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.

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.

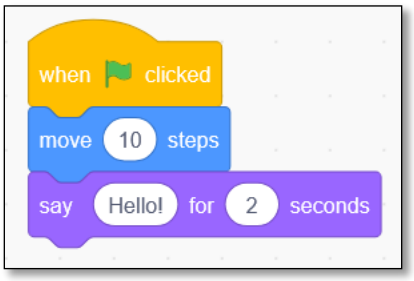
The Code Ladder

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 1.1 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 and experiment with the program. The Cat moves and speaks.
6. Change the numbers to see the effect. Try a negative number for movement.
7. Take a screenshot and paste it into a Word document.



Tutorial 1.2 Move the Cat with Variables

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

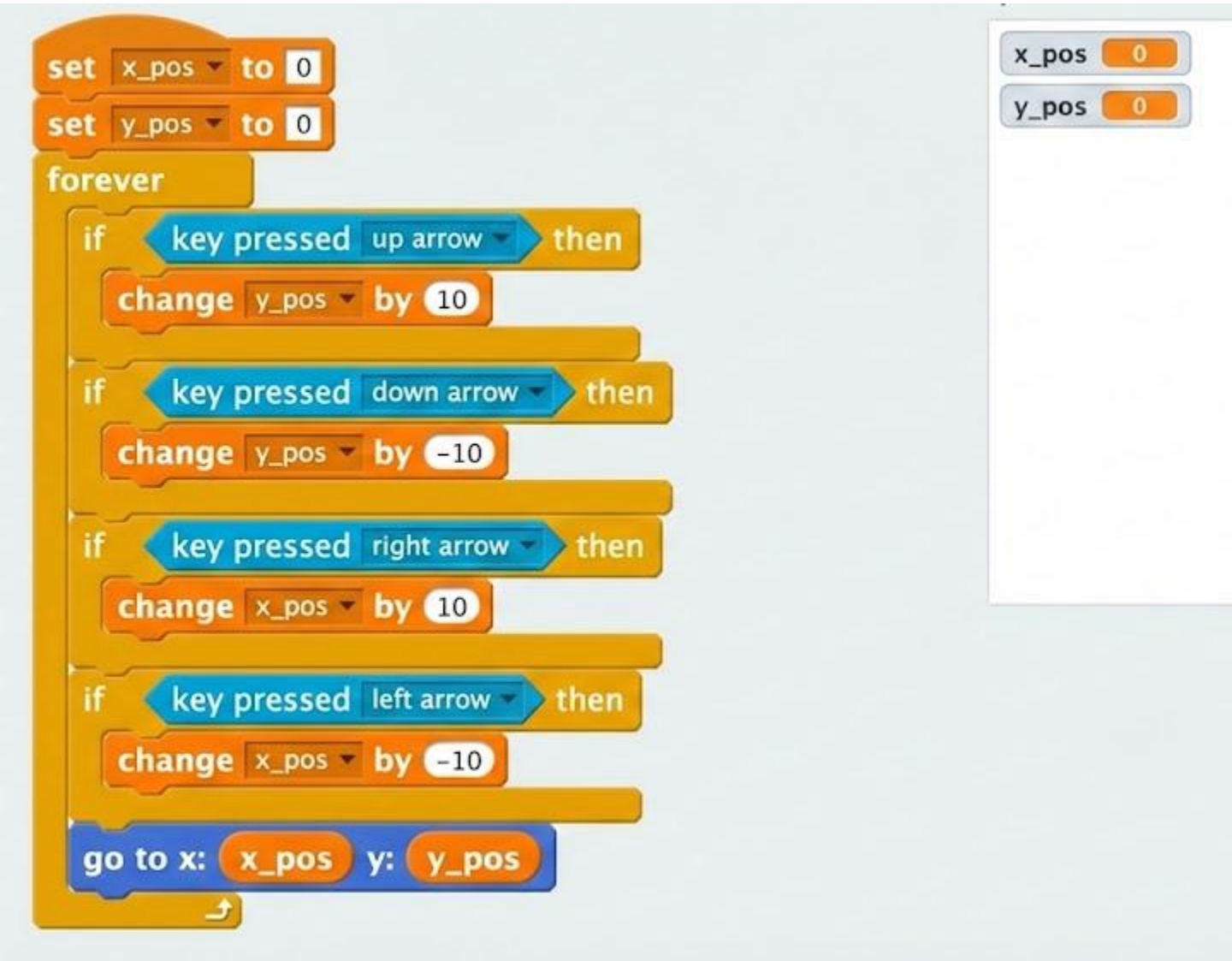
- **Output:** The computer holding up a sign for you to read.

- **Input:** The computer handing 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.



3. Conditionals (The Fork in the Road)

Logic is about making decisions. 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:

Plaintext

```
if <(age) > (18)> then
  say [Vote!]
else
  say [Wait a few years.]
```

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.

---

4. Loops (The Hamster Wheel)

Computers are great at doing boring tasks repeatedly without complaining. 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:

Plaintext

```
repeat (5)
  say [Hello]
```

Python:

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:

Plaintext

```
repeat until <(answer) = [yes]>
  ask [Are we there yet?] and wait
```

Python:

Python

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

---

5. Lists (The Egg Carton)

Sometimes you have too many variables to name them all individually (score1, score2, score3...). Instead, we use a **List** (called an **Array** or **List** in Python).

The Analogy

A variable is a single box. A list is a pill organizer or an egg carton. It's one object that holds many separate slots, numbered 0, 1, 2, 3...

The Code Ladder

Concept	Scratch Block	Python Code
Create List	(Make a List named "Grocery")	grocery = ["Eggs", "Milk", "Bread"]
Add Item	add [Apples] to [Grocery v]	grocery.append("Apples")
Get Item	item (1) of [Grocery v]	grocery[0]

Warning: Scratch starts counting at **1**. Python (and almost all programming languages) starts counting at **0**.

- Scratch: Item 1 = Eggs
- Python: Item 0 = Eggs

Summary Checklist

As you move from Scratch to Python, keep this translation guide in mind:

1. **Green Flag** $\rightarrow$ def main(): (The start of your program)
2. **Orange Blocks** $\rightarrow$ variables = value
3. **Yellow Blocks** $\rightarrow$ if, while, for (Control Flow)
4. **Blue "Ask"** $\rightarrow$ input()
5. **Purple "Say"** $\rightarrow$ print()

Your First Python Assignment

Write a Python script that asks the user for a password.

- **If** the password is "secret", **print** "Access Granted."
- **Else**, **print** "Access Denied."

Assignment Submission

1. Attach your Word document with screenshots showing the code for each program.
2. Submit in Blackboard.