Assignment 2

August 7, 2023

Q1. How do you comment code in Python? What are the different types of comments?

ANS: There are three types of comments: single-line, multi-line, and docstring comments. The syntax of comments varies depending on the type.

Single-Line Comments in Python: Single-line comments begin with the "#" character. Anything that is written in a single line after '#' is considered as a comment.

Multi-Line Comments in Python: Python does not provide the option for multiline comments. However, there are different ways through which we can write multiline comments. Multiline comments using multiple hashtags (#) We can multiple hashtags (#) to write multiline comments in Python. Each and every line will be considered as a single-line comment.

Docstring in Python: Python docstring is the string literals with triple quotes that are appeared right after the function. It is used to associate documentation that has been written with Python modules, functions, classes, and methods. It is added right below the functions, modules, or classes to describe what they do. In Python, the docstring is then made available via the **doc** attribute.

[2]:
```python
# Print "Single-Line Comments !" to console
print("Single-Line Comments")
```

```
Single-Line Comments
```

[3]:
```python
# Python program to demonstrate
# multiline comments
print("Multiline comments")
```

```
Multiline comments
```

[4]:
```python
def multiply(a, b):
    """Multiplies the value of a and b"""
    return a*b
# Print the docstring of multiply function
print(multiply.__doc__)
```

```
Multiplies the value of a and b
```

[ ]:
```python

```

Q2. What are variables in Python? How do you declare and assign values to variables?

ANS: Variable is a name that is used to refer to memory location. Python variable is also known as an identifier and used to hold value. In Python, we don't need to specify the type of variable because Python is a infer language and smart enough to get variable type. Python does not bind us to declare a variable before using it in the application. It allows us to create a variable at the required time.We don't need to declare explicitly variable in Python. When we assign any value to the variable, that variable is declared automatically.

In Python, we need not declare a variable with some specific data type. Python has no command for declaring a variable. A variable is created when some value is assigned to it. The value assigned to a variable determines the data type of that variable. Thus, declaring a variable in Python is very simple. ->Just name the variable ->Assign the required value to it ->The data type of the variable will be automatically determined from the value assigned, we need not define it explicitly.

The assignment operator, denoted by the "=" symbol, is the operator that is used to assign values to variables in Python.

[7]:
```
a = 5

# printing value of a
print ("The value of a is: " + str(a))
```

```
The value of a is: 5
```

[ ]:
```

```

Q3. How do you convert one data type to another in Python?

ANS: Python defines type conversion functions to directly convert one data type to another which is useful in day-to-day and competitive programming. There are two types of Type Conversion in Python:

1.Implicit Type Conversion 2.Explicit Type Conversion

->Implicit Type Conversion: In Implicit type conversion of data types in Python, the Python interpreter automatically converts one data type to another without any user involvement.

->Explicit Type Conversion: In Explicit Type Conversion, the data type is manually changed by the user as per their requirement. With explicit type conversion, there is a risk of data loss since we are forcing an expression to be changed in some specific data type. Various forms of explicit type conversion are there.

1. int(a, base): This function converts any data type to integer. 'Base' specifies the base in which string is if the data type is a string.
2. float(): This function is used to convert any data type to a floating-point number.
3. ord() : This function is used to convert a character to integer.
4. hex() : This function is to convert integer to hexadecimal string.
5. oct() : This function is to convert integer to octal string.
6. tuple() : This function is used to convert to a tuple.
7. set() : This function returns the type after converting to set.
8. list() : This function is used to convert any data type to a list type.
9. dict() : This function is used to convert a tuple of order (key,value) into a dictionary.
10. str() : Used to convert integer into a string.

11. complex(real,imag) : This function converts real numbers to complex(real,imag) number.
12. chr(number): This function converts number to its corresponding ASCII character.

[10]:
```python
# Python code to demonstrate Type conversion
# using int(), float()

# initializing string
a = "10010"

# printing string converting to int base 2
b = int(a,2)
print ("After converting to integer base 2 : ", end="")
print (b)

# printing string converting to float
c = float(a)
print ("After converting to float : ", end="")
print (c)
```

```
After converting to integer base 2 : 18
After converting to float : 10010.0
```

[11]:
```python
# Python code to demonstrate Type conversion
# using  ord(), hex(), oct()

# initializing integer
a = '4'

# printing character converting to integer
b = ord(a)
print ("After converting character to integer : ",end="")
print (b)

# printing integer converting to hexadecimal string
c = hex(56)
print ("After converting 56 to hexadecimal string : ",end="")
print (c)

# printing integer converting to octal string
d = oct(56)
print ("After converting 56 to octal string : ",end="")
print (d)
```

```
After converting character to integer : 52
After converting 56 to hexadecimal string : 0x38
After converting 56 to octal string : 0o70
```

[12]:
```
# Python code to demonstrate Type conversion
# using  tuple(), set(), list()

# initializing string
m = 'geeks'

# printing string converting to tuple
a = tuple(m)
print ("After converting string to tuple : ",end="")
print (a)

# printing string converting to set
b = set(m)
print ("After converting string to set : ",end="")
print (b)

# printing string converting to list
c = list(m)
print ("After converting string to list : ",end="")
print (c)
```

```
After converting string to tuple : ('g', 'e', 'e', 'k', 's')
After converting string to set : {'g', 's', 'e', 'k'}
After converting string to list : ['g', 'e', 'e', 'k', 's']
```

[13]:
```
# Python code to demonstrate Type conversion
# using  dict(), complex(), str()

# initializing integers
a = 1
b = 2

# initializing tuple
tup = (('a', 1) ,('f', 2), ('g', 3))

# printing integer converting to complex number
k = complex(1,2)
print ("After converting integer to complex number : ",end="")
print (k)

# printing integer converting to string
u = str(a)
print ("After converting integer to string : ",end="")
print (u)

# printing tuple converting to expression dictionary
r = dict(tup)
```

```
print ("After converting tuple to dictionary : ",end="")
print (r)
```

```
After converting integer to complex number : (1+2j)
After converting integer to string : 1
After converting tuple to dictionary : {'a': 1, 'f': 2, 'g': 3}
```

[14]:
```
# Convert ASCII value to characters
a = chr(76)
b = chr(77)

print(a)
print(b)
```

```
L
M
```

[ ]:
```

```

Q4. How do you write and execute a Python script from the command line?

ANS: It's quite easy to run Python scripts from the command line.

1 1.Verify your command prompt can run Python.

2 2.Create a Python script that is error-free.

3 3.Use python your/file/name.py to run your script from the terminal.

Make Sure Command Prompt Can Run Python: To start, we need to make sure the command line application we are using has access to our Python installation. To do this, open the command prompt, type python and press 'Enter'. we should see a message that documents the Python version that is being used followed by »>, which indicates the next code we will type will be executed by the Python interpreter.

Create a Python Script: The Python script (hello.py) will simply print out a statement that lets us know the code in the script has run.

Run the Python Script from command line. Once our Python script is created it's super easy to run it from the command line. All we need to do is type python followed by the script name. we'll need to make sure that our terminal's working directory is the directory that contains our python script, or give the full path to the script.

[ ]:
```

```

Q5. Given a list my_list = [1, 2, 3, 4, 5], write the code to slice the list and obtain the sub-list [2, 3].

ANS: # 1.start is the index of the list where slicing starts. # 2.stop is the index of the list where slicing ends. # 3.step allows you to select nth item within the range start to stop.

[15]:
```
my_list = [1, 2, 3, 4, 5]
print(my_list)
```

```
[1, 2, 3, 4, 5]
```

[16]:
```
my_list = [1, 2, 3, 4, 5]
print(my_list[1:3])
```

```
[2, 3]
```

[ ]:
```

```

Q6. What is a complex number in mathematics, and how is it represented in Python?

ANS: A number which can be represented in the form of x+iy, where 'i' is an imaginary number, is called a complex number. By the expression, we can conclude that the complex number is a combination of a real number and an imaginary number. For example, 3+5i is a complex number, where 3 is the real number and 5i is imaginary.

Complex numbers are created from two real numbers. You can create it directly or you can use the complex function. It is written in the form of (x + yj) where x and y are real numbers and j is an imaginary number which is the square root of -1.

[21]:
```
z = complex(5, 7)
print("Output:", z)

z = complex(3)
print("Output:", z)

z = complex()
print("Output:", z)

z = complex('1+1j')
print("Output:", z)

z = complex(1, -4)
print("Output:", z)
```

```
Output: (5+7j)
Output: (3+0j)
Output: 0j
Output: (1+1j)
Output: (1-4j)
```

[22]:
```
z  = 3 + 4j
```

```
print('Real part:', z.real)
print('Imaginary part:', z.imag)
print('Conjugate value:', z.conjugate())
```

```
Real part: 3.0
Imaginary part: 4.0
Conjugate value: (3-4j)
```

[ ]:

Q7. What is the correct way to declare a variable named age and assign the value 25 to it?

ANS:The correct way to declare a variable named age and assign the value 25 to it is:

[23]:
```
Age=25
```

[26]:
```
Age
```

[26]:
```
25
```

[ ]:

Q8. Declare a variable named price and assign the value 9.99 to it. What data type does this variable belong to?

[27]:
```
price=9.99
print('price',type(price))
```

```
price <class 'float'>
```

[ ]:

Q9. Create a variable named name and assign your full name to it as a string. How would you print the value of this variable?

[29]:
```
Name="Anurag Tiwari"
print(Name)
```

```
Anurag Tiwari
```

[ ]:

Q10. Given the string "Hello, World!", extract the substring "World".

[30]:
```
str="Hello, World!"
print(str[7:12])
```

```
World
```

```
[ ]:
```

Q11. Create a variable named “is_student” and assign it a boolean value indicating whether you are currently a student or not.

```
[31]: is_student=True
```

```
[32]: is_student
```

```
[32]: True
```

```
[ ]:
```