# Regular Expressions

### **Task 1: Finding Patterns in Text**

**Objective:** Learn how to find specific patterns in a text using regular expressions.

**Instructions:**

1. Write a Python function that takes a string and a pattern as input.
2. Use the re.findall method to find all occurrences of the pattern in the string.
3. Return the list of matches.

**Example:**

import re

def find\_patterns(text, pattern):

matches = re.findall(pattern, text)

return matches

# Test the function

text = "The rain in Spain falls mainly in the plain."

pattern = r'\bin\b'

print(find\_patterns(text, pattern)) # Output: ['in', 'in', 'in']

### **Task 2: Replacing Patterns in Text**

**Objective:** Learn how to replace specific patterns in a text with another string.

**Instructions:**

1. Write a Python function that takes a string, a pattern, and a replacement string as input.
2. Use the re.sub method to replace all occurrences of the pattern with the replacement string.
3. Return the modified string.

**Example:**

import re

def replace\_patterns(text, pattern, replacement):

modified\_text = re.sub(pattern, replacement, text)

return modified\_text

# Test the function

text = "The rain in Spain falls mainly in the plain."

pattern = r'\bin\b'

replacement = 'out'

print(replace\_patterns(text, pattern, replacement)) # Output: "The raout out Spaout falls mainly out the plaout."

### **Task 3: Validating Email Addresses**

**Objective:** Learn how to validate email addresses using regular expressions.

**Instructions:**

1. Write a Python function that takes an email address as input.
2. Use the re.match method to check if the email address matches a pattern for a valid email.
3. Return True if the email is valid, otherwise return False.

**Example:**

import re

def validate\_email(email):

pattern = r'^[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$'

if re.match(pattern, email):

return True

return False

# Test the function

print(validate\_email("test@example.com")) # Output: True

print(validate\_email("test@.com")) # Output: False

### **Task 4: Extracting Phone Numbers**

**Objective:** Learn how to extract phone numbers from a text using regular expressions.

**Instructions:**

1. Write a Python function that takes a string as input.
2. Use the re.findall method to find all phone numbers in the string. Assume phone numbers are in the format (xxx) xxx-xxxx or xxx-xxx-xxxx.
3. Return the list of phone numbers.

**Example:**

import re

def extract\_phone\_numbers(text):

pattern = r'\(?\d{3}\)?[-.\s]?\d{3}[-.\s]?\d{4}'

phone\_numbers = re.findall(pattern, text)

return phone\_numbers

# Test the function

text = "Call me at (123) 456-7890 or 987-654-3210."

print(extract\_phone\_numbers(text)) # Output: ['(123) 456-7890', '987-654-3210']

### **Task 5: Removing Non-Alphanumeric Characters**

**Objective:** Learn how to remove all non-alphanumeric characters from a string using regular expressions.

**Instructions:**

1. Write a Python function that takes a string as input.
2. Use the re.sub method to replace all non-alphanumeric characters with an empty string.
3. Return the cleaned string.

**Example:**

import re

def remove\_non\_alphanumeric(text):

cleaned\_text = re.sub(r'[^a-zA-Z0-9\s]', '', text)

return cleaned\_text

# Test the function

text = "Hello, World! Welcome to NLP 101."

print(remove\_non\_alphanumeric(text)) # Output: "Hello World Welcome to NLP 101"