From Blocks to Text: A Scratch-to-Python Bridge

Contents

Welcome to your first step into computer science! Before we dive into the syntax of Python, we are going to use **Scratch** - a visual block-based language developed by MIT - to understand the logic.

Think of **Scratch** as building with LEGOs: the pieces snap together in obvious ways, and you can see the structure immediately. **Python** is like writing the architectural blueprints for that same structure: it requires more precision, but it allows you to build much more complex skyscrapers.

Tutorial 1.1 Move the Cat

Video Walkthrough: [Move the Cat](#)

1. Go to https://scratch.mit.edu/

2. Click **Start Creating**. (If you wish to save your Scratch projects → Click Join)

3. At the Top of the window → Click **Tutorials** → **Getting Started**.

4. A tutorial window will show at the bottom of the window.

5. Create the program as directed.

6. Click the **Green Flag** to run the program.

7. The Cat moves and speaks.

8. Change the numbers to see the effect. Try a negative number for movement.

9. Take a screenshot of your program and paste it into a Word document.
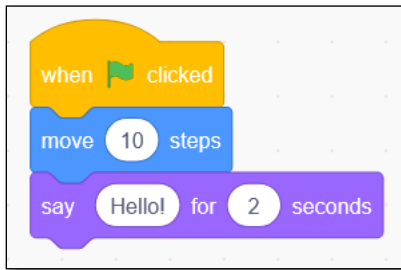
The Variable (The Storage Box)

In programming, we constantly need to save information to use later. We call these containers **Variables**.

The Analogy

Imagine a cardboard box. You can write a label on the outside (the variable name) and put an object inside (the value).

- **Scratch:** You literally make a block called "Score" and use an orange block to "Set Score to 0."

- **Python:** You simply write the name, an equals sign, and the value.

The Code Ladder

| Concept	Scratch Block	Python Code
Creating/Assigning	set [my_score v] to (10)	my_score = 10
Updating	change [my_score v] by (1)	my_score = my_score + 1

Key Takeaway: The = in Python doesn't mean "equals" in the mathematical sense. It means "assign." Think of it as an arrow pointing left: my_score <--- 10.

Tutorial 1.2 Move the Cat with Variables

Video Walkthrough: [Variables](#)

The only difference between the previous program and this one is that we are using variables to store the steps and seconds.

1. Go to the **Variables** palette.

2. Click **Make a Variable**.

3. Create 2 variables: **steps** and **seconds**.

4. Drag the **move (10) steps** block away from the **when clicked** block.

5. In the space drag 2 **set my variable** blocks.

6. Click the drop down of **set my variable**.

7. Change one to **steps** and other to **seconds**.

8. Drag the variable **steps** to the move steps block.

9. Drag **seconds** to the say Hello Block.

10. Change the value stored in each variable.

11. Your program should look like this.
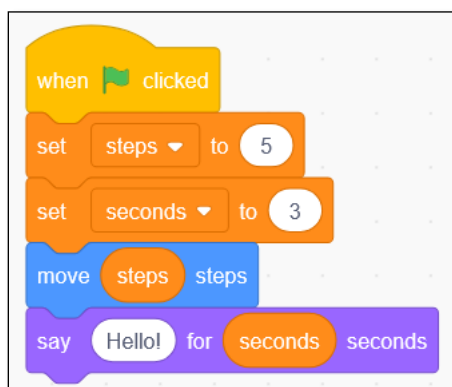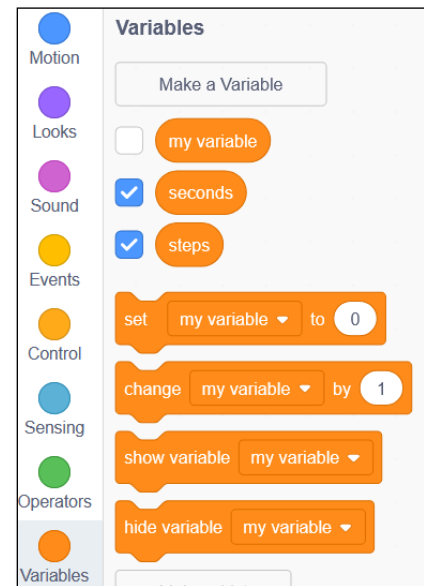
12. Experiment and change the **steps** values from negative to positive.

13. Change the **seconds** values with different positive numbers. Watch how long the **Hello!** stays on the screen.

14. Take a screenshot of your program and paste it into a Word document.

Input and Output (Talking to the User)

A program isn't useful if it can't interact with you. This is **I/O** (Input/Output).

The Analogy

- **Output:** The computer holds up a sign for you to read.
- **Input:** The computer ihands you a form to fill out.

The Code Ladder

Concept	Scratch Block	Python Code
Output (Say)	say [Hello World!]	print("Hello World!")
Input (Ask)	ask [What is your name?] and wait	name = input("What is your name?")

Note: In Python, input() grabs what the user types, and we immediately store it in a variable (like name) so we don't lose it.

Tutorial 1.3: Hello Cat World! (Input)
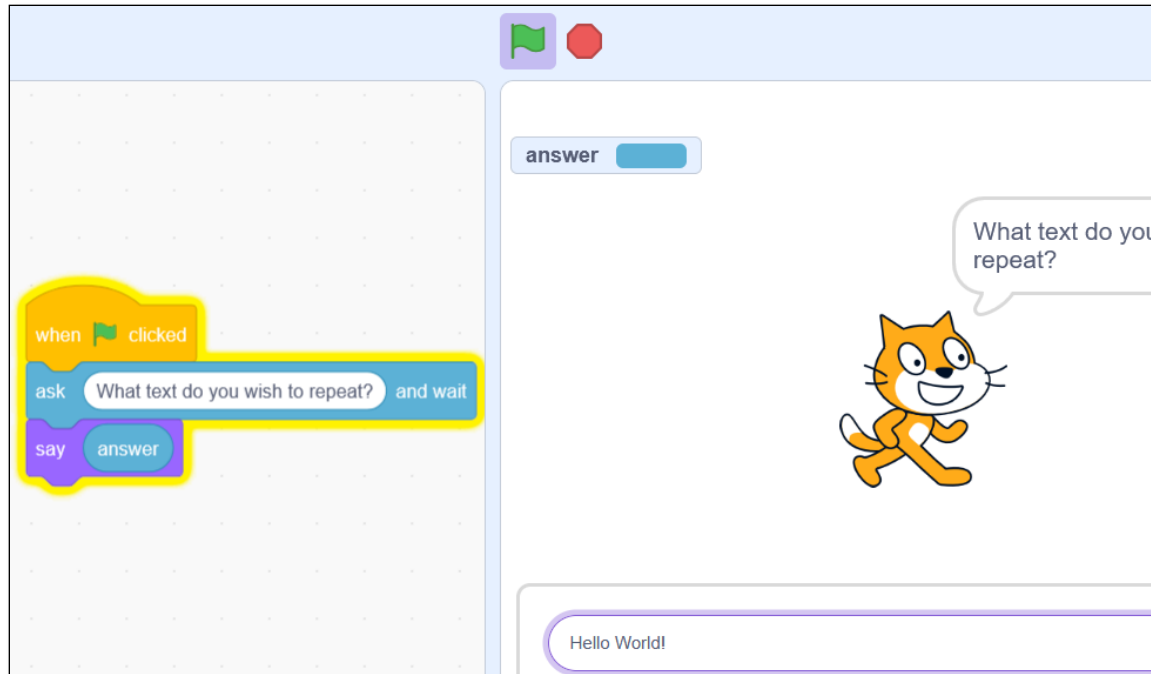
Video Walkthrough: Input

Hello World! is traditionally the first program a programmer writes in a new language. The first tutorial we did was Hello! That is close enough. However, to be a real coder, we have to write the traditional Hello World! program as a rite of passage. We are going one step further, we are going to use input!

1. The **Teal Sensing** tab contain the **ask and wait** block.

2. Click the checkmark in front of answer to create a variable **answer** to store the user input.

The finished program.
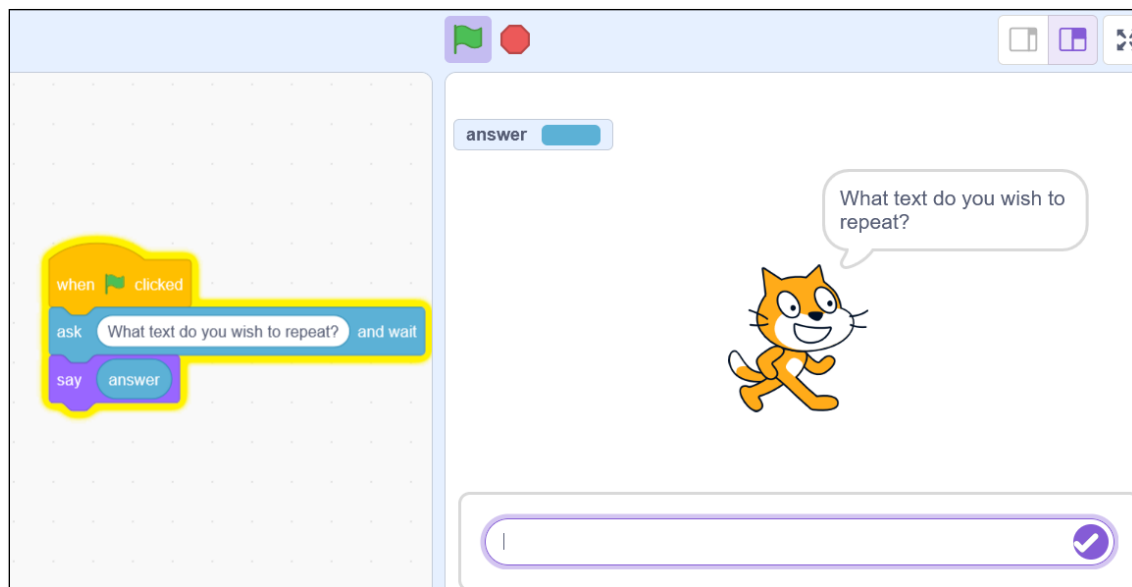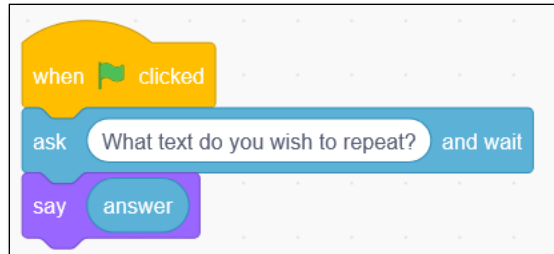
Experiment with the program.

NOTE: All programs follow the following steps: **Input** -> **Storage** → **Process** → **Display**

Tutorial 1.4: Decisions, Decisions, Decisions (The Fork in the Road)

Video Walkthrough: [Decisions]

Logic is about making decisions. If we never made decisions, we would always do the same thing everyday. In programming, we use **If/Else** statements to control the flow.

The Analogy

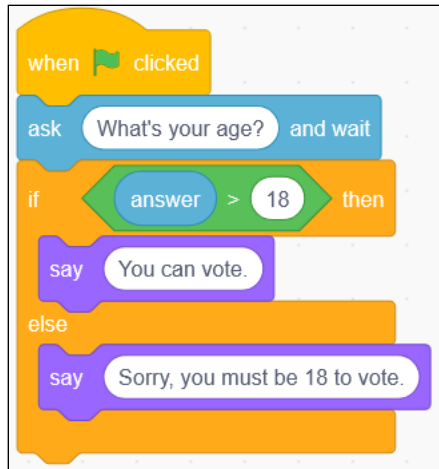Imagine walking down a path. You encounter a sign:

"If you are over 18, go Left.
Else, go Right."

You can only take one path.

The Code Ladder

Scratch:

Python:

```python
if age > 18:
    print("Vote!")
else:
    print("Wait a few years.")
```

Crucial Syntax Note: Notice the indentation in Python?

- In Scratch, the "mouth" of the yellow block wraps around the code to show it belongs inside.

- In Python, we use a **colon (:)** and **indentation (tabs/spaces)** to show that code belongs inside the if statement.

Tutorial 1.5: Loops (The Hamster Wheel)

Video Walkthrough: Loops

Computers are great at doing boring tasks repeatedly without complaining. So are hamsters. We call this **Looping** or **Iteration**.
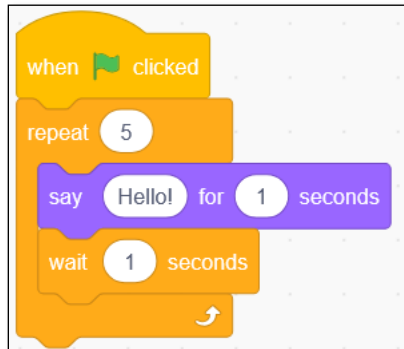
The Analogy

- **Counted Loop (For Loop):** "Do 10 pushups." You count 1, 2, 3... until you hit 10.

- **Conditional Loop (While Loop):** "Keep running **until** you are tired." You don't know how many steps it will take, you just check the condition (Are you tired?) constantly.

The Code Ladder

A. The "Repeat" (For Loop)

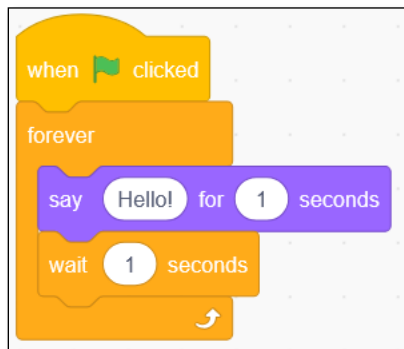**Scratch:**

Python:

```
for i in range(5):
    print("Hello")
```

Note: range(5) generates the numbers

0, 1, 2, 3, 4.

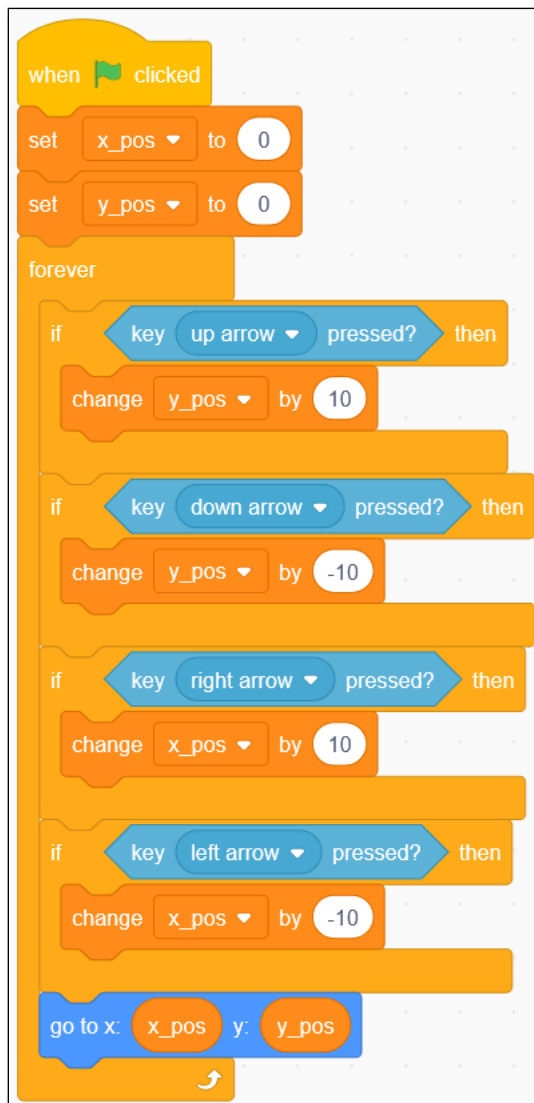### B. The "Forever" or "Until" (While Loop)

Scratch:

Python:

```
answer = ""
while answer != "yes":
    answer = input("Are we there yet?")
```

Cat Driving Finale!

Video Walkthrough: Move the Cat

This program combines everything we have learned up to this point. Drive the Cat crazy and drive around the screen!

Assignment Submission

1. Attach your Word document with screenshots showing the code for each program.

2. Submit in Blackboard.