In [1]:
```
print('nandini')
print(r'nandini') #we can use r or R
print(r'nandini'=='nandini')
```

```
nandini
nandini
True
```

In [2]:
```
print(r'the \n the')
```

```
the \n the
```

In [3]:
```
print('the \n the')
```

```
the
 the
```

In [4]:
```
import re #importing regex
n = '''the ip addresses are as follows 172.45.78.109
comp1: 10.3.20.32
comp2: 10.4.20.34
comp3: 10.5.20.36
comp4: 11.5.20.44
comp5: 12.4.20.66'''
```

In [5]:
```
s1 = re.findall(r'\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3}',n)
print(f'ip addesses are:{s1}')
```

```
ip addesses are:['172.45.78.109', '10.3.20.32', '10.4.20.34', '10.5.20.36', '11.5.20.
44', '12.4.20.66']
```

In [6]:
```
s1 = re.findall(r'1[0-1].\d{1,3}.\d{1,3}.\d{1,3}',n) #also we can written as[0\1]
print(f'ip addesses are:{s1}')
```

```
ip addesses are:['10.3.20.32', '10.4.20.34', '10.5.20.36', '11.5.20.44']
```

In [7]:
```
print("find all matches for format month day")
matches = re.findall(r'[A-Z][a-z]+\s\d{1,2}',"these are match dates January 4, May 20,
print(f"given month date format.{matches}")
```

```
find all matches for format month day
given month date format.['January 4', 'May 20', 'December 10']
```

In [8]:
```
print("find all matches for format month day")
matches = re.findall(r'[A-Z][a-z]+\s(\d{1,2})',"these are match dates January 4, May 2
print(f"dates .{matches}")
```

```
find all matches for format month day
dates .['4', '20', '10']
```

In [9]:
```
print("find all matches for format month day")
matches = re.findall(r'([A-Z][a-z]+)\s(\d{1,2})',"these are match dates January 4, May
print(f"tuple month date format.{matches}")
```

```
find all matches for format month day
tuple month date format.[('January', '4'), ('May', '20'), ('December', '10')]
```

In [10]:
```
#filtering mails from data
g = "kalanandini18@gmail.com arjunredddy17@gmail.com abcd@gmail.com hello@yahoo.com"
```

```
g1 = re.findall(r"\w+@\w+.\w+",g)
print(g1)
```

```
['kalanandini18@gmail.com', 'arjunredddy17@gmail.com', 'abcd@gmail.com', 'hello@yaho
o.com']
```

In [11]:
```
#Search: it returns only the first match element from the target string
sr = "Emma is a girl \n Emma knows AI "
sr1 = re.search(r'\w{4}',sr)
print(sr1)
```

```
<re.Match object; span=(0, 4), match='Emma'>
```

In [12]:
```
st = "If you are strong enough, no one can stop you"
match_object = re.search('If',st)
print(f'type is object{match_object}')
```

```
type is object<re.Match object; span=(0, 2), match='If'>
```

In [13]:
```
match_object1 = re.search('no',st)
print(f'type is object{match_object1}')
```

```
type is object<re.Match object; span=(19, 21), match='no'>
```

In [14]:
```
match_object.start()
```

Out[14]:
```
0
```

In [15]:
```
match_object1.start()
```

Out[15]:
```
19
```

In [16]:
```
match_object.span()
```

Out[16]:
```
(0, 2)
```

In [17]:
```
match_object1.span()
```

Out[17]:
```
(19, 21)
```

In [18]:
```
source_str = "we need to inform him with the latest information"
info = re.search('inform',source_str)
info
```

Out[18]:
```
<re.Match object; span=(11, 17), match='inform'>
```

In [19]:
```
randomstr = "i am \\nandini"
print(randomstr)
re.search(r'\\nandini',randomstr)
```

Out[19]:
```
i am \nandini
<re.Match object; span=(5, 13), match='\\nandini'>
```

In [20]:
```
st = "If you are strong enough, no one can stop you"
print(st)
sb = re.sub('e','E',st)
sb
```

```
If you are strong enough, no one can stop you
```

Out[20]:
```
'If you arE strong Enough, no onE can stop you'
```

In [21]:
```
sb1 = re.sub('e','E',st,2)
sb1
```

Out[21]:
```
'If you arE strong Enough, no one can stop you'
```

In [22]:
```
# complie:
# It changes the string pattern in to a re.pattern object that can work upon
a = 'hat mat rat pat'
reg = re.compile('[r]at') # once created can be used multiple times
reg
re.compile(r'[r]at', re.UNICODE)
rplce = reg.sub('FOOD',a)
print(rplce)
# or we can simply use replace as used eariler
print()
rplc1 = re.sub('rat','FOOD',a)
print(rplc1)
```

```
hat mat FOOD pat

hat mat FOOD pat
```

In [23]:
```
# working with white spaces
chang = '''keep the blue flag
flying high
dear'''
chang
'keep the blue flag\nflying high \ndear'
new_str = re.sub('\n','',chang)
new_str
'keep the blue flagflying high dear'
```

Out[23]:
```
'keep the blue flagflying high dear'
```

In [24]:
```
# by using compile
cam = re.compile('\n')
cam1 = cam.sub(' ', chang)
cam1
```

Out[24]:
```
'keep the blue flag flying high  dear'
```

In [25]:
```
# match:
abc = "jessy loves python and pandas"
pattern = r"\w{6}"
result = re.match(pattern,abc)
print(result)
```

```
None
```

In [26]:
```
#search:
result1 = re.search(pattern,abc)
print(result1.group())
```

```
python
```

In [27]:
```
result23 = re.findall(pattern,abc)
print(result23)
```

```
['python', 'pandas']
abc = "jessy loves python and pandas"
pattern = r"\b\w{5}\b" # \b- boundary(only between that it takes)
result23 = re.findall(pattern,abc)
print(result23)
```

```
['python', 'pandas']
['jessy', 'loves']
```