Tutorial II - Control Structures

Programming with Python

Introduction

Just like in the previous tutorial, you will likely find solutions to most exercises online. However, I still strongly encourage you to work on these exercises independently without searching for answers.

Understanding someone else's solution is very different from developing your own. Use the lecture notes and try to solve the exercises on your own. This approach will significantly enhance your learning and problem-solving skills.

Remember, the goal is not just to complete the exercises, but to understand the concepts and improve your programming abilities. If you encounter difficulties, review the lecture materials, experiment with different approaches, and don't hesitate to ask for clarification during class discussions. Don't worry, I won't repeat this section again and again.

Slicing secret messages

In this exercise, we'll practice string manipulation and slicing. You'll work with a secret message encoded using various string operations and learn how to apply some new methods like `title()`, `replace()` and `count()`. By following a series of steps, you'll gradually decode the message.

String Manipulation

First, here are some commonly used string methods we could use to decode the message:

- `upper()`: Converts all characters in the string to uppercase
- `lower()`: Converts all characters in the string to lowercase
- `title()`: Converts first character of each word to uppercase
- `strip()`: Removes leading and trailing whitespace
- `replace()`: Replaces a substring with another substring
- `find()`: Finds first substring and returns its index
- `count()`: Counts the number of occurrences of a substring

> **Tip**
>
> You can use the `dir()` function to list all methods and attributes of an object. Here, we could see that `upper()`, `lower()`, `title()`, `strip()`, `replace()`, `find()`, and `count()` are all methods of the string object `str`.

Let's begin with the encoded message and work through each decoding step:

```
# Decode a secret message by following a series of instructions.
# Each instruction requires you to use different operations and methods.
# The encoded message is:
secret_message = "!nohtyyP gnidoc nrael ot nuf si tI"

# a) TODO: Reverse the string
# Hint: You can use slicing to reverse a string
```

```
# Your code here

# b) TODO: Remove the exclamation mark at the beginning
# Hint: Use string slicing
# Your code here

# c) TODO: Replace 'yP' with 'Py'
# Hint: Use the replace() method
# Your code here

# d) TODO: Convert the string to title case
# -> Capitalize the first letter of each word
# Hint: Use the title() method
# Your code here

# e) TODO: Add spaces around the word 'coding'
# Hint: Use the replace() method
# Your code here

# f) TODO: Count how many times the letter 'n' appears in the decoded message
# Hint: Use the count() method
# Your code here
```

> **Tip**
>
> Use the `help()` function to get more information about a method. For example, typing `help(str.replace)` in the shell will show the documentation for the `replace()` method. To exit the documentation, press `q`.