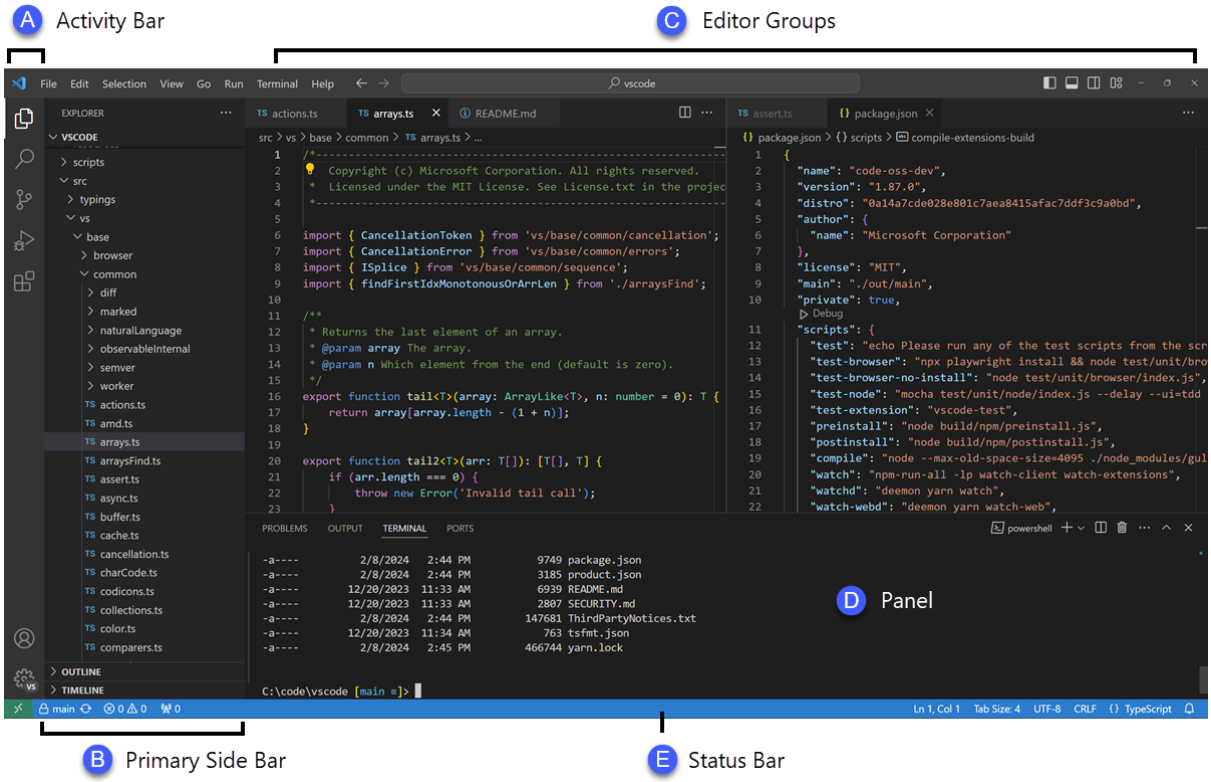
COMP 627 Neural Networks and Applications – Python Tutorial 1

1. Introduction to Python

Installing Software:

Visual Studio Code (VS Code) is the main development environment for this series of tutorials. Use the following link which would guide you to install VS Code on your computer.

User Interface:



Create a directory from which to read and write python files. It is recommended that you create a new directory in your personal computer, keep all python related files in this directory. Please create one for each laboratory session.

Installing Visual Studio Code (VS Code)

1. **Download VS Code**:
   - Go to the official VS Code website. (https://code.visualstudio.com)
   - Click on the download button suitable for your operating system (Windows, macOS, or Linux).
2. **Install VS Code**:
   - Once the download is complete, run the installer.
   - Follow the installation instructions provided by the installer.
   - Launch VS Code after installation completes.

Setting Up Python in VS Code

3. **Install Python**:

- If Python isn't installed on your system, download it from the official Python website (https://www.python.org/downloads) and follow the installation instructions.

4. **Install the Python Extension for VS Code**:
   - Open VS Code.
   - Go to the Extensions view by clicking on the square icon on the left sidebar or by pressing `Ctrl+Shift+X`.
   - Search for "Python" in the Extensions Marketplace.
   - Click on "Install" for the extension provided by Microsoft.
5. **Select Python Interpreter**:
   - Open a new terminal in VS Code by pressing `Ctrl+` (backtick) or navigating to `View -> Terminal`.
   - Type `python --version` to check if Python is installed correctly and note down the version.
   - In VS Code, press `Ctrl+Shift+P` (Windows/Linux) or `Cmd+Shift+P` (macOS) to open the command palette.
   - Type and select "Python: Select Interpreter".
   - Choose the interpreter that matches your installed Python version (e.g., `Python <version>`, like `Python 3.9.6 64-bit`).

Create a directory from which to read and write python files. It is recommended that you create a new directory in your personal computer, keep all python related files in this directory. Please create one for each laboratory session.

2. Programming Basics:

Please refer to the lecture materials in COMP 636.

2.1 Writing and running code in visual studio code

- Open Visual Studio Code
  - Launch Visual Studio Code on your computer.
- Create a New File
  - Click on the File menu at the top-left corner of the window.
  - Select New File or use the shortcut Ctrl+N (Windows/Linux) or Cmd+N (macOS).
  - Select Python as file type
  - This will open a blank file in the editor area.
- Write Your Code:
  - In the new file, write your code.
- Save the File
  - Click on the **File** menu again.
  - Select **Save As...** or use the shortcut Ctrl+Shift+S (Windows/Linux) or Cmd+Shift+S (macOS).
  - In the dialog that appears, navigate to the directory where you want to save your file.
  - Enter a name for your file with a suitable file extension (e.g., .py for Python files).
  - Click **Save**.

- **Run Your Code**:
  - Use the **Run** command to run the file.
- **View Output**:
  - The output of your code will be displayed in the terminal panel below.

2.2 Special Operations

Execute these in the command window:
is used for comments

```
> 7+3    # Addition
> 8-2    # Subtraction
> 3*2    # Multiplication
> 21/3   # Division
> 5%2    # Modulus
```

Create a new python file and save it as Exercise1.py
Try following code with logical operators.

```
x = 3
print(x == 3)                    # relational operator EQUAL
print(x != 3)                    # relational operator NOT EQUAL
print((x >= 3) and (x < 5))      # logical Operator AND
print((x >= 3) or (x < 5))       # logical Operator OR
print(not(x <2))                 # logical Operator NOT
```

2.3 Variable Assignment

```
x=3                  # a number
y = 'hello world'    # a text
```

2.4 Arrays, Vectors and Matrices

One-dimensional array: numpy (often abbreviated as np) is a powerful library for numerical operations in Python. To utilize it,

First, ensure NumPy is installed:

1. Open Visual Studio Code.
2. Open the terminal (`Terminal > New Terminal`).
3. Type the following command and press Enter: `pip install numpy`

This command installs NumPy, enabling you to create and manipulate arrays, vectors, and matrices efficiently in Python.

```
# Load library
import numpy as np

# Create a vector as a row
vector_row = np.array([1, 2, 3])

# Create a vector as a column
vector_column = np.array([[1], [2], [3]])

# Print the vectors
print("Vector as a row:")
print(vector_row)

print("\nVector as a column:")
print(vector_column)
```

Expected Result:

Vector as a row:
[1 2 3]
Vector as a column:
[[1]
 [2]
 [3]]

Two-dimensional array:

```
# Load library
import numpy as np

# Create a matrix
matrix = np.array([[1, 2],
                   [1, 2],
                   [1, 2]])

# Print the matrix
print("Matrix:")
print(matrix)
```

Expected Result:

Matrix:
[[1 2]
 [1 2]
 [1 2]]

Selecting one or more elements in a vector or matrix:

```
# Load library
import numpy as np

# Create row vector
vector = np.array([1, 2, 3, 4, 5, 6])

# Create matrix
matrix = np.array([[1, 2, 3, 4],
                   [5, 6, 7, 8],
                   [9, 10, 11, 12]])

# Select third element of vector
third_element = vector[2]

# Print vector, matrix, and selected element
print("Vector:")
print(vector)

print("\nMatrix:")
print(matrix)

print("\nThird element of vector:", third_element)
```

Expected Result:

Vector:
[1 2 3 4 5 6]

Matrix:
[[1 2 3 4]
 [5 6 7 8]
 [9 10 11 12]]

Third element of vector: 3

2.5 Important syntax:

1. Select all elements of a vector
   Syntax → vector[:]
   Output → array([1, 2, 3, 4, 5, 6])

2. Select everything up to and including the third element
   Syntax → vector[:3]
   Output → array([1, 2, 3])

3. Select everything after the third element
   Syntax → vector[3:]
   Output → array([4, 5, 6])

4. Select the last element
   Syntax → vector[-1]
   Output → 6

5. Select the first two rows and all columns of a matrix
   Syntax → matrix[:2,:]
   Output → array([[1, 2, 3, 4],
   [5, 6, 7, 8]])

6. Select all rows and the second column
   Syntax → matrix[:,1:2]
   Output → array([[2],
   [6],
   [10]])

7. View number of rows and columns
   Syntax → matrix.shape
   Output → (3, 4)

8. View number of elements (rows * columns)
   Syntax → matrix.size
   Output → 12

9. View number of dimensions
   Syntax → matrix.ndim
   Output → 2

10. Finding the Maximum and Minimum Values
    Syntax → np.max(matrix)
    Output → 12

    Syntax → np.min(matrix)
    Output → 1

11. Return mean
    np.mean(matrix)

12. Return variance
    np.var(matrix)

13. Return standard deviation
    np.std(matrix)

2.6 Pandas DataFrames

A Pandas DataFrame is a 2 dimensional data structure, like a 2 dimensional array, or a table with rows and columns.

To work with Pandas DataFrames, first, ensure NumPy and Pandas are installed:

1. Open Visual Studio Code.
2. Open the terminal (`Terminal > New Terminal`).
3. Type the following commands one by one and press Enter

```
pip install pandas
```

These commands install both NumPy (used for numerical operations) and Pandas (used for data manipulation and analysis), allowing you to work effectively with DataFrames and other data structures in Python.

```
import pandas as pd

data = {
  "calories": [420, 380, 390],
  "duration": [50, 40, 45]
}

#load data into a DataFrame object:
df = pd.DataFrame(data)

print(df)
```

Expected Result:

	calories	duration
0	420	50
1	380	40
2	390	45

Locate row:

Syntax → print(df.loc[0])

Output →
```
calories   420
duration    50
Name: 0, dtype: int64
```

Return row 0 and 1:

Syntax → print(df.loc[[0, 1]])

Output →

	calories	duration
0	420	50
1	380	40

Locate *'calories'* column:

Syntax → print(df[['calories']]) OR print(df.iloc[:, [0]])

Output →

	calories
0	420
1	380
2	390

Return column 0 and 1:

Syntax → print(df[['calories','duration']]) OR print(df.iloc[:, [0,1]])

Output →

	calories	duration
0	420	50
1	380	40
2	390	45

Concatenate two dataframes into one:

Syntax → data = pd.concat([df, df1], axis=1, join='inner')

Adding a column to a dataframe:

Syntax → df.insert(2, "Age", [21, 23, 24], True)

Load Files Into a DataFrame:

```
import pandas as pd

df = pd.read_csv('data.csv')

print(df)
```

Expected Result:

	Mathematics	Physics	Biology
0	69.0	63	33.0
1	52.0	56	NaN
2	NaN	58	38.0
3	29.0	28	56.0
4	96.0	100	88.0
5	32.0	95	64.0
6	24.0	23	54.0
7	77.0	53	79.0
8	31.0	76	45.0
9	74.0	48	100.0
10	75.0	60	29.0
11	68.0	54	56.0
12	21.0	64	94.0
13	32.0	95	64.0
14	77.0	53	79.0

COMP 627 Neural Networks and Applications – Python Tutorial 1

Read a csv file with headers:

```
import pandas as pd

df = pd.read_csv('data.csv', header=0)

print(df)
```

Expected Result:

	Mathematics	Physics	Biology
0	69.0	63	33.0
1	52.0	56	NaN
2	NaN	58	38.0
3	29.0	28	56.0
4	96.0	100	88.0
5	32.0	95	64.0
6	24.0	23	54.0
7	77.0	53	79.0
8	31.0	76	45.0
9	74.0	48	100.0
10	75.0	60	29.0
11	68.0	54	56.0
12	21.0	64	94.0
13	32.0	95	64.0
14	77.0	53	79.0

Viewing the Data:

```
import pandas as pd

df = pd.read_csv('data.csv', header=0)

print(df.head(10))  # prints first 10 rows of the dataframe
```

Expected Result:

	Mathematics	Physics	Biology
0	69.0	63	33.0
1	52.0	56	NaN
2	NaN	58	38.0
3	29.0	28	56.0
4	96.0	100	88.0
5	32.0	95	64.0
6	24.0	23	54.0
7	77.0	53	79.0
8	31.0	76	45.0
9	74.0	48	100.0

Finding Relationships in data:
This returns a table with a lot of numbers that represents how well the relationship is between two columns.

```
import pandas as pd

df = pd.read_csv('data.csv', header=0)

print(df.corr())    # prints correlation between columns
```

Expected Result:

	Mathematics	Physics	Biology
Mathematics	1.000000	0.126403	0.149994
Physics	0.126403	1.000000	0.118230
Biology	0.149994	0.118230	1.000000

2.7 Cleaning data

Remove rows that contain empty cells:
Syntax → df.dropna(inplace = True)

Replace empty cells with the number 130:
Syntax → df.fillna(130, inplace = True)

Remove all duplicates:
Syntax → df.drop_duplicates(inplace = True)

Delete a column:
Syntax → df.drop(columns=['Mathematics', 'Biology'], inplace = True)
Or
df.drop(['Mathematics', 'Biology'], axis=1, inplace = True)

COMP 627 Neural Networks and Applications – Python Tutorial 1

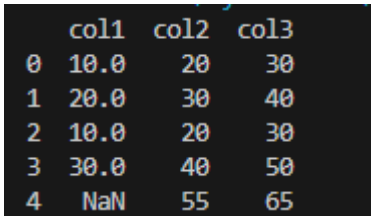
Laboratory Exercise:

Follow the following steps to develop a python code. Refer to the expected output at each step if specified.

Step 1: Copy the given CSV file 'Test1.csv' into the working folder and create a new python file to load the CSV file using Pandas DataFrame.

Step 2: Print the first 5 rows of the dataset.

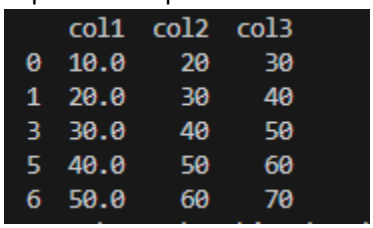
Expected output:



	col1	col2	col3
0	10.0	20	30
1	20.0	30	40
2	10.0	20	30
3	30.0	40	50
4	NaN	55	65

Step 3: Clean the dataset by removing empty cells and duplicates. Print the first 5 rows of the dataset.

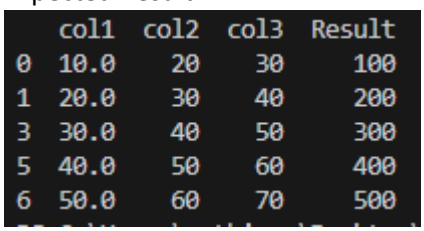
Expected Output:



	col1	col2	col3
0	10.0	20	30
1	20.0	30	40
3	30.0	40	50
5	40.0	50	60
6	50.0	60	70

Step 4: Add a new column "Result" with the values 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, and 1200. . Print the first 5 rows of the dataset.

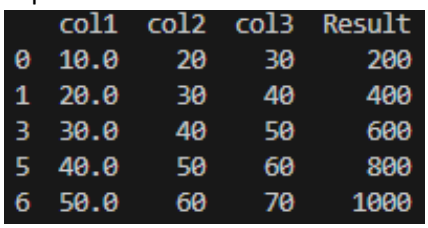
Expected Result:



	col1	col2	col3	Result
0	10.0	20	30	100
1	20.0	30	40	200
3	30.0	40	50	300
5	40.0	50	60	400
6	50.0	60	70	500

Step 5: Multiply the values in the 'Result' column by 2.

Expected Result:



	col1	col2	col3	Result
0	10.0	20	30	200
1	20.0	30	40	400
3	30.0	40	50	600
5	40.0	50	60	800
6	50.0	60	70	1000