***Variables and Data Types in Python:***

***Variables****:* Variables are like containers that holds data. Very similar to our containers in Kitchen that holds Sugar, Salt, Rice etc.

Creating a Variable is like creating a place holder in memory and assigning it some value. In python its as easy as writing:

A = 1

B = True or False

C = “Ravi”

D = None

***Data Type****s:* Data type species the type of data that variable holds. This is required in programming to do various operations without causing an error.

By default, Python provides the following built in data types:

1. *Numeric Data Type:*

Int (Integer) 3, -8, 0

Float (Decimal) 1.2, 35.85, -0.258

Complex 8+2i

1. *Text Data (Strings)*

Examples are “Hello, World”, “Welcome to Python programming”

1. Boolean

Boolean data contains a value like True or False

1. Sequenced Data: list, tuple

List: A list is a ordered collection of data with elements separated by commas and are enclosed within Square brackets. Lists are mutable and can be modified after creation.

List1= [0, 2, 3, [-4,5], [“apple”, “banana”]]

Print(list1)

Tuple: A list is a ordered collection of data with elements separated by commas and are enclosed within parentheses. Tuples are immutable and cannot be modified after creating.

Tupel1 = {{“Parrot”, “Sparrow”} , {“Lion” , “Tiger”}}

Print(tuple1)

1. Mapped Data: dict

Dict: A dictionary is an unordered collection of data containing a key:value pair. The Key:Value pairs are enclosed within curly brackets.

Dict1={“name”:”Ravi” , “Age”:30, “CanVote”: True}

Print(dict1)

***Finding the Type of Data*:**

In Python, we can print the type of any operator using type function:

A = 1

Print (type(a))

B = True

Print (type(b))

***Arithmetic Operators:***

|  |  |  |
| --- | --- | --- |
| Operator | Operator Name | Example |
| + | Addition | 15+7 |
| - | Subtraction | 15-7 |
| \* | Multiplication | 15\*7 |
| \*\* | Exponential | 15 to the power 7 |
| / | Division | 15/7 |
| % | Modules | 15%7 |
| // | Floor Division | 15//7 |

***TypeCasting in Python:***

The conversion of one data type into other data type is known as Type Casting in Python or Type conversion in Python.

Python supports a wide variety of function or methods like int(), float(), str(), ord(), hex(), oct(), tuple(), set(), list(), dict() etc for the type casting in Python.

Two types of Typecasting:

1. Explicit Conversion (Explicit type casting in Python)
2. Implicit Conversion (Implicit type casting in Python)

**Explicit Type casting:**

The conversion of one data type into another data type done via developer or programmer’s intervention or manually as per the requirement is know as explicit type conversion.

It can be achieved with the help of Python’s built-in type conversion functions such as int(), float(), hex(), oct(), str(), etc.

Example of explicit type casting:

String = “15”

Number = 7

String\_number = int(string)

Sub = number + string\_number

Print(“addition of two numbers is”, sub)

**Output:** Addition of two numbers is 22

**Implicit Type casting:**

Data types in Python do not have the same level i.e/. ordering of dat types is not the same in Python. Some of the data types have higher order and some have lower order. While performing any operations on variables with different data types in Python, one of the variables data types will be changed to the higher data type. According to the level, on data type is converted into other by the Python interpreter itself (automatically). This is called Implicit type casting in Python.

Python converts a small data type to a higher data type to prevent data loss.

Example of Implicit type casting:

A = 7

Print (type(a))

B = 3.0

Print (type(b))

C = a + b

Print©

Print (type©)

**User Input in Python**

In Python, we can take user input directly by using input() function. This input function gives a return value as string/character hence we have them into a variable.

Syntax:

Variable = input()

But input function returns the value as string. Hence we have to typecast them whenever required to another datatype/

Example:

Variable=int(input())

Variable=float(input())

We can also display a text using input function. This will make input() function take user input and display a message as well

Example:

A=input(“Enter the name:”

Print(“Hello”, a”.Welcome to Pyton Practice sessions”)

**Strings:**

In Python, anything that you enclose between single or double quotations marks is considered as String. A string is essentially a sequence or array of textual data. Strings are used when working with Unicode characters.

Example:

Name = “Ravi”

Print(“Hello, ” + name)

Output: Hello, Ravi

**Note:** It doesn’t matter whether you enclose your strings in single or double quotes. The output remains the same.

Sometimes, the user might need to put quotation marks in between the strings.

Example, consider the sentence: He said “I want to eat an apple”.

How will you print this statement in Python?. He said, “He want to eat an apple”

Print (‘He said, “He want to eat an apple”.’

Multiline Strings:

If our string has multiple line, we can create them like

Sentence = ‘’’Hi, I am Ravi.

How are you?

Hope everything with Python practice session is going right.’’’

Print(sentence)

**Accessing characters of a string:**

In Python, string is like an array of characters. We can access parts of the string by using its index which starts from 0.

Square brackets can be used to access elements of the string.

Print (name[0])

Print (name[1])

**Looping through the string**

We can loop through strings using a for loop like this

For character in name:

Print(character)

**String Slicing and Operations on string**

Length of a string:

We can find the length of a string using len() function.

**Example**:

Fruit = “Mango”

Len1 = len (fruit)

Print (“Mango is a”, len1,” character word”)

**Output**: Mango is a 5 character word

**String as an Array**

A string is essentially a sequence of characters also called as array. Thus we can access the elements of this array.

Example:

Pie = “Applepie”

Print (pie[0:5])

Print (pie[6])

**String Methods**

Python provides a set of built-in methods that we can use to alter and modify the strings.

***Upper():***

The upper() method converts a string to upper case

**Example**

Str1 = “AbcdeFGHij”

Print (str1.upper())

**Output**: ABCDEFGHIJ

***Lower()***:

The lower() method converts a string to lower case

**Example**

Str1 = “AbcdeFGHij”

Print (str1.lower())

**Output**: abcdefghij

***Strip()***

The strip() method removes any white spaces before and after the string

Example:

Str2 = “ Silver spoon “

Print (str2.strip())

Output: Silver spoon

***Rstrip()***

The rstrip() removes any trailing characters.

Example:

Str3 = “Hello |||”

Print (str3.rstrip(“|”))

Output : Hello

***Replace()***

The replace() method replaces all occurrences of a string with another string.

Example:

Str2 = “Silver spoon”

Print (str2.replace(“Sp” , “M”)

***Split()***

The split method splits the given string at the specified instance and returns the separated strings as list items.

**Example:**

Str2 = “Silver spoon”

Print (str2.split(“ “))

Output : [‘Silver’ , ‘spoon’]

***Capitalize*** ():

The capitalize() method turns only the first character of the string to uppercase and the rest other characters of the string are turned into lower case. The string has no effect if the first character is already uppercase.

Example:

Str1 = “Hello”

Print(str1.capitalize())

Str2 = “Hello WorlD”

Print (str2.capitalize())

Output: Hello, Hello world

***Center()***:

The center() method aligns the string to the centre as per the parameter given by the user.

Example:

Str1 = “!!!Welcome to the console!!!”

Print (str1.center(100))

OutPut: !!!Welcome to the console!!!

***Count():***

The count() method returns the number of times the given value has occurred within the given string

**Example**:

Str1 = “Abracadabra”

countStr = str1.count(“)

print (countstr)

**Output**: 4

***Endswith():***

The endswith() method checks if the string ends with a give value. If yes then returns True, else return False.

**Example**:

Str1 = “Welcome to Console !!!”

Print (str1.endswith(“!!!”))

**Output**: True

We can even also check for a value in-between the string by providing start and end index positions.

**Example**:

Str1 = “Welcome to Console !!!”

Print (str1.endswith(“to” ,4,10))

**Output**: True

***Find():***

The find() method searches for the first occurrence of the given value and returns the index where it is present. If given value is absent from the string then returns -1.

**Example**:

Str1 = “His name is Don. He is an honest person.”

Print (str1.find(“is”)

**Output**: 10

***Index():***

The index() method searches for the first occurrence of the give value and returns the index where it is present. If given value is absent from the string then raise an exception.

**Example**:

Str1 = “Hi, I am Ravi. Welcome to Python practice sessions”

Print (str1.index(“Ravi))

**Output**: 10

As we can see this method is somewhat similar to the find() method. The major difference being that index() raises an exception if value is absent whereas find() does not.

**Example**:

Str1 = “Hi, I am Ravi. Welcome to Python practice sessions”

Print (str1.index(“Rahul))

**Output**: substring not found

Isalnum():

The isalnum() {alpha numeric method) method returns True only if the entire string only consists of A-Z, a-z, 0-9. If any other characters or punctuations are present, then it returns false.

Example:

Str1 = “welcometopythonprogramming”

Print(str1.isalnum())

Output: True

Isalpha():

The isalpha() method returns True only if the entire strng only contains of A-Z, a-z. If any other characters or punctuations or numbers 0-9 are present then it returns False.

Example1:

Str1 = “welcometopyhtonprogramming”

Print (str1.isalpha())

Output: True

Example2:

Str1 = “Welcome2PythonProgramming”

Print (str1.isalpha())

Output: False

Islower():

The islower() method return True if all the characters in the string are lower case , else it returns False.

Example:

Str1 = “welcome to python programming”

Print (str1.islower())

Output: True

Isprintable():

The isprintable() method returns True if all the value within the given string are printable, if now then returns False.

Example:

Str1 = “We wish you a merry Christmas”

Print (str1.isprintable())

Output: True

Isspace():

The isspace() method retuns True only if the string contains white spaces, else returns False.

Example:

Str1 = “ ”

Print (str1.isspace())

Output: True

Istitle():

The istitle() method returns true if the first letter of each word of the string is capitalized, else it returns False.

Example:

Str1 = “Welcome To Python”

Print (str1.istitle())

Output: True

Swapcase():

The swapcase() method changes the character casing of the string. Upper cases are converted to lower case and lower case to upper case.

Example:

Str1 = “ Welcome To Python”

Print (str1.swapcase())

Title()

The Title() method capitalizes each letter of the word within the string.

Str1 = “Welcome to python”

Print (str1.title())