

Practical-18

May 19, 2025

0.1 Practical 18:-

Name: Sarthak Sanay

Enrollment No: 230031101611051

0.1.1 Problem Statement 1:-

RegEx Practicals :- 1.(A) Write a Python program to check that a string contains only a certain set of characters (in this case a-z, A-Z and 0-9).

```
[1]: import re

def check_alphanumeric(s):
    return bool(re.fullmatch(r'[a-zA-Z0-9]+', s))

print(check_alphanumeric("Python123"))
print(check_alphanumeric("Hello_World"))
```

True

False

1.(B) Write a Python program that matches a string that has an a followed by one or more b's.

```
[2]: def match_ab1(s):
    return bool(re.fullmatch(r'ab+', s))

print(match_ab1("ab"))
print(match_ab1("abbbbb"))
print(match_ab1("a"))
```

True

True

False

1.(C) Write a Python program that matches a string that has an a followed by two to three 'b'.

```
[4]: def match_ab2to3(s):
    return bool(re.fullmatch(r'ab{2,3}', s))

print(match_ab2to3("abb"))
```

```
print(match_ab2to3("abbb"))
print(match_ab2to3("abbbb"))
```

True
True
False

1.(D) Write a Python program that matches a string that has an 'a' followed by anything, ending in 'b'.

```
[5]: def match_a_any_b(s):
      return bool(re.fullmatch(r'a.*b', s))

print(match_a_any_b("acb"))
print(match_a_any_b("a123b"))
print(match_a_any_b("ab"))
```

True
True
True

1.(E) Write a Python program that matches a word containing 'z'.

```
[6]: def contains_z(s):
      return bool(re.search(r'\bz\w*|\w*z\b|\w*z\w*', s))

print(contains_z("amazing"))
print(contains_z("hello"))
```

True
False

1.(F) Write a Python program where a string will start with a specific number.

```
[8]: def starts_with_number(s, num):
      pattern = f"^{num}"
      return bool(re.match(pattern, s))

print(starts_with_number("5hello", 5))
print(starts_with_number("9test", 5))
```

True
False

1.(G) Write a Python program to search the numbers (0-9) of length between 1 to 3 in a given string.

```
[9]: def search_numbers(s):
      return re.findall(r'\b\d{1,3}\b', s)
```

```
print(search_numbers("He paid 5 for 120 pens and 9999 for laptops."))
```

```
['5', '120']
```

0.1.2 Problem Statement 2:-

Write a Python program to find the substrings within a string.

Sample Text: 'Python exercises, PHP exercises, C# exercises'

Pattern: 'exercises'

```
[11]: import re

text = input("Enter text: ")
words = re.findall(r'\b\w+\b', text)
counts = {}
for word in words:
    counts[word] = counts.get(word, 0) + 1

repeated = [word for word, count in counts.items() if count > 1]

if repeated:
    print("Pattern(s):-")
    for pat in repeated:
        print(pat)
else:
    print("No pattern found.")
```

Enter text: Python exercises, PHP exercises, C# exercises

Pattern(s):-

exercises

0.1.3 Problem Statement 3:-

Extracting Email Addresses:

Problem Statement: Given a large text file containing various lines of data, extract all the email addresses present in the file.

Example: "Contact us at support@example.com or sales@example.org for more information."

It should yield ["support@example.com", "sales@example.org"].

```
[19]: import re

def extract_emails(filename):
    emails = []
    email_pattern = r'[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'

    with open(filename, 'r') as file:
        for line in file:
            found_emails = re.findall(email_pattern, line)
```

```

        emails.extend(found_emails)
    return emails

filename = input("Enter the filename: ")
result = extract_emails(filename)
print(result)

```

Enter the filename: email.txt

```
['support@example.com', 'sales@example.org', 'marketing@mycompany.net', 'hr-dept@company.co.uk', 'hr.manager@company.co.uk', 'intern123@university.edu']
```

0.1.4 Problem Statement 4:-

Validating Phone Numbers:

Problem Statement: Create a function to validate whether a given string is a valid phone number in the format (XXX) XXX-XXXX or XXX-XXX-XXXX.

Example: “Phone number (123) 456-7890 is valid, but 123-45-67890 is not.” It should yield True for the first phone number and False for the second.

```

[20]: import re

def validate_phone_number(phone):
    pattern = r'^(\(\d{3}\) |\d{3}-)\d{3}-\d{4}$'
    if re.match(pattern, phone):
        return True
    else:
        return False

print(validate_phone_number("(123) 456-7890"))
print(validate_phone_number("123-456-7890"))
print(validate_phone_number("123-45-67890"))
print(validate_phone_number("1234567890"))

```

True

True

False

False

0.1.5 Problem Statement 5:-

Finding Dates:

Problem Statement: Write a script to find all the dates in a text document, where dates are in the format DD/MM/YYYY.

Example: “Important dates are 12/05/2021 and 23/08/2022.” It should yield [“12/05/2021”, “23/08/2022”].

```

[26]: import re

```

```
def find_dates(text):
    pattern = r'\b(0[1-9]|[12][0-9]|3[01])/(0[1-9]|1[0-2])/(\d{4})\b'
    dates = re.findall(pattern, text)
    extracted_dates = [f"{day}/{month}/{year}" for day, month, year in dates]
    return extracted_dates

text = input("Enter text: ")
print(find_dates(text))
```

Enter text: Important dates are 12/05/2021 and 23/08/2022.

['12/05/2021', '23/08/2022']

0.1.6 Problem Statement 6:-

Replacing URLs:

Problem Statement: Given a string containing multiple URLs, replace all the URLs with a placeholder text [LINK].

Example: “Visit <https://example.com> or <http://example.org> for more info.”

It should yield “Visit [LINK] or [LINK] for more info.”.

```
[27]: import re

def replace_urls(text):
    url_pattern = r'https?://[^\s]+'
    replaced_text = re.sub(url_pattern, '[LINK]', text)
    return replaced_text

text = input("Enter text: ")
print(replace_urls(text))
```

Enter text: Visit <https://example.com> or <http://example.org> for more info.

Visit [LINK] or [LINK] for more info.

0.1.7 Problem Statement 7:-

Extracting Hashtags:

Problem Statement: Extract all the hashtags from a social media post.

Example: “Loving the #sunshine and #beachlife!”

It should yield [“#sunshine”, “#beachlife”].

```
[28]: import re

def extract_hashtags(text):
    pattern = r'#\w+'
    hashtags = re.findall(pattern, text)
    return hashtags
```

```
post = input("Enter post caption: ")  
print(extract_hashtags(post))
```

Enter post caption: Loving the #sunshine and #beachlife!

```
['#sunshine', '#beachlife']
```

[]: