PYTHON ADVANCED PROGRAMMING

Chitra G M and P Rama Devi
Department of Computer Science Engineering

PYTHON ADVANCED PROGRAMMING

P Rama Devi

Department of Computer Science and Engineering

PYTHON ADVANCED PROGRAMMING

Character Set:

Uses [ ] with characters we want to match

Example: re.compile(r'\d\d\d[-.]\d\d\d[-.]\d\d\d\d')

This will match any character either . Or – given in charater set

If input is 321--431-1234

Above will not be matched since [-.] will match either – or . But not second – given in input.

PYTHON ADVANCED PROGRAMMING

If we want to match phone numbers starting from 800 or 900

Example: 800-123-6721
900-213-4567

pattern=re.compile(r'[89]00[-.]\d\d\d[-.]\d\d\d\d')

Output:

800-123-6721
900-213-4567

PYTHON ADVANCED PROGRAMMING

Example: matches digits between 1 to 5

pattern=re.compile(r'[1-5]')

In the above 1-5 defines range

re.compile(r'[a-z]')

re.compile(r'[a-zA-Z]')

re.compile(r'^[A-Z][a-z]')

Above one will negate everything not in character set. i.e not a lower case , upper case

Example:
Cat
Mat
Pat
Bat

Print all which end with at but not bat

So we can use re.compile(r'[^B]at')

Output:

Cat
Mat
Pat

Quantifiers:

IN previous examples for phone numbers we used \d multiple times instead we can use quantifiers

* - 0 or 1
\+ - 1 or more
? - 0 or one
{3} – exact number
{3,4} – range of numbers (minimum , maximum)

PYTHON ADVANCED PROGRAMMING

Example: to match names
text='''Mr. Ram
Mr Kevin
Ms Kavitha
Mrs. Devi
Mr. T'''

pattern=re.compile(r'Mr\.')
matches = pattern.finditer(text)
for match in matches:
        print(match)

Output will give 2 'Mr.' but not 'Mr'

PYTHON ADVANCED PROGRAMMING

So we should tell '.' is optional

Therefore re.compile(r'Mr\.?')

Will give all 'Mr'

re.compile(r'Mr\.?\s[A-Z]\w*')

Output:

 Mr. Ram
 Mr Kevin
 Mr. T

This did not match Ms, Mrs. Lets see use of groups in next class where we can use groups concept to print all names

THANK YOU

Chitra G M and P Rama Devi
Department of CSE

pramadevi@pes.edu
chitragm@pes.edu