
# Python: Functions and Files

**Day 3**

**FinTech**

**Lesson 2.3**

# Class Objectives

By the end of this class, you will be able to:

01 Define time value of money and explain how it's used.

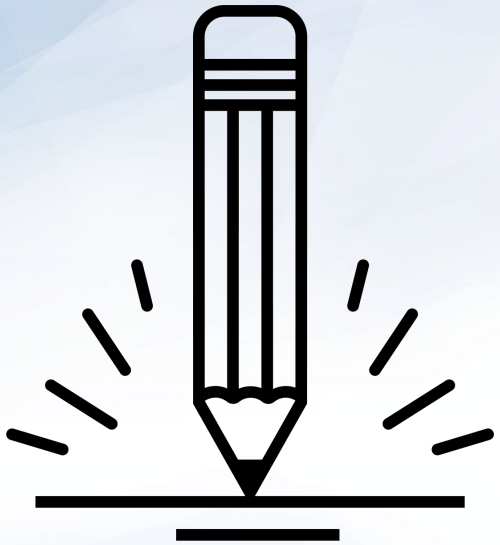
02 Perform basic financial analysis from user-defined financial functions (NPV).

03 Import standard and custom Python libraries.

04 Read and write text files.

05 Identify tabular data and its form.

06 Read and write CSV files.



# Activity: Refresher (15 min)

In this activity, you will apply skills learned in the previous lessons to a financial use case. You will act as an analyst who categorizes customers based on revenue and assign each customer a business tier: platinum, gold, silver, or bronze.

(Instructions sent via Slack)

**Suggested Time:**

**Time's Up!** Let's Review.

# Time Value of Money

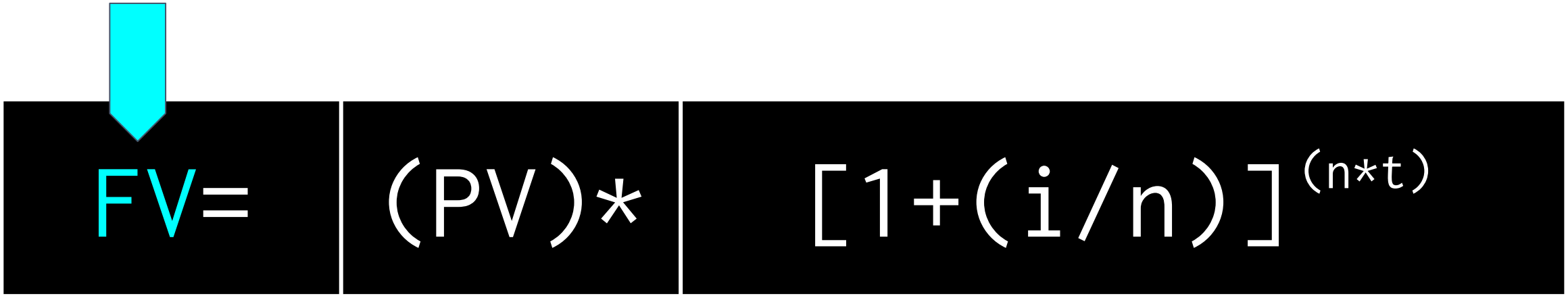
Would you rather have $50 today or in two weeks? **Why?**

# Money today is worth more than money tomorrow, due to the fact that money today can be invested and grown.

# Time Value of Money

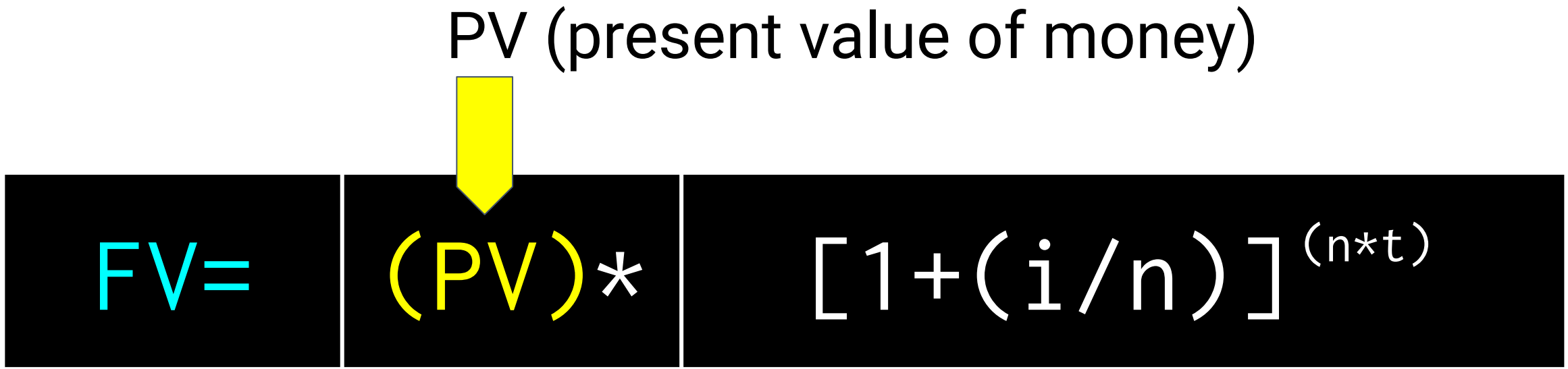
What does “time value of money” mean?

FV (future value of money)



# Time Value of Money

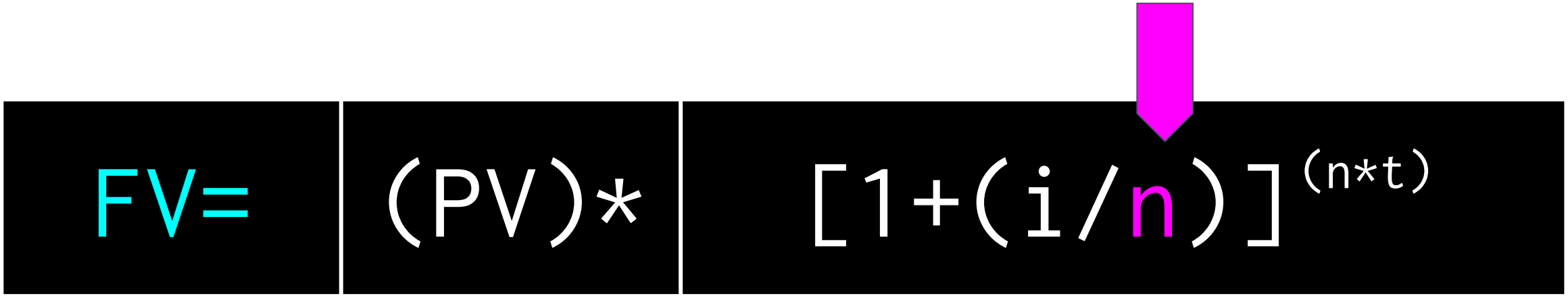
What does “time value of money” mean?



# Time Value of Money

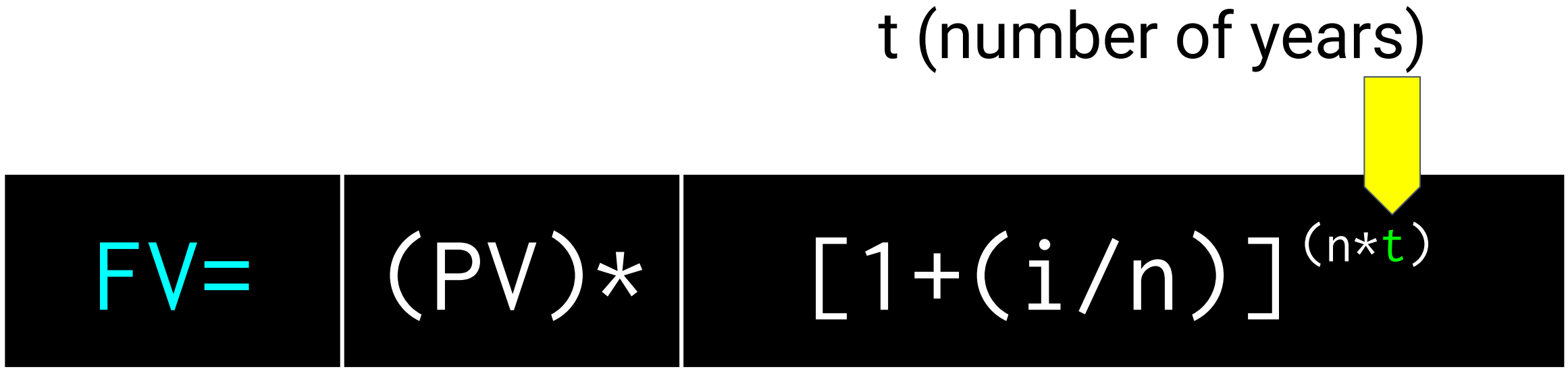
What does “time value of money” mean?

n (number of compounding periods per year)



# Time Value of Money

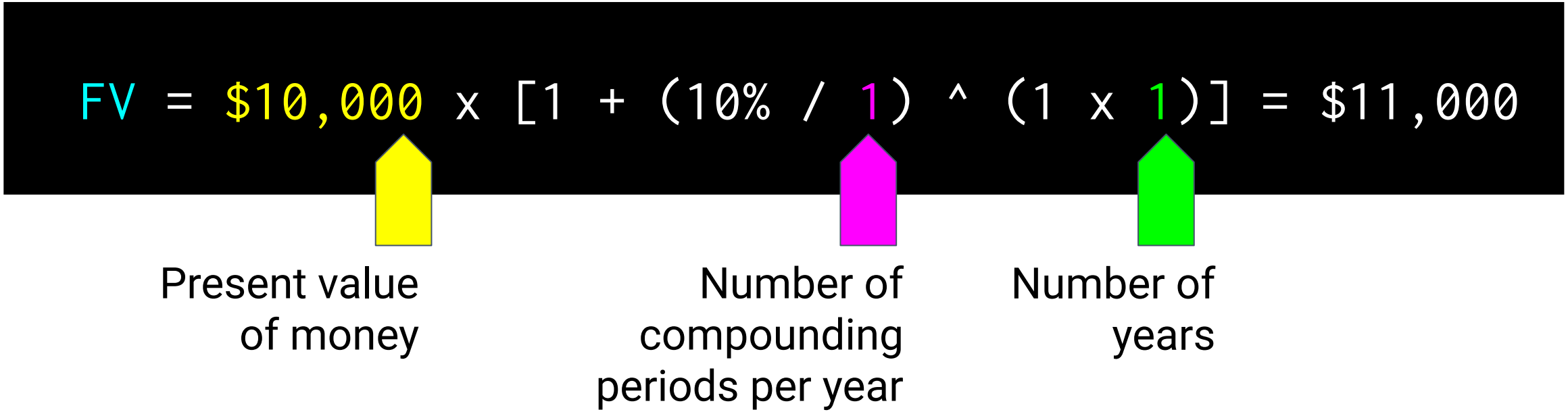
What does “time value of money” mean?



# Time Value of Money

How does it apply?

Assume a sum of $10,000 is invested for one year at 10% interest.
The future value of that money is:



# Time Value of Money

How does it apply?

Now let's compare the differences in compounding periods for FV (future value):

| | |
|---|---|
| **Yearly** Compounding | `FV = $10,000 x [1 + (10% / 1) ^ (1 x 1)] = $11,000` |
| **Quarterly** Compounding | `FV = $10,000 x [1 + (10% / 4) ^ (4 x 1)] = $11,038` |
| **Monthly** Compounding | `FV = $10,000 x [1 + (10% / 12) ^ (12 x 1)] = $11,047` |
| **Daily** Compounding | `FV = $10,000 x [1 + (10% / 365) ^ (365 x 1)] = $11,052` |

# Time Value of Money: Other Considerations

# Ponder This Statement

Manipulating the future value formula allows us to calculate the present value, with all else given.

# **Time Value of Money:** Other Considerations

**Which option would you choose?**

Receive
**$14,000.00**
today

Receive
**$15,000.00**
in 1 year

# **Time Value of Money:** Other Considerations

**Which option would you choose?**

Receive
**$14,000.00**
today

Receive
**$15,000.00**
in 1 year

You should take the $14,000 today, as the PV of $15,000 one year from now is worth $13,636.36 today.

---

By doing so, you would be able to invest the higher amount at 10.00% for 1 year, which would end up being more than $15,000.00.

# **Time Value of Money:** Other Considerations

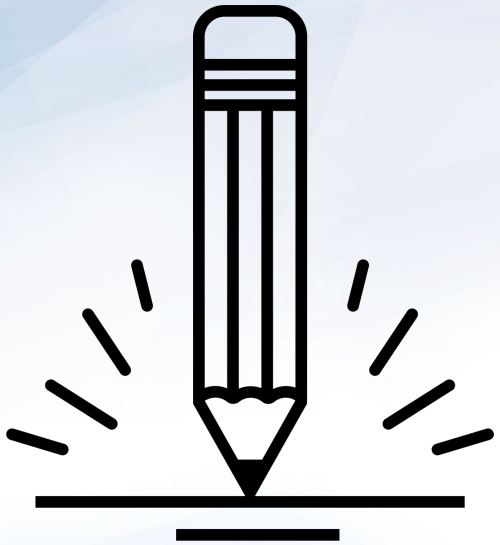
The value today of receiving **$15,000.00** in 1 year:

```
PV = $15,000 / (1 + (10% / 1) ^ (1 x 1)) = $13,636
```

**Difference in Investment:**

13,636 x 1.1 = $15,000

14,000 x 1.1 = $15,400



# Activity: Zero Coupon Bonds (25 min)

In this activity, you will use the concept of time value of money (TVM) to discount the future value of a zero-coupon bond to determine its present value. You will also compare the present value to its current selling price in order to decide whether or not to purchase the bond.

(Instructions sent via Slack)

**Suggested Time:**

**Time's Up!** Let's Review.

# Net Present Value

# Net Present Value

What does net present value mean?

**Net present value (NPV)** is the difference between the sum of present values of future cash flows and the absolute value of an initial investment.

$$\mathbf{NPV = \Sigma(Cash\ Flow\ /\ (1 + i)^{t}\ )- |\ Initial\ Investment\ |}$$

**i = Discount rate or return that could be earned in alternative investments**
**t = Number of years**

# Net Present Value

## So how is NPV used?

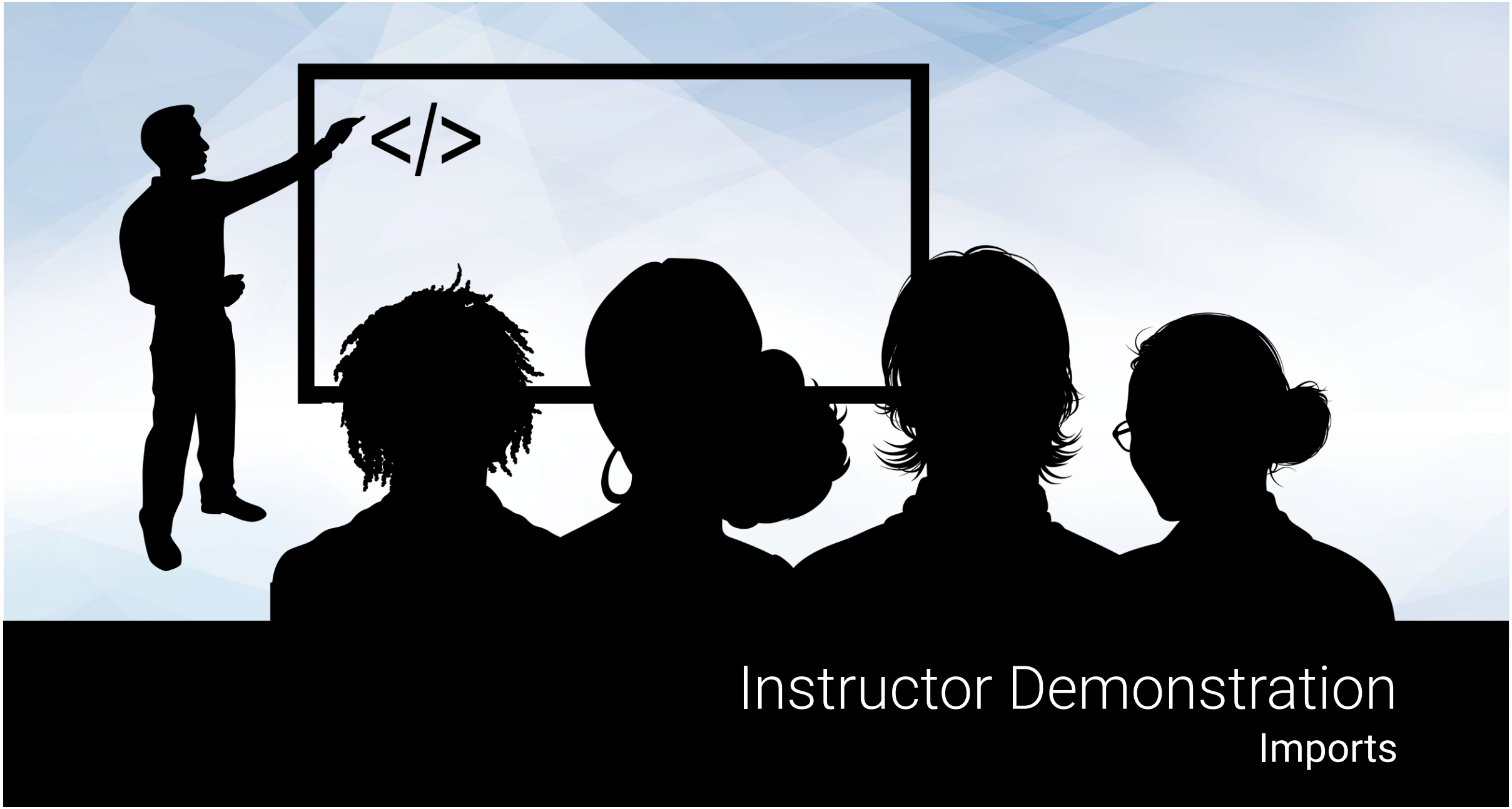
**Net present value (NPV)** can help financial analysts determine the profitability of potential projects that involve initial investments and projected future cash flows, and, therefore, the action to take.

**Example:**

**A proposed project costs $1,000 and is projected to return cash flows of $400 in Year 1, $400 in Year 2, $400 in Year 3, and $400 in Year 4. Assume the discount rate is 10%. Is this project worth it?**

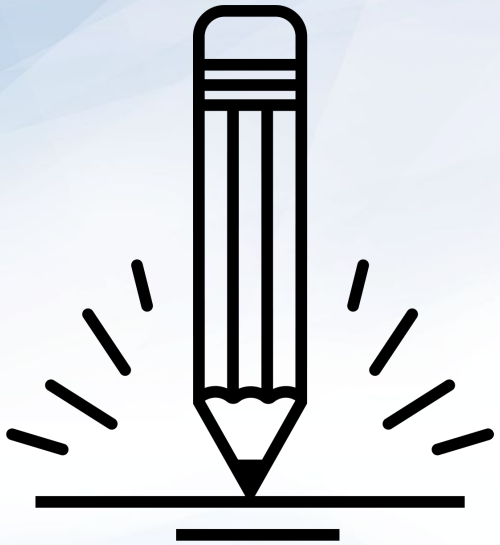
NPV = $400 / $(1 + 0.10)^1$ + $400 / $(1 + 0.10)^2$ + $400 / $(1 + 0.10)^3$ + $400 / $(1 + 0.10)^4$ - $1,000 = $267.94617853971704

**Therefore, according to the NPV calculation, this project is PROFITABLE and SHOULD be undertaken.**




# Instructor Demonstration

## Imports



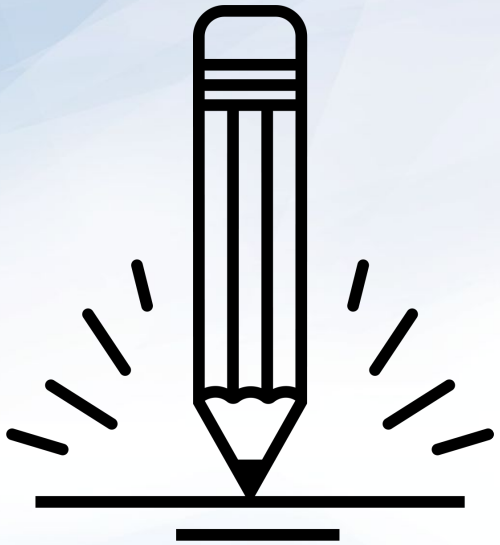
# Activity: Net Present Value (15 min)

In this activity, you will conduct financial analysis on three potential company projects that are categorized as conservative, neutral, and aggressive. Then, you will use the NPV function to determine which project scenario is the optimal choice.

(Instructions sent via Slack)

**Suggested Time:**

**Time's Up!** Let's Review.



# Activity: File Manipulation (15 min)

In this activity, you will parse a text file. You will also sum the total number of customers and count of days in the text file to calculate the daily average of customer traffic for your e-commerce business. Assume each line is equal to a day's worth of customers.

(Instructions sent via Slack)

**Suggested Time:**

**Time's Up!** Let's Review.


Countdown timer
40:00
(with alarm)
Break

# Tabular Data

# What Is Tabular Data?

Quite simply, tabular data is data organized in a table format, in which there are rows, columns, and values at the intersection of those rows and columns.

Value “Coghill” at the intersection of Row 12 and Column Salesperson

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Country | Salesperson | Order Date | OrderID | Units | Order Amount |
| 2 | USA | Fuller | 1/01/2011 | 10392 | 13 | 1,440.00 |
| 3 | UK | Gloucester | 2/01/2011 | 10397 | 17 | 716.72 |
| 4 | UK | Bromley | 2/01/2011 | 10771 | 18 | 344.00 |
| 5 | USA | Finchley | 3/01/2011 | 10393 | 16 | 2,556.95 |
| 6 | USA | Finchley | 3/01/2011 | 10394 | 10 | 442.00 |
| 7 | UK | Gillingham | 3/01/2011 | 10395 | 9 | 2,122.92 |
| 8 | USA | Finchley | 6/01/2011 | 10396 | 7 | 1,903.80 |
| 9 | USA | Callahan | 8/01/2011 | 10399 | 17 | 1,765.60 |
| 10 | USA | Fuller | 8/01/2011 | 10404 | 7 | 1,591.25 |
| 11 | USA | Fuller | 9/01/2011 | 10398 | 11 | 2,505.60 |
| 12 | USA | Coghill | 9/01/2011 | 10403 | 18 | 855.01 |
| 13 | USA | Finchley | 10/01/2011 | 10401 | 7 | 3,868.60 |
| 14 | USA | Callahan | 10/01/2011 | 10402 | 11 | 2,713.50 |
| 15 | UK | Rayleigh | 13/01/2011 | 10406 | 15 | 1,830.78 |
| 16 | USA | Callahan | 14/01/2011 | 10408 | 10 | 1,622.40 |
| 17 | USA | Farnham | 14/01/2011 | 10409 | 19 | 319.20 |
| 18 | USA | Farnham | 15/01/2011 | 10410 | 16 | 802.00 |

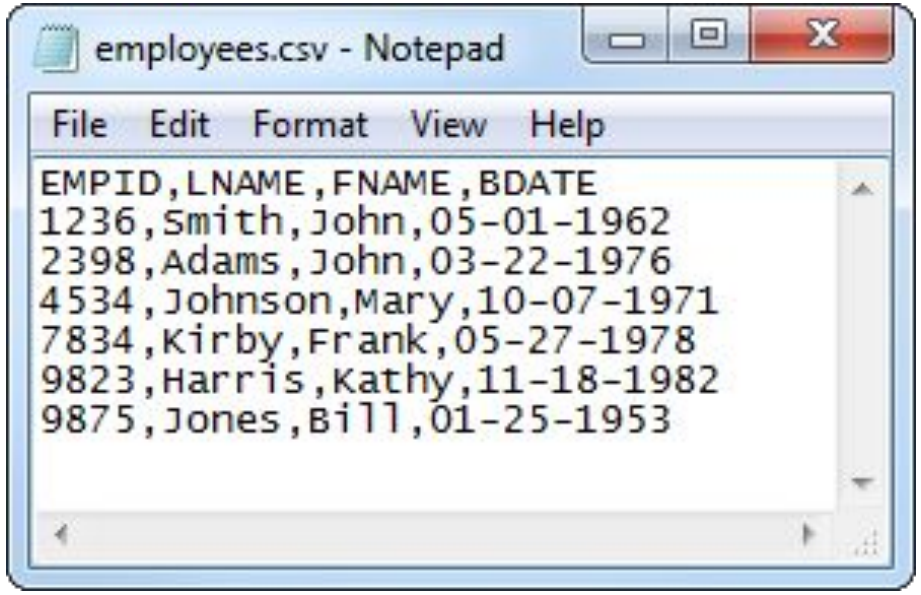
# Tabular Data Files

There are two common types of tabular data files:

01

## CSV (Comma-Separated Values)

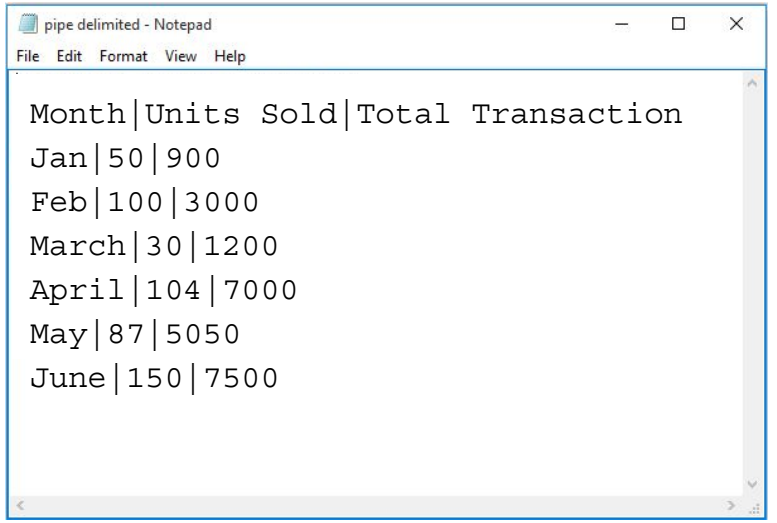
Each row has values separated by a comma.

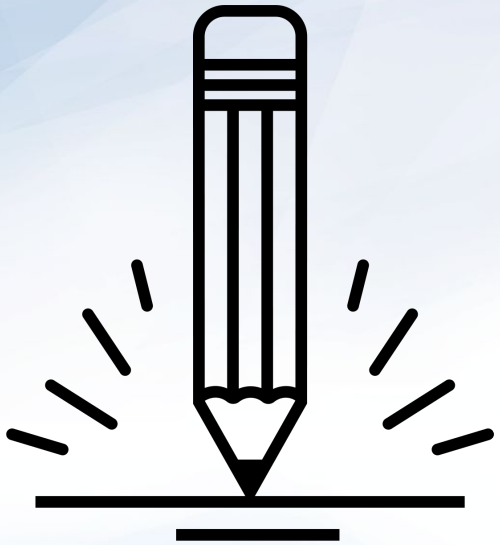



02

## Delimited Files

Each row has values separated by a specific **delimiter**. This example shows values delimited by a pipe character '|'.





# Activity: Sales Analysis (15 min)

In this activity, you will read in the `sales.csv` file to calculate customer revenue averages.

(Instructions sent via Slack)

**Suggested Time:**

**Time's Up!** Let's Review.

# Questions?

# Appendix

# Function Anatomy

# Function Anatomy

Defining the function, parameters, and return values:

```
1
2
3   def create_phase(name):
4       greeting = f"Hi {name} it's nice to meet you!"
5       return greeting
6
7   sentence = create_phase("Andrew")
8   print(sentence)
9
```

# Function Anatomy

Defining the function, parameters, and return values:

Defines the function `name` as well as the parameters it supports.

```
1
2
3   def create_phase(name):
4       greeting = f"Hi {name} it's nice to meet you!"
5       return greeting
6
7   sentence = create_phase("Andrew")
8   print(sentence)
9
```

# Function Anatomy

Defining the function, parameters, and return values:

The function uses the name parameter as a substitute for a dynamically created `greeting` string.

```
1
2
3   def create_phase(name):
4       greeting = f"Hi {name} it's nice to meet you!"
5       return greeting
6
7   sentence = create_phase("Andrew")
8   print(sentence)
9
```

# Function Anatomy

Defining the function, parameters, and return values:

The function returns the greeting variable.

```
1
2
3   def create_phase(name):
4       greeting = f"Hi {name} it's nice to meet you!"
5       return greeting
6
7   sentence = create_phase("Andrew")
8   print(sentence)
9
```

# Function Anatomy

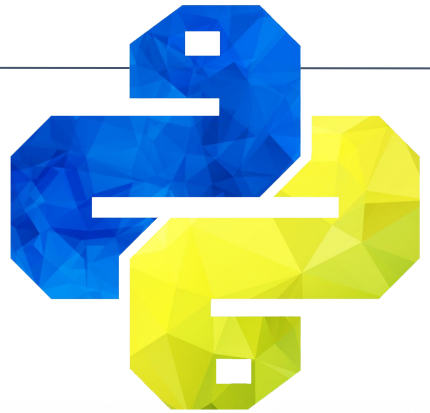
Defining the function, parameters, and return values:

Calls the function and assigns the return value to the variable `sentence`.

```
1
2
3   def create_phase(name):
4       greeting = f"Hi {name} it's nice to meet you!"
5       return greeting
6
7   sentence = create_phase("Andrew")
8   print(sentence)
9
```

# Function Anatomy

Showcasing the Python function and output:



```
1
2
3   def create_phase(name):
4       greeting = f"Hi {name} it's nice to meet you!"
5       return greeting
6
7   sentence = create_phase("Andrew")
8   print(sentence)
9
```