

EXPERIMENT 1

Harsh Kasliwal

2003085

C21

AIM: Exploring the basics of Python

THEORY:

print() Function: print() function is used to print a string or variable values etc. For string in print statement single quote (' ') double quote (" ") triple quotes (" " " ") and triple double quotes (" " " " " " " ") can be used for printing the variable name along with its value calculated is used

Example 1: $a = 12$

$b = 13$

print(a+b)

O/P → 25

By default python's print() function ends with a newline. print() function comes with a parameter called 'end'. By default, the value of this parameter is '\n', i.e. the new line character. You can end a print statement with any character/string using this parameter.

Example 2: printing two combine statements

print("My name is", end= ' ')

print("Harsh!")

O/P → My name is Harsh!

Output formatting:

Formatting output using the format method:

The format method of strings requires more manual effort. Users use {} to mark where a variable will be substituted and can provide detailed formatting directives, but the user also needs to provide the information to be formatted.

Syntax:

```
string.format(value1, value2...)
```

Example:

```
txt1 = "My name is {name}, I'm {age}".format(name = "John", age = 36)
```

Formatting output using String modulo operator (%):

It interprets the left argument much like a printf()-style format as in C language string to be applied to the right argument. The modulo operator % is overloaded by the string class to perform string formatting. Therefore, it is often called a string modulo (or sometimes even called modulus) operator.

Syntax:

%d → for integers

%e → for character

%s → for string

%f → for float

if-else statement: Our compiler will execute the if statement to check whether it is true or false now; if it's true, the compiler will execute the code in the "if" section of the program and skip the bunch of code written in "elif"(else if) and "else." But if the "if" condition is false, then the compiler will move towards the elif section and keep on running the code until it finds a true statement(there could be multiple elif statements). If this does not happen, then it will execute the code written in the "else" part of the program.

Example:

```
print("What is your age?")
```

```
age = int(input())
```

```
if age<18: # condition
```

```
    print("You cannot drive")
```

```
elif age==18:
```

```
    print("We will think about you")
```

```
else:
```

```
    print("You can drive")
```

For loop:

For loop basically depends upon the elements it has to iterate instead of the statement being true or false. We just have to declare a variable so we can print the output through it during different iterations and use the keywords “for” and “in”

Example:

```
items = [int, float, "HaERRY", 5,3, 3, 22, 21, 64, 23, 233, 23, 6]
```

```
for item in items:
```

```
    if str(item).isnumeric() and item>=6:
```

```
        print(item)
```

While loop:

“A while loop in python runs a bunch of code or statements again and again until the given condition is true when the condition becomes false, the loop terminates its repetition.”

The syntax for a while loop is simple and very much like for loop. We have to use the keyword “while”, along with it, we have to put a condition in parenthesis, and after that, a colon is placed. The condition could be either true or false. Until the condition is true, the loop will keep on executing again and again. If we use a certain condition in our while loop that never becomes false, the program will continue running endlessly until we stop it by force

Syntax:

```
while condition_is_true:
```

```
    Code inside the loop body
```

CODE:

1. Write a Python program to print the following string in a specific format

Twinkle, twinkle, little star,
 "How I wonder what you are! "
 Up above the world so high,
 Like a diamond in the sky.
Twinkle, 'twinkle ', little star,
 How I wonder what you are
Using only one print() function.

Code:

```
print('''Twinkle, twinkle little star,\n\t"How I wonder what you are!"\n\t\tUp above the world so high,\n\t\tLike a diamond in the sky.\nTwinkle, 'twinkle', little star,\n\tHow I wonder what you are''')
```

Output:

```
PS D:\Harsh\SEM 4\PYTHON> python -u "d:\Harsh\SEM 4\PYTHON\Exp1.1.py"
Twinkle, twinkle little star,
    "How I wonder what you are!"
        Up above the world so high,
        Like a diamond in the sky.
Twinkle, 'twinkle', little star,
    How I wonder what you are
PS D:\Harsh\SEM 4\PYTHON>
```

2. Program to show output formatting take two values and display them using single print function using

- str.format()
- % operator

Code:

```
name = input('What is your name: ')
age = int(input('How old are you: '))
#Using str.format()
str = "Your name is '{}' and you are '{}' years old."
out = str.format(name, age)
print(out)
#Using % operator
print("Your name is %s and your age is %d"%(name,age))
```

Output:

```
> python -u "d:\Harsh\SEM 4\PYTHON\Exp1.2.py"
What is your name: Harsh
How old are you: 20
Your name is 'Harsh' and you are'20' years old.
Your name is Harsh and your age is 20
PS D:\Harsh\SEM 4\PYTHON> █
```

3. Program to check the leap year using nested if:

Code:

```
year = int(input("Enter the year: "))
if year%4==0:
    if year%100==0:
        if year%400==0:
            print(year, 'is a leap year')
        else:
            print(year, 'is not a leap year')
    else:
        print(year, 'is a leap year')
else:
    print(year, 'is not a leap year')
```

Output:

```
PS D:\Harsh\SEM 4\PYTHON> python -u "d:\Harsh\SEM 4\PYTHON\te
Enter the year: 1900
1900 is not a leap year
PS D:\Harsh\SEM 4\PYTHON> █
```

4. Program to print all armstrong number in range 1 to 1000.

Code:

```
for n in range(1, 1001, 1):
    sum = 0
    x = n
    while x!=0:
        r = x%10
        sum = sum + r ** 3
        x = x//10

    if n == sum:
        print(n)
```

Output:

```
> python -u "d:\Harsh\SEM
1
153
370
371
407
PS D:\Harsh\SEM 4\PYTHON> []
```


5. Program to find fibonacci series of n terms

Code:

```
n = int(input("How many terms ?: "))
first=0
second=1
i=0
if n<=0:
    print("Please enter a positive integer")
elif n==1:
    print("Fibonacci sequence upto",n,":")
    print(first)
else:
    print("Fibonacci sequence: ")
    while n>i:
        print(first)
        next = first + second
        first = second
        second = next
        i += 1
```

Output:

```
PS D:\Harsh\SEM 4\PYTHON> python -u "d:\Harsh\SEM 4\PYTHON\Exp1.5
How many terms ?: 5
Fibonacci sequence:
0
1
1
2
3
PS D:\Harsh\SEM 4\PYTHON> █
```

6. Program on pattern

Code: A

```
# Triangle using alphabet
size = 5

ascii_value = 65

for i in range(size):
    for j in range(i+1):
        alphabet = chr(ascii_value)
        print(alphabet, end=" ")

    ascii_value += 1
    print(" ")
```

Output:

```
PS D:\Harsh\SEM 4\PYTHON> python -u "d:\Harsh\SEM 4\PYTHON\Exp
A
B B
C C C
D D D D
E E E E E
PS D:\Harsh\SEM 4\PYTHON>
```

Code: B

```
# Downward triangle using stars
size = 5

k = size

for i in range(size, -1, -1):

    for j in range(k, 0, -1):
        print(end=" ")

    for j in range(0, i):

        print("*", end=" ")

    print(" ")
```

Output:

```
> python -u "d:\Harsh\SEM 4\PYTHON"

* * * * *
* * * *
* * *
* *
*

PS D:\Harsh\SEM 4\PYTHON>
```

Code: C

```
size = 5
for i in range(1,size+1):
    for j in range(1, size+1-i):
        print(' ', end=' ')
    for j in range(1,i+1):
        print(j, end=' ')
    for j in range(i-1,0,-1):
        print(j, end=' ')
    print()
```

Output:

```
> python -u "d:\Hars
    1
   1 2 1
  1 2 3 2 1
 1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
PS D:\Harsh\SEM 4\PYTHON>
```

Code: D

```
# *****Pyramid using stars*****\

size = 5
m = size
for i in range(0, size):
    for j in range(0, m):
        print(end=" ")
    m = m - 1 # decrementing m after each loop
    for j in range(0, i + 1):

        # printing full Triangle pyramid using stars

        print("* ", end=' ')
    print(" ")
```

OUTPUT:

```
PS D:\Harsh\SEM 4\PYTHON> python -u "d:\Harsh\SEM 4\PYTHON\
    *
   * *
  * * *
 * * * *
* * * * *
PS D:\Harsh\SEM 4\PYTHON>
```