1. Python – Check whether a string starts and ends with the same character or not.

In [1]:
```
import re

regex = r'^[a-z]$|^([a-z]).*\1$'

def check(string):

    if(re.search(regex, string)):
        print("Valid")
    else:
        print("Invalid")

if __name__ == '__main__' :

    sample1 = "abba"
    sample2 = "a"
    sample3 = "abcd"

    check(sample1)
    check(sample2)
    check(sample3)
```

```
Valid
Valid
Invalid
```

2. Python regex to find sequences of one upper case letter followed by lower case letters

In [2]:
```
import re

def match(text):

        pattern = '[A-Z]+[a-z]+$'

        if re.search(pattern, text):
                return('Yes')
        else:
                return('No')

print(match("Geeks"))
print(match("geeksforGeeks"))
print(match("geeks"))
```

```
Yes
Yes
No
```

3. Python Program to Remove duplicate words from Sentence

In [3]:
```
from collections import Counter

def remov_duplicates(input):

    input = input.split(" ")

    UniqW = Counter(input)

    s = " ".join(UniqW.keys())
    print (s)

if __name__ == "__main__":
    input = 'Python is great and Java is also great'
    remov_duplicates(input)
```

```
Python is great and Java also
```

4. Python | Remove all characters except letters and numbers.

In [4]:
```
import re

ini_string = "123abcjw:, .@! eiw"

print ("initial string : ", ini_string)


result = re.sub('[\W_]+', '', ini_string)

print ("final string", result)
```

```
initial string :  123abcjw:, .@! eiw
final string 123abcjweiw
```

5. Python Regex | Program to accept string ending with alphanumeric character.

In [5]:
```
import re

regex = '[a-zA-z0-9]$'

def check(string):

    if(re.search(regex, string)):
        print("Accept")

    else:
        print("Discard")

if __name__ == '__main__' :

    string = "ankirai@"

    check(string)

    string = "ankitrai326"
    check(string)

    string = "ankit."
    check(string)

    string = "geeksforgeeks"
    check(string)
```

```
Discard
Accept
Discard
Accept
```

6. Python Regex – Program to accept string starting with vowel.

In [6]:
```
import re

regex = '^[aeiouAEIOU][A-Za-z0-9_]*'

def check(string):

    if(re.search(regex, string)):
        print("Valid")

    else:
        print("Invalid")

if __name__ == '__main__' :

    string = "ankit"

    check(string)

    string = "geeks"
    check(string)

    string = "sandeep"
    check(string)
```

```
Valid
Invalid
Invalid
```

7. Python Program to check if a string starts with a substring using regex.

In [7]:
```
import re

def find(string, sample) :

  if (sample in string):

      y = "^" + sample

      x = re.search(y, string)

      if x :
          print("string starts with the given substring")

      else :
          print("string doesn't start with the given substring")

  else :
      print("entered string isn't a substring")

string = "geeks for geeks makes learning fun"
sample = "geeks"

find(string, sample)

sample = "makes"

find(string, sample)
```

```
string starts with the given substring
string doesn't start with the given substring
```

8. Python Program to Check if an URL is valid or not using Regular Expression.

In [9]:
```
import re

def isValidURL(str):

    regex = ("((http|https):\/\/)(www.)?" +
             "[a-zA-Z0-9@:%._\\+~#?&\/\=]" +
             "{2,256}\\.[a-z]" +
             "{2,6}\\b([-a-zA-Z0-9@:%" +
             "._\\+~#?&\/\=]*)")

    p = re.compile(regex)

    if (str == None):
        return False

    if(re.search(p, str)):
        return True
    else:
        return False

url = "https://www.wikipedia.org"

if(isValidURL(url) == True):
    print("Yes")
else:
    print("No")
```

```
Yes
```

9. Parsing and Processing URL using Python – Regex.

In [11]:
```
import re

s = 'https://www.wikipedia.org/'

obj1 = re.findall('(\w+)://',
                  s)
print(obj1)

obj2 = re.findall('://www.([\w\-\.]+)',
                  s)
print(obj2)
```

```
['https']
['wikipedia.org']
```

10. Python Program to validate an IP address using ReGex.

In [12]:
```
import re

regex = "^((25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])\.){3}(25[0-5]|2[0-4][0-9]|1[0-9][0-9]|[1-9]?[0-9])$"

def check(Ip):

    if(re.search(regex, Ip)):
        print("Valid Ip address")

    else:
        print("Invalid Ip address")

if __name__ == '__main__' :

    Ip = "192.168.0.1"

    check(Ip)

    Ip = "110.234.52.124"
    check(Ip)

    Ip = "366.1.2.2"
    check(Ip)
```

```
Valid Ip address
Valid Ip address
Invalid Ip address
```

In [ ]: