Basic of python programming

Note Book Owner: Emdadul Hoque

program 01: Write a program to display the messages “Hello”, “World” and “Good Bye”. Each of the three messages should get displayed on a different line.

In [2]:

```
print("Hello")
print("World")
print("Good Bye")
```

```
Hello
World
Good Bye
```

In the above program, we have displayed each message in a different line. In short, the print function automatically prints a linefeed (\n) to cause the output to advance to the next line. However, if you want to display the messages “Hello” “World” and “Good Bye” in one line without using a single print statement, then you can invoke the print function by passing a special argument named end=’ ‘. The following program illustrates the use of the end argument within the print function

Program 02: Write a basic program to make use of the end key and display the messages “Hello” “World” and “Good Bye” in one line

In [5]:

```
print("Hello ", end='')
print(" World", end='')
print(" Good Bye", end='')
```

```
Hello  World Good Bye
```

Multiple assignments

In [7]:

```
# Normal Assignemnt
```

In [8]:

```
p = 30
q = 20
```

In [9]:

```
p
```

Out[9]:

```
30
```

In [10]:

```
q
```

Out[10]:

```
20
```

In [11]:

```
p, q = 20, 30
```

In [12]:

```
p
```

Out[12]:

```
20
```

In [13]:

```
q
```

Out[13]:

```
30
```

```
In [15]: #swap number

         p = 20
         q = 30
         temp = p
         p = q
         q = temp
```

```
In [16]: p
```

```
Out[16]: 30
```

```
In [17]: q
```

```
Out[17]: 20
```

```
In [18]: #using python special

         p = 30
         q = 20

         p, q = q, p
```

```
In [19]: p
```

```
Out[19]: 20
```

```
In [20]: q
```

```
Out[20]: 30
```

Program 03: Write a simple program to calculate the area of a rectangle in Python

```
In [1]: # formula: area = lenght * breadth

        length = int(input("lenght of a rectangle: "))
        breadth = int(input("Breadth of a rectangle: "))

        area = length * breadth

        print("Area of a rectangle is: ", area)
```

```
lenght of a rectangle: 5
Breadth of a rectangle: 10
area of a rectangle is:  50
```

The eval() Function

eval() function ব্যবহারের মাধ্যেম আমরা ইউজার থেকে এক্সপেক্টেড ইনপুট নিতে পারি। যেমন এর আগের প্রোগ্রামের দেখতে পাচ্ছি ইউজারের থেকে কোন সংখ্যা ইনপুট নেওয়ার সময় input() ফাংশনকে আমরা int() ফাংশনের মধ্যে রাখতে হয়েছিলো। যাতে করে আমরা ইউজার ইন্টিজার টাইপের ভ্যালু পাই। এর কারণ হচ্ছে আমরা যখন ইনপুট ফাংশন ব্যবহার করি তখন এই ফাংশন ইউজারের কাছ থেকে স্ট্রিং টাইপের ইনপুট নেয়। তবে আমরা eval() ফাংশন ব্যবহার করে আমরা ইউজারের কাছ থেকে এক্সপেক্টেড ভ্যালু নিতে পারি। অর্থাৎ ইউজার যদি সংখ্যা ইনপুট দেয় তাহলে আমরা সংখ্যা পাব এবং ইউজার যদি ক্যারেক্টার দেয় তাহলে আমরা ক্যারেক্টার টাইপের ভ্যালু পাবো।

```
In [6]: number = eval(input("Enter a number: "))
        string = input("Enter a string: ")

        print(type(number))
        print(type(string))
```

```
Enter a number: 12.45
Enter a string: sfsdfasd
<class 'float'>
<class 'str'>
```

Formatting number and strings

```
In [28]: x = 20.345123
```

```
In [13]: x = format(x, ".3f")
```

```
In [14]: x
```

```
Out[14]: '10.345'
```

```
In [41]: x = 20.345123
         x = format(x, "7.3f")
```

```
In [42]: x
```

```
Out[42]: ' 20.345'
```

```
In [44]: x = 20.345123
         x = format(x, "<7.3f")
```

```
In [49]: x # here format function return a string value
```

```
Out[49]: '20.345 '
```

```
In [50]: type(x)
```

```
Out[50]: str
```

```
In [51]: x = 20.345123
         x = eval (format(x, "<7.3f"))
```

```
In [52]: type(x)
```

```
Out[52]: float
```

```
In [ ]: 
```

```
In [46]: 
```

```
314.42%
```

```
In [47]: print(format(3.14423, ".3%"))
```

```
314.423%
```

```
In [60]: x = 3.1416
```

```
In [61]: x = format(x, ".2%")
```

```
In [62]: x
```

```
Out[62]: '314.16%'
```

```
In [63]: print(format(3.14423, ".2%"))
```

```
314.42%
```

```
In [64]: type(print(format(3.14423, ".2%")))

         314.42%

Out[64]: NoneType
```

Specifier Format --------- -- 10.2f ---> floating point number with precision 2 and width 10. <10.2f ---> Left Justify the floating point number. >10.2 ---> fRight Justify the formatted item. 10X ---> Format integer in hexadecimal with width 10 20s ---> Format String with width 20 10.2% ---> Format the number in decimal

Python Inbuilt Function

ceil(x) ----> Round X to nearest integer and returns that integer. floor(x) ----> Returns the largest value not greater than X exp(x) ----> Returns the exponential value for ex log(x) ----> Returns the natural logarithmic of x (to base e) log(x, base) ----> Returns the logarithmic of x to the given base sqrt(x) ----> Return the square root of x Sin(x) ----> Return the sin of X, where X is the value in radians asin(x) ----> Return the angle in radians for the inverse of sine cos(x) ----> Return the sin of X, where X is the value in radians aCos(x) ----> Return the angle in radians for the inverse of cosine tan(x) ----> Return the tangent of X, where X is the value in radians degress(x) ----> Convert angle X from to radians to degrees Radians(x) ----> Convert angle X from to radians to degrees

```
In [67]: import math

         11

In [68]: print(math.ceil(10.23))

         11

In [69]: print(math.floor(10.23))

         10

In [70]: print(math.exp(10.23))

         27722.51006805505

In [71]: print(math.log(2.7))

         0.9932517730102834

In [72]: print(math.log(10, 10))

         1.0

In [73]: print(math.sqrt(16))

         4.0
```

The ord and chr Functions

```
In [77]: ord("A") #Returns ACII value of Character 'A'

Out[77]: 65

In [78]: ord("z")

Out[78]: 122

In [79]: ord("x")

Out[79]: 120

In [80]: chr(65)
```

```
Out[80]: 'A'
```

```
In [81]: chr(122)
```

```
Out[81]: 'z'
```

```
In [82]: chr(120)
```

```
Out[82]: 'x'
```

Summary

1. Python breaks each statement into a sequence of lexical components called tokens. 2.Literals are numbers, strings or characters that appear directly in a program. 3.Python offers an inbuilt method called type to know the exact type of any value. 4.Keywords are reserved words. 5.Keywords cannot be used as identifiers or variables. An identifier is a name used to identify a variable, function, class or other objects. 6.Everything in Python is an object. 7.The int function converts a string or a number into a whole number or integer. 8.The float function converts a string into a floating-point number. 9.The Boolean data type is represented in Python as of type bool. 10.print function is used to display contents on the screen. 11.input() function is used to accept input from the user. 12.format() function can be used to return a formatted string. 13. Python Tokens: keywords, identifiers/Variables, Operators, Delimiters, literals 14. Interger Literal: 18, Floating Point Literal: 21.98, "Q" : Character literal, "Hello": String Literal 15. keywords: and, as, assert, break, class, continue, def, del, elif, else, except, False, finally, for, from, global, if, import, in, is, lambda, None, nonlocal, not, or, pass, raise, return, True, try, while, with, yield 16. Operators: + - * / // % ** ---> Arithmetic Operator == != <> <= >= ---> Relational Operator and not or ---> Logical Operator & | ~ ^ << >> ---> Bitwise Operator 17. Delimiter: Delimiters are symbols that perform a special role in Python like grouping, punctuation and assignment. Python uses the following symbols and symbol combinations as delimiters. ( ) [ ] { } , : . ‘ = ; += -= *= /= //= %= &= |= ^= >>= <<= **= 18. Identifier: • Is a sequence of characters that consists of letters, digits and underscore • Can be of any length • Starts with a letter which can be either lower or upper case • Can start with an underscore ‘_’ • Cannot start with a digit • Cannot be a keyword. Some examples of valid identifiers are Name, Roll_NO, A1, _Address etc.

Keyterms

chr(): Returns a character for a given ASCII value end(): Used as argument with print() function format(): Formats string and integer Identifier: Name to identify a variable Inbuilt Math Functions: abs(), max(), round(), ceil(), log(), exp(),sqrt(), sin(), asin(), acos(), atan(),cos(), degrees(), radians() and floor(). input(): Used to accept data from the user int(): Used to convert string or float into integer ord(): Returns ASCII value of a character print(): Prints contents on the screen str(): Used to convert a number into string type(): Used to know the exact type of any value Tokens: Breaks each statement into a sequence of lexical components

Excercise

```
In [1]: round(1.5)
```

```
Out[1]: 2
```

```
In [2]: round(-1.5)
```

```
Out[2]: -2
```

```
In [3]: print({:,}.format(‘100000’))
```

```
  Cell In[3], line 1
    print({:,}.format(‘100000’))
                      ^
SyntaxError: invalid character '‘' (U+2018)
```

```
In [ ]: Name = MyName
In [ ]: Sum = 10 + ‘10’
```

```
In [ ]: PriNt(“Hello Python!”)
```

#True or False

1. Python breaks each statement into a sequence of lexical components known as tokens.(True) 2. Keywords are tokens of Python.(True) 3. Operators are not a part of tokens.(False) 4. Python keywords do not have fixed meaning.(False) 5. Keywords can be used as identifiers or variables.(False) 6. Strings are part of literals.(True) 7. An identifier is a name used to identify a variable, function etc.(True) 8. Python classifies different kinds of objects into types.(True) 9. The float function converts a string into a whole integer number. (True) 10. The str function is used to convert a number into a string.(True)

#Excercise Question

What will be the output of the following statements if all of them are executed in Python interactive mode? a. abs(-2) b. min(102,220,130) c. max(-1,-4,-10) d. max(‘A’,’B’,’Z’) e. max(‘a’,’B’,’Z’) f. round(1.6) g. math.ceil(1.2) h. math.floor(1.8) i. math.log(16,2) j. math.exp(1) k. math. l. cos(math.pi) m. math.cos(math.pi)

```
In [11]: abs(-2)

Out[11]: 2
```

```
In [12]: min(102,220,130)

Out[12]: 102
```

```
In [13]: max(-1,-4,-10)

Out[13]: -1
```

```
In [14]: max(‘A’,’B’,’Z’)

  Cell In[14], line 1
    max(‘A’,’B’,’Z’)
        ^
SyntaxError: invalid character '‘' (U+2018)
```

```
In [15]: max(‘a’,’B’,’Z’)

  Cell In[15], line 1
    max(‘a’,’B’,’Z’)
        ^
SyntaxError: invalid character '‘' (U+2018)
```

```
In [18]: import math
         math.ceil(1.2)

Out[18]: 2
```

```
In [19]: math.floor(1.2)

Out[19]: 1
```

3. What will be the output of the following statements if all of them are executed in Python interactive mode? a. ord(‘a’) b. ord(‘F’) c. ord(‘f’) d. chr(97) e. chr(100)

```
In [20]: ord('a')

Out[20]: 97
```

```
In [21]: ord('f')
```

```
Out[21]:  102
```

```
In [22]:  ord('F')
```

```
Out[22]:  70
```

```
In [23]:  chr(97)
```

```
Out[23]:  'a'
```

```
In [24]:  chr(100)
```

```
Out[24]:  'd'
```

4. Identify the error in the following piece of code. Explain how you will fix it. num1 = ‘10’ num2 = 20.65 sum = num1 + num2

```
In [26]:  #where erro is num1 = '10'

          num1 = 10
          num2 = 20.65
          sum = num1 + num2
```

```
In [27]:  sum
```

```
Out[27]:  30.65
```

State the output of following statements. a. print(format(16,’x’)) b. print(format(10,’x’)) c. print(format(10+10,’x’)) d. print(format(10+ord(‘a’),’x’)) e. print(format(20,’o’)) f. print(format(100,’b’)) g. print(format(10,’b’))

```
In [28]:  print(format(16,’x’))
```

```
  Cell In[28], line 1
    print(format(16,’x’))
                    ^
SyntaxError: invalid character '’' (U+2019)
```

```
In [29]:  print(format(10,’x’))
```

```
  Cell In[29], line 1
    print(format(10,’x’))
                    ^
SyntaxError: invalid character '’' (U+2019)
```

```
In [30]:  print(format(10+ord(‘a’),’x’))
```

```
  Cell In[30], line 1
    print(format(10+ord(‘a’),’x’))
                        ^
SyntaxError: invalid character '‘' (U+2018)
```

Programming Assignments

1. Write a program to print ‘F’ to ‘A’ in five different lines. 2. Write a program to read and store the name of three different cities in three different variables and print all the contents of variables on the console. 3. Write a program to prompt the user to enter and display their personal details, such as name, address and mobile number. 4. By making use of five different print statements, write a program to print ‘A’ to ‘F’ in one single line. 5. Write a program to read an integer as string. Convert the string into integer and display the type of value before and after converting to int. 6. Write a program initialize the string “hello world” to a variable Str1 and convert the string into upper case. 7. Translate the following algorithm into Python code. Step 1: Initialize variable named Pounds with value 10. Step 2: Multiply Pounds by 0.45 and assign it to a variable Kilogram. Step 3: Display the value of variable Pounds and Variable. 8. Write a program to read the radius of a circle and print the area of the circle.

```
In [32]:  # 1. Write a program to print ‘F’ to ‘A’ in five different lines.
```

```
print("F")
print("E")
print("D")
print("C")
print("B")
print("A")
```

```
F
E
D
C
B
A
```

In [33]:

```
"""2. Write a program to read and store the name of three different cities in three d
variables and print all the contents of variables on the console."""

city_name1 = input("Type your city name ")
city_name2 = input("Type your city name ")
city_name3 = input("Type your city name ")

print(city_name1)
print(city_name2)
print(city_name3)
```

```
Type your city nameDhaka
Type your city namenoakhali
Type your city nameshylet
Dhaka
noakhali
shylet
```

In [35]:

```
"""Write a program to prompt the user to enter and display their personal details, su
name, address and mobile number."""

name = input("Type your name: ")
address = input("Type your address: ")
mobile_number = input("Type your mobile number: ")

print("Your name: ", name)
print("Your address: ", address)
print("Your Mobile number: ", mobile_number)
```

```
Type your name: Emdadul
Type your address: mohammadpur, Dhaka
Type your mobile number: 01521212572
Your name:  Emdadul
Your address:  mohammadpur, Dhaka
Your Mobile number:  01521212572
```

In [36]:

```
"""4. By making use of five different print statements, write a program to print ‘A’
single line"""

print("F", end=" ")
print("E", end=" ")
print("D", end=" ")
print("C", end=" ")
print("B", end=" ")
print("A")
```

```
F E D C B A
```

In [38]:

```
"""5. Write a program to read an integer as string. Convert the string into integer a
type of value before and after converting to int."""

integer_as_string = input("input your value: ")
print(type(integer_as_string))
```

```
convert_into_integer = int(integer_as_string)
print(convert_into_integer)
```

```
input your value: 123345
<class 'str'>
123345
```

In [39]:

```
"""6. Write a program initialize the string “hello world” to a variable Str1 and conv
into upper case."""

str1 = "Hello World"
print(str1.upper())
```

```
HELLO WORLD
```

7. Translate the following algorithm into Python code. Step 1: Initialize variable named Pounds with value 10. Step 2: Multiply Pounds by 0.45 and assign it to a variable Kilogram. Step 3: Display the value of variable Pounds and Variable.

In [41]:

```
pounds = 10
kilogram = 10*0.45

print(pounds)
print(kilogram)
```

```
10
4.5
```

In [45]:

```
"""8. Write a program to read the radius of a circle and print the area of the circle

radius = eval(input("Input radius of a circle: "))
area = math.pi * radius * radius
print("Area of the circle:", area)
```

```
Input radius of a circle: 1234
Area of the circle: 4783879.062809778
```

In [ ]:

```

```