Python Strings Assignment (Basic to Advanced)

Date: May 9, 2025 (Friday)
Topic: String Manipulation in Python

Instructions

This assignment progresses from basic string operations to more advanced string manipulation techniques. Complete all sections, showing your work for each problem. Comment your code to explain your approach.

Section 1: Creating & Accessing Strings (Basic)

1. Create three different strings using different quotation methods (single quotes, double quotes, and triple quotes).
2. Create a string variable containing your full name and write code to:
   - Print the first character
   - Print the last character
   - Print the length of the string
3. Given the string `s = "Python Programming"`:
   - Access and print the letter 'P' in "Programming" using positive indexing
   - Access and print the letter 'P' in "Programming" using negative indexing

---

Section 2: Slicing & Extended Slicing (Basic to Intermediate)

1. Using the string `s = "Python Programming"`:
   - Extract and print "Python"
   - Extract and print "Programming"
   - Extract and print "gram"
2. Given `s = "0123456789"`:
   - Extract every even-indexed character
   - Extract every odd-indexed character
   - Reverse the string using slicing
3. Create a function that takes a string and returns a "rotated" version where the first character is moved to the end. For example, "Python" becomes "ythonP".

Section 3: Immutability & Interning (Intermediate)

1. Demonstrate string immutability with an example. Try to change a character in a string and explain what happens.
2. Write code to check if two strings with the same content point to the same memory location. Test this with:
   - Two string literals with the same content
   - Two string variables assigned the same string literal
   - Two string variables created using string operations that result in the same content
3. Create a function that efficiently concatenates a large number of strings. Compare your approach with using the `+` operator in a loop.

Section 4: String Methods (Intermediate)

1. Given the string `s = " Python is Amazing! "`:
   - Remove leading and trailing whitespace
   - Convert to all uppercase
   - Convert to all lowercase
   - Replace "Amazing" with "Awesome"
2. Write a function that counts the occurrences of each character in a string and returns a dictionary with the results.
3. Create a function that checks if a string is a palindrome (reads the same backward as forward), ignoring case, spaces, and punctuation.
4. Given `s = "python,java,c++,javascript,ruby"`:
   - Split the string into a list of programming languages
   - Join the list with a different separator (e.g., " | ")

Section 5: Escape Sequences & Raw Strings (Intermediate)

1. Create a string that includes tab characters, newlines, and quotes using escape sequences.
2. Explain when you would use raw strings in Python and provide an example with file paths.
3. Create a function that formats a multi-line address with proper newlines and indentation using escape sequences.
4. Write a program that takes a Windows file path (with backslashes) as input and correctly processes it using raw strings.

Section 6: Unicode & Multiline Strings (Intermediate to Advanced)

1. Create strings containing characters from at least three different writing systems (e.g., Latin, Cyrillic, Arabic, CJK).
2. Write a function that counts the number of characters in a Unicode string, taking into account combining characters and surrogate pairs.
3. Create a multi-line string containing a short poem or quote with proper formatting.
4. Create a function that takes a multi-line string and returns the line with the most characters.

Section 7: String Formatting (Advanced)

1. Format the same data using all three main formatting methods (f-strings, `.format()`, and `%` formatting):
   - Format a floating-point number to two decimal places
   - Format an integer with leading zeros
   - Format a string with fixed width and alignment
2. Given a dictionary with student information (name, ID, grades), create formatted output using all three formatting methods.
3. Create a function that generates a table of data with proper alignment using f-strings.
4. Write a program that formats dates and times in different regional formats using string formatting.

Submission Guidelines

- Submit your solutions as a single Python file with clear sections to sudhanshu@euron.one mail id

Good luck!