Data Types

- Data types specify the type of data like numbers and characters to be stored and manipulated within a program. Basic data types of Python are
  - Numbers
  - Boolean
  - Strings
  - None

- **Numbers**
- Integers, floating point numbers and complex numbers fall under Python numbers category. They are defined as int, float and complex class in Python.
- Integers can be of any length; it is only limited by the memory available.
- A floating point number is accurate up to 15 decimal places.
- Integer and floating points are separated by decimal points.
- Complex numbers are written in the form, x + yj, where x is the real part and y is the imaginary part.

Data Types

- **Boolean**
  - Booleans -useful in conditional statements. since a condition is really just a yes-or-no question, the answer to that question is a Boolean value, either True or False.
- The Boolean values, True and False are treated as reserved words.
- **Strings**
- A string consists of a sequence of one or more characters, which can include letters, numbers, and other types of characters.
- A string can also contain spaces.
- You can use single quotes or double quotes to represent strings and it is also called a string literal.
- Multiline strings can be denoted using triple quotes, ''' or """. These are fixed values, not variables that you literally provide in your script.

Click to add title

- print('ECE')
- print("ENGG")
- print('"ECE BENGALURU"')
- s=(''' ECE
- ENGG
- Bengaluru ''')
- print(s)
- print(''' ECE 'ENGG' Bengaluru''')

None

- *None* is another special data type in Python. *None* is frequently used to represent the absence of a value.
- money = None

Comments

- Comments are an important part of any program. A comment is a text that describes what the program or a particular part of the program is trying to do and is ignored by the Python interpreter.
- Comments are used to help you and other programmers understand, maintain, and debug the program.
- Python uses two types of comments: single-line comment

#This is single line Python comment

- multiline comments.

#This is

#multiline comments

#in Python

Reading Input

- In Python, *input()* function is used to gather data from the user. The syntax for input function is,

variable_name = input([prompt])

- >>> person = input("What is your name?")
- What is your name? Carrey
- >>> person
- 'Carrey'

Print Output

- The *print()* function allows a program to display text onto the console.
- print("Hello World!!")

  Hello World!!

- Two major string formats which are used inside the *print()* function to display the contents onto the console
- 1. str.format()
- 2. f-strings

str.format() Method

- The syntax for *format()* method is, ***str.format(p0, p1, ..., k0=v0, k1=v1, ...)***

p0, p1,... are called as positional arguments and, k0, k1,... are keyword arguments with their assigned values of v0, v1,... respectively.

- Formatted strings or f-strings were introduced in Python 3.6. A *f-string* is a string literal that is prefixed with "f". These strings may contain replacement fields, which are expressions enclosed within curly braces {}. The expressions are replaced with their values.

Example:

```
USN = input("Enter your USN: ")
Name = input("Enter your Name: ")
print("Student USN is {0} and  name {1}".format(USN, Name))
print("Student name is {1} and USN is {0}".format(USN, Name))
print("Student belongs to {Section}, ECE".format(Section = "6ABCD"))
print(f"Student USN {USN} and {Name}")
```

Type Conversions

You can explicitly cast, or convert, a variable from one type to another.

The *int()* Function

The *float()* Function

The *str()* Function

The *chr()* Function

The *complex()* Function

The *ord()* Function

The *hex()* Function

The *oct()* Function

The *type()* Function and is Operator

type(object)

The *type()* function returns the data type of the given object

type(6.4)

<class 'float'>