1. **Introduction**

**Goal of this Note:**

1. Understand Python Division, Floor Division and Modulo
2. Explore Python on the Jupyter.
3. Variables
4. Lists
5. Conditionals
6. **How to Approach Python for the First Time**

Recall that Python is a high-level programming language, but fortunately its purpose is to be friendly to beginners. So, here’s a helpful tip that might sound childish but is very effective if you have never coded before in your life: pretend that Python is not a program but an unintelligent robot. You can command this robot to perform tasks but since the robot happens to be so doltish, you reduce your commands to generic statements in English that lack grammar.

1. **Division, Floor Division and Modulo**

**A picture containing text, screenshot, monitor, screen

Description automatically generated**

Thanks to CS 61A, we have the wonderful table above that summarizes the functionality of these operations. True Division (/) is the division we carry out normally. Floor division (//) returns the quotient. The modulo(%) returns the modulus of a number i.e., it returns the remainder.

[DEMO on JUPYTER]

1. **Variables**

Variables are containers that store the data we want to keep track of. In a very ‘*obnoxious’* manner, variables have types. To find the variable type, we use the type() function where the variable is passed in as an input to the function. Mr. Python – an otherwise doleful robot – spares us with declaring our data types. Sadly, this is no act of mercy. While not declaring our data type can seem like a good thing, the caveat is that our variable type can change dynamically. If I declare a as an integer, I can change it to be a Boolean value in the very next step. So, we have to be a little careful with what data type our variables are. Coming to the data types themselves, refer to the table below to find the different data types:

|  |  |  |
| --- | --- | --- |
| Data Type | Explanation | Example |
| Integer (Int) | Positive or negative integer | A = 5 |
| Float | Decimal (floating point) number | A = 3.1415926 |
| Boolean (Bool) | True or False | A = True or A = False |
| List | Ordered List of Value | A = [1, 2, 3] |
| String (Str) | A List of Characters (text) | A = ‘Marvin the depressed robot’ |
| Dictionary (Dict) | A mapping of keys and values | A = {‘e’:2.718, ‘pi’, 3.141} |
| None | Nonetype (null or missing value) | A = None |

1. **Print Statement and Operations**

Naturally, print statement prints out results. If you are printing a string, enclose them with brackets and put single quotation marks around them for strings or type in the variable name you want to print.

Next, let’s review a few basic mathematical operations Python can carry out:

|  |  |  |
| --- | --- | --- |
| Operator | Name | Example |
| + | Addition | A + B |
| - | Subtraction | A - B |
| \* | Multiplication | A\*B |
| \*\* | Exponentiation | A\*\*B |
| == | Equal | A == B |
| != | Not equal | A!=B |
| > | Greater Than | A > B |
| < | Less than | A < B |
| >= | Greater Than or Equal To | A >= B |
| <= | Less than Or Equal To | A <= B |
| and | As a condition, it returns true if both statements are true.  If you enter raw conditions without control statements (if clauses), if there is one false statement, returns first False statement. If both are True, returns the last True Statement. | A > B and B > C |
| or | As a condition, it returns true if both statements are true.  If you enter raw conditions without control statements (if clauses), if there is one false statement, returns first True statement. If both are True, returns the last True Statement. | A > B or B > C |
| not | If result is True, returns false | not (A < B) |

[DEMO on JUPYTER]

1. **Lists**

A list is a data structure that acts like a container and stores multiple elements. Each element can be of any type, even a list itself. We write a list as a comma-separated list of expressions in square brackets.

|  |  |  |  |
| --- | --- | --- | --- |
| **List Elements** | 1 | 2 | 3 |
| **Forward Index** | 0 | 1 | 2 |
| **Reverse Index** | -3 | -2 | -1 |

Lists are accessed with their index. Usually, we use forward index, but if you want to reverse your list, you can use reverse index too. Forward Index starts from the start of the list and Reverse Index starts from the back of the list. It is the forward index subtracted by the length of the list

We can also slice lists. Slicing a list creates a copy of part or all of list.

The syntax to slice a list is ‘list[<start index>:<end index>:<step size>]’. This expression evaluates to a new list containing the elements of list:

1. Starting at and including the element at <start index>.
2. Up to but not including the element at <end index>.
3. With <step size> as the difference between indices of elements to include.

If the start, end, or step size are not explicitly specified, Python has default values for them. A negative step size indicates that we are stepping backwards through a list when including elements.

There are various methods you can use on List. If you want to add a single element to a python list, you use the append() method and if you want to add multiple elements to a list, you use the extend() method.

[DEMO on JUPYTER]

1. **Conditionals**

Conditionals are also known as control statements because they literally control the flow of your code. It has the following structure:

If CONDITION:

[Do Something]

elif CONDITION:

[Do Something Else]

else:

[When everything else fails, do this]

[DEMO on JUPYTER]

1. **Great Rule of Equality**

What happens when I instantiate a variable as another variable?

Table

Description automatically generated with medium confidence

For instance, we want to run the following code above. What is happening?

1. I instantiate a to be an integer 3.
2. I instantiate b to be an integer 3
3. I instantiate c to be a list [1,2,3].
4. I instantiate d to be c.
5. I make a copy of c and assign it to e.

My environment diagram/output of the code run above looks like the following:

Diagram

Description automatically generated

The environment diagram above keeps track of the state of the program. Notice how c and d point to the same list. But since I created a copy of c and assigned that to e, e and c do not point to the same list. So while c and e have the same contents and return true when I run the command print( c==e), c is not e!

1. **The Go-Do-Your-Homework-Instead Section**
2. *The modulo is used in RSA encryption*
3. *Print actually returns None*
4. *Guido van Rossum named ‘Python’ while reading the Monty Python’s Flying Circus script so that the language sounded ‘cool and mysterious.’ Python was actually just a ‘hobby project’ for Guido van Rossum.*
5. *If you type ‘import this’ into your Python IDLE, you get a poem called ‘The Zen of Python’ by Tim Peters.*
6. *The name ‘git’ was given to it by Linus Torvalds who wrote the very first version. He described this version control system as “the stupid content tracker” and named it after a British-English slang for ‘unpleasant person’: git. It also stands for Global Information Tracker, and it does just that – track information between several local computers/networks using a global repository.*
7. *If you want a more in-depth exploration of terminal, console, command line, and shell, refer to the link below:*

[*https://askubuntu.com/questions/506510/what-is-the-difference-between-terminal-console-shell-and-command-line*](https://askubuntu.com/questions/506510/what-is-the-difference-between-terminal-console-shell-and-command-line)

1. [*https://mac.install.guide/homebrew/3.html*](https://mac.install.guide/homebrew/3.html) *is a guide to installing Homebrew.*
2. *If at any point during the installation, you get errors, don’t panic. Equip yourself with the mighty power of Stack Overflow! Almost any error you face has an answer on Stack Overflow. Whether you like it or not, most of coding is just looking up solutions for installation errors on Stack Overflow. However, the most important thing you need from that installation guide is just Anaconda so you must ensure that Anaconda is working on your computer.*