Tutorials for Week 03-02

Functions:

- Python functions are blocks of reusable code that perform a specific task.
- They are defined using the `def` keyword and can accept parameters and return data.
- Functions can provide default values for parameters, allow for variable numbers of arguments (*args), and support keyword arguments (**kwargs).

File Handling:

- File handling in Python is done through the built-in `open()` function, which returns a file object.
- Reading from a file is commonly done with the `read()`, `readline()`, or `readlines()` methods.
- Writing to a file is done using the `write()` or `writelines()` methods, and it's important to manage file resources properly, often using the `with` statement to automatically handle file closing.

Exceptions:

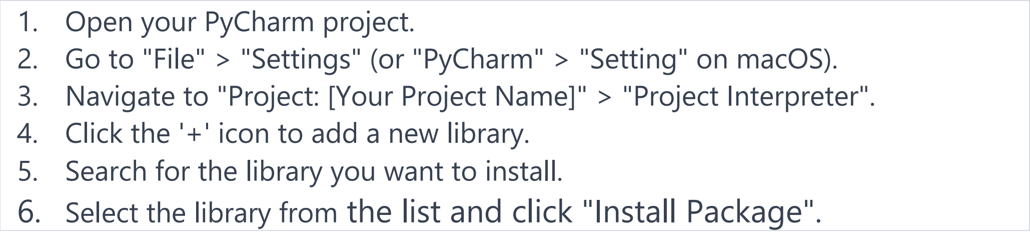
- Exceptions are raised when an error occurs, interrupting the normal flow of a program.
- The `try` block is used to catch exceptions, followed by one or more `except` blocks to handle specific error types.
- The `finally` block is executed no matter what, used for clean-up actions that must be executed under all circumstances.

Introduction to Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Here's a brief outline for a tutorial on Matplotlib:

Introduction to Matplotlib:

- Explain the purpose of data visualization and the role of Matplotlib.
- Install Matplotlib using `pip install matplotlib`.
- For installing a Library in PyCharm, use the following steps:



1. Open your PyCharm project.
2. Go to "File" > "Settings" (or "PyCharm" > "Setting" on macOS).
3. Navigate to "Project: [Your Project Name]" > "Project Interpreter".
4. Click the '+' icon to add a new library.
5. Search for the library you want to install.
6. Select the library from the list and click "Install Package".

Basic Plotting:

- Introduce the `pyplot` module.
- Demonstrate creating simple line plots, histograms, scatter plots, and bar charts.
- Show how to label axes, add a title, and customize colors and styles.

Figure and Subplots:

- Discuss the `Figure` object and how to create subplots using `plt.subplots()`.
- Customize the size of the figure with the `figsize` parameter.

Customization Techniques:

- Explain the customization of plots with linewidths, markers, and linestyle.
- Show how to use the `rcParams` to set default styles.

Working with Text and Annotations:

- Demonstrate how to add text inside the plot using `plt.text()`.
- Show how to annotate elements with `plt.annotate()`.

Advanced Plot Types:

- Introduce more complex plots like boxplots, violin plots, and pie charts.
- Discuss the use of `pandas` with Matplotlib for plotting directly from DataFrames.

Saving Figures:

- Show how to save figures to files using `plt.savefig()`.

Interactive Plots:

- Briefly touch on making interactive plots with `mpl_toolkits`.