

## practical-2

February 3, 2025

```
[1]: # Categorize words based on their length
text = "Short words are simple, and long words are complex."
words = text.split()
categories = {"short": [], "medium": [], "long": []}
for word in words:
    if len(word) <= 4:
        categories["short"].append(word)
    elif len(word) <= 7:
        categories["medium"].append(word)
    else:
        categories["long"].append(word)
for category, words in categories.items():
    print(f"{category.capitalize()} words: {words}")
```

Short words: ['are', 'and', 'long', 'are']  
Medium words: ['Short', 'words', 'simple,', 'words']  
Long words: ['complex.']

```
[2]: # Capitalize the first and last word of each sentence in the text
text = "Python is fun. Programming is a skill."
sentences = text.split('. ')
capitalized_sentences = []
for sentence in sentences:
    words = sentence.split()
    if len(words) > 1:
        words[0] = words[0].capitalize()
        words[-1] = words[-1].capitalize()
    capitalized_sentences.append(' '.join(words))
capitalized_text = '. '.join(capitalized_sentences)
print("Text with first and last word capitalized:", capitalized_text)
```

Text with first and last word capitalized: Python is Fun. Programming is a Skill.

```
[3]: # Check if all words in the sentence are in alphabetical order
text = "Python is a great language"
words = text.split()
if words == sorted(words):
```

```

    print("The words are in alphabetical order.")
else:
    print("The words are not in alphabetical order.")

```

The words are not in alphabetical order.

```

[4]: # Create a summary by showing the first and last word of each sentence
text = "Python is great for beginners. It is also widely used in various fields.
↪"
sentences = text.split('. ')
summary = []
for sentence in sentences:
    words = sentence.split()
    if len(words) > 1:
        summary.append(f"{words[0]}...{words[-1]}")
    else:
        summary.append(words[0])
text_summary = '. '.join(summary)
print("Text summary:", text_summary)

```

Text summary: Python...beginners. It...fields.

```

[5]: # Check if a given word is a palindrome (ignoring case)
word = "madam"
if word.lower() == word.lower()[::-1]:
    print(f"'{word}' is a palindrome.")
else:
    print(f"'{word}' is not a palindrome.")

```

'madam' is a palindrome.

```

[3]: # Find all words in a text that have a specific length
text = "Python is a versatile programming language hhhjjj ."
length = 6
words_of_length = [word for word in text.split() if len(word) == length]
print(f"Words with {length} characters:", words_of_length)

```

Words with 6 characters: ['Python', 'hhhjjj']

```

[7]: # Count the number of words in each sentence of a paragraph
text = """Python is a very popular programming language. It is widely used in
↪data science, web development, and automation.
Python is easy to learn, and it has a vast collection of libraries that make
↪coding fun."""
sentences = text.split('. ')
word_counts = []

```

```

for sentence in sentences:
    words = sentence.split()
    word_count = len(words)
    word_counts.append(f"Sentence: '{sentence.strip()}' - Word Count: {word_count}")

for count in word_counts:
    print(count)

```

Sentence: 'Python is a very popular programming language' - Word Count: 7  
 Sentence: 'It is widely used in data science, web development, and automation. Python is easy to learn, and it has a vast collection of libraries that make coding fun.' - Word Count: 28

```

[8]: # Reverse each word in a sentence but keep the word order intact
text = "Python is simple to use and very powerful"
words = text.split()
reversed_words = [word[::-1] for word in words]
reversed_text = ' '.join(reversed_words)
print("Reversed words but keeping order:", reversed_text)

```

Reversed words but keeping order: nohtyP si elpmis ot esu dna yrev lufrewop

```

[9]: # Count the number of vowels and consonants in the text
text = "Python is a versatile language for developers"
vowels = "aeiouAEIOU"
vowel_count = 0
consonant_count = 0

for char in text:
    if char.isalpha():
        if char in vowels:
            vowel_count += 1
        else:
            consonant_count += 1

print(f"Vowel count: {vowel_count}")
print(f"Consonant count: {consonant_count}")

```

Vowel count: 16  
 Consonant count: 23

```

[10]: # Check if a given word exists in the text and find its position
text = "Python is a great programming language used for various applications."
word_to_find = "great"

if word_to_find in text:

```

```

    position = text.index(word_to_find)
    print(f"'{word_to_find}' found at position {position}.")
else:
    print(f"'{word_to_find}' not found in the text.")

```

'great' found at position 12.

```

[11]: # Check if a given word is present at the start or end of a sentence
text = "Python is a wonderful programming language"
word_to_check = "Python"

if text.startswith(word_to_check):
    print(f"The word '{word_to_check}' is at the start of the sentence.")
elif text.endswith(word_to_check):
    print(f"The word '{word_to_check}' is at the end of the sentence.")
else:
    print(f"The word '{word_to_check}' is neither at the start nor the end.")

```

The word 'Python' is at the start of the sentence.

```

[12]: # Extract the first and last character of each word in a sentence
text = "Python is a powerful programming language"
words = text.split()
first_last_chars = []

for word in words:
    first_last_chars.append(word[0] + word[-1])

print("First and last characters of each word:", first_last_chars)

```

First and last characters of each word: ['Pn', 'is', 'aa', 'pl', 'pg', 'le']

```

[13]: # Find and display the longest and shortest words in a sentence
text = "Python is a highly versatile programming language"
words = text.split()
longest_word = max(words, key=len)
shortest_word = min(words, key=len)

print("Longest word:", longest_word)
print("Shortest word:", shortest_word)

```

Longest word: programming

Shortest word: a

```

[14]: # Swap the first and last word in a sentence
text = "Python is a very flexible language"
words = text.split()
words[0], words[-1] = words[-1], words[0]

```

```
swapped_text = ' '.join(words)
print("Text after swapping first and last words:", swapped_text)
```

Text after swapping first and last words: language is a very flexible Python

```
[15]: # Create a simple word frequency count chart from user input
text = input("Please enter a text: ").lower()
words = text.split()
word_frequency = {}

for word in words:
    word_frequency[word] = word_frequency.get(word, 0) + 1

# Display frequency chart as text
print("Word Frequency Chart:")
for word, freq in word_frequency.items():
    print(f"{word}: {'#' * freq} ({freq})")
```

Please enter a text: hello hello ravi g

Word Frequency Chart:

hello: ## (2)

ravi: # (1)

g: # (1)

```
[16]: # Create a histogram of word lengths from user input
sentence = input("Please enter a sentence: ")
words = sentence.split()
word_lengths = [len(word) for word in words]

# Display histogram of word lengths
print("Histogram of word lengths:")
for length in sorted(set(word_lengths)):
    count = word_lengths.count(length)
    print(f"Length {length}: {'#' * count} ({count})")
```

Please enter a sentence: hello ravi g

Histogram of word lengths:

Length 1: # (1)

Length 4: # (1)

Length 5: # (1)

```
[17]: # Check the strength of a password entered by the user
password = input("Please enter a password: ")

length_check = len(password) >= 8
lower_check = any(char.islower() for char in password)
```

```

upper_check = any(char.isupper() for char in password)
digit_check = any(char.isdigit() for char in password)

if length_check and lower_check and upper_check and digit_check:
    print("The password is strong.")
else:
    print("The password is weak.")

```

Please enter a password: 2333

The password is weak.

```

[18]: # Replace every word in the sentence starting with a specific letter with
      ↪ another word
sentence = input("Please enter a sentence: ")
letter = input("Enter the letter to search for: ")
replacement_word = input("Enter the word to replace with: ")

words = sentence.split()
modified_words = [replacement_word if word.lower().startswith(letter.lower())
                  ↪ else word for word in words]

modified_sentence = ' '.join(modified_words)

print("Modified sentence:", modified_sentence)

```

Please enter a sentence: hello hi

Enter the letter to search for: hi

Enter the word to replace with: ravi

Modified sentence: hello ravi

[ ]: