**20/04/2019 Lecture26**

**Regular Expressions**

**Match**

>>> import re >>> re.match(t) Traceback (most recent call last): File "<stdin>", line 1, in <module> NameError: name 't' is not defined le "<stdin>", line 1, in <module> AttributeError: 'module' object has no attribute 'fullmatch' >>> match=re.match('\w+',"tiger") >>> match.start(),match.end(),match.group(0) (0, 5, 'tiger') >>> match=re.match('t',"ti,ger") >>> match.start(),match.end(),match.group(0) (0, 1, 't') >>> match=re.match('(\w+)\,(\w+)',"tig.er") >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2) Traceback (most recent call last): File "<stdin>", line 1, in <module> AttributeError: 'NoneType' object has no attribute 'start' >>> match=re.match("(\w+)\,(\w+)","tig.er") >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2) Traceback (most recent call last): File "<stdin>", line 1, in <module> AttributeError: 'NoneType' object has no attribute 'start' >>> match=re.match('(\w+)\.(\w+)',"tig.er") >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2) (0, 6, 'tig.er', 'tig', 'er') >>> match=re.match("(\w+)\,(\w+)","tiger") >>> match=re.match("(\w+)\,(\w+)","ti,g.er") >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2) (0, 4, 'ti,g', 'ti', 'g') >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2),match.group(3) Traceback (most recent call last): File "<stdin>", line 1, in <module> IndexError: no such group >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2) (0, 4, 'ti,g', 'ti', 'g') >>> match=re.match('(\w+)\.(\w+)',"tig.er") >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2) (0, 6, 'tig.er', 'tig', 'er') >>> match=re.match("(\w+)\,(\w+)\.","ti,g.er") >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2) (0, 5, 'ti,g.', 'ti', 'g') >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2),match.group(3) Traceback (most recent call last): File "<stdin>", line 1, in <module> IndexError: no such group >>> match=re.match("(\w+)\,(\w+)\.(\w+)","ti,g.er") >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2),match.group(3) (0, 7, 'ti,g.er', 'ti', 'g', 'er') >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2) (0, 4, 'ti,g', 'ti', 'g')

**Search** >>> match=re.search("(\w+)\.(\w+)","ti,g.er") >>> match.start(),match.end(),match.group(0),match.group(1),match.group(2) (3, 7, 'g.er', 'g', 'er') >>>

**Iterator**

>>> iter=re.finditer("(\w+)\.(\w+)","swapnil$tig.er") >>> for it in iter: ... it.start(),it.end(),it.group(0),it.group(1),it.group(2) ... (8, 14, 'tig.er', 'tig', 'er') >>> iter=re.finditer("(\w+)\@(\w+)\.(\w+)","swa1312@gmail.com") >>> for it in iter: ... it.start(),it.end(),it.group(0),it.group(1),it.group(2) ... (0, 17, 'swa1312@gmail.com', 'swa1312', 'gmail') >>>

**#WAP to accept email id from user and print if it is valid or not**

import re

def checkEmail(checkEmailList):

valid=[]

invalid=[]

for email in checkEmailList:

emailCheck=re.match("(\w+)\@(\w+)\.(\w+)",email)

if emailCheck:

valid.append(email)

print(email,"Valid Email")

else:

invalid.append(email)

print("Invalid Email")

print("Valid Email",valid)

print("invalid email",invalid)

def main():

checkEmailList=[]

while(True):

email=input("Enter email id to stop enter n")

if (email=='n'):

break;

else:

emailID=email

checkEmailList.append(email)

checkEmail(checkEmailList)

if \_\_name\_\_=="\_\_main\_\_":

main()

#WAP to read file from user and print all email ids in it.

#Refine re expression for email to check dots in string before@

**Compile**

compile is faster that all other re methods

>>> regExObj=re.compile("(\w+)\.(\w+)") >>> regExObj.match("ti.ger")

>>> regExObj.match("li.on")

>>> regExObj.sub('hello',"ti.ger li.on")

'hello hello'

>>> re.sub("\w+\.\w+",'hello',"ti.ger li.on swapnil")

'hello hello swapnil'

>>> re.subn("\w+\.\w+","hello","ti.ger li.on swapnil",2)

('hello hello swapnil', 2)

>>> re.subn("\w+\.\w+","hello","ti.ger li.on swapnil",1)

('hello li.on swapnil', 1)

**#RE with flag**

import re

text="this is some text -- with punctuation"

pattern='T\w+'

with\_case= re.compile(pattern)

without\_case=re.compile(pattern,re.IGNORECASE)

print("Text:",text)

print("Pattern:",pattern)

print("Case-sensitive : ",with\_case.findall(text))

print("Case insenitive : ",without\_case.findall(text))

**Split**

>>> re.split("@","swa@lo")

['swa', 'lo']

>>> re.split("@","swa@gmail")

['swa', 'gmail']

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Special characters

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

>>> re.search("a.b", "a\*b")

<\_sre.SRE\_Match object; span=(0, 3), match='a\*b'>

>>> re.search("a.b", "a~b")

<\_sre.SRE\_Match object; span=(0, 3), match='a~b'>

>>> re.search("a.b", "ab")

>>> re.search("a.b", "abb")

<\_sre.SRE\_Match object; span=(0, 3), match='abb'>

>>> re.search("a.b", "a b")

<\_sre.SRE\_Match object; span=(0, 3), match='a b'>

>>> re.search("^a.b", "a\*b")

<\_sre.SRE\_Match object; span=(0, 3), match='a\*b'>

>>> re.search("^a.b", "a~b")

<\_sre.SRE\_Match object; span=(0, 3), match='a~b'>

>>> re.search("^a.b", "ab")

>>> re.search("^a.b", "a b")

<\_sre.SRE\_Match object; span=(0, 3), match='a b'>

<\_sre.SRE\_Match object; span=(0, 3), match='a b'>

>>> re.search("a.b", "kadfkjgasa~b")

<\_sre.SRE\_Match object; span=(9, 12), match='a~b'>

>>> re.search("^a.b", "kadfkjgasa~b")

##### dont return anything as it searches at start

>>> re.search("a.b$", "kadfkjgasa~b")

<\_sre.SRE\_Match object; span=(9, 12), match='a~b'>

>>> re.search("a.b$", "kadfa~bkjgas")

##### dont return anything as it searches at end

>>> for match in re.finditer('a\*','babababababaaabbaaab'):

... print(match)

...

<\_sre.SRE\_Match object; span=(0, 0), match=''>

<\_sre.SRE\_Match object; span=(1, 2), match='a'>

<\_sre.SRE\_Match object; span=(2, 2), match=''>

<\_sre.SRE\_Match object; span=(3, 4), match='a'>

<\_sre.SRE\_Match object; span=(4, 4), match=''>

<\_sre.SRE\_Match object; span=(5, 6), match='a'>

<\_sre.SRE\_Match object; span=(6, 6), match=''>

<\_sre.SRE\_Match object; span=(7, 8), match='a'>

<\_sre.SRE\_Match object; span=(8, 8), match=''>

<\_sre.SRE\_Match object; span=(9, 10), match='a'>

<\_sre.SRE\_Match object; span=(10, 10), match=''>

<\_sre.SRE\_Match object; span=(11, 14), match='aaa'>

<\_sre.SRE\_Match object; span=(14, 14), match=''>

<\_sre.SRE\_Match object; span=(15, 15), match=''>

<\_sre.SRE\_Match object; span=(16, 19), match='aaa'>

<\_sre.SRE\_Match object; span=(19, 19), match=''>

<\_sre.SRE\_Match object; span=(20, 20), match=''>

>>> for match in re.finditer('a+','bababaaaabbaaab'):

... print(match)

...

<\_sre.SRE\_Match object; span=(1, 2), match='a'>

<\_sre.SRE\_Match object; span=(3, 4), match='a'>

<\_sre.SRE\_Match object; span=(5, 9), match='aaaa'>

<\_sre.SRE\_Match object; span=(11, 14), match='aaa'>

>>> for match in re.finditer('a+?','bababaaaabbaaab'):

... print(match)

...

<\_sre.SRE\_Match object; span=(1, 2), match='a'>

<\_sre.SRE\_Match object; span=(3, 4), match='a'>

<\_sre.SRE\_Match object; span=(5, 6), match='a'>

<\_sre.SRE\_Match object; span=(6, 7), match='a'>

<\_sre.SRE\_Match object; span=(7, 8), match='a'>

<\_sre.SRE\_Match object; span=(8, 9), match='a'>

<\_sre.SRE\_Match object; span=(11, 12), match='a'>

<\_sre.SRE\_Match object; span=(12, 13), match='a'>

<\_sre.SRE\_Match object; span=(13, 14), match='a'>

>>> for match in re.finditer('a.','bababaaaabbaaab'):

... print(match)

...

<\_sre.SRE\_Match object; span=(1, 3), match='ab'>

<\_sre.SRE\_Match object; span=(3, 5), match='ab'>

<\_sre.SRE\_Match object; span=(5, 7), match='aa'>

<\_sre.SRE\_Match object; span=(7, 9), match='aa'>

<\_sre.SRE\_Match object; span=(11, 13), match='aa'>

<\_sre.SRE\_Match object; span=(13, 15), match='ab'>

>>> for match in re.finditer('a\*?','bababaaaab'):

... print(match)

...

<\_sre.SRE\_Match object; span=(0, 0), match=''>

<\_sre.SRE\_Match object; span=(1, 1), match=''>

<\_sre.SRE\_Match object; span=(2, 2), match=''>

<\_sre.SRE\_Match object; span=(3, 3), match=''>

<\_sre.SRE\_Match object; span=(4, 4), match=''>

<\_sre.SRE\_Match object; span=(5, 5), match=''>

<\_sre.SRE\_Match object; span=(6, 6), match=''>

<\_sre.SRE\_Match object; span=(7, 7), match=''>

<\_sre.SRE\_Match object; span=(8, 8), match=''>

<\_sre.SRE\_Match object; span=(9, 9), match=''>

<\_sre.SRE\_Match object; span=(10, 10), match=''>

>>> for match in re.finditer('a{1,3}','bababaaaab'):

... print(match)

...

<\_sre.SRE\_Match object; span=(1, 2), match='a'>

<\_sre.SRE\_Match object; span=(3, 4), match='a'>

<\_sre.SRE\_Match object; span=(5, 8), match='aaa'>

<\_sre.SRE\_Match object; span=(8, 9), match='a'>

>>> for match in re.finditer('a{1,4}','bababaaaab'):

... print(match)

...

<\_sre.SRE\_Match object; span=(1, 2), match='a'>

<\_sre.SRE\_Match object; span=(3, 4), match='a'>

<\_sre.SRE\_Match object; span=(5, 9), match='aaaa'>

>>> for match in re.finditer('a{1,3}?','bababaaaab'):

... print(match)

...

<\_sre.SRE\_Match object; span=(1, 2), match='a'>

<\_sre.SRE\_Match object; span=(3, 4), match='a'>

<\_sre.SRE\_Match object; span=(5, 6), match='a'>

<\_sre.SRE\_Match object; span=(6, 7), match='a'>

<\_sre.SRE\_Match object; span=(7, 8), match='a'>

<\_sre.SRE\_Match object; span=(8, 9), match='a'>

>>> for match in re.finditer('a[0-9]','baba1baa4aa3b'):

... print(match)

...

<\_sre.SRE\_Match object; span=(3, 5), match='a1'>

<\_sre.SRE\_Match object; span=(7, 9), match='a4'>

<\_sre.SRE\_Match object; span=(10, 12), match='a3'>

>>> for match in re.finditer('a[12]','baba1ba2a4aa3b'):

... print(match)

...

<\_sre.SRE\_Match object; span=(3, 5), match='a1'>

<\_sre.SRE\_Match object; span=(6, 8), match='a2'>

>>> for match in re.finditer('a[12]','baba1ba2a4aa12b'):

... print(match)

...

<\_sre.SRE\_Match object; span=(3, 5), match='a1'>

<\_sre.SRE\_Match object; span=(6, 8), match='a2'>

<\_sre.SRE\_Match object; span=(11, 13), match='a1'>

>>> for match in re.finditer('a[1][2]','baba1ba2a4aa12b'):

... print(match)

...

<\_sre.SRE\_Match object; span=(11, 14), match='a12'>

>>> for match in re.finditer('a[a-zA-Z0-9]','baba1ba2a4aa12b'):

... print(match)

...

<\_sre.SRE\_Match object; span=(1, 3), match='ab'>

<\_sre.SRE\_Match object; span=(3, 5), match='a1'>

<\_sre.SRE\_Match object; span=(6, 8), match='a2'>

<\_sre.SRE\_Match object; span=(8, 10), match='a4'>

<\_sre.SRE\_Match object; span=(10, 12), match='aa'>

>>> for match in re.finditer('[^a-zA-Z0-9]+','baba1ba2$@%$a4aa12b'):

... print(match)

...

<\_sre.SRE\_Match object; span=(8, 12), match='$@%$'>

##### ^ here is used as negation which will display string not containing 'a-zA-Z0-9' (given RE).

>>> for match in re.finditer('a|b','baba1baa'):

... print(match)

...

<\_sre.SRE\_Match object; span=(0, 1), match='b'>

<\_sre.SRE\_Match object; span=(1, 2), match='a'>

<\_sre.SRE\_Match object; span=(2, 3), match='b'>

<\_sre.SRE\_Match object; span=(3, 4), match='a'>

<\_sre.SRE\_Match object; span=(5, 6), match='b'>

<\_sre.SRE\_Match object; span=(6, 7), match='a'>

<\_sre.SRE\_Match object; span=(7, 8), match='a'>

Read next slide - special sequences