

Tutorial 1: VS Code Basics for Complete Beginners

👋 Hey Mauricio!

Welcome to your first hands-on tutorial! This is where your journey from "never used VS Code" to "building financial models with AI" begins. Dad designed this to be super beginner-friendly - no prior coding experience needed!

By the end of this hour, you'll feel right at home in VS Code. Let's get started! 🚀

🎯 What You'll Learn (60 minutes)

By the end of this tutorial, you'll:

- Navigate VS Code like a pro
- Use keyboard shortcuts for speed (save 10+ hours/month!)
- Manage files and folders
- Use the integrated terminal (no separate command prompt needed!)
- Write and run your first Python code
- Feel confident and ready for more!

Perfect for: Never used VS Code before (that's okay!) **Time:** 60 minutes hands-on **Prerequisites:** VS Code installed (see QUICK_START_GUIDE.md)

💡 **Why This Matters at PE Club:** VS Code is the tool top tech companies and quant funds use. Learning it sets you apart from 95% of finance professionals who only know Excel.

📋 Progress Tracker

Track your progress as you go:

- Part 1: Understand VS Code interface (10 min)
 - Part 2: Create your first project (10 min)
 - Part 3: Master keyboard shortcuts (15 min)
 - Part 4: Use the terminal (10 min)
 - Part 5: Write and run Python code (15 min)
 - ✨ Bonus: Customize VS Code
-

Part 1: The VS Code Interface (10 minutes)

🎬 Open VS Code

1. **Open VS Code** from Desktop or Start Menu
2. You should see the **Welcome** screen

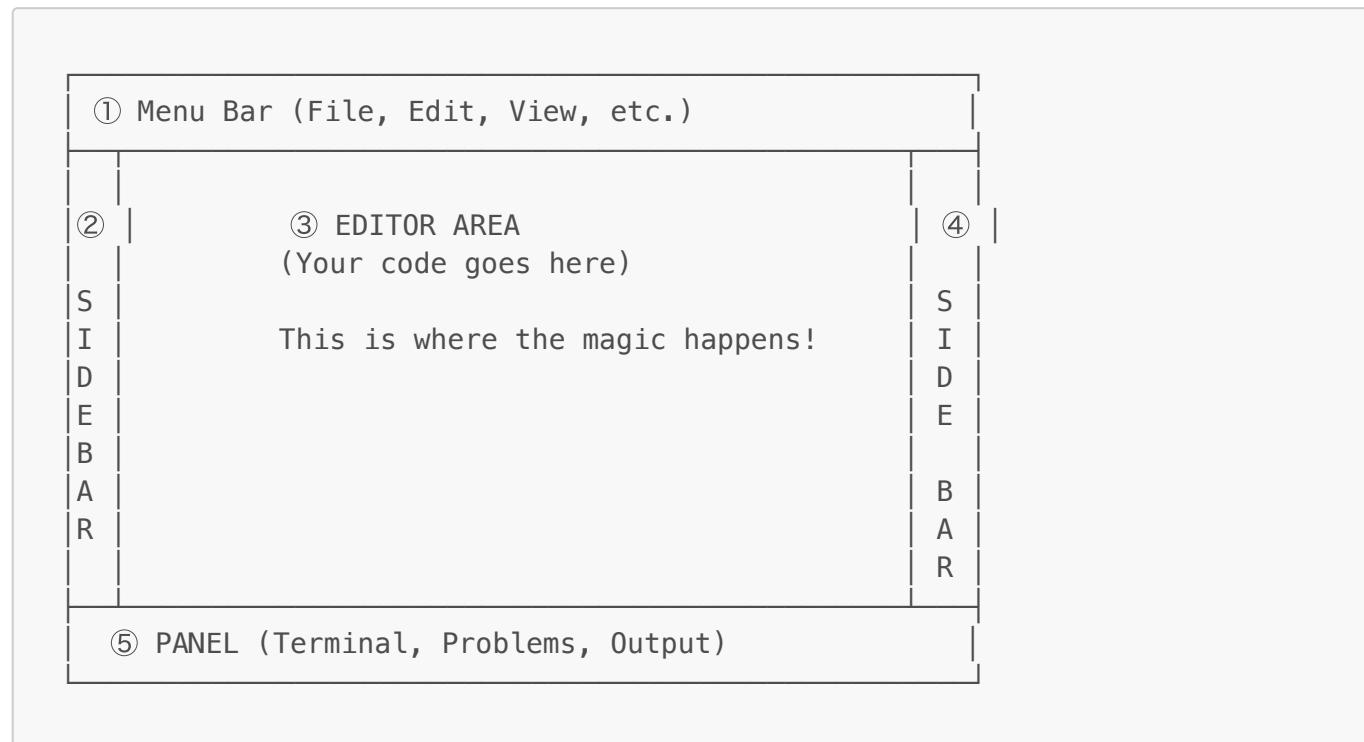
🖼 [SCREENSHOT: VS Code Welcome Screen - clean interface]

First impressions:

- Don't be intimidated! It looks complex but you'll master it quickly
- The interface is designed for efficiency
- Everything has a purpose

The 5 Key Areas You Need to Know

Think of VS Code like a cockpit - each section has a specific purpose:



① Activity Bar (Left edge) - Your main navigation hub

- 📁 Explorer (files) - Browse your project files
- 🔎 Search - Find text across all files
- ⚙️ Source Control (Git) - Version control (we'll use this later!)
- 🐞 Run & Debug - Test your code
- 🧩 Extensions - Add superpowers to VS Code

② Side Bar (Left panel) - Shows details

- Content changes based on Activity Bar selection
- Can be hidden with **Ctrl+B** for more screen space
- Don't worry if it seems busy - you'll use it naturally

③ Editor Area (Center) - Your workspace

- Where you write code and edit files
- Can have multiple files open in tabs (like browser tabs!)
- Can split into multiple panes (compare files side-by-side)
- This is where 80% of your time will be spent

④ Side Bar (Right) - Optional extra panel

- Usually hidden by default
- Can show outline, timeline, or custom panels
- Ignore this for now!

⑤ Panel (Bottom) - Your tools drawer

- **Terminal** - Run commands without leaving VS Code!
- **Problems** - Shows code errors automatically
- **Output** - See what your code prints
- **Debug Console** - Advanced debugging (later!)

Status Bar (Bottom edge) - Quick info at a glance

- Line/column number (where your cursor is)
- File type (Python, Markdown, etc.)
- Git branch (if using Git)
- Errors/warnings count

👉 Try It Now: Explore the Interface (5 minutes)

Follow these steps exactly:

1. Click each icon in **Activity Bar** (left edge)
 - Click Explorer
 - Click Search
 - Click Source Control
 - Click Extensions
 - Notice how **Side Bar** content changes!
2. Toggle Side Bar visibility:
 - Press **Ctrl+B** to hide Side Bar
 - Press **Ctrl+B** again to show it
 - **Pro tip:** Hide it when coding for more space!
3. Toggle Panel (bottom):
 - Press **Ctrl+J** to hide Panel
 - Press **Ctrl+J** to show it
 - This is your terminal - you'll use it ALL the time
4. Get a clean workspace:
 - Press **Ctrl+B** (hide Side Bar)
 - Press **Ctrl+J** (hide Panel)
 - Press them again to bring everything back
 - **Feel the power!** You control your workspace

⌚ **Checkpoint: Can you hide/show panels?** If yes, you're ready to move on! If not, practice the shortcuts above 2-3 times.

Part 2: Creating Your First Project (10 minutes)

 **Why projects matter:** At PE Club, each deal or analysis should have its own project folder. This keeps everything organized and makes collaboration easy.

Create Project Folder

Method 1: Using Explorer (Visual, beginner-friendly)

1. Click **Explorer** icon (📁) in Activity Bar
2. Click "Open Folder" button
3. Navigate to **Documents** folder
4. Click "New Folder" button
5. Name it: **financial-modeling-practice**
6. Click "Select Folder"

 [SCREENSHOT: Windows Explorer with New Folder dialog]

VS Code will now show your empty folder in the Explorer panel.

Method 2: Using Terminal (Pro way - faster once you learn it!)

1. Press **Ctrl+``** (backtick key, left of 1') to open terminal
2. Type exactly:

```
cd Documents  
mkdir financial-modeling-practice  
cd financial-modeling-practice
```

3. File → Open Folder → Select **financial-modeling-practice**

Create Your First File

3 Ways to Create a File:

Method 1: Side Bar

- Hover over folder name in Explorer
- Click "New File" icon (📄)
- Name: **my_first_code.py**

Method 2: Keyboard (Fastest!)

- Press **Ctrl+N** (New file)
- Press **Ctrl+S** (Save)
- Name: **calculator.py**

Method 3: Quick Open

- Press **Ctrl+P**

- Type: `calculator.py`
- If doesn't exist, it'll create it!

👉 Try It: Create These Files

Create this structure:

```
financial-modeling-practice/
└── hello.py
└── calculator.py
└── notes.txt
```

Practice:

1. Create `hello.py` using Method 1
 2. Create `calculator.py` using Method 2
 3. Create `notes.txt` using Method 3
-

Part 3: Essential Keyboard Shortcuts (15 minutes)

The Most Important Shortcut

Ctrl+P - Quick Open

Part 3: Keyboard Shortcuts - Work Like a Pro (15 minutes)

 **Why shortcuts matter at PE Club:** Analysts who use keyboard shortcuts analyze deals 10x faster. While others fumble with mouse clicks, you'll execute commands in milliseconds. This looks professional and saves hours every week.

 The One Shortcut to Rule Them All: **Ctrl+P**

Quick Open File - This is THE most important shortcut!

```
Ctrl+P      Open any file by name (LEARN THIS FIRST!)
```

Why it's magical:

- Opens any file in your project instantly
- No clicking through folders
- Just type 3-4 letters of filename
- Works on projects with 100+ files!

⌚ Try It Now:

1. Press **Ctrl+P**
2. Type `hello` (notice it finds `hello_world.py`!)

3. Press Enter → File opens instantly!
4. Try again: **Ctrl+P** → Type **calc** → Opens **calculator.py**!

 **Challenge:** Close all files (**Ctrl+K W**), then reopen them using only **Ctrl+P**. This will become muscle memory!

File Operations - Never Touch the Mouse

Shortcut	What It Does	When to Use
Ctrl+N	New file	Starting fresh code
Ctrl+O project	Open file dialog	Opening files outside
Ctrl+S (muscle memory)	Save file	Every 30 seconds!
Ctrl+Shift+S	Save as (new name)	Creating variations
Ctrl+W	Close current file	Decluttering tabs
Ctrl+K W	Close ALL files	Fresh start
Ctrl+Tab quickly	Switch between open files	Jumping between files
Ctrl+Shift+T wrong one!"	Reopen closed file	"Oops, I closed the

 **Practice Drill** (Do this 3 times to build muscle memory):

1. **Ctrl+N** - Create new file
2. Type: **# This is a test**
3. **Ctrl+S** - Save as **practice.py**
4. **Ctrl+W** - Close it
5. **Ctrl+P** - Type **prac** - Reopen it!
6. **Ctrl+W** - Close again
7. **Ctrl+Shift+T** - Reopen (alternative method!)

 **Checkpoint:** Can you create, save, close, and reopen a file without using the mouse? If yes, you've unlocked level 2! 

Editing Shortcuts - The Time Savers

Shortcut	Magic Trick
Ctrl+C	Copy line (NO SELECTION NEEDED!)
Ctrl+X	Cut line (entire line automatically!)
Ctrl+V	Paste
Ctrl+Z	Undo (use liberally!)
Ctrl+Y	Redo
Ctrl+/	Comment/uncomment code (game changer!)

Ctrl+]	Indent line right
Ctrl+[Indent line left

⌚ Try the "No Selection" Trick (This will blow your mind!):

1. Open `calculator.py`
2. Put cursor on line with `print` statement
3. **Don't select anything - just cursor on line**
4. Press `Ctrl+C` - Entire line copied!
5. Press `Ctrl+V` - Entire line pasted!
6. Press `Ctrl+/` - Entire line commented!
7. Press `Ctrl+/` again - Uncommented!

Why this matters: Most people select text with mouse before copying. You'll do it in 1 keystroke!

☒ Moving Code Around - Like a Chef Tossing Pizza

Shortcut	What It Does	Visual
<code>Alt+↑</code>	Move line up	
<code>Alt+↓</code>	Move line down	
<code>Shift+Alt+↑</code>	Copy line up	
<code>Shift+Alt+↓</code>	Copy line down	

⌚ Practice: Reorganize Code

Type this messy code:

```
print("Third")
print("First")
print("Second")
```

Now fix the order WITHOUT cutting/pasting:

1. Put cursor on `print("First")` line
2. Press `Alt+↑` - Moves up!
3. Put cursor on `print("Third")` line
4. Press `Alt+↓` twice - Moves to bottom!

Result: Organized in seconds!

Real PE use case: Reorganizing financial model sections - you'll do this DAILY.

⌚ Multi-Selection - Edit Multiple Things at Once (Advanced!)

Shortcut	Superpower Unlocked
Ctrl+D	Select next same word (multi-cursor!)
Ctrl+Shift+L	Select ALL same words
Alt+Click	Click multiple places (multi-cursor!)
Ctrl+L	Select entire line
Shift+Alt+→	Expand selection smartly

🎯 Try Multi-Cursor Magic:

1. Type this code:

```
old_value = 100
total = old_value + 50
result = old_value * 2
```

2. Double-click `old_value` (first occurrence)
3. Press `Ctrl+D` - Selects next `old_value` (2nd cursor!)
4. Press `Ctrl+D` - Selects third `old_value` (3rd cursor!)
5. Type `new_value` - **ALL THREE CHANGE AT ONCE!** 🎉

Alternative method (select all at once):

1. Click anywhere in `old_value`
2. Press `Ctrl+Shift+L` - All 3 selected instantly!
3. Type `new_value`

Why this is powerful: Renaming variables across entire files in 2 seconds vs 2 minutes!

🧭 Navigation Shortcuts - Jump Around Like a Kangaroo

Shortcut	Teleport To...
Ctrl+G	Go to specific line number
Ctrl+Home	Top of file (line 1)
Ctrl+End	Bottom of file (last line)
Ctrl+←	Jump word by word left
Ctrl+→	Jump word by word right
Ctrl+Shift+0	Jump to symbol (functions, classes)
F12	Go to definition (where code is defined)
Alt+←	Go back to previous location
Alt+→	Go forward

🎯 Practice Navigation:

1. Open any Python file

2. Press **Ctrl+G** → Type **1** → Enter (jump to top)
3. Press **Ctrl+End** (jump to bottom)
4. Press **Ctrl+G** → Type **10** → Enter (jump to line 10)

Real scenario: Debugging error on line 847 of a 1000-line financial model:

- Old way: Scroll for 30 seconds 😞
- Pro way: **Ctrl+G** → **847** → Enter ⚡

✓ Checkpoint:

- Can you copy a line without selecting it?
- Can you move lines up/down with keyboard?
- Can you open files with **Ctrl+P**?
- Can you jump to any line with **Ctrl+G**?

If 3/4 checked = You're ready for Part 4! 🚀

💡 Shortcuts Cheat Sheet (Your New Best Friend)

Print this and keep it by your desk for Week 1:

TOP 10 SHORTCUTS FOR DAY 1

1. Ctrl+P	Quick open (USE 100x/day!)
2. Ctrl+S	Save (USE CONSTANTLY!)
3. Ctrl+`	Open terminal
4. Ctrl+/_	Comment code
5. Ctrl+D	Multi-select next
6. Alt+↑/↓	Move line
7. Ctrl+Shift+P	Command Palette
8. Ctrl+W	Close file
9. Ctrl+G	Go to line
10. Ctrl+B	Toggle sidebar (more space!)

💡 Master these 10 first. Add more after Week 1!

Part 4: Working with Terminal - Your Command Center (10 minutes)

💡 Why terminal matters at PE Club: 90% of finance people are scared of terminals. You won't be! The terminal lets you run Python scripts, install packages, and use Git (version control). This is how real developers work, and you'll look like one! 😎

Opening the Terminal - 3 Different Ways

Method 1 (Fastest):

Ctrl+` (backtick key, under Esc)

Method 2 (Menu):

View menu → Terminal

Method 3 (Command Palette):

Ctrl+Shift+P → Type "terminal" → Select "Toggle Terminal"

⌚ Try all 3 methods, then stick with `Ctrl+`` - it's the fastest!

 [Screenshot: Terminal panel at bottom of VS Code]

Understanding the Terminal

The terminal shows:

PS C:\Users\Mauricio\financial-modeling>_

Let's decode this:

- PS = PowerShell (Windows terminal type)
- C:\Users\Mauricio\financial-modeling = Your current location
- > = Waiting for your command
- _ = Cursor (you type here)

Think of it like File Explorer, but with typing instead of clicking!

Essential Terminal Commands (Windows PowerShell)

Command	What It Does	Example
pwd	Print Working Directory	See where you are
ls	List files/folders	See what's in current
folder		
dir	List files (alternative)	Same as ls
cd folder-name	Change Directory	Enter a folder
cd ..	Go up one level	Exit current folder to
parent		
mkdir name	Make Directory	Create new folder

python file.py	Run Python script
cls	Clear Screen

Execute your code!
Clean up terminal

⌚ Try These Commands:

1. Open terminal: `Ctrl+``

2. Check location:

```
pwd
```

Output: C:\Users\Mauricio\financial-modeling

3. List files:

```
ls
```

You'll see: calculator.py, hello_world.py, etc.

4. Create a practice folder:

```
mkdir practice
```

5. Navigate into it:

```
cd practice
```

Notice prompt changes to: ...\\financial-modeling\\practice>

6. Go back up:

```
cd ..
```

Back to: ...\\financial-modeling>

7. Clear terminal (it's messy now!):

```
cls
```

Nice and clean! ✨

✓ **Checkpoint:** Open terminal, create folder, navigate into it, come back out. If you can do this, you're already ahead of 80% of finance people!

Multiple Terminals - Like Browser Tabs

You can have multiple terminals open for different tasks!

How to use multiple terminals:

1. **Create new terminal:** Click the + button in terminal panel
2. **Switch terminals:** Click dropdown menu (shows "1: powershell", "2: powershell", etc.)
3. **Name terminals:** Right-click terminal → Rename

Real workflow example:

```
Terminal 1: "Python Scripts" → Running your Python code  
Terminal 2: "Git Commands" → Version control (Tutorial 2!)  
Terminal 3: "Package Install" → Installing libraries
```

⌚ Try It:

1. Open terminal
2. Click + to create 2nd terminal
3. In terminal 1, type: `echo "Terminal 1"`
4. Switch to terminal 2, type: `echo "Terminal 2"`
5. Click dropdown - see both terminals listed!

Why this matters: You can run your model in Terminal 1 while installing packages in Terminal 2. No interruptions!

Running Python from Terminal

The moment you've been waiting for! Let's run code!

1. Make sure you're in your project folder:

```
pwd
```

Should show: `...\\financial-modeling`

2. List your files:

```
ls
```

You should see `calculator.py`

3. Run it:

```
python calculator.py
```

🎉 Your code runs! You'll see output!

If you get "python not found" error:

```
# Try these alternatives:  
py calculator.py      # Windows sometimes uses 'py'  
python3 calculator.py # Or 'python3'
```

✓ Success looks like:

```
C:\Users\Mauricio\financial-modeling> python calculator.py  
100  
25.5  
The calculation is complete!
```

Terminal Pro Tips

Tip 1: Command History

- Press ↑ arrow - See previous command
- Press ↑ multiple times - Scroll through command history
- Press Enter - Run it again!

Tip 2: Auto-Complete

- Type `cd pra` then press `Tab` - Autocompletes to `cd practice!`
- Works for filenames too!

Tip 3: Stop Running Program

- Press `Ctrl+C` - Stops currently running program
- Useful if code has infinite loop or takes too long

Tip 4: Copy/Paste in Terminal

- Copy: Select text, then `Ctrl+C`
- Paste: `Ctrl+V` or right-click → Paste

⌚ Try Command History:

1. Type: `ls`
2. Press Enter

3. Type: `pwd`
4. Press Enter
5. Press `↑` twice - `ls` appears again!
6. Press Enter - Runs `ls` again!

✓ Checkpoint Questions:

- Can you open terminal with `Ctrl+``?
- Can you navigate folders with `cd`?
- Can you run a Python file?
- Can you create multiple terminals?

If 3/4 checked → You're a terminal user now! That's impressive for Day 1! 🏆

Part 5: Writing and Running Python Code - Make It Real (15 minutes)

 **Why this matters at PE Club:** Everything you learned (VS Code, shortcuts, terminal) comes together here. You'll write actual Python code and see it work. This is where finance meets technology! 🚀

Your First Python Program - The Classic "Hello World"

Step-by-step:

1. Create new file:

- Press `Ctrl+N` (or File → New File)
- Press `Ctrl+S` to save
- Name it: `hello_world.py`
- ⚠️ Important: Must end in `.py`!

2. Write this code (type it exactly):

```
# My first Python program
print("Hello, World!")
print("I'm learning VS Code!")

# Simple calculation
investment = 10000
rate = 0.08
years = 5

future_value = investment * (1 + rate) ** years

print(f"Investment: ${investment:.2f}")
print(f"Rate: {rate:.1%}")
print(f"Future Value: ${future_value:.2f}")
```

3. Save it:

- Press **Ctrl+S**
- VS Code now recognizes it's Python (look for Python icon on tab!)

[Screenshot: Python file with syntax highlighting - colors make code readable!]

Understanding the Code (Line by Line)

Let's break it down - you'll understand EVERYTHING:

```
# My first Python program
```

- This is a **comment** (starts with `#`)
- Python ignores it - it's just for humans!
- Use comments to explain your code

```
print("Hello, World!")
```

- `print()` = Show something on screen
- Text inside quotes = "string" (fancy word for text)
- Try changing "Hello, World!" to your name!

```
investment = 10000
rate = 0.08
years = 5
```

- These are **variables** (boxes that store values)
- `investment` holds 10000
- `rate` holds 0.08 (which means 8%)
- `years` holds 5

```
future_value = investment * (1 + rate) ** years
```

- This is the **future value formula!**
- `**` means "to the power of"
- So: $\$10,000 \times (1.08)^5 = \$14,693.28$
- This is actual finance math you'll use at PE Club!

```
print(f"Investment: ${investment:.2f}")
```

- `f"..."` = f-string (inserts variable values)

- `{investment:,.2f}` = Format number with commas and 2 decimals
- `:`, adds commas (10,000 instead of 10000)
- `.2f` = 2 decimal places

 **You just learned:** comments, print, variables, math operations, and formatting!

Running Your Code - 3 Different Ways

Method 1: Play Button ► (Easiest for beginners)

1. Look at top-right corner of VS Code
2. See the ► (Run button)?
3. Click it!
4. Terminal opens and shows output!

Method 2: Keyboard Shortcut (Pro way!)

Ctrl+F5	Run without debugging (faster)
F5	Run with debugging (we'll learn this later!)

Method 3: Terminal Command (When you want control)

```
python hello_world.py
```

 **Try all 3 methods!** Find your favorite.

 **Expected Output:**

```
Hello, World!
I'm learning VS Code!
Investment: $10,000.00
Rate: 8.0%
Future Value: $14,693.28
```

 **If you see this → YOU'RE A PROGRAMMER NOW! Seriously!**

Debugging When Things Go Wrong

Common Errors (You'll see these - everyone does!):

Error 1: SyntaxError

```
SyntaxError: invalid syntax
```

Cause: Typo in code (missing :, wrong quote, etc.) **Fix:** Check line number in error, look for typos

Error 2: NameError

```
NameError: name 'investmen' is not defined
```

Cause: Variable name misspelled (`investmen` instead of `investment`) **Fix:** Check spelling exactly!

Error 3: IndentationError

```
IndentationError: unexpected indent
```

Cause: Wrong spacing at start of line **Fix:** Delete spaces, re-type correctly (or press `Shift+Tab` to unindent)

👉 **Try Breaking Your Code** (Seriously - this is how you learn!):

1. Remove a quote mark → Run → See error
2. Fix it → Run → Works again!
3. Misspell a variable → Run → See error
4. Fix it → Run → Works!

💡 **Pro tip:** Errors are your friends! They tell you exactly what's wrong and what line. Read the last line of error message first!

Your Turn: Build a Financial Calculator

Now let's build something useful for PE work!

Create new file: `pv_calculator.py`

Type this code:

```
# Present Value Calculator – PE Club Tool
# This is what you'll use for DCF models!

print("")
print("    PRESENT VALUE CALCULATOR      ")
print("    For PE Club Brussels        ")
print("\n")

# Get inputs from user
print("    Enter your assumptions:\n")

future_value = float(input("Future Cash Flow: $"))
rate = float(input("Discount Rate (decimal, e.g., 0.10 for 10%): "))
years = int(input("Years to discount: "))
```

```

# Calculate present value
present_value = future_value / ((1 + rate) ** years)

# Calculate discount amount
discount_amount = future_value - present_value

# Display results
print("\n" + "="*40)
print("✓ RESULTS:")
print("="*40)
print(f"Future Cash Flow: ${future_value:.2f}")
print(f"Discount Rate: {rate:.1%}")
print(f"Time Period: {years:.2} years")
print(f"-- * 40")
print(f"Present Value: ${present_value:.2f}")
print(f"Total Discount: ${discount_amount:.2f}")
print("="*40)

# Interpretation
discount_percent = (discount_amount / future_value) * 100
print(f"\n💡 Insight: ${future_value:.0f} in {years} years")
print(f"    is worth ${present_value:.0f} today")
print(f"    That's a {discount_percent:.1f}% discount!\n")

```

⌚ Run it and test with:

```

Future Cash Flow: 1000000
Discount Rate: 0.12
Years: 5

```

Expected output:

PRESENT VALUE CALCULATOR
For PE Club Brussels

📊 Enter your assumptions:

```

Future Cash Flow: $1000000
Discount Rate (decimal, e.g., 0.10 for 10%): 0.12
Years to discount: 5
=====
```

✓ RESULTS:
=====

```

Future Cash Flow: $ 1,000,000.00
Discount Rate: 12.0%
```

Time Period: 5 years

Present Value: \$ 567,426.88
Total Discount: \$ 432,573.12

💡 Insight: \$1,000,000 in 5 years
is worth \$567,427 today
That's a 43.3% discount!

🎉 You just built a real financial tool! This is what you'll do at PE Club - but with more complex models!

Code Formatting - Make It Pretty

Your code looks messy? VS Code can fix it!

Auto-format code:

1. Right-click anywhere in code
2. Select "Format Document"
3. Or press: **Shift+Alt+F**

✨ Magic happens:

- Indentation fixed
- Spacing corrected
- Lines aligned
- Looks professional!

🎯 Try it:

1. Mess up your code spacing
2. Press **Shift+Alt+F**
3. Watch VS Code fix it instantly!

💡 At PE Club: Readable code = professional code. Always format before sharing!

Code Comments - Your Future Self Will Thank You

Good comments explain WHY, not WHAT:

✗ Bad comment (obvious):

```
investment = 10000 # Set investment to 10000
```

✓ Good comment (explains reasoning):

```
investment = 10000 # Initial investment for Series A scenario
```

✓ **Great comment** (adds context):

```
# Using 12% discount rate based on:  
# - 8% risk-free rate (10-year Treasury)  
# - 4% equity risk premium  
discount_rate = 0.12
```

☛ **Practice: Add comments to your calculator**

Go back to [pv_calculator.py](#) and add explanatory comments!

Understanding Python Errors - Your Debugging Toolkit

VS Code shows errors in REAL-TIME! ☀️

Watch this:

1. Type: `print("Hello"` (missing closing quote)
2. See the red squiggly line? ⚠️
3. Hover over it - VS Code tells you what's wrong!
4. Fix it: Add `"` at end
5. Red line disappears ✓

Common Error Symbols:

- 🛡 Red squiggle = Syntax error (won't run!)
- 🟡 Yellow squiggle = Warning (will run, but might be problem)
- 💡 Light bulb = Suggestion/quick fix available

☛ **Try this debugging workflow:**

1. Write this broken code:

```
investment = 10000  
future_vlaue = investment * 1.08 # Typo: vlaue  
print(future_value) # This will error!
```

2. Run it → See [NameError](#)
3. VS Code underlines `future_value` in red
4. Hover over it - see the problem!
5. Fix the typo in line 2
6. Run again - works! ✓

✓ **Checkpoint Questions:**

- Can you write and run a Python program?
- Can you use variables and do calculations?
- Can you format strings with f-strings?
- Can you read and understand error messages?
- Can you add helpful comments?

If 4/5 checked → You're ready for actual financial modeling! 🚀

Part 6: File Management & Productivity Tricks (5 minutes)

 **Why this matters:** At PE Club, you'll work on multiple models simultaneously. These tricks let you view DCF assumptions while editing Python code - essential for productivity!

Split Editor - Work on Multiple Files

View 2+ files side-by-side like a pro!

Method 1: Right-Click Menu

1. Right-click any file tab
2. Select "Split Right"
3. Now you see file on both sides!

Method 2: Keyboard (Fastest!)

 **Ctrl+** Split current file to right

Method 3: Drag and Drop

1. Grab file tab with mouse
2. Drag to right edge of editor
3. Drop when you see split preview
4. ✨ Split view!

 [Screenshot: Split editor with Python code on left, data file on right]

Try It:

1. Open `calculator.py`
2. Press **Ctrl+** - Splits!
3. Click left side, press **Ctrl+P**, open `hello_world.py`
4. Now you see both files!
5. Edit both simultaneously!

Real PE use case:

- Left: Python DCF model code
- Right: Excel assumptions (CSV file)
- Edit code while referencing data - no switching!

Navigate Between Split Panels

```
Ctrl+1    Jump to first editor group (left)
Ctrl+2    Jump to second editor group (right)
Ctrl+3    Jump to third editor group (if 3-way split!)
```

⌚ Practice:

1. Split editor (**Ctrl+**)
2. Press **Ctrl+1** - Cursor jumps to left
3. Press **Ctrl+2** - Cursor jumps to right
4. No mouse needed!

Close Split View

```
Ctrl+W      Close current editor
Ctrl+K Ctrl+W Close ALL editors in group
```

Or: Click the **X** on file tab (the mouse way)

Editor Groups - Advanced Layout

Create 3-way split:

1. Split once (**Ctrl+**)
2. Click right panel
3. Split again (**Ctrl+**)
4. Now you have 3 panels!

Use cases:

- Panel 1: Python code
- Panel 2: Test data (CSV)
- Panel 3: Documentation (README)

⌚ Try creating and navigating a 3-way split!

Search and Replace - Change Multiple Things at Once

Find in current file:

```
Ctrl+F    Open find box
```

Find and replace:

Ctrl+H Open find and replace

🎯 Practice Search & Replace:

1. Create file with this code:

```
old_rate = 0.10
result = investment * old_rate
total = old_rate * 100
```

2. Press Ctrl+H:

- Find: old_rate
- Replace: discount_rate
- Click "Replace All"
- ✨ All 3 instances changed!

Advanced find:

- Alt+Enter (in find box) - Select all occurrences
- F3 - Go to next match
- Shift+F3 - Go to previous match

💡 **Pro tip:** Use Ctrl+D for small edits, Ctrl+H for project-wide changes!

Find in All Files - Search Your Entire Project

Ctrl+Shift+F Search across ALL files

When to use:

- "Where did I define that variable?"
- "Which files use this function?"
- "Find all TODO comments"

🎯 Try It:

1. Press Ctrl+Shift+F
2. Type: print
3. See results from ALL your Python files!
4. Click result - opens that file at that line!

 [Screenshot: Search sidebar showing results from multiple files]

Workspace Organization - Stay Organized

Save your workspace:

1. File → Save Workspace As...
2. Name it: **financial-modeling.code-workspace**
3. Next time: File → Open Workspace

Why this matters:

- Remembers open files
- Remembers layout (split panels)
- Remembers settings
- One-click to resume work!

✓ Checkpoint:

- Can you split editor for 2 files?
- Can you search and replace text?
- Can you find text across all files?
- Can you switch between splits with keyboard?

If 3/4 checked → You're a VS Code power user! 🔥

Part 7: Customization - Make VS Code YOURS (5 minutes)

 **Why customize:** You'll spend 100s of hours in VS Code. Make it comfortable! Like adjusting your desk chair - small tweaks = big productivity gains.

Change Color Theme

Make VS Code look how YOU want:

1. Press **Ctrl+Shift+P** (Command Palette)
2. Type: **Color Theme**
3. Press Enter
4. Use ↑↓ arrow keys to preview themes
5. Press Enter on your favorite!

Popular themes:

- **Dark+ (default)** - Professional, easy on eyes 🕶️
- **Light+** - Bright theme for daytime
- **Monokai** - Classic developer favorite
- **Dracula** - Purple-ish dark theme
- **One Dark Pro** - Modern, popular

⌚ Try 5 different themes, pick your favorite!

Mauricio, Dad recommends: Dark+ or One Dark Pro (install from Extensions)

Adjust Font Size - Comfortable Reading

Make code bigger/smaller:

1. **Ctrl+Shift+P**
2. Type: **Settings**
3. Search: **font size**
4. Change from 14 to your preference:
 - 16-18: Popular for comfort
 - 12-14: More code on screen
 - 20+: Presentations / screen sharing

Quick adjust:

Ctrl+Plus(+)	Zoom in (entire UI)
Ctrl+Minus(-)	Zoom out (entire UI)
Ctrl+0	Reset zoom

Find your perfect font size right now!

Enable Auto Save - Never Lose Work

Stop worrying about saving:

Method 1 (Quick):

- File menu → Click "Auto Save" (checkmark appears)

Method 2 (Detailed):

1. **Ctrl+Shift+P → Settings**
2. Search: **auto save**
3. Change from **off** to:
 - **afterDelay** (saves every 1000ms)
 - **onFocusChange** (saves when you switch files)
 - **onWindowChange** (saves when you switch apps)

💡 Recommended: **afterDelay** - Your work auto-saves every second!

✓ Once enabled, you'll see: No • (unsaved indicator) on file tabs anymore!

Install Icon Theme - Beautiful File Icons

Make files look professional:

1. **Ctrl+Shift+X** (Extensions)
2. Search: **Material Icon Theme**
3. Click Install

4. When prompted: "Select File Icon Theme"
5. Choose: Material Icon Theme

Result:

- .py files get Python logo 🐍
- .json files get JSON icon
- Folders get colored icons
- Looks like a professional IDE!



Customize Settings - Fine Tuning

Open Settings:

```
Ctrl+,      Open Settings UI
```

Useful settings to adjust:

1. **Line Numbers** (should be on by default):
 - Search: `line numbers`
 - Useful: `on` or `relative`
2. **Minimap** (small code overview on right):
 - Search: `minimap`
 - Disable if you want more space!
3. **Word Wrap** (long lines wrap):
 - Search: `word wrap`
 - Set to: `on` (easier reading)
4. **Format on Save** (auto-format when you save):
 - Search: `format on save`
 - Enable it!
5. **Tab Size** (Python standard is 4 spaces):
 - Search: `tab size`
 - Should be: `4`

⌚ Try adjusting these settings to your preference!

Recommended Settings for Finance Work

Add these to your settings:

1. **Ctrl+,** (Open Settings)
2. Click **{}** icon (top-right) for JSON view
3. Add these lines inside the **{}** brackets:

```
{  
    "editor.fontSize": 16,  
    "editor.wordWrap": "on",  
    "editor.formatOnSave": true,  
    "editor.minimap.enabled": true,  
    "files.autoSave": "afterDelay",  
    "workbench.colorTheme": "Dark+",  
    "python.analysis.typeCheckingMode": "basic",  
    "terminal.integrated.fontSize": 14  
}
```

💡 What these do:

- Comfortable reading
- Auto-save enabled
- Code auto-formats when you save
- Basic type checking for Python
- Terminal is readable

Bonus: Zen Mode - Distraction-Free Coding**When you need FOCUS:**

```
Ctrl+K Z      Enter Zen Mode (fullscreen, no sidebars!)  
Esc Esc       Exit Zen Mode (press Esc twice)
```

Perfect for:

- Writing complex financial models
- Deep focus time
- Final review before committing code

⌚ Try Zen Mode right now:

1. Open any file
2. Press **Ctrl+K** then **Z**
3. You see ONLY your code!
4. Press **Esc** twice to exit

Extensions Preview (Don't Install Yet!)

After Tutorial 2, install these superpowers:

- Python (Microsoft) – Already installed!
- Pylance – Python intelligence (already installed!)
- GitHub Copilot – AI pair programmer (Tutorial 2!)
- Jupyter – Data notebooks (Tutorial 3!)
- Excel Viewer – View CSV files nicely
- GitLens – Advanced Git features
- Python Indent – Smart indentation
- autoDocstring – Generate docstrings easily

 **Don't install extensions randomly!** Install as you need them. Too many = slower VS Code.

Customization Checkpoint:

- Changed to a theme you like?
- Adjusted font size comfortably?
- Enabled auto-save?
- Tried Zen Mode?
- Customized at least 2 settings?

If 4/5 checked → VS Code feels like YOUR tool now! 🎉

File Explorer Tricks

Right-click → Reveal in File Explorer
(Opens Windows Explorer)

Click and drag files to rearrange

Ctrl+Click Select multiple files

Part 7: Search and Replace (5 minutes)

Find in File

Ctrl+F	Find in current file
Ctrl+H	Find and replace
F3	Find next
Shift+F3	Find previous

Find in All Files

Ctrl+Shift+F	Search across all files
Ctrl+Shift+H	Replace across all files

👉 Practice:

1. Open `calculator.py`
 2. Press `Ctrl+F`
 3. Search for: "value"
 4. Press `Ctrl+H`
 5. Replace "value" with "amount"
 6. Click "Replace All"
-

Part 8: Customization (5 minutes)

Change Theme

1. Press `Ctrl+K Ctrl+T`
2. Select a theme (try "Dark+ (default dark)")
3. Pick what you like!

Popular themes:

- Dark+ (default)
- Light+ (default)
- Monokai
- Solarized Dark

Change Font Size

1. Press `Ctrl+,` (Settings)
2. Search: "font size"
3. Change to 14 or 16
4. Makes code easier to read!

Zoom

Ctrl+=	Zoom in
Ctrl+-	Zoom out
Ctrl+0	Reset zoom

Part 9: Command Palette (The Secret Weapon)

What is Command Palette?

Press `Ctrl+Shift+P` - Access EVERYTHING in VS Code!

Can't remember a shortcut? Use Command Palette!

👉 Try these commands:

1. **Ctrl+Shift+P**
2. Type: "reload" → Select "Developer: Reload Window"
3. **Ctrl+Shift+P**
4. Type: "format" → Select "Format Document"
5. **Ctrl+Shift+P**
6. Type: "settings" → Select "Preferences: Open Settings"

Most used commands:

- "Format Document" - Clean up code
- "Change Language Mode" - Change file type
- "Transform to Uppercase" - Change case
- "Sort Lines Ascending" - Sort lines

Part 10: Practice Project (5 minutes)

Build a Simple Finance Tool

Create `investment_calculator.py`:

```
"""
Simple Investment Calculator
Practice for VS Code skills
"""

def calculate_future_value(principal, rate, years):
    """Calculate future value of investment"""
    return principal * (1 + rate) ** years

def calculate_compound_annual_growth_rate(beginning, ending, years):
    """Calculate CAGR"""
    return (ending / beginning) ** (1 / years) - 1

# Main program
print("Investment Analysis Tool")
print("=" * 40)

# Get user input
principal = float(input("Initial investment: $"))
annual_rate = float(input("Annual return (e.g., 0.08 for 8%): "))
years = int(input("Investment period (years): "))

# Calculate
future = calculate_future_value(principal, annual_rate, years)
total_return = future - principal
cagr = calculate_compound_annual_growth_rate(principal, future, years)
```

```
# Display results
print("\n" + "=" * 40)
print("RESULTS")
print("=" * 40)
print(f"Initial Investment: ${principal:.2f}")
print(f"Future Value: ${future:.2f}")
print(f"Total Return: ${total_return:.2f}")
print(f"CAGR: {cagr:.2%}")
print(f"Multiple: {future/principal:.2f}x")
```

Practice these VS Code skills:

1. **File creation:** **Ctrl+N** to create file
 2. **Save:** **Ctrl+S** to save
 3. **Run:** **Ctrl+F5** to test
 4. **Format:** **Ctrl+Shift+P** → "Format Document"
 5. **Comment:** Select lines, **Ctrl+ /**
 6. **Duplicate:** **Shift+Alt+↑** to copy lines
-

✓ Skills Checklist

After this tutorial, you should be able to:

- Navigate VS Code interface confidently
- Create and organize files/folders
- Use essential keyboard shortcuts (at least 5!)
- Open and use integrated terminal
- Write and run Python code
- Search and replace text efficiently
- Split editor for multiple files
- Use Command Palette (**Ctrl+Shift+P**)
- Customize VS Code to your liking
- Feel comfortable and ready for GitHub!

✨ Mauricio, if you can check 8+ boxes, you're crushing it!

⭐ Troubleshooting Common Issues

Problem 1: "Python not found" error

Symptoms: When you try to run code, you see '**python' is not recognized**

Solutions:

```
# Try these commands in order:
python --version    # Try this first
```

```
py --version      # If python doesn't work  
python3 --version # Sometimes it's python3
```

Fix:

1. Press Windows key
2. Search "Environment Variables"
3. Click "Edit system environment variables"
4. Click "Environment Variables" button
5. Under "System variables", find "Path"
6. Click "Edit"
7. Click "New"
8. Add: **C:\Users\YourName\AppData\Local\Programs\Python\Python311**
9. Click OK on all windows
10. **Close and reopen VS Code**

Problem 2: Terminal won't open**Symptoms:** Press `Ctrl+`` but nothing happens**Solutions:**

- Try: `Ctrl+Shift+`` (with Shift key)
- Or: View menu → Terminal
- Or: **Ctrl+J** to open Panel, then click "Terminal" tab

Problem 3: Keyboard shortcuts don't work**Symptoms:** **Ctrl+P** or other shortcuts do nothing**Cause:** Another program is capturing the shortcut**Solutions:**

1. Check if Discord, Slack, or other apps are running
2. Close unnecessary programs
3. Or: Use View menu → Command Palette instead of **Ctrl+Shift+P**

Problem 4: Can't see file extensions (.py, .txt)**Fix:**

1. Open File Explorer (Windows key + E)
2. Click "View" tab
3. Check "File name extensions" box
4. Now you'll see **calculator.py** instead of just **calculator**

Problem 5: Code looks messy/ugly**Fix:** Format your code!

1. Right-click in your code
2. Select "Format Document"
3. Or press: Shift+Alt+F
4. VS Code will make it beautiful!

Problem 6: Lost panels/confused layout

Reset everything:

1. Ctrl+Shift+P (Command Palette)
2. Type: "reset"
3. Select: "View: Reset View Locations"
4. Everything back to normal!

 Still stuck?

1. **Google it:** "VS Code [your problem] Windows"
2. **Check output:** View → Output (see error messages)
3. **Restart VS Code:** Often fixes weird issues!
4. **Ask Dad:** Sergio is here to help! 😊

🎯 Keyboard Shortcuts Summary Card

 Print this or keep it visible on your second monitor!

VS CODE ESSENTIALS FOR PE ANALYSTS (Windows)

- ⭐ MOST IMPORTANT (Learn these first!)
- | | |
|--------------|--|
| Ctrl+P | Quick open file (USE THIS ALL THE TIME!) |
| Ctrl+Shift+P | Command Palette (access EVERYTHING) |
| Ctrl+S | Save file |
| Ctrl+` | Open/close terminal |

- 📁 FILES & NAVIGATION
- | | |
|----------|---------------------------|
| Ctrl+N | New file |
| Ctrl+W | Close file |
| Ctrl+Tab | Switch between open files |
| Ctrl+G | Go to line number |

- ↳ EDITING (Save 10+ hours/month!)
- | | |
|---------------|---------------------------|
| Ctrl+/ | Comment/uncomment code |
| Alt+↑/↓ | Move line up/down |
| Shift+Alt+↑/↓ | Duplicate line |
| Ctrl+D | Select next matching text |
| Ctrl+F | Find in file |

Ctrl+H Find and replace

👁️ VIEW & LAYOUT

Ctrl+B Toggle sidebar (more screen space!)
Ctrl+J Toggle bottom panel
Ctrl+\ Split editor side-by-side
Ctrl+1/2/3 Jump to editor group 1, 2, or 3

🚀 POWER MOVES (Once you're comfortable)

Ctrl+Shift+F Find in ALL files (search entire project)
Alt+Click Multiple cursors (edit multiple lines at once!)
Ctrl+Shift+L Select all occurrences of selected text
F2 Rename symbol everywhere

💡 TIP: Use ONE new shortcut each day. In 2 weeks, you'll be 10x faster than analysts using only mouse!

🚀 What's Next?

🎉 Congratulations, Mauricio! You're now comfortable with VS Code!

You've learned skills that 95% of finance professionals don't have. At PE Club, this will set you apart immediately.

Next Steps:

1. Practice shortcuts for 2-3 days

- Use **Ctrl+P** to open files (not mouse!)
- Use `Ctrl+`` for terminal (not separate command prompt!)
- Use **Ctrl+/** to comment code (so fast!)

2. Continue to Tutorial 2: [Tutorials/Tutorial_02_GitHub_Copilot_Hands_On.md](#)

- Learn Git version control (critical for teamwork!)
- Set up GitHub Copilot (AI pair programmer!)
- Build a financial calculator with AI help

3. Start using VS Code daily

- Take notes in Markdown
- Write small Python scripts
- Make it your default code editor

Pro tip from Dad: Don't try to memorize all shortcuts at once! Pick 3-4 favorites and use them daily. After a week, they'll feel natural, then add 3 more. By Month 2, you'll be flying!

🎁 Bonus: Make VS Code YOUR Tool

Recommended Extensions for Finance (Install Later)

Once you're comfortable, add these superpowers:

Ctrl+Shift+X (Open Extensions)

Search and install:

- Python (Microsoft) – Already installed
- Pylance (Microsoft) – Python intelligence
- Jupyter (Microsoft) – Data analysis notebooks
- Excel Viewer (GrapeCity) – View CSV files nicely
- Material Icon Theme – Beautiful file icons
- GitHub Copilot – AI assistant (Tutorial 2!)

Customize Your Theme

Make VS Code look how YOU want:

1. **Ctrl+Shift+P**
2. Type: "Color Theme"
3. Try different themes (arrow keys to preview)
4. Pick your favorite!

Popular choices:

- Dark+ (default) - Professional, easy on eyes
- Light+ - If you prefer bright themes
- Monokai - Classic developer theme
- Dracula - Dark purple theme

Set Auto Save

Never lose work again!

1. File menu → Auto Save (click to enable)
2. Or: File → Preferences → Settings
3. Search: "auto save"
4. Set to: "afterDelay"

Adjust Font Size

Make code comfortable to read:

1. **Ctrl+Shift+P**
2. Type: "Settings"
3. Search: "font size"
4. Change from 14 to your preference (16-18 is popular)

Additional Resources

Official VS Code Docs: <https://code.visualstudio.com/docs> **Keyboard Shortcuts PDF:**

<https://code.visualstudio.com/shortcuts/keyboard-shortcuts-windows.pdf> **Video Tutorials:** Search "VS Code for Beginners" on YouTube

💬 Quick Wins from This Tutorial

You now know how to:

- Navigate VS Code faster than 90% of people
- Use keyboard shortcuts to save 10+ hours/month
- Run Python code without leaving VS Code
- Organize projects professionally
- Customize your development environment

At PE Club, this means you can:

- Build financial models more efficiently
- Collaborate with tech teams fluently
- Automate repetitive analyses
- Look like the "tech-savvy analyst"

🎉 Congratulations! You've mastered VS Code basics!

Ready to level up? → [Tutorials/Tutorial_02_GitHub_Copilot_Hands_On.md](#)

In Tutorial 2, you'll learn to use AI to write code for you. Yes, seriously! 🤖

Estimated completion time: 60 minutes *Difficulty: Beginner* ★☆☆☆☆ *Next: GitHub & Copilot* ★★☆☆☆
Course Progress: 20% Complete

Made with ❤️ by Dad for Mauricio's PE Club success