Lesson Plan

**More on operators, user input &**

**Control statements**



Summary

1. Subject(s): Comparison and Boolean opearators, Input function and control flow
2. Objective: To get students familiar with the above concepts
3. Time Allotment: 2 Hrs

Implementation

# Learning Context

We have seen what programming is, why we need to learn it both as a developer or other field specialist.

Then we looked what programming languages are, why we chose Python, where Python is best utilized.

Finally, we wrote some code and understood variables assignment and arithmetic operators as well as print statement with f string.

So today we will dig a little further and see the input function, comparison operators, Booleans, and controlling the flow of our code.

# Procedure

1. Anticipatory Set

#########################

b. Direct Instruction

**Input function and Casting**

name = abebe

print(f"Hello, {name}")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

In the above code from our previous class, we declared a variable assigned a value to it and printed it. But this won’t be practical, because our code won’t be flexible. We want it to work for any name of any user not just abebe. So let’s modify it a little

name = input("What's your name? ")

age = input("And your age? ")

print(f"Hello, {name}\nHow is {age} years old treating you.")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

As for casting:

age = int(input("How old are you? "))

length = int(input("How long is it? "))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The above code takes input from users, converts the input in to integers then saves that integer into the respective variables.

This way we will be able to accept input from users, convert it if it needs casting, store that input into a variable to use it later on. Okay now let’s move on to something different. We’ll get back to how it all connects but for now let me take you to operators.

**Comparison Operators**

We had seen assignment and arithmetic operators. But what if we needed to compare to variables? Say the age of abebe and kebede. That is where comparative operators come in. The main ones are:

* **==,** - **!=**
* **>=,** - **>**
* **<=,** - **<**

Here are a few examples:

a = 2

b = 3

a == b

b >= a

b < a

name1 = "Abebe"

name2 = "abebe"

name1 != name2

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Booleans**

Okay so what do you thing is the result of the above lines of code? It’s going to be either true or false right. That’s what we call a Boolean, a data type that is either true or false.

a = 2

b = 3

a == b #False

b >= a #True

b < a #False

name1 = "Abebe"

name2 = "abebe"

name1 != name2 #True

**not**, **and**, **or**

**not** a == b #False

b >= a **and** name1 != name2 #True

a == b **or** b < a #False

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Alright any questions?** On Booleans, bool operators, comparison operatives and the input function.

**Conditionals: how we control flow**

Now what can we do with the Booleans we get from comparison operatives is use them as conditions for whether to execute a code or not. Look at the below code for example:

a = 2

b = 3

if b > a:

print("'B' is greater than 'A'")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

number = int(input("Input a number: "))

if number > 0:

print("The number is positive")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The print code lines only get executed if the above condition is true. If it turns out to be false the statements under an if condition is skipped. But what if we wanted to test multiple conditions?

number = int(input("Input a number: "))

if number > 0:

print("The number is positive")

else:

print("The number is not positive")

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Even now tho, I feel like there is something missing. Can we make it simpler? What is not positive? Negative? Then what about zero? Here’s a possible solution:

number = int(input("Input a number: "))

if number > 0:

print("The number is positive")

elif number < 0:

print("The number is negative")

else:

print("The number is Zero")

Now it is better! This is how we check for multiple conditions one at a time. If you want to check a single value against multiple conditions, there is an alternative way called match statements. I will leave that to you guys as an assignment. We can also use if statements in inline form. What that means is basically write it all in one line like this:

number = int(input("Input a number: "))

print("Positive") **if** number >= 0 **else** print("Negative")

c. Guided Practice

* Prompts user for his age, then either allow them to watch the movie or prohibit them depending on their age
* Prompts for the users class grade, then if it is 90 and + excellent, 80 and + very good, 70 and + good, 60 and + needs improvement, 59 and - Fail!

1. Check for Understanding
2. Independent Practice
3. Closing

Alright students we have seen how to get input from users, how to use that input to decide the flow of the code, how to convert one type of data to the other, how to compare to values and variables, and how Booleans work. This are fundamental concepts you will encounter everyday when programming so try to get used to them and do lot’s of practices. Next time we will see data structures such as lists, and for and while loops

Till then have a great time!