BHARATHIDASAN UNIVERSITY

Tiruchirappalli- 620024

Tamil Nadu, India

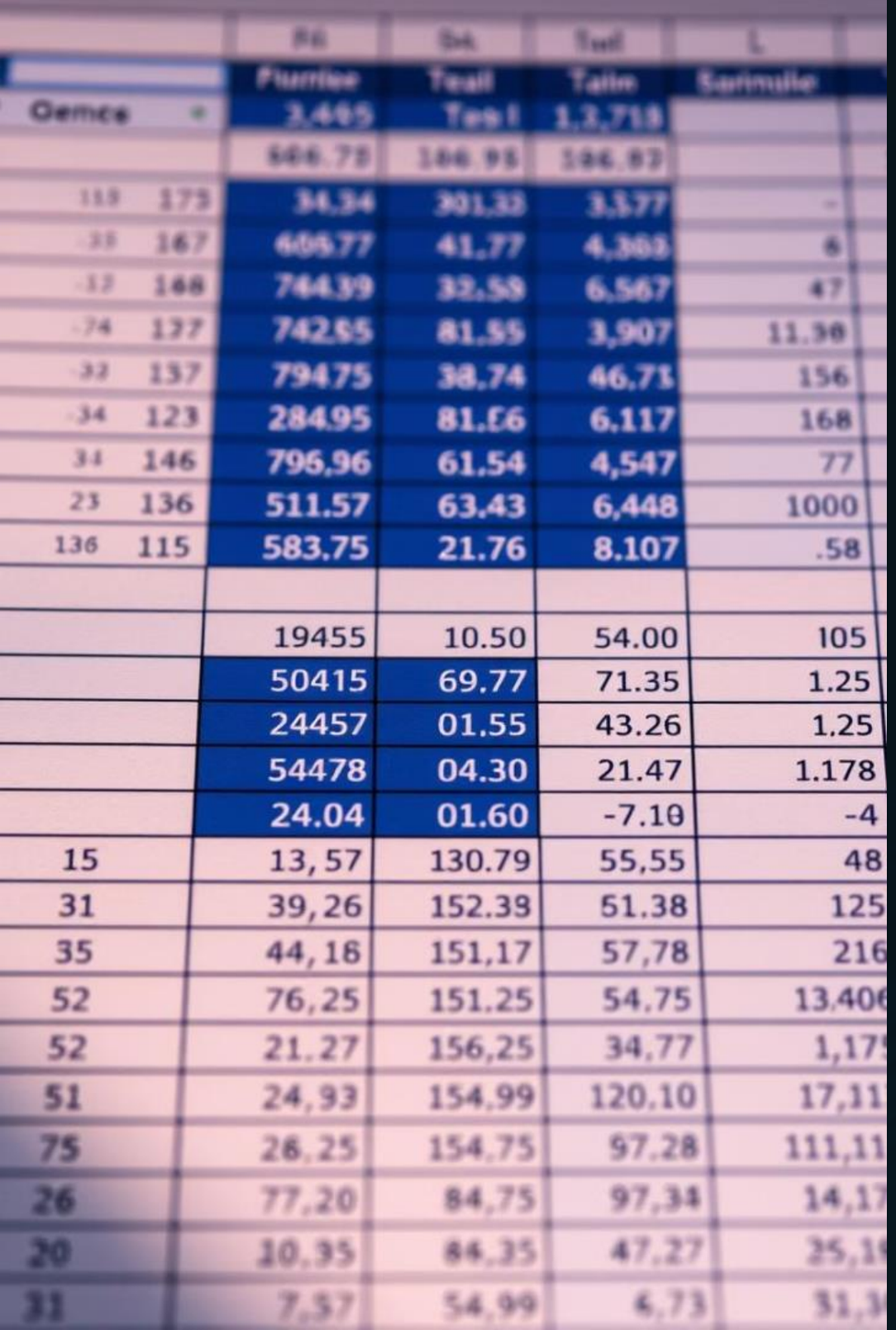
Programme: MBA (Financial Management)

Course Title : R and Python for Finance (NSE)

Course Code : FMEC2/24

Unit V : Financial Analytics and Development

Dr. M. KAMAL
Visiting Faculty
Asst. Professor of Computer Applications
Jamal Mohamed College
Tiruchirappalli-20



Spreadsheet Interaction in Python

This presentation explores the power of Python for interacting with spreadsheets, covering key libraries and practical examples.

by Dr. M. KAMAL
Asst. Prof. Computer Applications (Visiting Faculty)
Jamal Mohamed College, Tiruchrappalli-20

Key Libraries for Spreadsheet Interaction

openpyxl

Works with Excel files (.xlsx and .xlsm). Supports reading, writing, and modifying Excel workbooks.

pandas

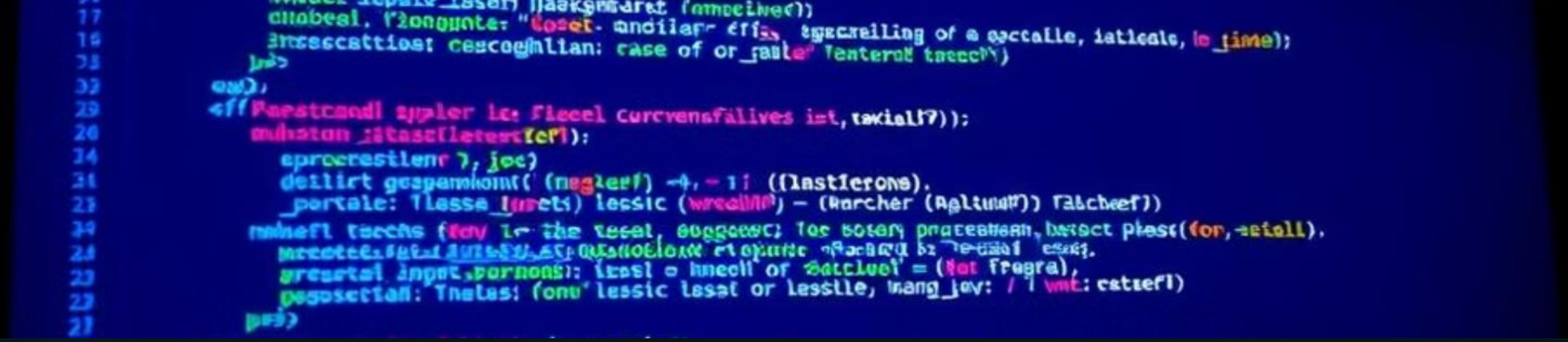
Ideal for working with tabular data in spreadsheets. Provides easy reading/writing of Excel files using read_excel and to_excel.

xlrd and xlwt

Used for older Excel formats (.xls). Deprecated in many cases.

Reading Excel Files with Pandas

```
import pandas as pd

# Read the spreadsheet into a DataFrame
data = pd.read_excel('example.xlsx')

# Display the first few rows
print(data.head())

# Display the last few rows
print(data.tail())
```

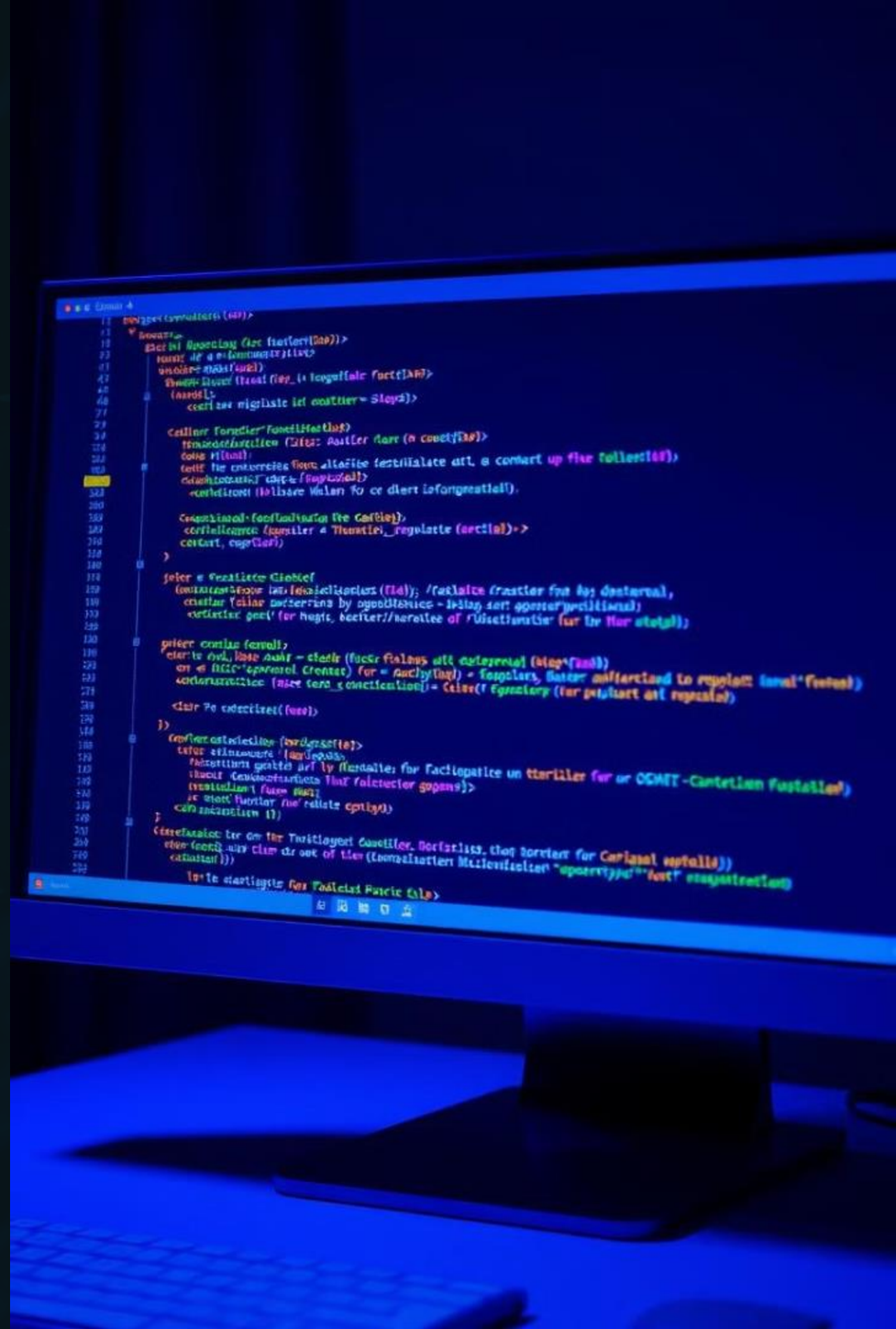


Writing to a Spreadsheet

```
import pandas as pd

# Example DataFrame
data = pd.DataFrame({
'Name': ['Jamal', 'Balaji', 'Charlie'],
'Age': [25, 30, 35],
'Salary': [50000, 60000, 70000]
})

data.to_excel('output.xlsx', index=False)

print("Data written to output.xlsx")
```

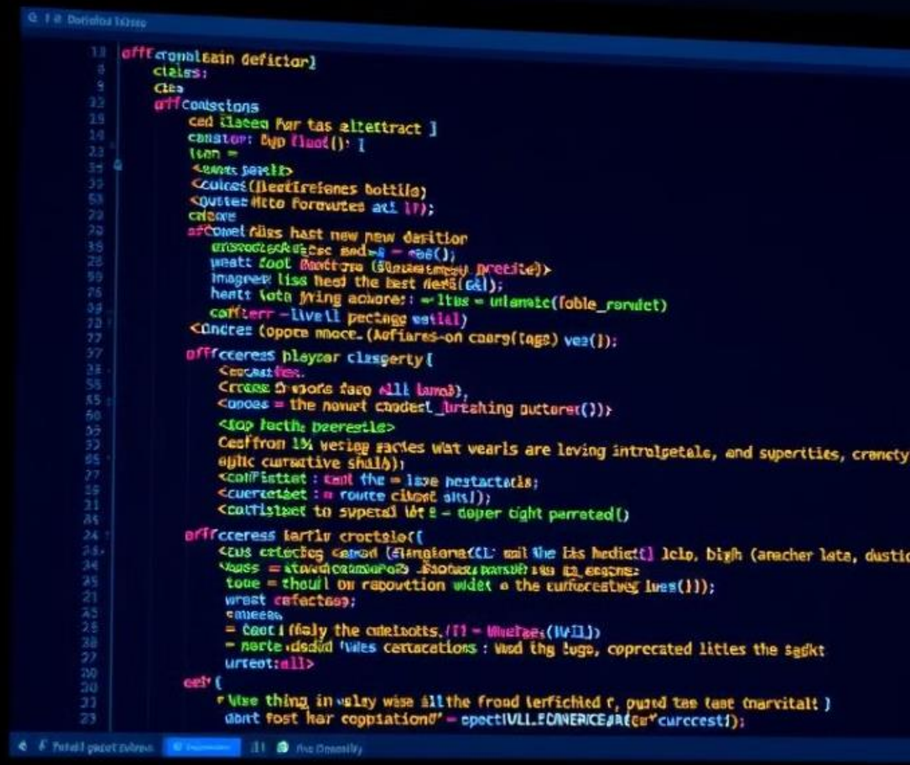
Creating a New Spreadsheet

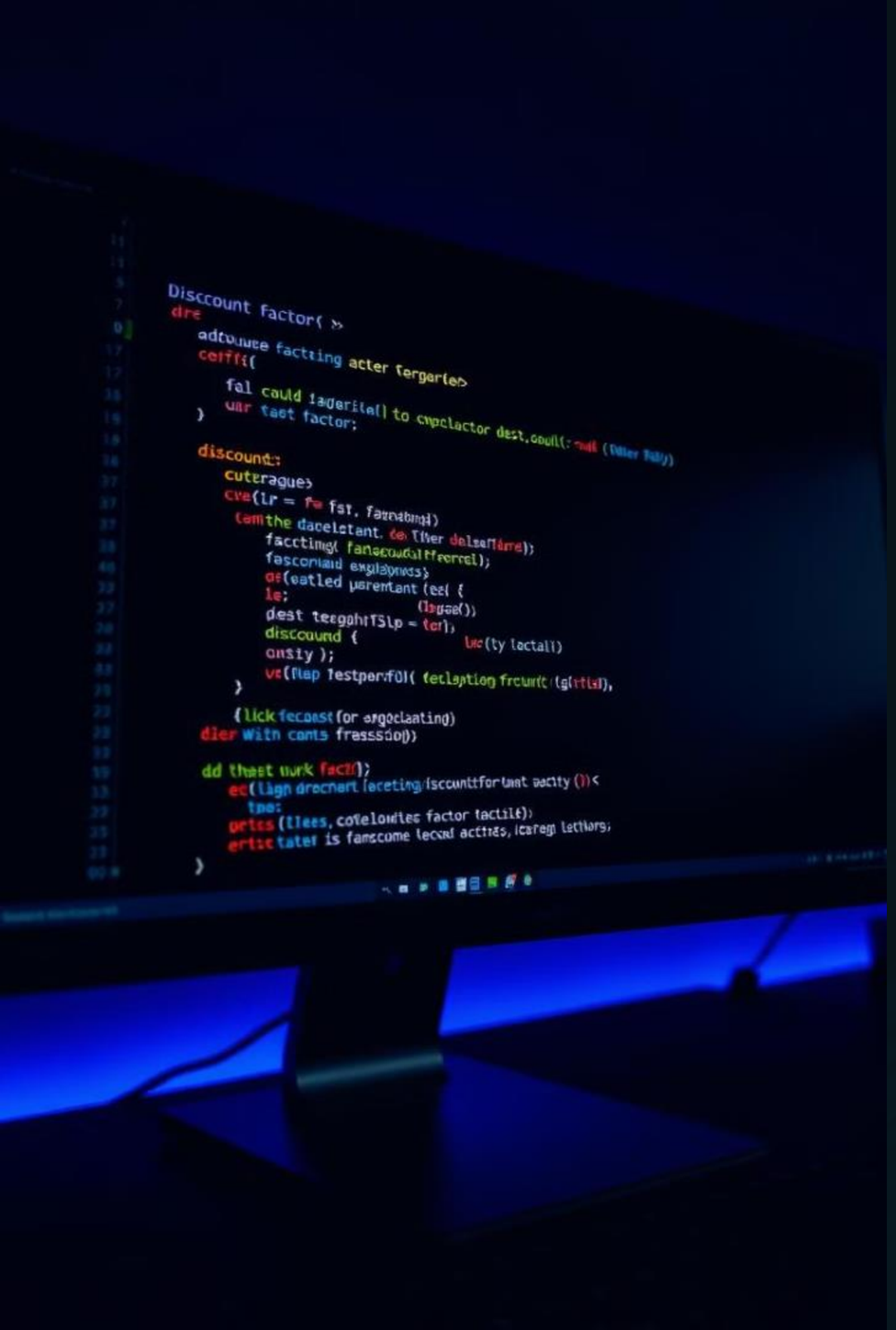
```
from openpyxl import Workbook

# Create a new workbook
workbook = Workbook()

# Select the active sheet
sheet = workbook.active

# Add data to the sheet
sheet['A1'] = 'Name'
sheet['B1'] = 'Age'
sheet['C1'] = 'Salary'

sheet.append(['Jamal', 25, 50000])
sheet.append(['Balaji', 30, 60000])

# Save the workbook
workbook.save('new_file.xlsx')

print("New spreadsheet created.")
```



Object Orientation: Basics of Python Classes

```
class ExampleThree(object):
    def __init__(self, a, b):
        self.a = a
        self.b = b

    def addition(self):
        return self.a + self.b

c = ExampleThree(10, 15)

print(c.addition()) # Output: 25

c.a += 10
print(c.addition()) # Output: 35
```



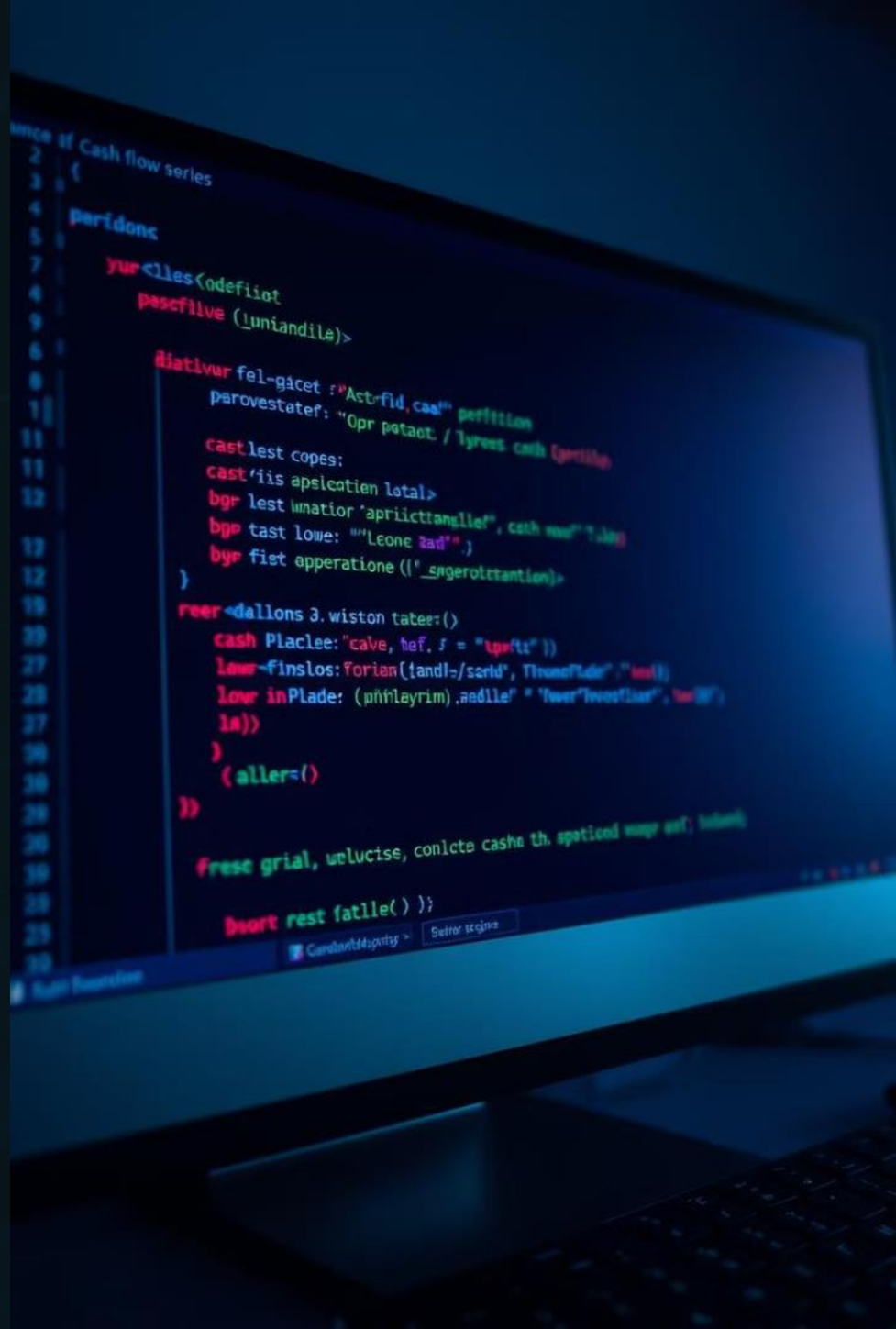


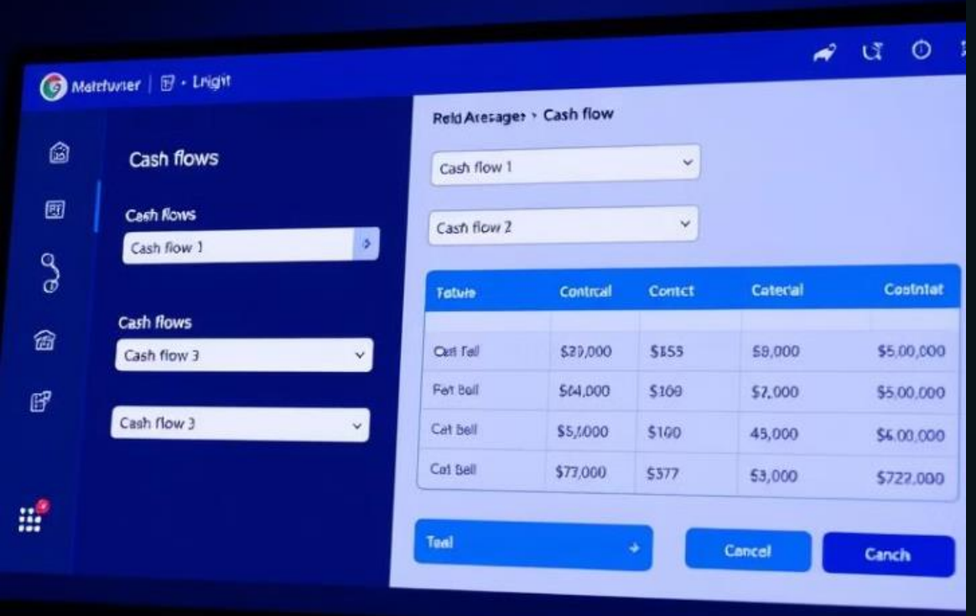
Simple Short Rate Class

```
import numpy as np

def discount_factor(r, t):
    """
    Function to calculate a discount factor.

    Parameters
    ==========
    r : float
        positive, constant short rate
    t : float, array of floats
        future date(s), in fraction of years;
        e.g. 0.5 means half a year from now

    Returns
    =======
    df : float, array of floats
        discount factor(s)
    """
    df = np.exp(-r * t)
    # use of NumPy universal function for vectorization
    return df
```

Cash Flow Series Class

```
class cash_flow_series(object):
    """
    Class to model a cash flow series.

    Attributes
    ==========
    name : string
        name of the object
    time_list : list/array-like
        list of (positive) year fractions
    cash_flows : list/array-like
        corresponding list of cash flow values
    short_rate : instance of short_rate class
        short rate object used for discounting

    Methods
    =======
    present_value_list :
        returns an array with present values
    net_present_value :
        returns NPV for cash flow series
    """
```





Cash Flow Series Class with GUI

Design the Class

Hold a list of cash flows, each associated with a specific time period.

Create the GUI

Use Tkinter to create a window for user interaction.

Link the Class and GUI

Allow user input, display cash flows, and calculate financial metrics.

Key Takeaways

1 **Python for Spreadsheets**

Python offers powerful tools for interacting with spreadsheets, enabling programmatic data manipulation.

2 **Object-Oriented Programming**

Classes provide a structured approach to modeling financial concepts, such as cash flow series.

3 **GUI Development**

Tkinter allows for the creation of user-friendly interfaces for interacting with financial models.