

# Functions and GDP Growth Calculation

## 1. Introduction to Functions

Functions are reusable blocks of code that perform a specific task. They help in organizing your code, making it more readable, and reducing repetition.

## 2. Basic Syntax of Functions

The basic structure of a Python function is:

```
def function_name(parameters):  
    # Code block  
    return value
```

### Key Components:

1. **def**: Used to define a function.
2. **function\_name**: The name of the function (should be descriptive).
3. **parameters**: Inputs to the function (optional).
4. **return**: Used to output the result (optional).

## 3. Why Use Functions?

- **Code Reusability**: Write once, use multiple times.
- **Modularity**: Divide a program into smaller, manageable parts.
- **Improved Readability**: Clear and organized code.

## 4. Example Dataset and Problem Statement

We are provided with a dataset of annual GDP values for various countries. The task is to:

1. Extract GDP data for a specific country.
2. Calculate year-over-year GDP growth using a function.
3. Handle missing or invalid data gracefully.

### GDP Growth Formula:

To calculate GDP growth:

$$\left[ \frac{\text{GDP}_{\text{current year}} - \text{GDP}_{\text{previous year}}}{\text{GDP}_{\text{previous year}}} \times 100 \right]$$

## 5. Writing a Function for GDP Growth Calculation

Below is an example of a Python function to calculate GDP growth.

```

def calculate_growth(data, country):
    """
    Calculates year-over-year GDP growth for a given country.

    Args:
        data (list): List of dictionaries containing GDP data.
        country (str): The country to calculate growth for.

    Returns:
        None
    """
    # Filter data for the specific country
    country_data = [row for row in data if row["Country"] == country]

    # Sort data by year
    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}%")

# Example Data

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},
]

# Test the function
calculate_growth(data, "Norway")
calculate_growth(data, "Portugal")

```

## 6. Practice Task

1. Modify the `calculate_growth` function to return a list of growth rates instead of printing them.
2. Write another function `average_growth(data, country)` to calculate the average GDP growth for a country.

## 7. Solution Example

### Function to Return Growth Rates:

```
def calculate_growth(data, country):  
    """  
    Calculates year-over-year GDP growth for a given country.  
  
    Args:  
        data (list): List of dictionaries containing GDP data.  
        country (str): The country to calculate growth for.  
  
    Returns:  
        list: List of year-over-year GDP growth rates.  
    """  
    # Filter data for the specific country  
    country_data = [row for row in data if row["Country"] == country]  
  
    # Sort data by year  
    country_data.sort(key=lambda x: x["Year"])  
  
    # Calculate GDP growth  
    growth_rates = []  
    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:  
            growth_rates.append(None)  
            continue  
  
        # Calculate growth rate  
        growth_rate = ((curr_gdp - prev_gdp) / prev_gdp) * 100  
        growth_rates.append(growth_rate)  
  
    return growth_rates
```

## 8. Summary

In this chapter, you learned:

- The syntax and purpose of functions in Python.
- How to define and use a function for GDP growth calculations.
- How to handle missing data gracefully within a function.