Loops, Conditions, and GDP Growth Calculation

1. Introduction

In this tutorial, we will learn how to use loops and conditions in Python by analyzing GDP data. The goal is to calculate the year-over-year GDP growth rate for specific countries.

2. Basic Syntax of Loops and Conditions

2.1 Loops

Loops allow us to execute a block of code repeatedly. In Python, the two most common loops are:

1. **For Loop**: Used to iterate over a sequence (like a list or a range of numbers).

```
for item in sequence:
    # Code block
```

2. **While Loop**: Repeats as long as a condition is `True`.

```
while condition:
    # Code block
```

2.2 Conditions

Conditions allow us to execute code based on specific criteria.

1. **If Statement**: Executes a block of code if the condition is true.

```
if condition:
    # Code block
```

2. **Elif and Else Statements**: Handle additional conditions or execute code when no conditions are met.

```
if condition1:
    # Code block 1
elif condition2:
    # Code block 2
else:
    # Default code block
```

3. Example Dataset and Problem Statement

We are provided with a dataset that includes annual GDP figures for various countries. Each row represents a year, and we aim to calculate the GDP growth rate for consecutive years.

GDP Growth Formula:

To calculate GDP growth:

[\text{Growth Rate} = \frac{\text{GDP}_{\text{current year}} - \text{GDP}_{\text{previous year}}}{\text{GDP}_{\text{previous year}}} \times 100]

4. Exercise: Calculate GDP Growth

Problem:

Write a Python program to:

1. Extract GDP data for a specific country.
2. Use a loop to calculate the year-over-year GDP growth.
3. Use conditions to handle missing or invalid data.

Code Example:

Here's a sample solution:

```
# Sample GDP data for demonstration
# Columns: Year, Country, GDP

data = [
    {"Year": 2020, "Country": "Norway", "GDP": 1000},
    {"Year": 2021, "Country": "Norway", "GDP": 1100},
    {"Year": 2022, "Country": "Norway", "GDP": None},  # Missing value example
    {"Year": 2020, "Country": "Portugal", "GDP": 800},
    {"Year": 2021, "Country": "Portugal", "GDP": 850},
]

# Filter data for a specific country
country = "Norway"
country_data = [row for row in data if row["Country"] == country]

# Sort data by year (if not already sorted)
country_data.sort(key=lambda x: x["Year"])

# Calculate GDP growth
print(f"Year-over-Year GDP Growth for {country}:")
for i in range(1, len(country_data)):
    prev_gdp = country_data[i - 1]["GDP"]
    curr_gdp = country_data[i]["GDP"]

    # Check for missing values
    if prev_gdp is None or curr_gdp is None:
        print(f"Year {country_data[i]['Year']}: Data unavailable")
```

```
        continue

    # Calculate growth rate
    growth_rate = ((curr_gdp - prev_gdp) / prev_gdp) * 100
    print(f"Year {country_data[i]['Year']}: {growth_rate:.2f}%")
```

5. Practice Task

Using the provided GDP dataset:

1. Write a function `calculate_growth(data, country)` that calculates and prints the year-over-year GDP growth for a given country.
2. Include error handling for missing or invalid data.
3. Test the function with at least two different countries.

6. Solution

The solution code is shown in the example above. Adapt and expand it to explore other countries and handle more data.

7. Summary

In this tutorial, you learned:

- How to use loops to iterate over data.
- How to use conditions to handle special cases.
- How to calculate GDP growth rates using Python.