PYTHON OOPs

Regular Expressions : A regular expression in a programming language is a special text string used for describing a search pattern.

· It is extremely useful for extracting information from text such as code, files, log, spreadsheets or even documents.

· While using the regular expression the first thing is to recognize is that everything is essentially a character, and we are writing patterns to match a specific sequence of characters also referred as string.

· Ascii or latin letters are those that are on your keyboards and Unicode is used to match the foreign text.

· It includes digits and punctuation and all special characters like $#@!%, etc.

· In Python, a regular expression is denoted as RE (REs, regexes or regex pattern) are imported through **re module**. Python supports regular expression through libraries. In Python regular expression supports various things like **Modifiers, Identifiers,** and **White space characters**.

Identifiers	Modifiers	Whitespace characters
\d =any number	\d represents a digit Ex:\d{1,5} it will declare digit between 1,5.	\n = new line
\D=anything but a number(non-digit)	+= matches 1 or more	\s = space
\s =space(tab,space,newline)	?= matches 0 or 1	\t =tab
\S =anything but space	*=0 or more	\e = escape
\w=letters(Matchingalphanumeric character,including "_")	$ match end of a string	\r=carriage return
\W=anything but letters(matches a non-alphanumeric character excluding "_")	^ match start of string	\f = form feed
= anything but letters(periods)	| matches either of x/y	
\b = any character except for new line	[] = range of "variance"	

Syntax :

Import re

RE :

• "re" module included with Python primarily used for string searching and manipulation.

• Also used frequently for web page "Scraping“.

Example :

"^": This expression matches the start of a string.

"w+": This expression matches the alphanumeric character in the string.

• Here we will see an example of how we can use w+ and ^ expression in our code.

Example :

```
import re

a = "digitallync111,education"

b = re.findall(r"^\w+",a)

print (b)
```

Output :

```
['digitallync111']
```

• for our string “digitallync111, education" if we execute the code with w+ and^, it will give the output " digitallync111 ".

Example :

```
import re

a = "digitallync111,education"
```

```
b = re.findall(r"^\w",a)

print (b)
```

Output :

```
['d']
```

· if you remove +sign from the w+, the output will change, and it will only give the first character of the first letter, i.e., [d].

\s expression in re.split : "s": This expression is used for creating a space in the string.

· To understand how this regular expression works in Python, we begin with a simple example of a split function. In the example, we have split each word using the "re.split" function and at the same time we have used expression \s that allows to parse each word in the string separately.

Example :

```
import re

print (re.split(r'\s','we are splitting the words'))
```

Output :

```
['we', 'are', 'splitting', 'the', 'words']
```

· Now, let see what happens if you remove "\" from s. There is no 's' alphabet in the output, this is because we have removed '\' from the string, and it evaluates "s" as a regular character and thus split the words wherever it finds "s" in the string.

Example :

```
import re

print (re.split(r's','we are splitting the words'))
```

Output :

```
['we are ', 'plitting the word', '']
```

• There are series of other regular expressions in Python that you can use in various ways in Python like \d,\D,$,\.,\b, etc.

Regular Expression Methods : The "re" package provides several methods to actually perform queries on an input string. The method we going to see are

• re.match()

• re.search()

• re.findall()

re.match() :

• The match function is used to match the RE pattern to string with optional flags. In this method, the expression "w+" and "\W" will match the words starting with letter ‘d' and thereafter, anything which is not started with ‘d' is not identified. To check match for each element in the list or string, we run the forloop.

Example :

```
import re

list = ['digital get','digital give','digital nice',]

for a in list:

        z = re.match("(d\w+)\W(n\w+)",a)

        if z:

                print (z.groups())
```

Output :

```
('digital', 'nice')
```

re.search() :

· A regular expression is commonly used to search for a pattern in a text.

· This method takes a regular expression pattern and a string and searches for that pattern with the string.

· In order to use search() function, you need to import re first and then execute the code. The search() function takes the "pattern" and "text" to scan from our main string and returns a match object when the pattern is found or else not match if the pattern is not found.

Example :

```
import re

patterns = ["python ","digital-lync"]

text = 'python is a high-level general-purpose
programming language. '

for pattern in patterns:

print('looking for "%s" in "%s"'%(pattern,text)),

        if re.search(pattern,text):

                print ("found a match")

        else:

                print("no match")
```

Output :

```
looking for "python " in "python is a high-level general-purpose
programming language. "

        found a match

looking for "digital-lync" in "python is a high-level general-
purpose programming language. "

        no match
```

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.

Example :

```
import re

a = "abc@digitallync.com, abc@digital-lync.com,digital-lync@yahoomail.com"

emails = re.findall(r'[\w\.-]+@[\w\.-]+',a)

for email in emails:

    print (email)
```

Output :

```
abc@digitallync.com

abc@digital-lync.com

digital-lync@yahoomail.com
```

Python Flags :

• Many Python Regex Methods and Regex functions take an optional argument called Flags. This flags can modify the meaning of the given Regex pattern. To understand these we will see one or two exampleof these Flags.

• Various flags that are used in python

Syntax for Regex Flags	What does flag do
[re.M]	Make begin/end consider each line
[re.I]	It ignores case
[re.S]	Make [.]
[re.U]	Make {\w,\W,\b,\B} follows Unicode rules
[re.L]	Make {\w,\W,\b,\B} follows locale
[re.X]	Allow Comment in Regex

• In multiline the pattern character [^] match the first character of the string and the beginning of each line (following immediately after the each newline).

• While expression small "w" is used to mark the space with characters. When you run the code the first variable "k1" only prints out the character ‘d' for word digital-lync, while when you add multiline flag, it fetches out first characters of all the elements in the string.

Example :

```
import re

a = """digital-lync

python

Datascience"""

k1 = re.findall(r"^\w",a)

k2 = re.findall(r"^\w",a , flags = re.MULTILINE)
```

```
print (k1)

print (k2)
```

Output :

```
['d']

['d', 'p', 'D']
```