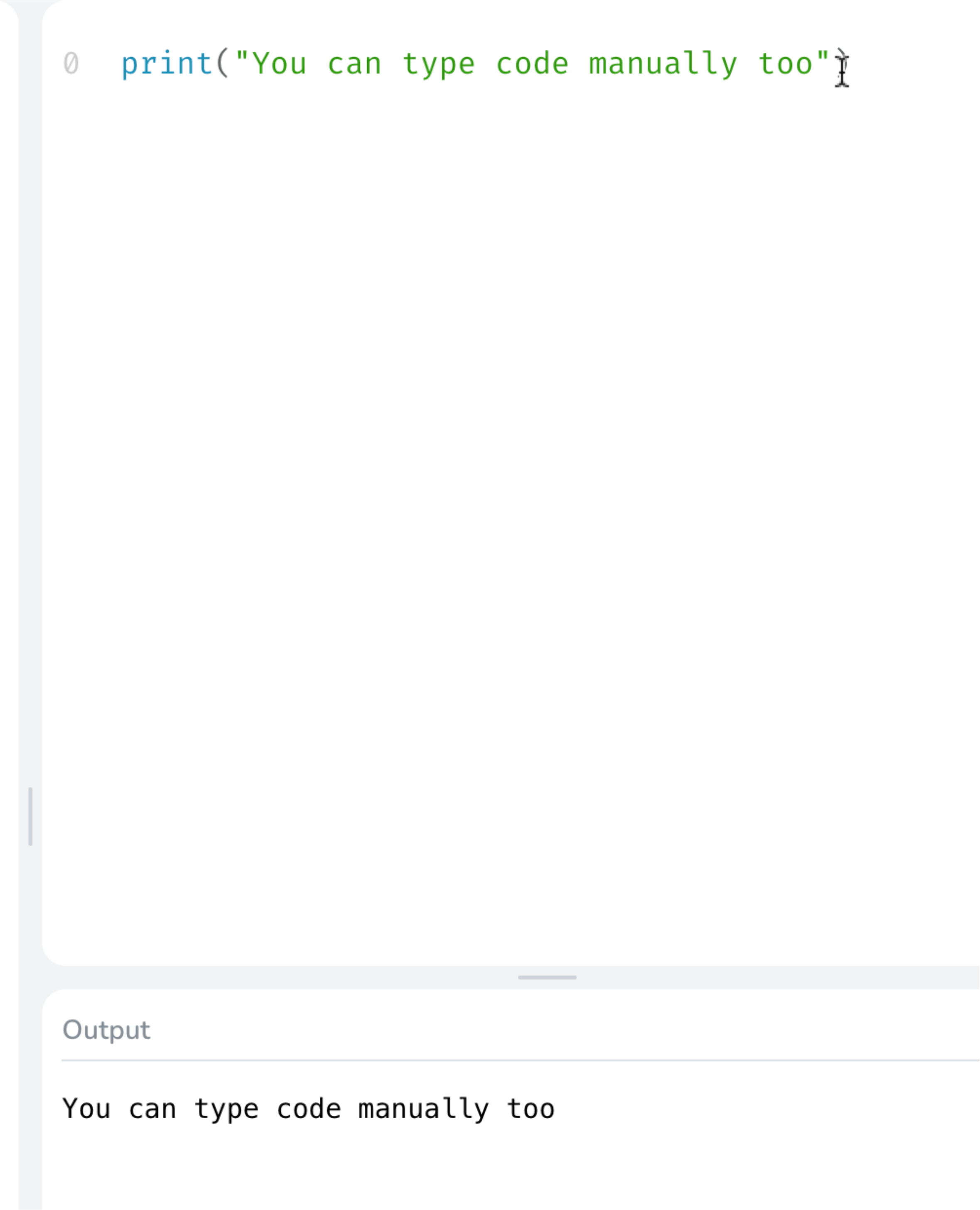
4.16 Going Back to the Projects Page

Click on SnapPy's logo to go back to the homepage:



4.17 Writing Code Manually

You can also write your programs by hand if you wish:

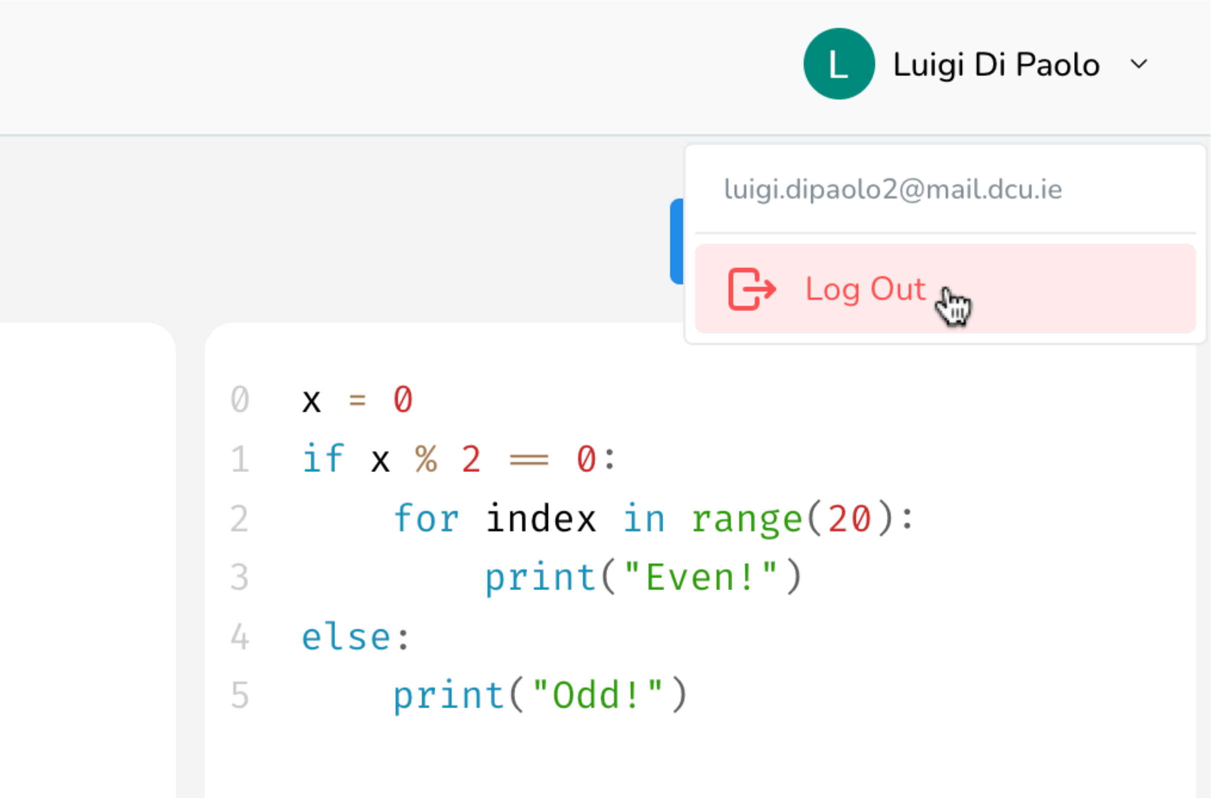


```
print("You can type code manually too")
```

```
You can type code manually too
```

4.18 Logging out

Click on your name on the top-right corner. It will open a menu with a Log Out option:



```python
x = 0
if x % 2 == 0:
    for index in range(20):
        print("Even!")
else:
    print("Odd!")
```