INTRODUCTION TO PYTHON

• PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. It is used for:

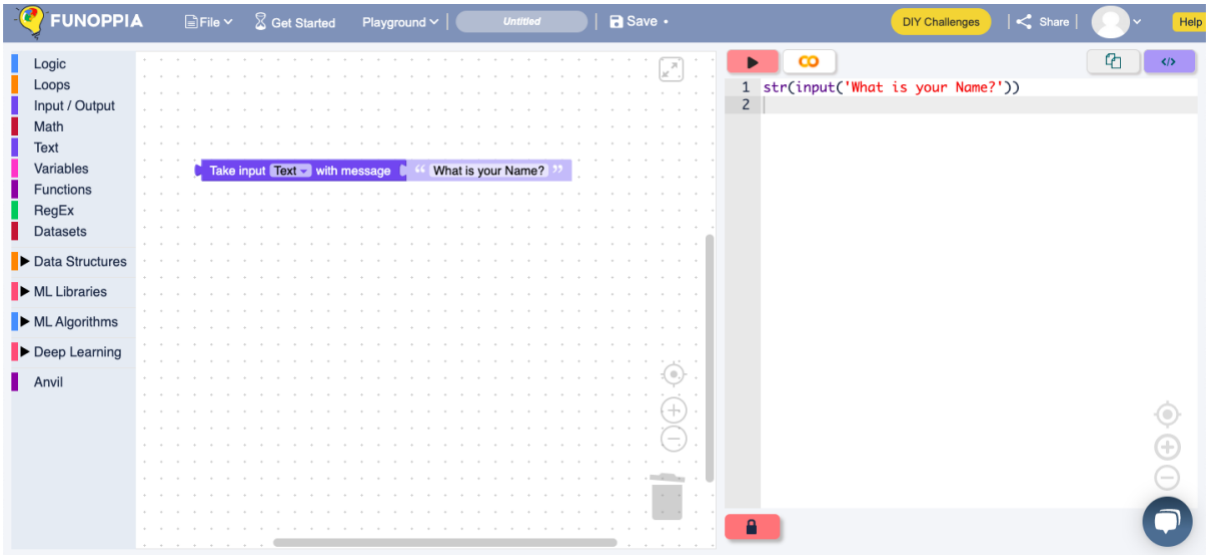
- web development (server-side),
- software development,
- mathematics,
- system scripting.

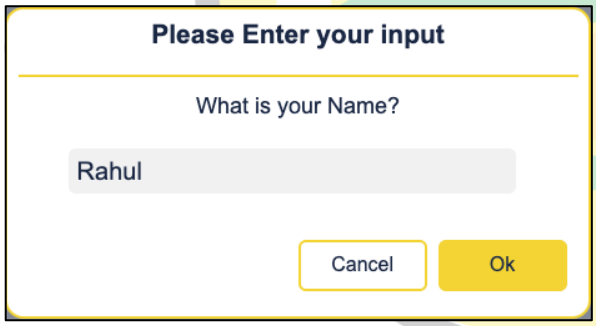
• BASICS OF PYTHON

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Python has a simple syntax similar to the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-oriented way or a functional way.

• INPUT

The program asks the user to enter something into the system is known as input.





As you can see in the above image on the right is the code whereas on the left is the block where I set it as a text Input.
On the right of this block we can see that it asks for our input which is 'What is your Name?', to which I have input a name.

This is known as input. The source code is written as :

str(input("What is your Name?"))

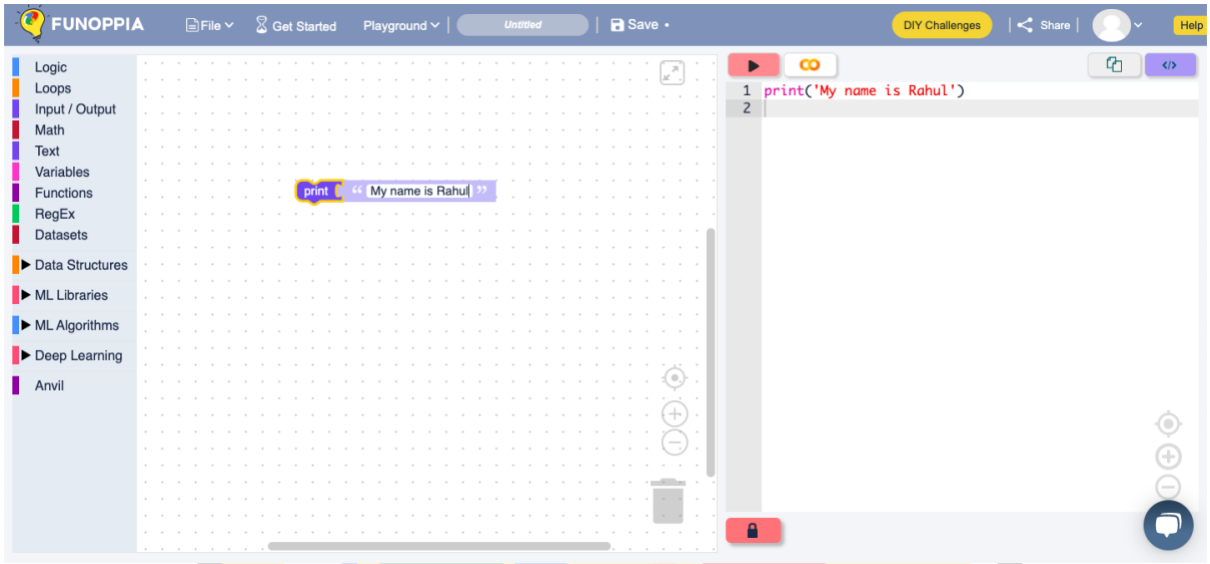
where str means string, a set of letters.

TASK – https://www.funoppia.com/funide/index.php

Try writing your name as well and your age as input.

• OUTPUT

We use the print() function to output the data on the output screen.



As you can see in the above image on the right is the code whereas on the left is the block where I set it as a text output.
On the right of this block we can see that it asks for our output which is 'My name is Rahul'.

This is known as input. The source code is written as :

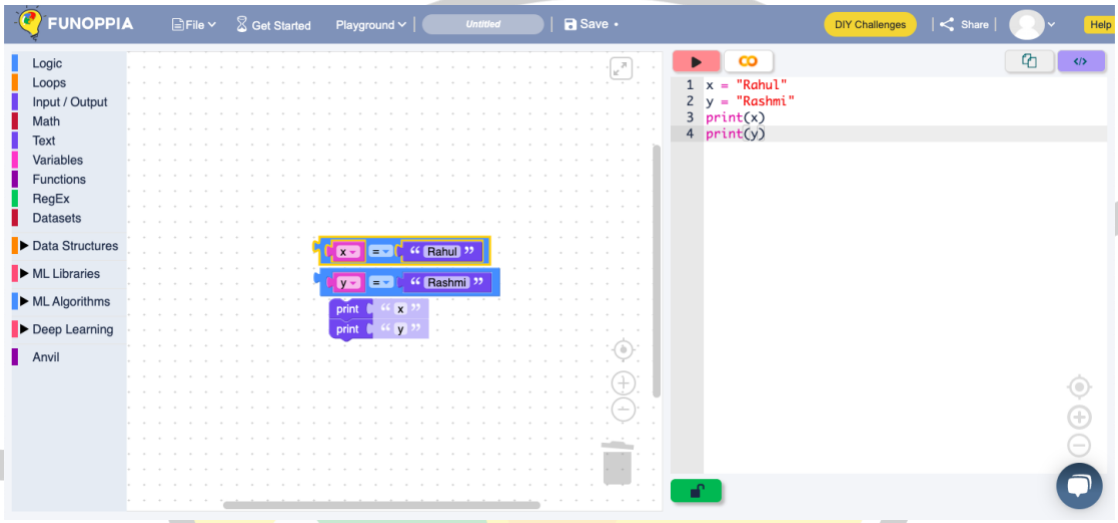
print("My name is Rahul")

TASK – https://www.funoppia.com/funide/index.php

print your name and age in different lines.

• VARIABLES

Variables are used to store any information which is provided to it.
Eg – There are different students in class, rather than calling them by their name they are called as x and y to make calling them out simpler.
There are 2 students called Rahul and Rashmi
We named Rahul as x and Rashmi as y.
So whenever we will call them they will be referred as x and y.



The output of the following code
Will be that the names are printed
As Rahul and Rashmi are denoted
as x and y.

```
Rahul
Rashmi
```

This is known as variables. The source code is written as :

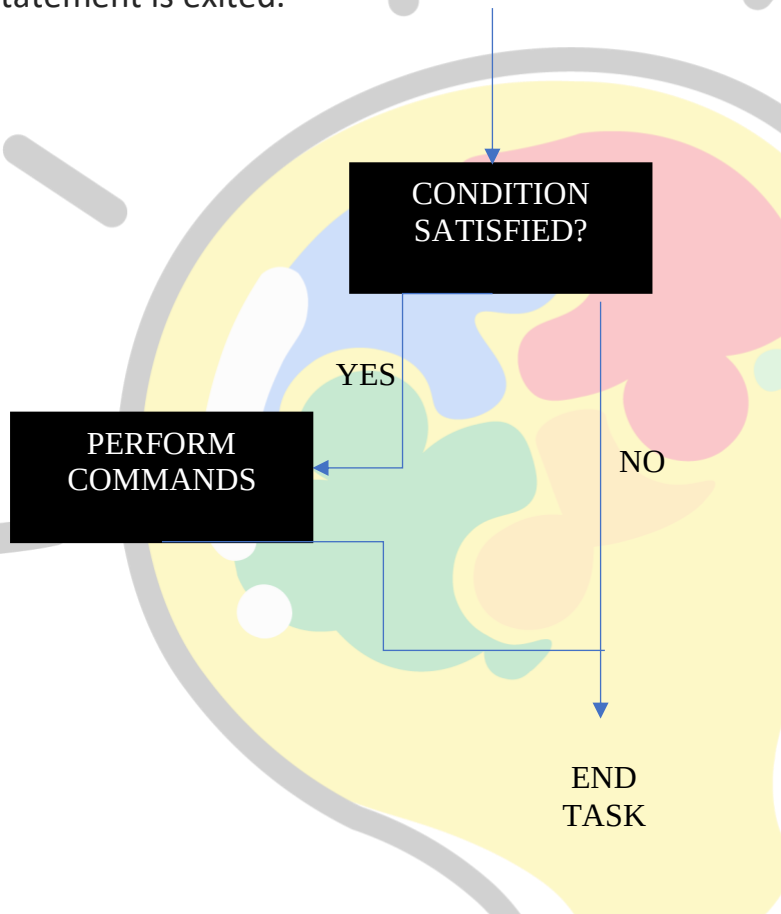
x=”Rahul”
y=”Rashmi”
print(x)
print(y)

TASK - https://www.funoppia.com/funide/index.php

Try putting in roll number and name of student and printing them out.

• CONDITIONAL STATEMENTS

Here is a flowchart to understand how a conditional statement works.
First the condition is seen then if it is true then the next set of conditions or instructions is followed, then again the condition is checked, if it is false then the statement is exited.



When a particular command has different types of sub-commands like different proper conditions.
There are 4 types of if statements –

1. If – 1 condition is given.
2. If Else – 2 conditions are given.
3. If If-Else if – More than 2 conditions are given
4. Nested if – Condition inside a condition.

1. If -

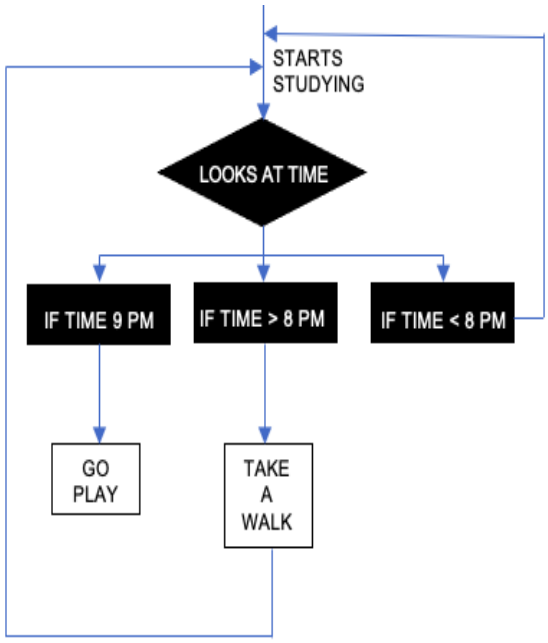
A boy is studying and doing his homework for a particular time. He sits to do his homework at 7 pm and his mother has told him to study till 9 pm. At 9 Pm he looks at the clock and sees the time. As the condition was to study till 9 pm he will get up and go to play.

2. If Else –

The boy is studying and he sees the clock and it's 8 pm he will continue to study as the condition was to study till 9 pm and it is not 9 pm currently he will continue studying.

3. If If-Else If –

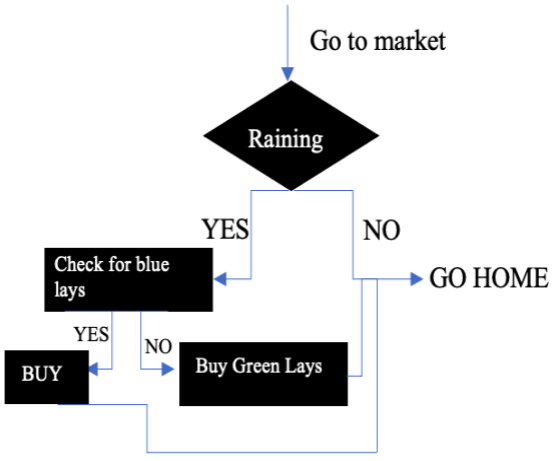
The mother of the boy had said that he can take a walk after 8 pm, He checks the time at 8:15 pm and hence he takes a walk, this was the third condition. If the time would have been 7:45 pm he would have to sit study.



4. Nested if –

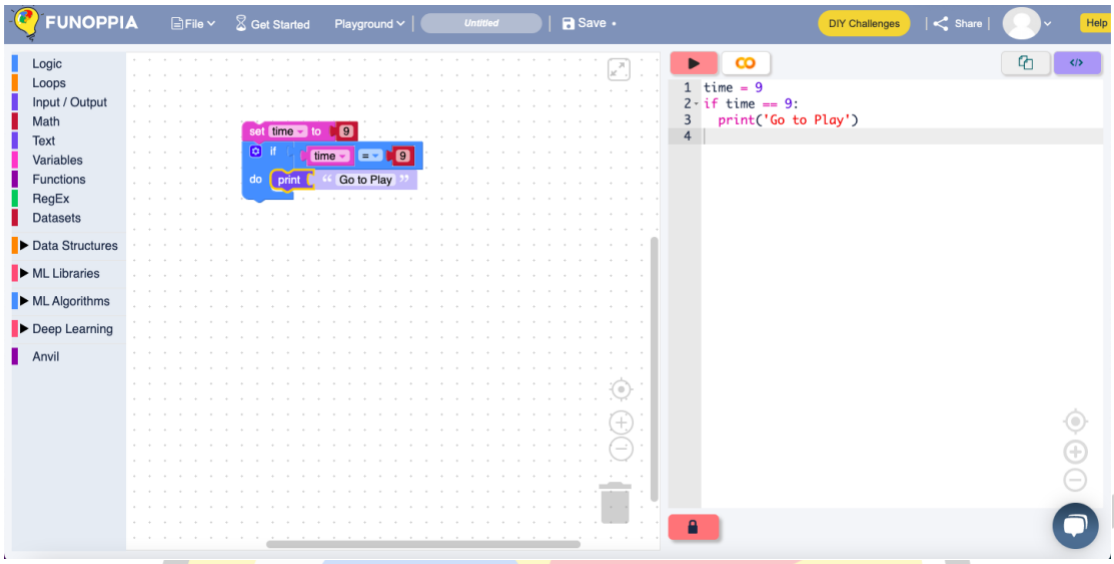
When there is a condition inside a condition.Your mom asked you to go to the market and get 1 packet of blue lays if it is raining Otherwise come home.Then check If Blue lays are available, if not then buy Green lays.

So you will go to the market and see if it is raining or not then check availability of particular packet of chips.



- **CODES**

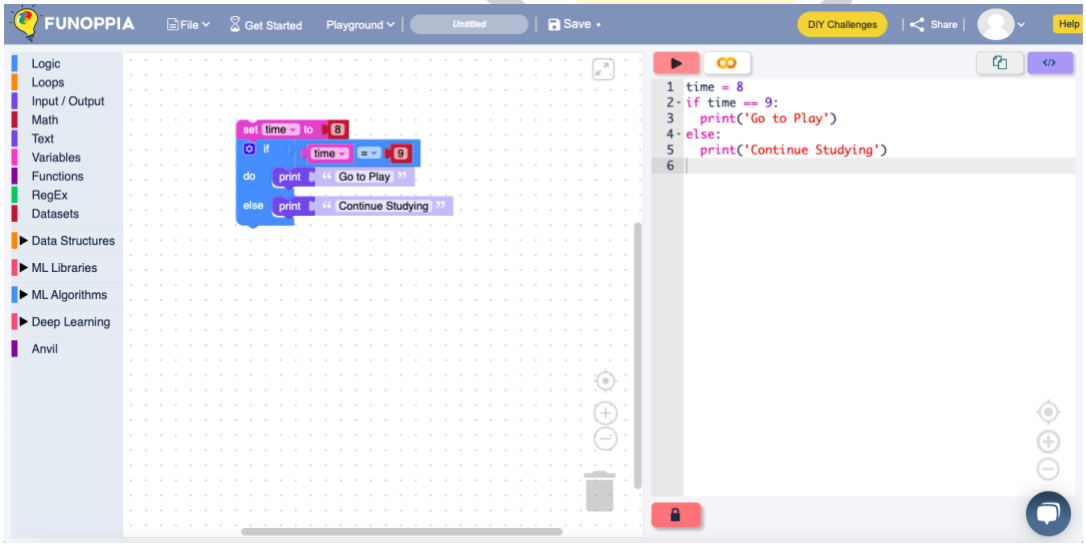
1. If statement

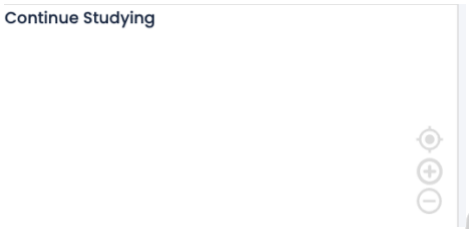



```
Go to Play
```

The above code shows the working, if the time would be equal to 9 pm then you can go to play, hence the output shows that you can go out to play.

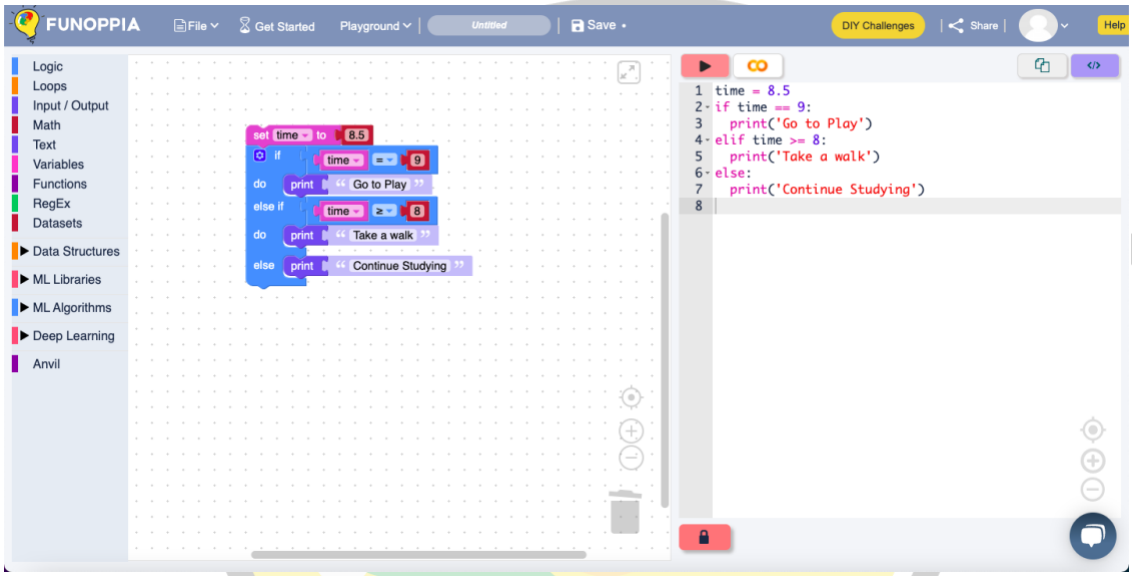
2. If – Else statement





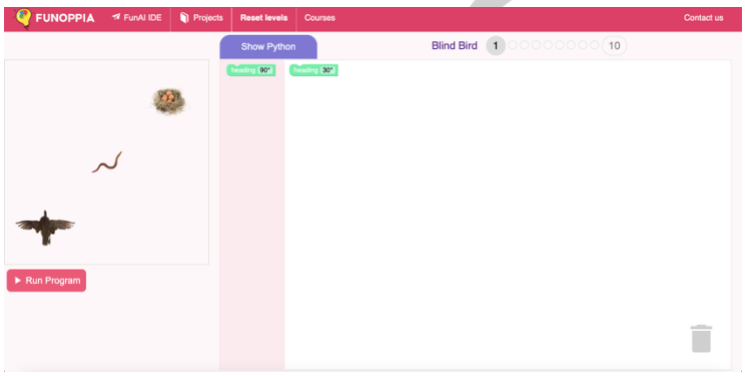
The above code Shows that if the time is 8 you have to continue studying, and the output is also continue studying.

3. If – Else - if statement



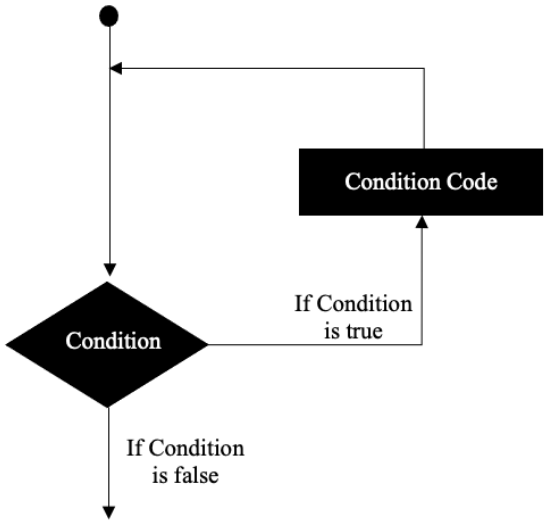
The above code Shows that if the time is greater than 8 but less than 9 then you can take a walk and continue studying.

TASK - https://funzone.funoppia.com/bird.html
Complete the Blind Bird Game



• LOOPS

What is a loop? A loop means to go round and round to do a particular task till a certain condition arises in it and the set of commands is followed in the same order as given.



Eg- Your mom tells you to go to the kitchen and get a glass of water for the guests, but you can only carry one glass at a time. So you will take multiple trips to the kitchen and keep getting water.

There are 2 types of loops –

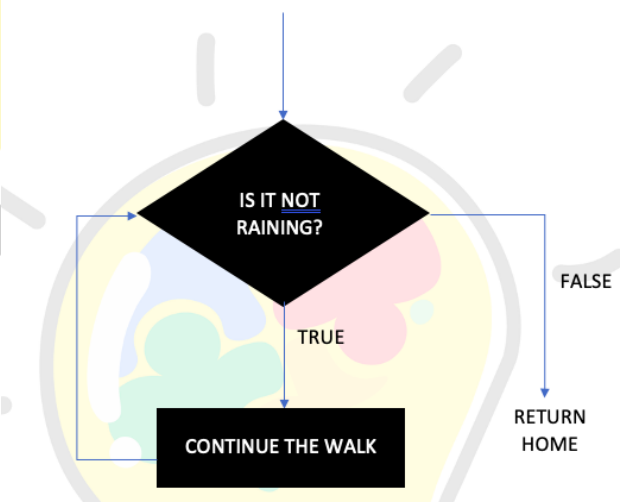
a) While loop
b) For loop

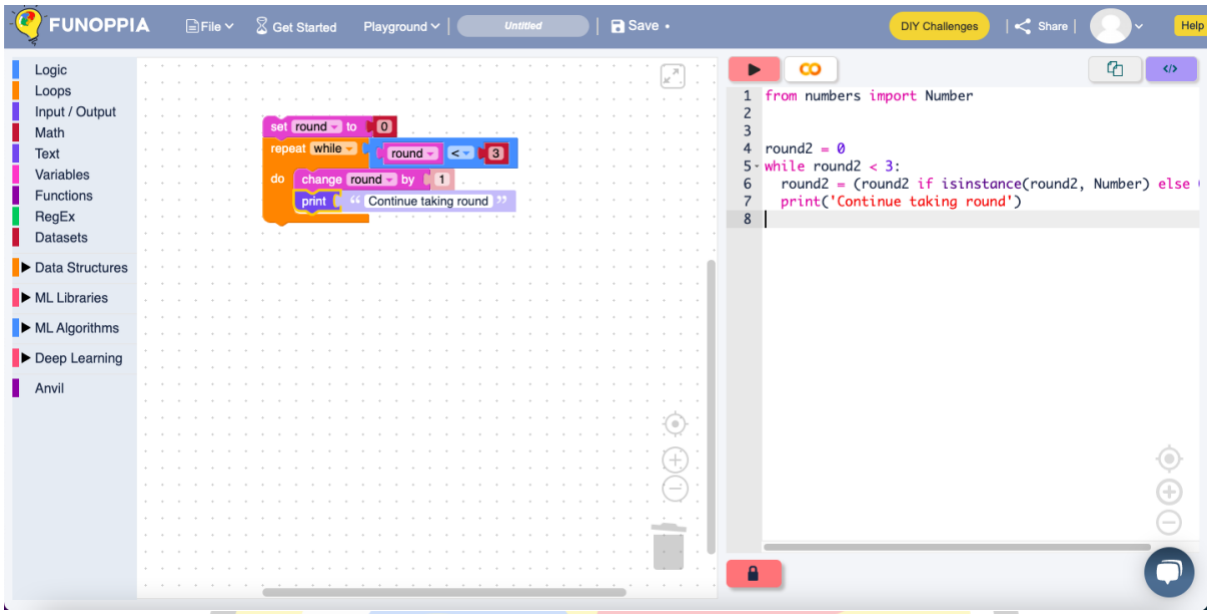
1. WHILE LOOP

In while loop, the set of instructions/commands is repeated until the condition given is TRUE, when the situation becomes FALSE, the loop stops/exits.

Eg- your mom has asked you to take a walk in your society till you complete 3 rounds. After completing one round you will check which round number it is if it is not round number 3 you will continue taking a round in your society. But if it is round number 3 you will return home.

See the flowchart for proper understanding





2.FOR LOOP

For loop refers to a particular condition which needs to be fulfilled till a particular number or times.

Eg – like the first example of getting water for the guests your mom asks you to get water for your relatives. There are 8 guests present in your house, you will continue to get water 8 times till everyone gets a glass of water. This is known as a for loop where you have to get a glass of water repeatedly till all the 8 guests get a glass.

It means that –

For (get glass of water)

Give glass to first guest

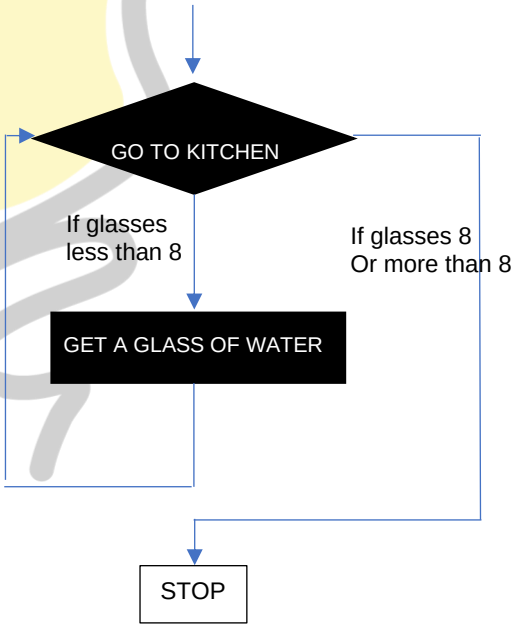
See if 8 glasses have been given

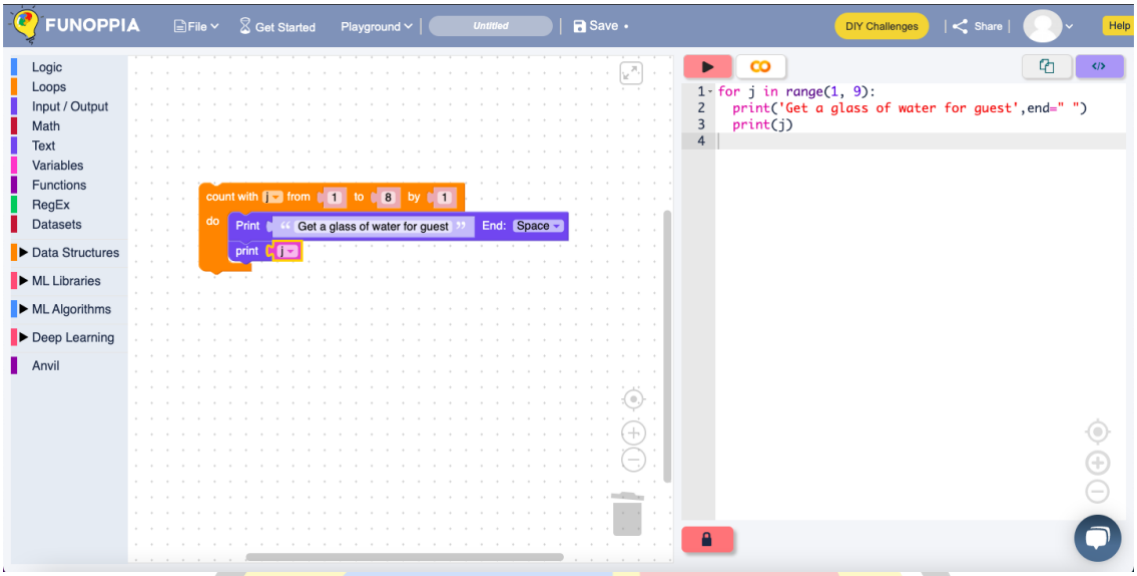
If yes then stop

If no go to the kitchen and get a glass

Repeated till everyone gets a glass of water

Therefore, you will give 8 glasses of water in total to the guests.

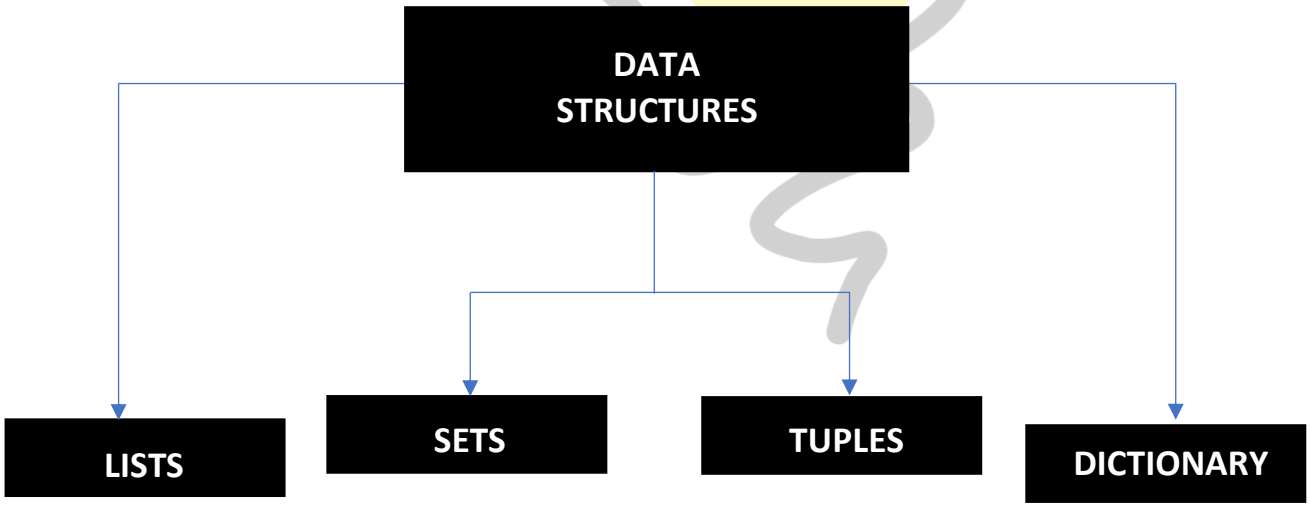




• DATA STRUCTURES

In python data structures include lists, sets, tuples and dictionary. Each of them are unique in their own way. They are containers that organize and group data according to the type.

Data structures are differentiated on the basis of mutability and order. Mutability refers to the ability to change an object after its creation. Mutable objects can be modified, added, or deleted after they have been created, while in immutable objects cannot be modified after the creation. Order in this context relates to whether the position of an element can be used to access the element.

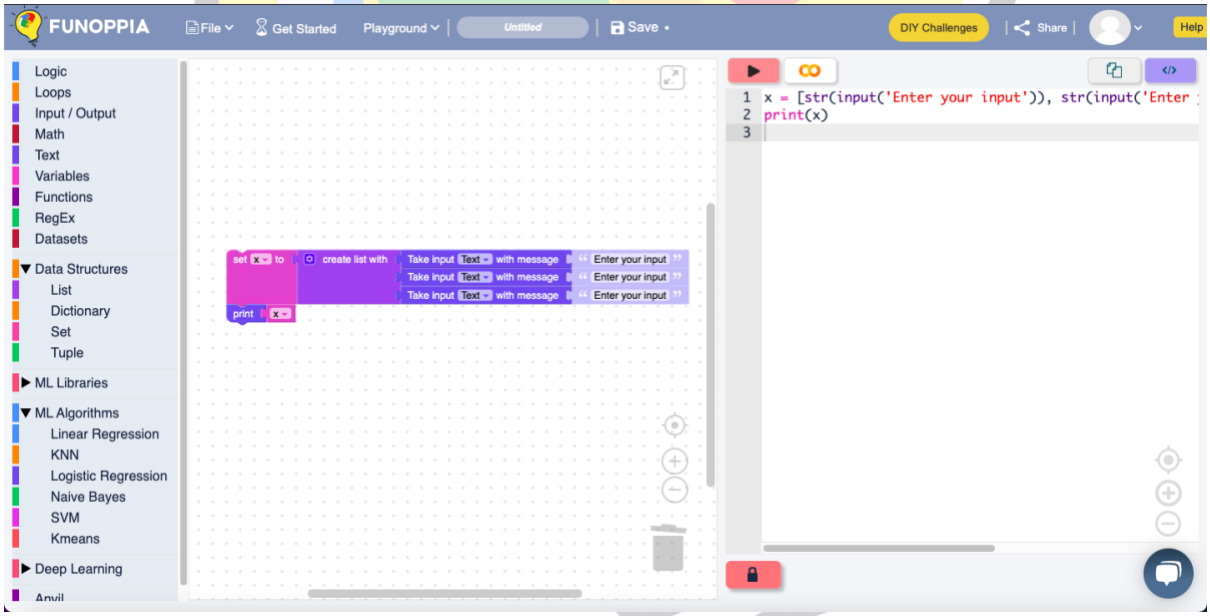


1. LISTS

A list is defined as an ordered collection of items, and it is one of the essential data structures when using Python to create a project. The term “ordered collections” means that each item in a list comes with an order that uniquely identifies them. The order of elements is an inherent characteristic that remains constant throughout the life of the list. Eg- your mom asks you to go to the grocery to get some daily needs for your home, she gives you a list of items properly ordered.

Lists are mutable in nature.
Lists can be created as –

List A = [item 1, item 2 … item n]



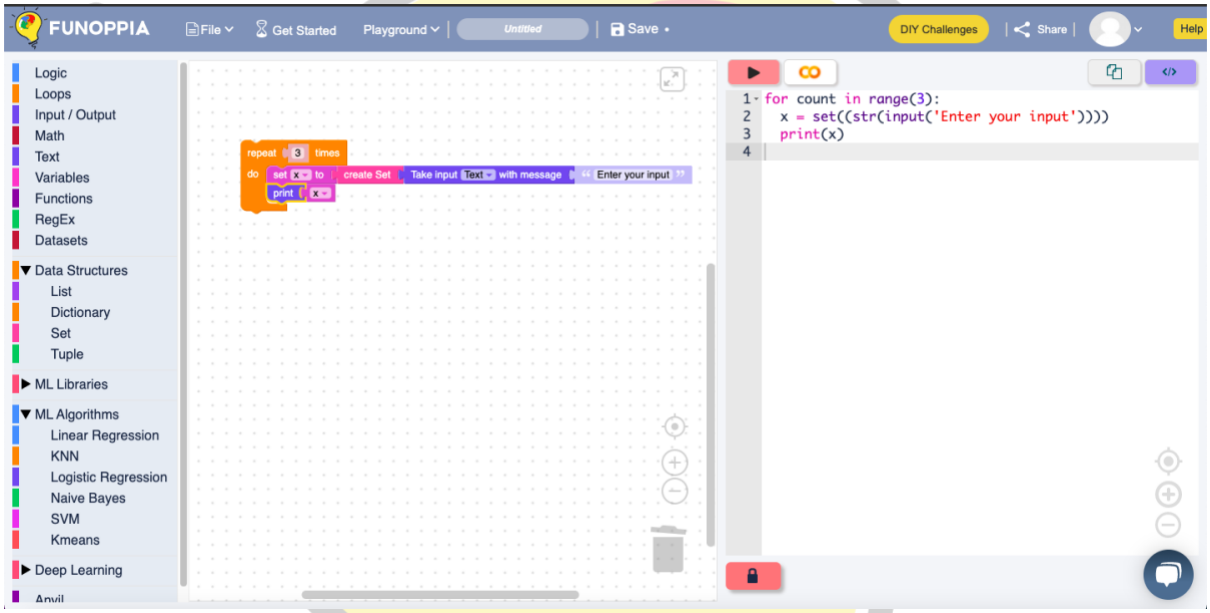

```
['2', '3', '4']
```

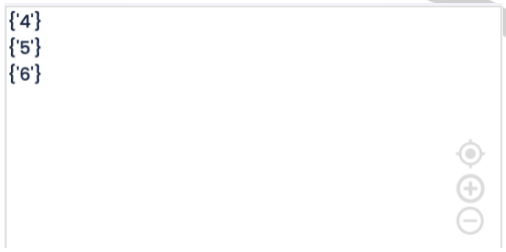
The above code shows how the list is created then after putting the inputs the list is printed out as it can be seen in the output on the left.

2. SETS

A set is defined as a unique collection of unique elements that do not follow a specific order. Sets are used when the existence of an object in a collection of objects is more important than the number of times it appears or the order of the objects. Unlike tuples, sets are mutable – they can be modified, added, replaced, or removed. A sample set can be represented as follows:

set_a = {“item 1”, “item 2”, “item 3”,….., “item n”}





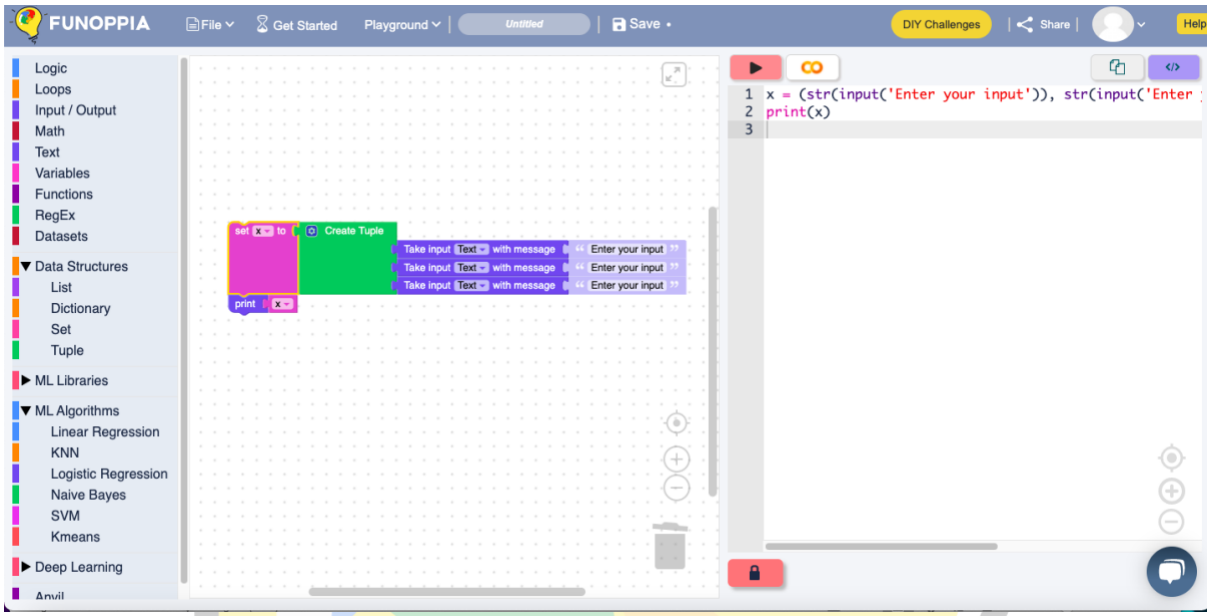
In the above code we can see that we have made a set and asked to repeat the same procedure three times and then print, hence 3 different sets are printed.

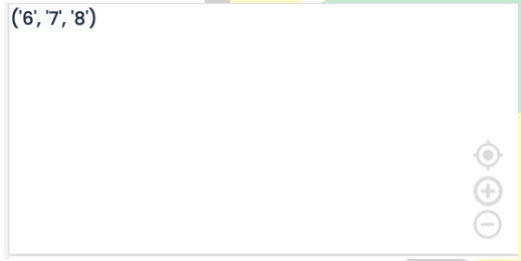
3. TUPLES

A tuple is a built-in data structure in Python that is an ordered collection of objects. Unlike lists, tuples come with limited functionality.
The primary differing characteristic between lists and tuples is mutability. Lists are mutable, whereas tuples are immutable. Tuples cannot be modified, added, or deleted once they’ve been created. Lists are defined by using parentheses to enclose the elements, which are separated by commas.

Tuples are immutable in nature.
Tuples can be created as –

tuple_A = (item 1, item 2, item 3,..., item n)



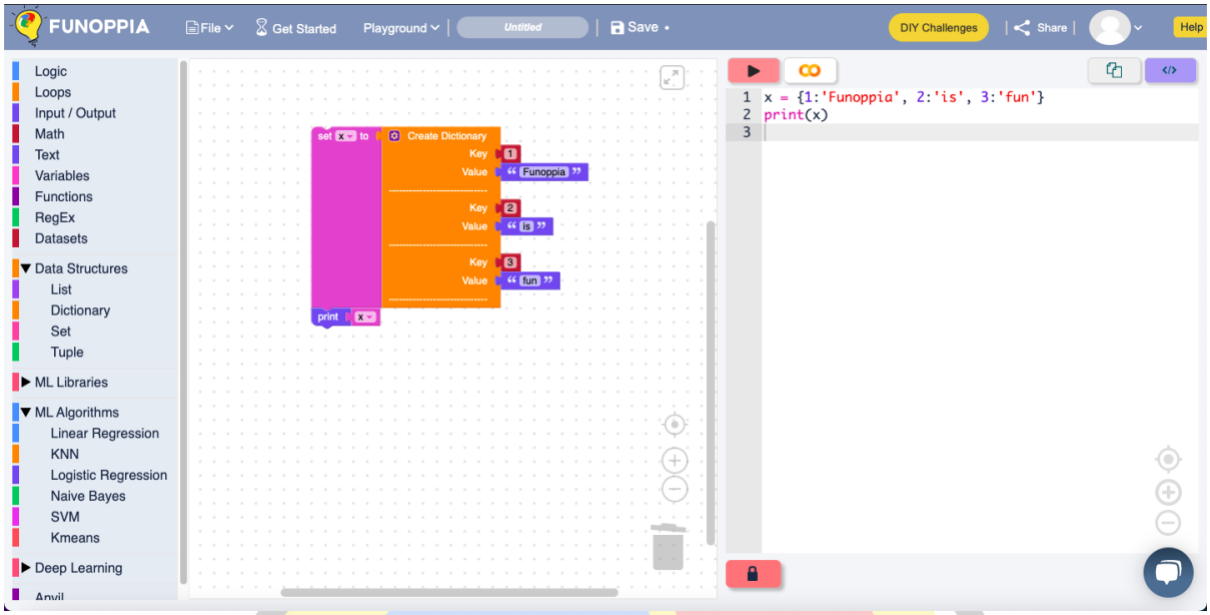


The above code shows how the tuple is created then after putting the inputs the list is printed out as it can be seen in the output on the left.

4. DICTIONARY

Dictionary in Python is an unordered collection of data values, used to store data values like a map, which, unlike other Data Types that hold only a single value as an element, Dictionary holds **key:value** pair. Key-value is provided in the dictionary to make it more optimized.
Dictionary keys are case sensitive, the same name but different cases of Key will be treated distinctly

DictA = dict({1: 'Funoppia', 2: 'is', 3:'fun'})

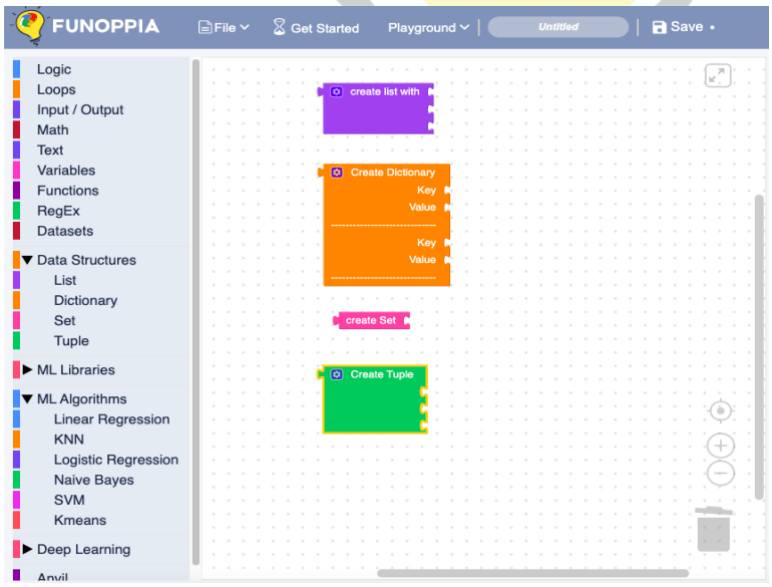



```
{1: 'Funoppia', 2: 'is', 3: 'fun'}
```

The above code shows how a dictionary is created and then it is printed, the output code is on the left.

TASK – https://www.funoppia.com/funide/index.php

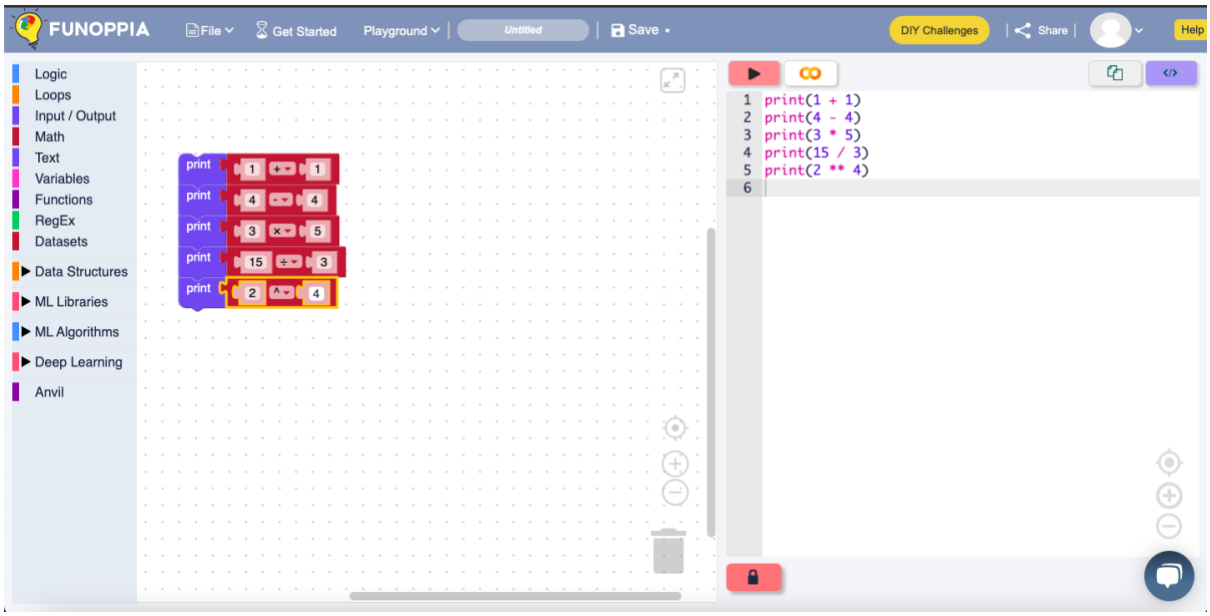
Try creating multiple lists, sets, tuples and dictionary and print them out in the IDE.



• TYPES OF OPERATORS

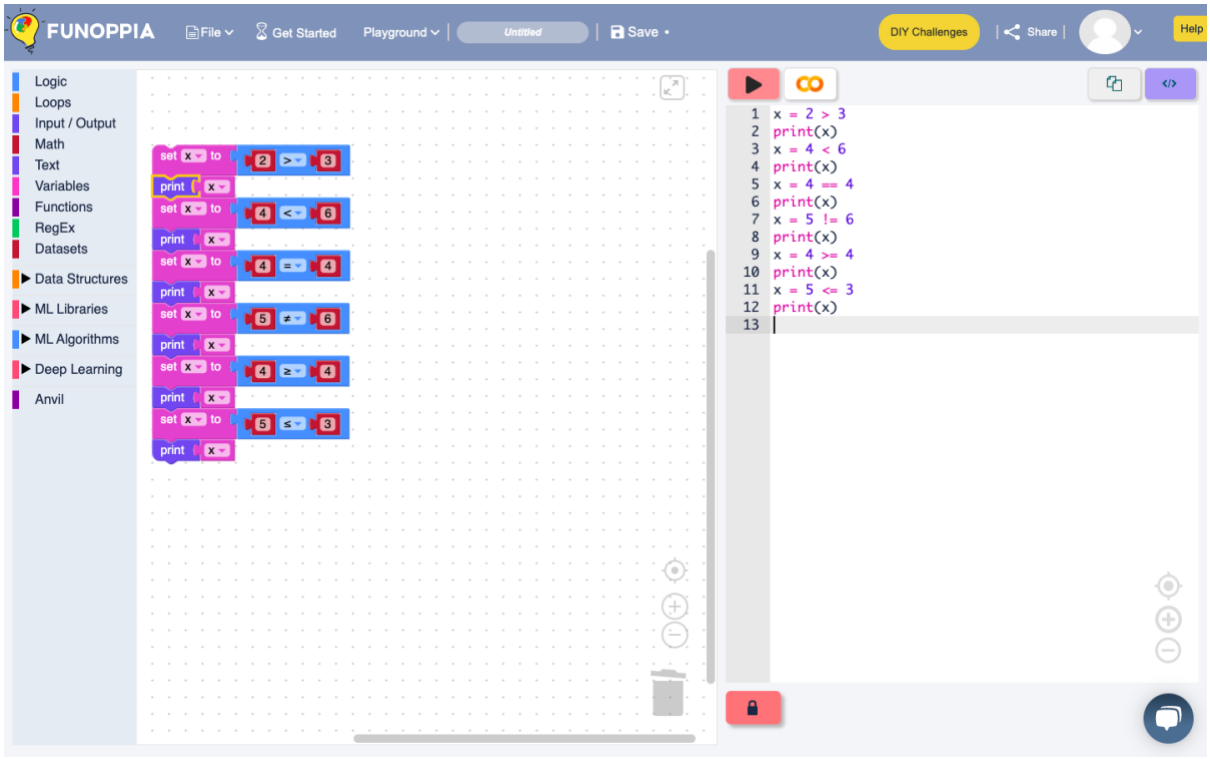
An operator is a symbol that will perform **mathematical** operations on **variables** or on **values**. Operators operate on **operands** (values) and return a **result**.
Python has 4 types of operators that you can use:

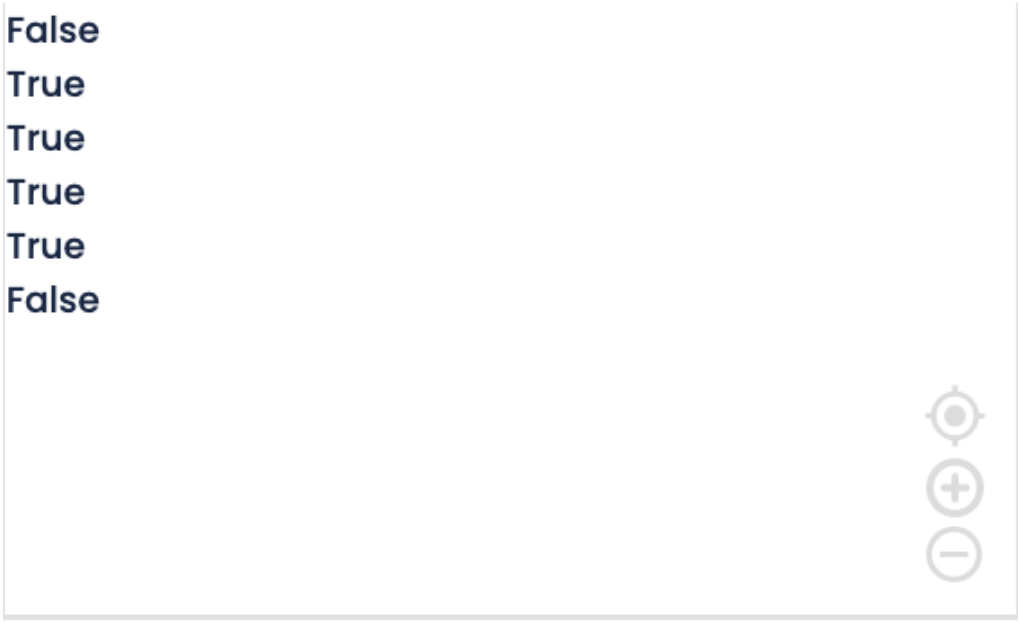
1. Arithmetic Operators – Basic maths operations.
    - + (Addition)
    - – (Subtraction)
    - * (Multiplication)
    - / (Division)
    - ** (Exponentiation)



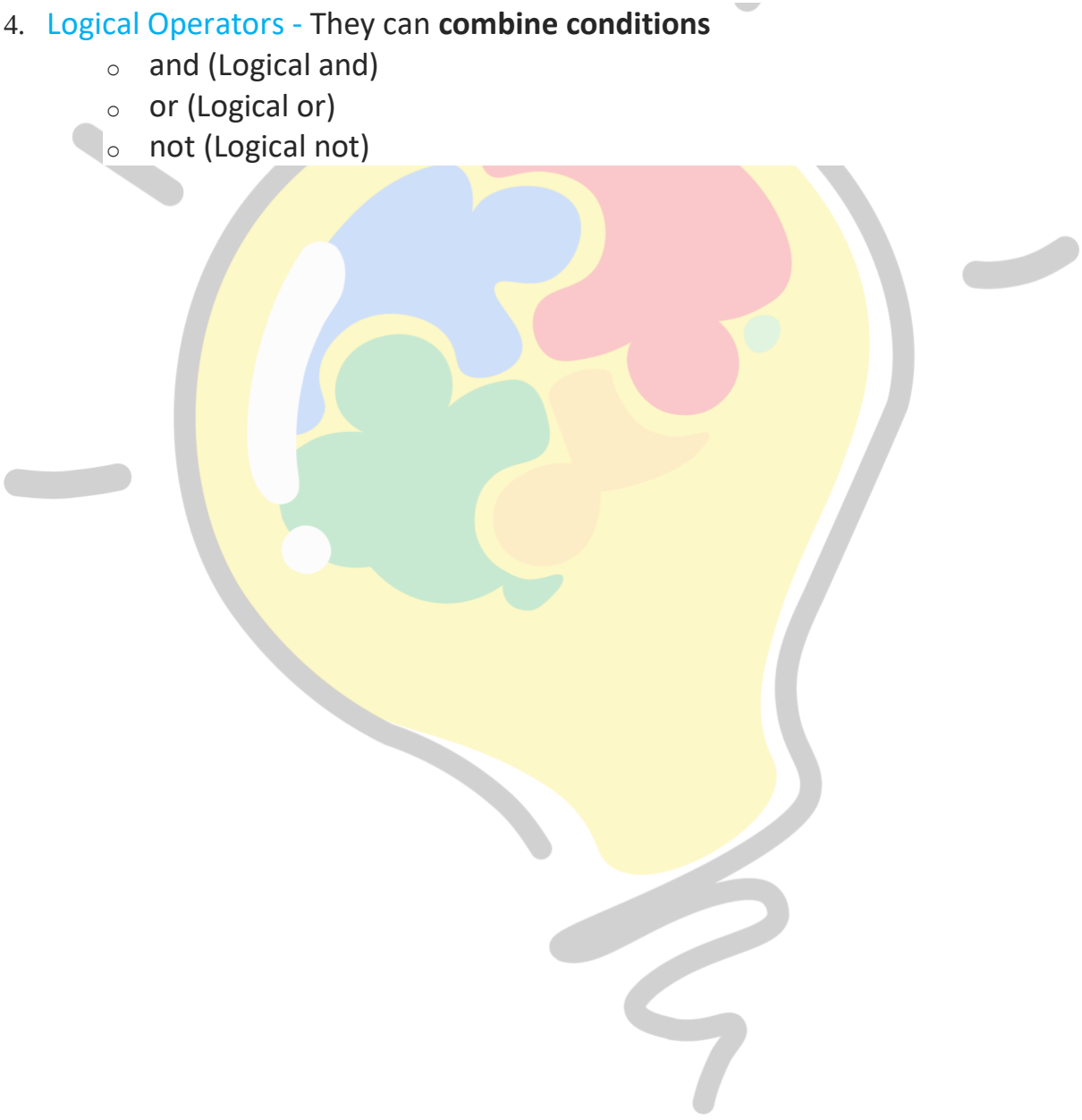

```
2
0
15
5.0
16
```

2. Relational Operators - They compare values.
    - > (Greater than)
    - < (Less than)
    - == (Equal to)
    - != (Not equal to)
    - >= (Greater than or equal to)
    - <= (Less than or equal to)





3. Assignment Operators - They perform an **operation** and **assign** a value.
    - = (Assign)
    - += (Add and assign)
    - -= (Subtract and assign)
    - *= (Multiply and assign)
    - /= (Divide and assign)
    - %= (Modulus and assign)
    -
4. Logical Operators - They can **combine conditions**
    - and (Logical and)
    - or (Logical or)
    - not (Logical not)



PRACTICE QUESTIONS

Q1. Give more examples of

a) If.
b) If Else
c) If If-Else if
d) Nested if

Q2. Give more examples of

a) While loop
b) For loop

Q3. Create code for the following

a) Lists
b) Sets
c) Tuples
d) Dictionary.