

Some problems and their python code with basic python syntax, regular expression, keyword searching code etc.

Details

- Course No: CSE 2206
- Course Title: Sessional based On CSE 2205
- Experiment No: 2
- Date of Experiment: 18-01-20
- Date of Submission: 02-02-20

Submitted By

KANCHON GHARAMI

Roll: 1703037

Section: A

Department: Computer Science and Engineering
Rajshahi University of Engineering & Technology

Submitted To

MD. FARUKUZZAMAN FARUK

LECTURER

Department of Computer Science & Engineering
Rajshahi University of Engineering & Technology

Topics Name

Some problems and their python code with basic python syntax, regular expression, keyword searching code etc.

Program No:

INDEX

PROG -RAM NO	TOPIC	PROGRAM TITLE	PAGE NO
01	Python Basic Programs	Working with python basic syntax, programs, loops, conditions, console input output etc.	03~05
02	numpy Module in Python	Using and analyzing numpy library in python.	06~07
03	matplotlib.pyplot Module in Python	Using and analyzing matplotlib.pyplot library in python.	08~09
04	re Module in Python	Using and analyzing re Module in Python.	10~11

Topic and problems are connected by content with class lecture 3A ~ class lecture 4E. Those parts were completed with python code in lab.

REFERENCES

Program No:

Python Basic Programs

Program Title: Working with python basic syntax, programs, loops, conditions, console input output etc.

PROBLEM DESCRIPTIONS:

Working with python basic syntax, programs, loops, conditions, console input output etc.

OBJECTIVE:

Getting knowledge about python basic programs.

THEORY:

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

1.1 print() : The print() function prints the specified message to the screen, or other standard output device.

CODE:

```
1. print("Welcome al to Python.automata")
```

OUTPUT:

```
Welcome al to Python.automata
```

In this program, we have used the built-in print() function to print the string Welcome al to Python.automata on our screen.

1.2 input() : This function first takes the input from the user and then evaluates the expression, which means Python automatically identifies whether user entered a string or a number or list. If the input provided is not correct then either syntax error or exception is raised by python.

CODE:

```
1. a= input("Enter whatever you want: ")
2. print(a)
3. print(type(a))
4.
5. aa= int(a)
6. print(aa)
7. print(type(aa))
```

OUTPUT:

Program No:

```
Enter whatever you want: 16
16
<class 'str'>
16
<class 'int'>
```

Whatever we enter as input, input function convert it into a string. if we enter an integer value still input() function convert it into a string. We need to explicitly convert it into an integer in our code using typecasting.

1.3 for loop : A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

This is less like the for keyword in other programming languages, and works more like an iterator method as found in other object-orientated programming languages.

With the for loop we can execute a set of statements, once for each item in a list, tuple, set etc..

CODE:

```
1. for i in range(0,10,1):
2.     print(i)
```

OUTPUT:

```
0
1
2
3
4
5
6
7
8
9
```

The range() function defaults to 0 as a starting value, however it is possible to specify the starting value by adding a parameter: range(2,10,1), which means values from 0 to 10 (but not including 10).

1.4 if condition : In Python, the body of the if statement is indicated by the indentation. Body starts with an indentation and the first unindented line marks the end. Python interprets non-zero values as True . None and 0 are interpreted as False.

CODE:

```
1. #for loop in python
2. for i in range(0,10,1):
3.     print(i)
4.     if i%2==0:
5.         print(i,"is even")
```

Program No:

OUTPUT:

```
0
0 is even
1
2
2 is even
3
4
4 is even
5
6
6 is even
7
8
8 is even
9
```

The body of if is executed only if this evaluates to True.

Program No:

numpy Module in Python

Program Title: Using and analyzing numpy library in python.

PROBLEM DESCRIPTIONS:

Using and analyzing numpy library in python.

OBJECTIVE:

Getting knowledge about numpy library in python.

THEORY:

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

1.1 Array and dimension: A NumPy array is a homogeneous block of data organized in a multidimensional finite grid. All elements of the array share the same data type, also called dtype (integer, floating-point number, and so on). The shape of the array is an n-tuple that gives the size of each axis..

CODE:

```
2. import numpy as np
3. mat1 = np.array( [ (1,3,5,7),(2,4,6,8),(1,1,2,3) ])
4. print(mat1)
5. dimention = mat1.shape
6. print(dimention)
7.
8. dimention_row = mat1.shape[0]
9. print(dimention_row)
10. dimention_row = mat1.shape[0]
11. print(dimention_row)
12.
```

OUTPUT:

```
[[1 3 5 7]
 [2 4 6 8]
 [1 1 2 3]]
(3, 4)
3
```

1.2 NumPy Multiplication Matrix : If both a and b are 2-D (two dimensional) arrays -- Matrix multiplication. If either a or b is 0-D (also known as a scalar) -- Multiply by using numpy. multiply(a, b) or a * b. If a is an N-D array and b is a 1-D array -- Sum product over the last axis of a and b..

CODE:

```
1. import numpy as np
2. mat1 = np.array( [ (1,3,5,7),(2,4,6,8),(1,1,2,3) ])
3. mat2 = np.array( [ (1,9,7),(4,5,6),(7,4,1),(2,5,8) ])
4.
```

Program No:

```
5. mat3 = np.dot(mat1,mat2)
6. print(mat3)
7.
8. mat4 = np.zeros((4,3))
9. print(mat4)
```

OUTPUT:

```
[[ 62  79  86]
 [ 76 102 108]
 [ 25  37  39]]
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]
```

Program No:

matplotlib.pyplot Module in Python

Program Title: Using and analyzing matplotlib.pyplot library in python.

PROBLEM DESCRIPTIONS:

Using and analyzing matplotlib.pyplot library in python..

OBJECTIVE:

Getting knowledge about matplotlib.pyplot library in python..

THEORY:

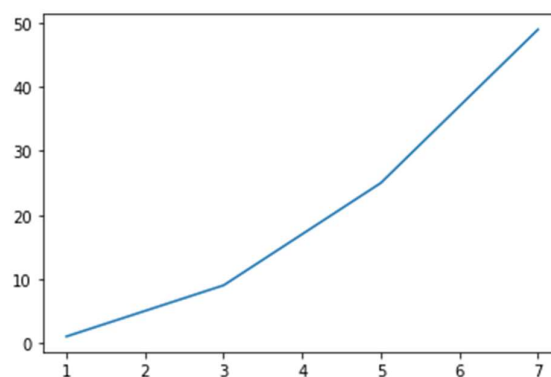
matplotlib.pyplot is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits. There are several toolkits which are available that extend python matplotlib functionality.

1.1 plot(): plot() is a versatile command, and will take an arbitrary number of arguments.

CODE:

```
1. #matplotlib
2.
3. %matplotlib inline
4. import numpy as np
5. import matplotlib.pyplot as plt
6.
7. x = np.array([1,3,5,7])
8. y = np.array(x**2)
9. plt.plot(x,y)
10. plt.show()
```

OUTPUT:



1.2 plot(): plot() is a versatile command, and will take an arbitrary number of arguments.

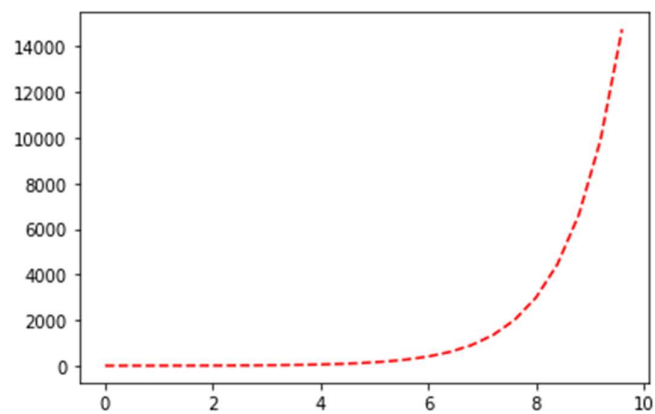
CODE:

```
1. #matplotlib
2.
3. %matplotlib inline
```


Program No:

```
4. import numpy as np
5. import matplotlib.pyplot as plt
6.
7. '''
8. x = np.array([1,3,5,7])
9. #y = np.array(x**2)
10. y = np.array(np.exp(x))
11.
12. '''
13. x = np.arange(0,10,0.4)
14. y = np.array(np.exp(x))
15.
16. plt.plot(x,y, 'r--')
17. plt.show()
```

OUTPUT:



Program No:

re Module in Python

Program Title: Using and analyzing re Module in Python.

PROBLEM DESCRIPTIONS:

Using and analyzing re module in python.

OBJECTIVE:

Getting knowledge about re module in python.

THEORY:

A regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. Regular expressions are widely used in UNIX world.

The Python module re provides full support for Perl-like regular expressions in Python. The re module raises the exception re.error if an error occurs while compiling or using a regular expression.

We would cover two important functions, which would be used to handle regular expressions. But a small thing first: There are various characters, which would have special meaning when they are used in regular expression. To avoid any confusion while dealing with regular expressions, we would use Raw Strings as r'expression'.

1.1 re.findall(): Re.findall() module is used when you want to iterate over the lines of the file, it will return a list of all the matches in a single step. For example, here we have a list of e-mail addresses, and we want all the e-mail addresses to be fetched out from the list, we use the re.findall method. It will find all the e-mail addresses from the list.

CODE:

```
1. import re
2. data = '''
3.     Hello everyone, good morning, I am Faruk and I am from RUET.
4.     I meet Ronaldo last 17/Jan/2020 and got this phone number +8801725345621 , his manage
5.     r gave me another phone number +8801955377859
6.     '''
7. date_pattern = re.compile(r'\d+/[A-Za-z]+/\d{4}')
8. dates = date_pattern.findall(data)
9. print(dates)
10.
11. phone_pattern = re.compile(r'\+[8][8][0][1][3|5|7|8|9]\d{8}')
12.
13. phone_number = phone_pattern.findall(data)
14. print(phone_number)
15.
16. name_pattern = re.compile(r'[A-Z][a-z]+')
17. name = name_pattern.findall(data)
18. print(name)
```

OUTPUT:

Program No:

```
['17/Jan/2020']
['+8801725345621', '+8801955377859']
['Hello', 'Faruk', 'Ronaldo', 'Jan']
```

1.2 re.compile(pattern, repl, string): We can combine a regular expression pattern into pattern objects, which can be used for pattern matching. It also helps to search a pattern again without rewriting it.

CODE:

```
1. import re
2. import numpy as np
3. numbers = np.array( [ '+8801838167090', '+88018381dgg090', '+876038167090', '+8801988167090'
, '+9801838167090' ])
4.
5. for i in numbers:
6.     phone_pattern = re.compile(r'\+[8][8][0][1][3|5|7|8|9]\d{8}')
7.     phone_number = phone_pattern.findall(i)
8.
9.     if phone_number == []:
10.         print("invalid number")
11.     else:
12.         print(phone_number, "valid")
```

OUTPUT:

```
['+8801838167090'] valid
invalid number
invalid number
['+8801988167090'] valid
invalid number
```