

Lab Series No. 12

Lab 12 – Introduction to Regular Expression, Random Numbers & Namespace and Scope

Objectives:

- Introduction to Regular Expression
- Metacharacters
- Functions in regular expressions
- Random Numbers
- Namespace and Scope

1. Introduction to Regular Expression

A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern. RegEx can be used to check if a string contains the specified search pattern. We use **RegEx Module** Python has a built-in package called **re**, which can be used to work with Regular Expressions. So you need to import library **re** before you can use regular expressions in Python.

Regular expressions use two types of characters:

- a) Meta characters: As the name suggests, these characters have a special meaning, similar to * in wild card.
- b) Literals (like a,b,1,2...)

The most common uses of regular expressions are:

- Search a string (search and match)
- Finding a string (findall)
- Break string into a sub strings (split)
- Replace part of a string (sub)

2. What are various methods of Regular Expressions?

The 're' package provides multiple methods to perform queries on an input string. Here are the most commonly used methods:

1. re.match()
2. re.search()

Student Name: _____

Roll No: _____

Section: _____

3. re.findall()
4. re.split()
5. re.sub()
6. re.compile()

3. Metacharacters

Metacharacters are characters with a special meaning:

Character	Description	Example
1. []	A set of characters	"[a-m]"
2. \	Signals a special sequence (can also be used to escape special characters) "\\d"	
3. .	Any character (except newline character)	"he..o"
4. ^	Starts with	"^hello"
5. \$	Ends with	"world\$"
6. *	Zero or more occurrences	"aix*"
+	One or more occurrences	"aix+"
7. {}	Exactly the specified number of occurrences	"al{2}"
8.	Either or	"falls stays"
9. ()	Capture and group	

Exercise 1: Write a program search the string to see if it starts with "Some" and ends with "companies".

```
import re
txt = "Some analysts say a Buffett-less Berkshire Hathaway could be a
candidate for being broken up into multiple companies."
x = re.search("^Some.*companies$", txt)
```

Output:

Student Name: _____

Roll No: _____

Section: _____

4. The findall() function:

The findall() function returns a list containing all matches.

Exercise 2: Write a code that will able to extract ou from all the words in the string.

```
import re
str = "Some analysts say a Buffett-less Berkshire Hathaway could be a
candidate for being broken up into multiple companies."

x = re.findall("ou", str)
print(x)
```

Output:

The list contains the matches in the order they are found. If no matches are found, an empty list is returned.

Exercise 3: Write a code that will able to extract ou from all the words in the string.

```
import re
str = "Some analysts say a Buffett-less Berkshire Hathaway could be a
candidate for being broken up into multiple companies."

x = re.findall("in to", str)
print(x)
```

Output:

Student Name: _____

Roll No: _____

Section: _____

5. The search() Function:

The `search()` function searches the string for a match, and returns a match object if there is a match.

If there is more than one match, only the first occurrence of the match will be returned:

Exercise 4: Write a code that will able to extract the first white-space character from the given string.

```
import re
str = "Some analysts say a Buffett-less Berkshire Hathaway could be a
candidate for being broken up into multiple companies."

x = re.search("\s", str)
print("The first white-space character is located in the position:" ,
x.start())
```

Output:

6. The split() Function:

The `split()` function returns a list where the string has been split at each match:

Exercise 5: Write a code that will help you to split each word by finding the blank character between them.

```
import re

str = "Buffett, who is worth almost $90 billion, still lives in a relatively
modest house about 10 minutes outside downtown Omaha that he bought in 1958."
x = re.split("\s", str)
print(x)
```

Output:

Student Name: _____

Roll No: _____

Section: _____

7. The sub() Function:

The `sub()` function replaces the matches with the text of your choice:

Exercise 6: Write a code that will help you to substitute the blank character and replace it with *.

```
import re

str = "Buffett, who is worth almost $90 billion, still lives in a relatively modest house about 10 minutes outside downtown Omaha that he bought in 1958."
x = re.sub("\s", "*" str)
print(x)
```

Output:

Exercise 7: Write a code that will help you to substitute the blank character and replace it with * only 4 times.

```
import re

str = "Buffett, who is worth almost $90 billion, still lives in a relatively modest house about 10 minutes outside downtown Omaha that he bought in 1958."
x = re.sub("\s", "*" str, 4)
print(x)
```

Output:

Student Name: _____

Roll No: _____

Section: _____

4. Introduction to Random Numbers

Python defines a set of functions that are used to generate or manipulate random numbers. This particular type of functions that are used in a lot of games, lotteries or any application requiring random number generation.

- a. choice()
- b. randrange(beg, end, step)
- c. random()
- d. seed()
- e. shuffle()
- f. uniform

Exercise 8: Write a program which uses all the random functions to show the difference between them.

```
import random

# using choice() to generate a random number from a given list of numbers.
print ("A random number from list is : ",end="")
print (random.choice([1, 3, 5, 7, 9]))

# using randrange() to generate in range from 10
print ("A random number from range is : ",end="")
print (random.randrange(10, 40, 5))

# using random() to generate a random number between 0 and 1
print ("A random number between 0 and 1 is : ", end="")
print (random.random())

# using seed() to seed a random number
random.seed(4)
print ("The mapped random number with 4 is : ", end="")
print (random.random())

lst_odd = [1,9,7,3,5]
print ("The list before shuffling is : ", end="")
for i in range(0, len(lst_odd)):
    print (lst_odd[i], end=" ")
print("\r")

# using shuffle() to shuffle the list
random.shuffle(lst_odd)
# Printing list after shuffling
print ("The list after shuffling is : ", end="")
```

Student Name: _____

Roll No: _____

Section: _____

```
for i in range(0, len(lst_odd)):
    print (lst_odd[i], end=" ")
print("\r")

# using uniform() to generate random floating number in range
# prints number between 5 and 10
print ("The random floating point number between 5 and 10 is: ",end="")
print (random.uniform(5,10))
```

Output:

5. Namespace and Scope

A namespace is a system to have a unique name for each and every object in Python. An object might be a variable or a method. Python itself maintains a namespace in the form of a Python dictionary. Let's go through an example, a directory-file system structure in computers. Needless to say, that one can have multiple directories having a file with the same name inside of every directory. But one can get directed to the file, one wishes, just by specifying the absolute path to the file.

Real-time example, the role of a namespace is like a surname. One might not find a single “Alice” in the class there might be multiple “Alice” but when you particularly ask for “Alice Lee” or “Alice Clark” (with a surname), there will be only one (time being don't think of both first name and surname are same for multiple students).

On the similar lines, Python interpreter understands what exact method or variable one is trying to point to in the code, depending upon the namespace. So, the division of the word itself gives little more information. Its Name (which means name, an unique identifier) + Space(which talks something related to scope). Here, a name might be of any Python method or variable and space depends upon the location from where is trying to access a variable or a method.

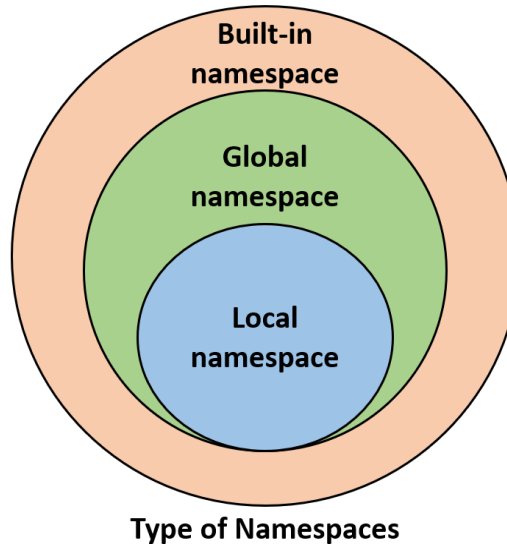
Student Name: _____

Roll No: _____

Section: _____

Types of namespaces:

When Python interpreter runs solely without and user-defined modules, methods, classes, etc. Some functions like `print()`, `id()` are always present, these are built in namespaces. When a user creates a module, a global namespace gets created, later creation of local functions creates the local namespace. The built-in namespace encompasses global namespace and global namespace encompasses local namespace.

**Lifetime of a namespace:**

A lifetime of a namespace depends upon the scope of objects, if the scope of an object ends, the lifetime of that namespace comes to an end. Hence, it is not possible to access inner namespace's objects from an outer namespace.

Exercise 8: Write a program to show different types of variables according to their scopes.

```
#uit_1 is in the global namespace
uit_1 = 5
def some_func():

    # cs_2 is in the local namespace
    cs2 = 6
    def some_inner_func():

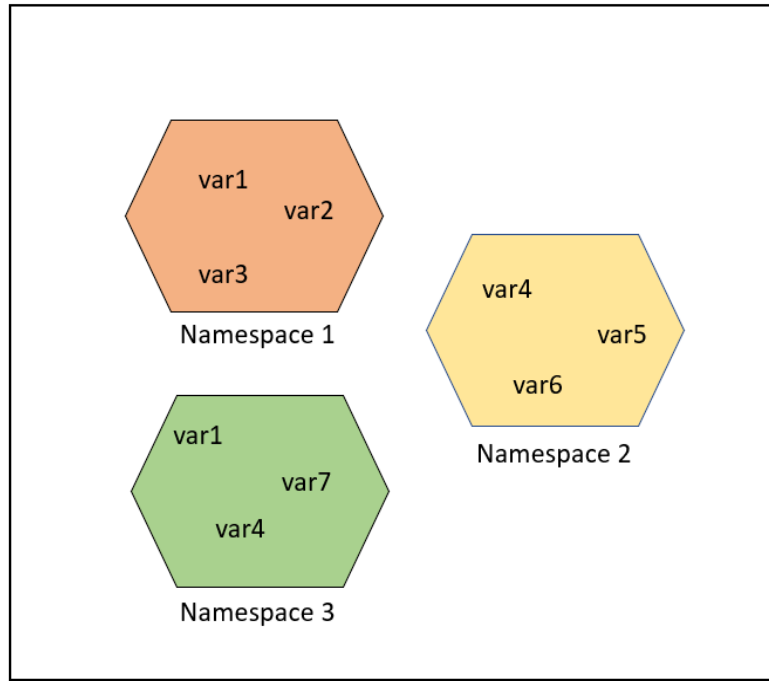
        # var3 is in the nested local
        # namespace
        se_3 = 7
```


Student Name: _____

Roll No: _____

Section: _____

As shown in the following figure, same object name can be present in multiple namespaces as isolation between the same name is maintained by their namespace.



But in some cases, one might be interested in updating or processing global variable only, as shown in the following example, one should mark it explicitly as global and the update or process.

Exercise 9: Write a program to show how global scope works and called inside the function. Similarly, the function updating will affect the value of global variable.

```
count = 5
def some_method():
    global count
    count = count + 1
    print(count)
some_method()
```

Student Name: _____

Roll No: _____

Section: _____

Programming Exercise

Task 1: Write what you understand by the different function used in random module. Elaborate your answer steps by steps. Define each function separately with example.

Task 2: Create a small application to generate game like dice mechanism in Luddo. You can create single dice and two separate dices as well.

Task 3: Create a GUI for using at least 15 most important functions of regular expressions. Use a text file and try to extract the output from the text file.

Task 4: Create a code which will use the given Txt_RE file and split each word, calculate number of vowels from them, find similar words, replace all small characters with the capital once after splitting and count total number of words.

Task 5: What is the purpose of using different types of scope. Show how you can change local and global values of a variable using function.