# **RegEx Project Solution**

1. Text ='Python,Exercises, PHP,exercises.'

substitute = re.sub(r"[ ,.]", ":", Text)

print(substitute)

Output: Python:Exercises::PHP:exercises:

1. import pandas as pd

import regex as re

dict = {'SUMMARY':['hello,world!', 'XXXXX test', '123four, five:; six...']}

df = pd.DataFrame(dict)

df['SUMMARY'] = df['SUMMARY'].apply(lambda x: re.sub(r'[^\w\s]', '', x))

print(df['SUMMARY'])

Output: 0 helloworld

1 test

1. 123four five six
2. string = "This is a sample string with words of different length"

def find\_words(string):

pattern = re.compile(r'\b\w{4}\b')

matches = pattern.findall(string)

return matches

result = find\_words(string)

print(result)

Output: ['This', 'with']

1. string = "This is a sample string with words of different length"

def find\_words(string):

pattern = re.compile(r'\b\w{3,5}\b')

matches = pattern.findall(string)

return matches

result = find\_words(string)

print(result)

Output: ['This', 'with', 'words']

1. sample\_text = ["example(.com)", "hr@fliprobo(.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]

def remove\_parenthesis(text):

for item in text:

re\_parenthesis=re.compile(r" ?\(| ?\)")

print(re\_parenthesis.sub(" ", item))

remove\_parenthesis(sample\_text)

Output: example .com

hr@fliprobo .com

github .com

Hello Data Science World

Data Scientist

1. with open("Sample\_Text\_1.txt") as file:

for items in file:

print(re.sub(r" ?\([^)]+\)", "", items), end= " ")

Output: ["example", "hr@fliprobo", "github", "Hello", "Data"]

1. text = "ImportanceOfRegularExpressionsInPython"

result = re.findall('[A-Z][^A-Z]\*' , text)

print(result)

Output: ['Importance', 'Of', 'Regular', 'Expressions', 'In', 'Python']

1. text = "RegularExpression1IsAn2ImportantTopic3InPython"

def insert\_spaces(text):

# Use regular expression to find words starting with numbers

pattern = r'([A-Za-z]+)(\d+)'

result = re.sub(pattern, r'\1 \2',text)

return result

output = insert\_spaces(text)

print(output)

Output: RegularExpression 1IsAn 2ImportantTopic 3InPython

1. text = "RegularExpression1IsAn2ImportantTopic3InPython"

def insert\_spaces(text):

# Use regular expression to find words starting with numbers

pattern = r'(\d+)([A-Za-z]+)'

result = re.sub(pattern, r' \1 \2', text)

return result

output = insert\_spaces(text)

print(output)

Output: RegularExpression 1 IsAn 2 ImportantTopic 3 InPython

1. url="https://raw.githubusercontent.com/dsrscientist/DSData/master/happiness\_score\_dataset.csv"

df = pd.read\_csv(url)

df['first\_five\_letters'] = df['Country'].apply(lambda x: x[:6])

print(df['first\_five\_letters'])

Output: 0 Switze

1 Icelan

2 Denmar

3 Norway

4 Canada

...

153 Rwanda

154 Benin

155 Syria

156 Burund

157 Togo

Name: first\_five\_letters, Length: 158, dtype: object

1. import re

string = "There is 404\_Error"

def match\_string(string):

pattern = r"^[a-zA-Z0-9\_]+$"

if re.match(pattern, string):

print("String matches the pattern")

else:

print("String does not match the pattern")

Output: String matches the pattern

1. string = '143 Data World'

number = 143

def check\_starting\_number(string, number):

if string.startswith(str(number)):

return True

else:

return False

if check\_starting\_number(string, number):

print("The string starts with the specified number.")

else:

print("The string does not start with the specified number.")

Output: The string starts with the specified number.

1. ip\_address = '188.122.008.004'

def remove\_leading\_zeros(ip\_address):

# Split the IP address into octets

octets = ip\_address.split('.')

# Remove leading zeros from each octet

octets\_without\_zeros = [str(int(octet))for octet in octets]

# Join the octets back into a string

ip\_address\_without\_zeros = ('.'.join(octets\_without\_zeros))

return ip\_address\_without\_zeros

ip\_address\_without\_zeros = remove\_leading\_zeros(ip\_address)

print(ip\_address\_without\_zeros)

Output: 188.122.8.4

1. with open("Text.txt") as file:

for line in file:

pattern = "([a-zA-Z]+) (\d+[a-z]+) (\d+)"

matched = re.search(pattern, line)

print ("%s" % (matched.group()))

Output: August 15th 1947

1. import re

patterns = [ 'fox', 'dog', 'horse' ]

text = 'The quick brown fox jumps over the lazy dog.'

for pattern in patterns:

if re.search(pattern, text):

print('%s words is in String' %pattern)

else:

print('%s words is not in string' %pattern)

Output: fox words is in String

dog words is in String

horse words is not in string

1. pattern = 'fox'

text = 'The quick brown fox jumps over the lazy dog.'

match = re.search(pattern, text)

s = match.start()

e = match.end()

print('Found "%s" in "%s" from %d to %d ' % \

(match.re.pattern, match.string, s, e))

Output: Found "fox" in "The quick brown fox jumps over the lazy dog." from 16 to 19

1. text = 'Python exercises, PHP exercises, C# exercises'

pattern = 'exercises'

for match in re.findall(pattern, text):

print('Found "%s"' % match)

Output: Found "exercises"

Found "exercises"

Found "exercises"

1. text = 'Python exercises, PHP exercises, C# exercises'

pattern = 'exercises'

for match in re.finditer(pattern, text):

s = match.start()

e = match.end()

print('Found "%s" at %d:%d' % (text[s:e], s, e))

Output: Found "exercises" at 7:16

Found "exercises" at 22:31

Found "exercises" at 36:45

1. def change\_date\_format(dt):

return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', dt)

dt1 = "2023-10-04"

print("Original date in YYY-MM-DD Format: ",dt1)

print("New date in DD-MM-YYYY Format: ",change\_date\_format(dt1))

Output: Original date in YYY-MM-DD Format: 2023-10-04

New date in DD-MM-YYYY Format: 04-10-2023

1. text= "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"

decimal\_with\_precision(text)

def decimal\_with\_precision(string):

deci\_num = re.compile(r"\b([0-9]+\.[0-9]{1,2})\b")

result = deci\_num.findall(text)

print(result)

Output: ['01.12', '145.8', '3.01', '27.25', '0.25']

1. text = "Virat is a cricket player 1988 who was born on november 5, 1988"

for m in re.finditer("\d+", text):

print(m.group(0))

print("Index position:", m.start())

Output: 1988

Index position: 26

5

Index position: 56

1988

Index position: 59

1. def extractMax(input):

numbers = re.findall('\d+',input)

print(numbers)

numbers = map(int,numbers)

print ("Maximum Numeric value is ",max(numbers))

input = 'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'

extractMax(input)

Output: ['947', '896', '926', '524', '734', '950', '642']

Maximum Numeric value is 950

1. import re

def insert\_spaces(text):

pattern = r'([A-Z][a-z]+)'

result = re.sub(pattern, r' \1', text)

result = result.strip()

return result

sample\_text = "RegularExpressionIsAnImportantTopicInPython"

expected\_output = insert\_spaces(sample\_text)

print(expected\_output)

Output: Regular Expression Is An Important Topic In Python

1. def match(text):

pattern = '[A-Z][a-z]\*$'

if re.search(pattern, text):

return('Yes')

else:

return('No')

print(match('Good Morning'))

print(match('good Morning'))

print(match('GOOD MORNING'))

print(match('Good morning'))

Output: Yes

Yes

Yes

No

1. import regex as re

sample\_text = "Hello hello world world"

pattern = r'\b(\w+)(?:\W+\1\b)+'

a = re.sub(pattern, r'\1', sample\_text)

print(a)

Output: Hello hello world

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

def check\_alpha\_numeric(string):

if(re.search(regex, string)):

print("The string is ending with an alphanumeric character. \n")

else:

print("The string is not ending with an alphanumeric character. \n")

check\_alpha\_numeric("Be aware")

check\_alpha\_numeric("Indiain1947")

check\_alpha\_numeric("15August.")

check\_alpha\_numeric("Independenceday&")

Output: The string is ending with an alphanumeric character.

The string is ending with an alphanumeric character.

The string is not ending with an alphanumeric character.

The string is not ending with an alphanumeric character.

1. Sample\_Text = """RT @kapil\_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the same has rendered USELESS <ed><U+00A0><U+00BD><ed><U+00B1><U+0089> "acquired funds" No wo"""

hashtags = re.findall(r"#\w+", Sample\_Text)

print("LinkedIn:", Sample\_Text)

print("\n Hashtag:", hashtags)

Output: LinkedIn: RT @kapil\_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the same has rendered USELESS <ed><U+00A0><U+00BD><ed><U+00B1><U+0089> "acquired funds" No wo

Hashtag: ['#Doltiwal', '#xyzabc', '#Demonetization']

1. Sample\_text= "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who are protesting #demonetization are all different party leaders"

removedsymbol\_text = re.sub(r"<U\+[A-Z0-9]+>", "", Sample\_text)

print("Sample Text:", Sample\_text)

print("\n Expected Text:", removedsymbol\_text)

Output: Sample Text: @Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who are protesting #demonetization are all different party leaders

Expected Text: @Jags123456 Bharat band on 28??<ed><ed>Those who are protesting #demonetization are all different party leaders

1. import re

with open("sample\_text.txt") as file:

for dates in file:

extract\_dates = re.findall(r"\d{2}-\d{2}-\d{4}", dates)

print(extract\_dates)

Output: ['12-09-1992', '15-12-1999']

1. def remove\_words(string):

shortword = re.compile(r'\b\w{2,4}\b ')

print(shortword.sub('',sample\_text))

sample\_text = "The following example creates an ArrayList with a capacity of 50 elements. 4 elements are then added to the ArrayList and the ArrayList is trimmed accordingly."

print(remove\_words(sample\_text))

Output: following example creates ArrayList a capacity elements. 4 elements added ArrayList ArrayList trimmed accordingly.

​

​

​

1. def decimal\_with\_precision(string):
3. deci\_num = re.compile(r"\b([0-9]+\.[0-9]{1,2})\b")
4. result = deci\_num.findall(text)
5. print(result)
7. text= "01.12 0132.123 2.31875 145.8 3.01 27.25 0.2