KHANZADI WAZIR ALI

00272819

ASSIGNMENT 01

Python Strings Assignment

Part 1: Understanding String Literals

1. Create three different strings using each type of quotation: o Single quotes ('example') o Double quotes ("example") o Triple quotes ("""example""")

```
# Single quotes
string1 = 'Hello, World!'

print(string1)
```

```
Hello, World!
```

```
#Double quotes
string2 = "I am a Khanzadi WazirAli Memon"

print(string2)
```

```
I am a Khanzadi WazirAli Memon
```

```
#Triple quotes
string3 = '''
Once upon a time, in a quiet little town, there lived a brilliant young girl
named Khanzadi.
She was passionate about coding and loved solving complex problems.

Every day, she would sit in front of her laptop, learning new techniques and
experimenting with creative ideas.

One evening, while working on an important project, Khanzadi encountered a
stubborn bug.
No matter what she tried, the error wouldn’t go away.
She attempted everything she could think of, but nothing worked.
Frustrated and exhausted, she decided to take a break.

Taking a deep breath, she made herself a cup of coffee and paused to think.
```

```
Then, suddenly, an idea struck her—what if the problem wasn’t in the code
itself but in the way she was approaching it?

She went back to her screen, carefully analyzing each line, searching for even
the tiniest mistake.
And there it was—a small, overlooked error.
With just one quick fix, the program ran perfectly.

That day, Khanzadi learned an important lesson:
"Patience and a fresh perspective can solve even the toughest problems."

And so, her journey toward becoming a great programmer continued...
'''

print(string3)
```

```
Once upon a time, in a quiet little town, there lived a brilliant young girl
named Khanzadi.
She was passionate about coding and loved solving complex problems.

Every day, she would sit in front of her laptop, learning new techniques and
experimenting with creative ideas.

One evening, while working on an important project, Khanzadi encountered a
stubborn bug.
No matter what she tried, the error wouldn’t go away.
She attempted everything she could think of, but nothing worked.
Frustrated and exhausted, she decided to take a break.

Taking a deep breath, she made herself a cup of coffee and paused to think.
Then, suddenly, an idea struck her—what if the problem wasn’t in the code
itself but in the way she was approaching it?

She went back to her screen, carefully analyzing each line, searching for even
the tiniest mistake.
And there it was—a small, overlooked error.
With just one quick fix, the program ran perfectly.

That day, Khanzadi learned an important lesson:
"Patience and a fresh perspective can solve even the toughest problems."

And so, her journey toward becoming a great programmer continued...
```

2. Explain in your own words: What is the advantage of each type of quotation?

In Python, there are three types of quotations:

Single Quotes (' ')

Double Quotes (" ")

Triple Quotes (''' ''' or """ """)

Each type has its own advantages:

1. Single Quotes (' ')

Used for short, simple strings.

Saves one keystroke compared to double quotes.

Best for cases where the string doesn’t contain an apostrophe.

Example:

```
name = 'Khanzadi'
print(name)
```

```
Khanzadi
```

2. Double Quotes (" ")

Works the same as single quotes but allows including apostrophes (') inside the string.

Useful when the string contains a single quote.

```
sentence = "It's a beautiful day!"
print(sentence)
```

```
It's a beautiful day!
```

3. Triple Quotes (''' ''' or """ """)

Used for multi-line strings.

Often used for writing docstrings (comments inside functions).

Can contain both single and double quotes without escaping.

Example:

```
message = """Hello!
This is a multi-line string.
Python is fun!"""
print(message)
```

```
Hello!
This is a multi-line string.
```

```
Python is fun!
```

Conclusion:

Use single (') or double (") quotes for short strings.

Use double quotes (") if the string has an apostrophe.

Use triple quotes (''' or """) for multi-line strings or docstrings.

3. Write a string that contains both single and double quotes inside it.

Example: She said,

"I'm going to the store."

```
text = """She said, "I'm going to the Shopping."""""
print(text)
```

```
She said, "I'm going to the Shopping.
```

4. Create a multi-line string using triple quotes that describes your favorite hobby.

```
hobby = """I love coding and building web applications.
It allows me to be creative while solving real-world problems.
Exploring new technologies and improving my skills makes it even more exciting!

Every project I work on teaches me something new, from debugging errors to
optimizing performance.
I enjoy experimenting with different programming languages, frameworks,
and design patterns.
The ability to turn an idea into a fully functional application is truly
rewarding.

Coding also helps me stay updated with the latest industry trends.
Whether it's working on frontend designs or backend logic, every aspect of
development keeps me engaged.
I love collaborating with other developers, sharing knowledge, and
contributing to open-source projects.

For me, programming is more than just a hobby—it's a passion that fuels my
curiosity and creativity every day!"""

print(hobby)
```

```
I love coding and building web applications.
It allows me to be creative while solving real-world problems.
```

```
Exploring new technologies and improving my skills makes it even more exciting!

Every project I work on teaches me something new, from debugging errors to
optimizing performance.
I enjoy experimenting with different programming languages, frameworks,
and design patterns.
The ability to turn an idea into a fully functional application is truly
rewarding.

Coding also helps me stay updated with the latest industry trends.
Whether it's working on frontend designs or backend logic, every aspect of
development keeps me engaged.
I love collaborating with other developers, sharing knowledge, and
contributing to open-source projects.

For me, programming is more than just a hobby—it's a passion that fuels my
curiosity and creativity every day!
```

Part 2: String Methods Practice

1. Create a variable full_name with your full name (first and last name). Then write code

to:

o Print your name in all uppercase letters

o Print your name in all lowercase letters

o Print your name with the first letter of each name capitalized

```python
# Create a variable with your full name
full_name = "Khanzadi WazirAli Memon"

# Print the name in all uppercase letters
print(full_name.upper())

# Print the name in all lowercase letters
print(full_name.lower())

print(full_name.title())
```

```
KHANZADI WAZIRALI MEMON
khanzadi wazirali memon
Khanzadi Wazirali Memon
```

2. Create a variable messy_text = " Python programming is fun! " Then write code

to:

o Remove all the extra spaces at the beginning and end o Replace "fun" with "amazing"

o Count how many times the letter 'i' appears

```
# Create a variable with extra spaces
message = " Python programming is fun! "

# Remove extra spaces at the beginning and end
text = message.strip()
print(text)

# Replace "fun" with "amazing"
updated = text.replace("fun", "amazing")
print(updated)

# Count how many times the letter 'i' appears
count = text.count('i')

print(count)
```

```
Python programming is fun!
Python programming is amazing!
2
```

3. Create a variable sentence = "The quick brown fox jumps over the lazy dog"

Then write code to:

o Split this sentence into a list of words

o Join the words back together with dashes between them

o Check if the sentence starts with "The"

o Find the position of the word "fox"

```
# Create the sentence variable
sentence = "The quick brown fox jumps over the lazy dog"

# 1. Split the sentence into a list of words
list = sentence.split()
print(list)

# 2. Join the words back together with dashes between them
join= "-".join(list)
print(join)

# 3. Check if the sentence starts with "The"
```

```
starts = sentence.startswith("The")
print(starts)

# 4. Find the position of the word "fox"
position = sentence.find("fox")

print(position)
```

```
['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']
The-quick-brown-fox-jumps-over-the-lazy-dog
True
16
```

Part 3: F-Strings

1. Create variables for:

o Your name

o Your age

o Your favorite programming language

```
# Creating variables
name = "Khanzadi Wazir Ali Memon"
age = 21
favorite_language = "Python"

print("Name:", name)
print("Age:", age)
print("Favorite Programming Language:", favorite_language)
```

```
Name: Evrenshah Khan
Age: 25
Favorite Programming Language: Python
```

Then use f-strings to create these sentences:

o "My name is {your_name} and I am {your_age} years old."

o "I enjoy programming in {language}, it's my favorite!"

o Create a math expression inside an f-string: "In 5 years, I will be {age + 5} years old."

```
# Creating variables
name = "Khanzadi Wazir Ali Memon"
age = 21
favorite_language = "Python"
```

```
# Using f-strings to create sentences
sentence1 = f"My name is {name} and I am {age} years old."
sentence2 = f"I enjoy programming in {favorite_language}, it's my favorite!"
sentence3 = f"In 5 years, I will be {age + 5} years old."

print(sentence1)
print(sentence2)
print(sentence3)
```

```
My name is Khanzadi Wazir Ali Memon and I am 21 years old.
I enjoy programming in Python, it's my favorite!
In 5 years, I will be 26 years old.
```

Part 4: Combining String Methods with F-Strings

Create a program that:

1. Asks for user input about their first name, last name, and birth year
2. Uses string methods to properly capitalize their name
3. Uses f-strings to create a profile message: "Profile: {First Last}, Age: {calculated age}, Username: {first initial + last name + birth year}"

```
# Ask for user input
first_name = input("Enter your first name: ").strip().capitalize()
last_name = input("Enter your last name: ").strip().capitalize()
birth_year = input("Enter your birth year: ").strip()

# Calculate age (assuming the current year is 2025)
current_year = 2025
age = current_year - int(birth_year)

# Generate a username (first initial + last name + birth year)
username = f"{first_name[0].lower()}{last_name.lower()}{birth_year}"

# Create the profile message using f-strings
profile_message = f"Profile: {first_name} {last_name}, Age: {age}, Username: {username}"

print(profile_message)
```

```
Enter your first name: Khanzadi
Enter your last name: Wazir Ali
Enter your birth year: 2004
Profile: Khanzadi Wazir ali, Age: 21, Username: kwazir ali2004
```