REGULAR EXPRESSION

1.

```
 import regex as re

sub = 'Python Exercises, PHP exercises.'

x = re.sub(r"[\s+$|^\s+.,]",':',sub,)

print(x)
```

2.

```
import pandas as pd

import re

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

df = pd.DataFrame(text)

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

print(df)
```

3.

```
import re

input = 'the quick brown fox jumps upon the lazy dog'

def find_words(text):

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

    return pattern.findall(text)

words = find_words(text)

print(words)
```

4.

```
import re
```

```
input = 'the quick brown fox jumps upon the lazy dog'

def find_words(text):

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

    return pattern.findall(text)

words = find_words(input)

print(words)
```

5.

```
import re

text = ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]

def remove_parentheses(strings):

    pattern = re.compile(r'[()]')  # Create a pattern to match parentheses

    cleaned_strings = [pattern.sub('', s) for s in strings]

    return cleaned_strings

output = remove_parentheses(text)

print(output)
```

6.

```
import re

input_text: ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]

with open('input.txt', 'r') as f:

    text = f.read()

pattern = re.compile(r'[()]')

cleaned_text = pattern.sub('', text)
```

```
with open('output.txt', 'w') as output_file:
    output_file.write(cleaned_text)


print("Output written to 'output.txt'")
```

7.

```
import re
text = "ImportanceOfRegularExpressionsInPython"
result = re.findall(r'[A-Z][^A-Z]*', text)
print(result)
```

8.

```
import re
text = "RegularExpression1IsAn2ImportantTopic3InPython"
def insert_spaces(text):
    # Use regular expression to find words starting with numbers and insert spaces
    return re.sub(r'(\d)', r' \1', text)
result = insert_spaces(text)
print(result)
```

9.

```
import re
text = "RegularExpression1IsAn2ImportantTopic3InPython"
def insert_spaces(text):
    return re.sub(r'(?=[A-Z0-9])', r' ', text).strip()
```

```
result = insert_spaces(text)

print(result)
```

10.

11.

```
import re

def match_string(text):

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

    if re.match(pattern, text):

        return 'Found a match!'

    else:

        return 'Not matched!'

print(match_string("Valid_String123"))

print(match_string("Invalid-String!"))
```

12.

```
import re


def starts_with_number(text, number):

    pattern = rf'^{number}'

    if re.match(pattern, text):

        return 'String starts with the specified number!'

    else:

        return 'String does not start with the specified number.'

print(starts_with_number("5-2345861", 5))
print(starts_with_number("6-2345861", 5))
```

13.

```
import re

def remove_leading_zeros(ip):

    return re.sub(r'\b0+(\d)', r'\1', ip)

ip_address = "100.020.003.400"

cleaned_ip = remove_leading_zeros(ip_address)

print(cleaned_ip)

ip_address = "001.200.001.004"

cleaned_ip = remove_leading_zeros(ip_address)

print(cleaned_ip)
```

14.

```
import re

sample_text = 'On August 15th 1947 that India was declared independent from British
colonialism, and the reins of control were handed over to the leaders of the Country.'

with open('sample_text.txt', 'w') as file:

    file.write(sample_text)

with open('sample_text.txt', 'r') as file:

    text = file.read()

pattern = r'\b([A-Z][a-z]+ \d{1,2}(?:st|nd|rd|th)? \d{4})\b'

matches = re.findall(pattern, text)

for match in matches:

    print(match)
```

15.

```
import re
def search_literals(text, patterns):
    for pattern in patterns:
        if re.search(pattern, text):
            print(f'Searching for "{pattern}" in "{text}" -> Matched!')
        else:
            print(f'Searching for "{pattern}" in "{text}" -> Not Matched!')
text = 'The quick brown fox jumps over the lazy dog.'
patterns = ['fox', 'dog', 'horse']
search_literals(text, patterns)
```

16.import re

```
def search_literal(text, pattern):
    match = re.search(pattern, text)
    if match:
        start = match.start()
        end = match.end()
        return f'Found "{pattern}" in "{text}" from {start} to {end}'
    else:
        return f'"{pattern}" not found in "{text}"'
text = 'The quick brown fox jumps over the lazy dog.'
pattern = 'fox'
result = search_literal(text, pattern)
```

```
print(result)

17.

import re

def search_literal(text, pattern):
    match = re.search(pattern, text)
    if match:
        start = match.start()
        end = match.end()
        return f'Found "{pattern}" in "{text}" from {start} to {end}'
    else:
        return f'"{pattern}" not found in "{text}
text = 'The quick brown fox jumps over the lazy dog.'
pattern = 'fox'
result = search_literal(text, pattern)
print(result)

18.

import re


def find_substrings(text, pattern):
    matches = re.finditer(pattern, text)
    results = [(match.start(), match.end()) for match in matches]
    return results
text = 'Python exercises, PHP exercises, C# exercises'
```

```
pattern = 'exercises'
results = find_substrings(text, pattern)
for start, end in results:
    print(f'Found "{pattern}" from {start} to {end}')
```

19.

```
from datetime import datetime
def convert_date_format(date_str):
    original_date = datetime.strptime(date_str, '%Y-%m-%d')
    new_date = original_date.strftime('%d-%m-%Y')
    return new_date
date_str = '2024-08-21'
converted_date = convert_date_format(date_str)
print(f'Original date: {date_str}')
print(f'Converted date: {converted_date}')
```

20.

```
import re
def find_decimals(text):
    pattern = re.compile(r'\b\d+\.\d{1,2}\b')
    matches = pattern.findall(text)
    return matches
text = "01.12 0132.123 2.31875 145.8 3.01 27.25 0.25"
result = find_decimals(text)
print(result)
```

21.

```
import re

def find_numbers_with_positions(text):

    matches = re.finditer(r'\d+', text)

    for match in matches:

        print(f"Number: {match.group(0)}, Position: {match.start()}")

text = "The house number is 123 and the zip code is 45678."

find_numbers_with_positions(text)
```

22.

```
import re

def extract_max_number(text):

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

    numbers = list(map(int, numbers))

    return max(numbers) if numbers else None

text = "The house number is 123 and the zip code is 45678."

max_number = extract_max_number(text)

print(f"The maximum number in the string is: {max_number}")
```

23.

```
import re

def extract_max_number(text):

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

    numbers = list(map(int, numbers))

    return max(numbers) if numbers else None
```

```
text = 'My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'
max_number = extract_max_number(text)
print(f"The maximum number in the string is: {max_number}")
```

24.

```
import re
def insert_spaces(text):
   result = re.sub(r'([a-z])([A-Z])', r'\1 \2', text)
   return result
sample_text = "RegularExpressionIsAnImportantTopicInPython"
output = insert_spaces(sample_text)
```

25.

```
import re
def find_uppercase_sequences(text):
   pattern = r'[A-Z][a-z]+'
   matches = re.findall(pattern, text)
   return matches
text = "This is a Sample Text with Multiple Matches"
sequences = find_uppercase_sequences(text)
print(sequences)
```

26.

```
import re
def check_string(text):
   pattern = r'[a-zA-Z0-9]$'
```

```
    if re.search(pattern, text):

        return "Accept"

    else:

        return "Discard"

sample_text1 = "ankitrai326"

sample_text2 = "ankitrai@"

print(f"Sample Text: '{sample_text1}' - {check_string(sample_text1)}")

print(f"Sample Text: '{sample_text2}' - {check_string(sample_text2)}")
```

27.

```
import re



def extract_hashtags(text):

    pattern = r'#\w+'

    hashtags = re.findall(pattern, text)

    return hashtags

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 = extract_hashtags(sample_text)

print(hashtags)
```

28.

```
import re

def remove_unicode_symbols(text):

    pattern = r'<U\+[0-9A-Fa-f]{4}>'
    cleaned_text = re.sub(pattern, '', text)

    return cleaned_text

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_output = remove_unicode_symbols(sample_text)

print(expected_output)
```

29.

```
import re

def extract_dates_from_file(file_path):

    with open(file_path, 'r') as file:

        text = file.read()

    date_pattern = r'\b\d{2}-\d{2}-\d{4}\b'

    dates = re.findall(date_pattern, text)

    return dates

file_path = 'sample.txt'

extracted_dates = extract_dates_from_file(file_path)

print("Extracted Dates:", extracted_dates)
```

30.

```
import re

def remove_short_words(text):

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

    cleaned_text = pattern.sub('', text)

    cleaned_text = re.sub(r'\s+', ' ', cleaned_text).strip()
    return cleaned_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."

result = remove_short_words(sample_text)

print("Expected Output:", result)
```