Regular expression

- A regular expression is a sequence of character that match a pattern.
- it is a symbol representation.
- regex can be used to check if a string contain the specified search pattern.
- python has a built-in package called re,which can be used to work with regex. -we have three methods i.e.,
  - match() - The re.match() method finds match if it occurs at start of the string.
  - search() -
  - findall()
- metacharacter

In [31]:

```
## example for set charcter []

import re
data ='welcome to python session'
res = re.findall('[a-h]',data)
```

In [3]:

```
print(res)
```

```
['e', 'c', 'e', 'h', 'e']
```

In [32]:

```
import re
data ='welcome to python session'
res = re.findall('[0-9]',data)
print(res)
```

```
[]
```

In [33]:

```
import re
data ='welcome to python session 12'
res = re.findall('[0-9]',data)
print(res)
```

```
['1', '2']
```

In [34]:

```
# example of sequence(\)
data = input()
res=re.findall('\d',data)
print(res)
```

```
this is 50 dollar
['5', '0']
```

In [10]:

```
# example of dot(.)
data = 'hello world'
res = re.findall('he..o',data)
print(res)
```

```
['hello']
```

In [11]:

```
# example of ^
data = 'hello world'
res = re.findall('^he.+',data)
print(res)
```

```
['hello world']
```

In [15]:

```
data = 'hello world'
res = re.findall('^hello+',data)
print(res)
```

```
['hello']
```

In [16]:

```
data = 'hello world'
res = re.findall('^hello.+',data)
print(res)
```

```
['hello world']
```

In [17]:

```
data = 'hello world'
res = re.findall('^world',data)
print(res)
```

```
[]
```

In [18]:

```
data = 'hello world'
res = re.findall('^world$',data)
print(data)
```

```
hello world
```

In [20]:

```
data = 'hello world'
res = re.findall('^hello$',data)
print(res)
```

```
[]
```

In [21]:

```
data = 'hello world'
res = re.findall('^hello$',data)
print(data)
```

```
hello world
```

In [22]:

```
# example of *
data = 'hello world hello'
res = re.findall('he*',data)
print(res)
```

```
['he', 'he']
```

In [35]:

```
data = 'hello world hello'
res = re.findall('p*',data)
print(res)
```

```
['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '']
```

In [7]:

```
# example of +
data = input()
res=re.findall('pyt+',data)
print(res)
```

```
python programming


---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-4e25a4c4efc3> in <module>
      1 # example of +
      2 data = input()
----> 3 res=re.findall('pyt+',data)
      4 print(res)

NameError: name 're' is not defined
```

In [6]:

```
data = input()
res=res.findall('pyt+',data)
print(res)
```

```
python programming


---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-6-db6d4cea7e9f> in <module>
      1 data = input()
----> 2 res=res.findall('pyt+',data)
      3 print(res)

NameError: name 'res' is not defined
```

In [12]:

```
# example of {}
data = input()
res=res.match('[0-9]',data)
print(res)
```

```
12345
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-12-f1b5037d84f4> in <module>
      1 # example of {}
      2 data = input()
----> 3 res=res.match('[0-9]',data)
      4 print(res)

NameError: name 'res' is not defined
```

In [11]:

```
data = input()
res=re.match('[0-9]{5}',data)
print(res)
```

```
12345
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-11-004637b3cd29> in <module>
      1 data = input()
----> 2 res=re.match('[0-9]{5}',data)
      3 print(res)

NameError: name 're' is not defined
```

In [26]:

```
## example of /
## examle of {}
data = input()
if(re.match('^[6-9][0-9]{9}$|[+][9][1][-][6-9][0-9]{9}$',data)):
    print(data)
else:
    print('invalid pno')
```

```
+91-6304483781
+91-6304483781
```

In [28]:

```
# example of ()
data = input()
res = re.search('([a-zA-Z]+)(\s)(\d+)',data)
print(res)
```

```
july 12
<re.Match object; span=(0, 7), match='july 12'>
```

The re.match()Mehod

In [36]:

```
re.match('k','avinash')
```

In [37]:

```
name = input().split()
for i in name:
    if re.match('k',i):
        print(i)
    else:
        print('Invalid name')
```

```
avinash rajesh ram sai
Invalid name
Invalid name
Invalid name
Invalid name
```

In [38]:

```
name = input().split()
for i in name:
    if re.match('a',i):
        print(i)
    else:
        print('Invalid name')
```

```
avinash
avinash
```

In [39]:

```
name = input().split()
for i in name:
    if re.match('a',i):
        print(i)
    else:
        print('Invalid name')

```

```
avinash avi
avinash
avi
```

In [40]:

```
name = input().split()
for i in name:
    if re.search('a',i):
        print(i)
```

```
avinash sai keerthana pavan
avinash
sai
keerthana
pavan
```

In [41]:

```
name = input().split()
for i in name:
    if re.search('a',i):
        print(i)
```

```
ssu sdm
```

task. 1-email starts with character(upper|lower|digit). 2-starts with characters(upper|lower|digits|.|_).

In [44]:

```
# email validation
data = input()
if(re.search('^[a-z]|[A-Z]|[0-9]|[.][_]{5-25}[@][a-z]{5-10}[.][a-z]{2-5}$',data)):
    print(data)
else:
    print('invalid email')
```

```
avinashgedela2001@gmail.com
avinashgedela2001@gmail.com
```

In [47]:

```
data = input()
pattern  = '^[a-zA-Z0-9][a-zA-Z0-9._]{5,25}[@][a-z]{5,10}[.][a-z]{2,5}$'
if re.match(pattern,data):
    print(data)
else:
    print('invaslid email')
```

```
avinashgedela2001@gmail.com
avinashgedela2001@gmail.com
```

In [ ]:

```

```