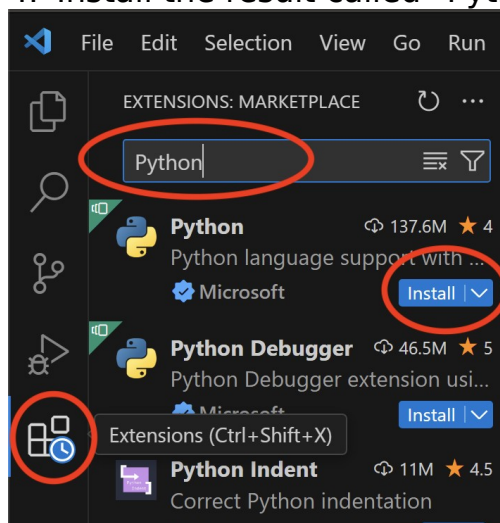


Week 1 – Introduction to Python

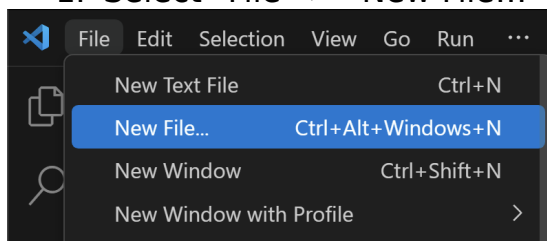
Initial setup

1. Download and Install VS Code: <https://code.visualstudio.com/>
2. Open VS Code and open the Extensions Tab on the left hand side
3. Search for Python
4. Install the result called “Python”

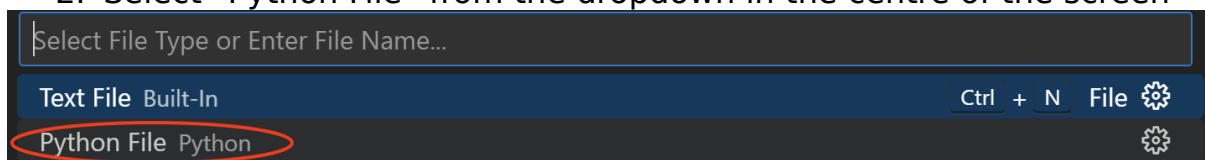


Create a new python file

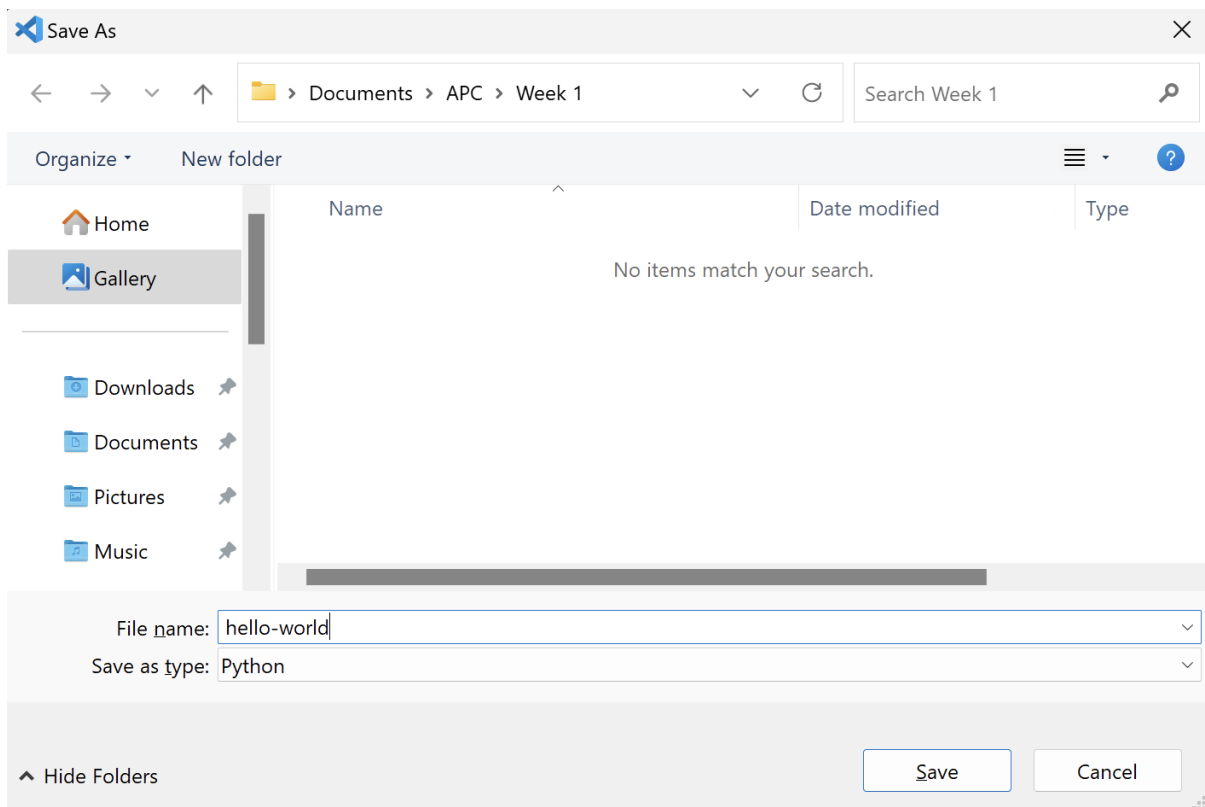
1. Select “File” > “New File...”



2. Select “Python File” from the dropdown in the centre of the screen



3. Select “File” > “Save”
4. Change the file name to “hello-world”
5. Click “Save”



Creating a Python program

We are going to start by creating a simple Hello World program to get you used to the basics before we start creating more complex programs.

1. Write a print function which will simply display whatever is contained inside the (""")
 - a. In your new "hello-world.py" file, type the following code:

```
print("Hello world")
```

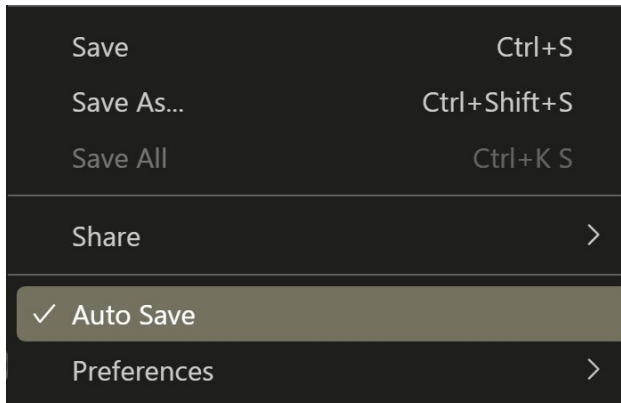
It is important that you use a lowercase 'p' in the word print. Function names should be lowercase, with words separated by underscores as necessary to improve readability.

If you are unsure to how to write something in Python, you can refer to the style guide: <https://peps.python.org/pep-0008> which will have everything you need.

2. Save your file

Top tip!

Turn on “Auto Save” so every time you make a change it will save automatically



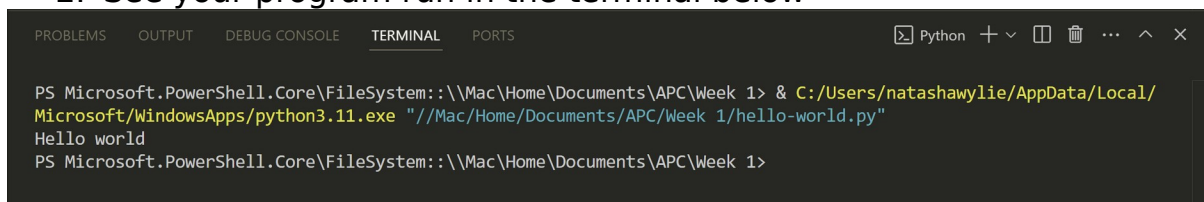
Select “File” > “Auto Save”

Make sure it has a tick next to it

Executing your Python program

To run your program:

1. Click this icon:  on the right of the window
2. See your program run in the terminal below

A screenshot of the VS Code terminal window. The terminal shows the command to run a Python script: `PS Microsoft.PowerShell.Core\FileSystem::\\Mac\Home\Documents\APC\Week 1> & C:/Users/natashawylie/AppData/Local/Microsoft/WindowsApps/python3.11.exe "//Mac/Home/Documents/APC/Week 1/hello-world.py"`. The output of the program is `Hello world`. The terminal title bar indicates it is running Python.

Troubleshooting

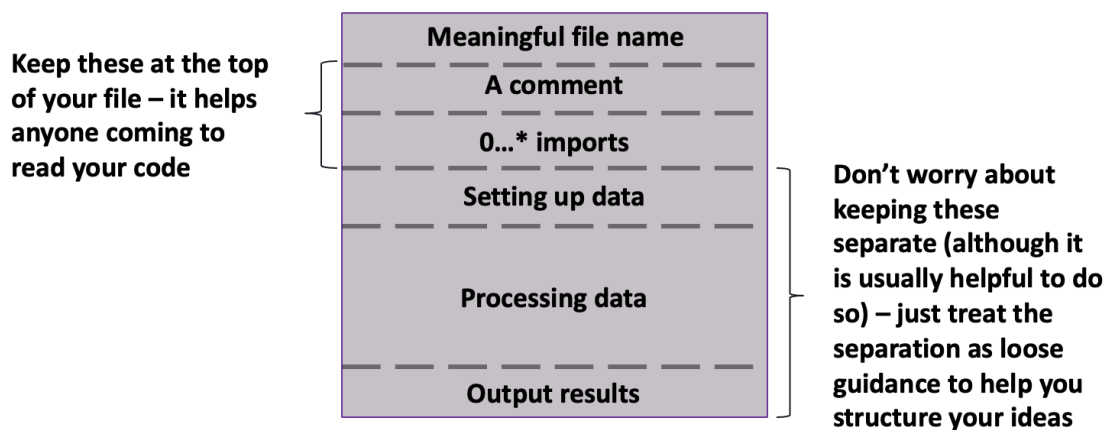
If your program does not run and saying there is no Python Interpreter installed, you may need to either restart VS Code or install Python on your machine via the Microsoft Store and select the Interpreter in VS Code. If you have this issue, please let your tutor know and they can work you through fixing it.

Variables, Data Types & Calculations in Python

Activity 1 – Calculate circle area

In this exercise, you will learn to use the `math` import and perform simple calculations in Python.

As shown in the lecture, we must arrange our code into the following structure:



1. Create a new Python file and call it “calculate-area.py”
2. Type a useful comment so you know what the program is going to do. To turn text into a comment you'll need to pre-fix the text with a `#` symbol, see example below:

```
#Calculate Circle Area
```

3. Import the `math` module, using the keywords `import math` so we can use this in our program.

```
#import standard math module  
import math
```

4. Now we need to do some calculations so we can get the area of our circle. To do this, we need to know the following:
 - circle radius
 - value of pi

To set these values we can create a variable and assign the value to the variable which we can use later in our program.

- 4.1. First value we are going to set is the circle's radius. We will add a variable called *radius* and set it to 30 with the following code: (note: we can also add another comment to explain what our code is going)

```
#Declare variables  
radius = 30
```

- 4.2. Then we need to set the value of pi. We can do this by using the *math* module we imported earlier. We will create a new variable called *pi* and set it to *math.pi*

```
pi = math.pi
```

- 4.3. To check pi has been set correctly, we can use the following to output the value to the console:

```
#Check value of pi is set  
print("The value of pi is", pi)
```

5. Now we are going to calculate the area using the variables we're just created. The formula to calculate the area of a circle is: $A = \pi r^2$

Where:

A = Area of the circle

π (pi) = 3.141592653589793

r = Radius of the circle

In order to do this in code, we need to create a new variable called *area* and then set that to πr^2 , see example below:

```
#Compute area  
area = radius*radius*pi #Formula
```

6. Then we can print the area value so it displays in the console:

```
#Display result  
print("The area of circle is:", area)
```

Activity 2 – Simple Maths (Portfolio task)

Project name: simple_math

Ensure you complete this task and add the source code *simple_math.py* and screenshot of the output to your week 1 folder (see Step 5 below the task)

Step 1: Create a new Python script file

- Create a new file called ***simple_math.py*** in your week 1 folder
- Type the following code into your file

```
pounds = 24
print(pounds)
pounds = 10
print(pounds)
pounds = pounds + 30
print("You have £: " + str(pounds))
```

Step 2: Run your program

- Run the program and make sure that you are getting the output that you are expecting

Step 3: Modify your code

- Change the type of ***pounds*** on the third line to a float (you are not to do this by changing 21 to 21.0 directly – find another way).

Step 4: Run your program

- Run the program again and the output should be different (if it isn't different then ask your tutor to help you to identify the problem)
- Why would the output be different?

Step 5: Take a screenshot of the output

- Take a screenshot of the application window showing the program output
- Paste the screenshot into Paint and save it in your project folder as ***simple_math.jpg***.

Activity 3 – Breakeven (Portfolio task)

Project name: breakeven

Ensure you complete this task and add the source code breakeven.py and screenshot of the output to your week 1 folder (see Step 5 below the task)

The break-even point for a business is defined as the point where the total revenue equals the business expenses (fixed costs). The following values are required to calculate this:

- What is the cost to produce each item
- What is the sale price per item (item cost + profit per item)
- What are the fixed costs

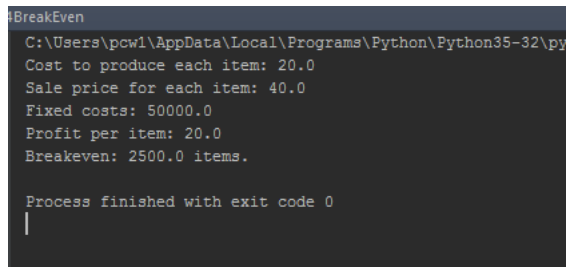
Dividing the fixed costs by the profit per item (the difference between the cost to produce each item and the sale price) gives the number of items that must be sold before "breaking even".

For example, if it costs £50.00 to produce an item which is then sold at £100.00, there is £50.00 profit per item. If the fixed costs are £1000.00, the company would need to sell 20 items to break even.

Exercise data:

- Cost to produce each item = £20.00
- The sale price per item (item cost + profit per item) = £40.00
- Fixed costs = £50000.00

Write a program to output the data with a meaningful label for each item and calculate and output the number of items we need to sell to breakeven. The output should look similar to the following (cropped for convenience).



```
BreakEven
C:\Users\powl\AppData\Local\Programs\Python\Python35-32\py
Cost to produce each item: 20.0
Sale price for each item: 40.0
Fixed costs: 50000.0
Profit per item: 20.0
Breakeven: 2500.0 items.

Process finished with exit code 0
|
```

Hint:

$$break_even = \frac{fixed_costs}{sale_price - item_cost}$$

Step 1: Analyse the problem

- Make sure that you have read through, and understand, the problem
- Decide the type of data being used

Step 2: Create a Python script file

- Create a new file called **breakeven.py**
- Make sure that you put it in your **week 1** folder

Step 3: Write source code

- Using your analysis from Step 1, write code to produce the required outputs
- Ensure that you use appropriate names for all your variables

Step 4: Run and test your application

- Run your application with the example data to ensure that you get the correct answer
- Identify and correct any errors in your source code if you don't get the output that you were expecting

Step 5: Take a screenshot of the output

- Press [ALT + Print Screen] for your output screen
- Paste your screenshot into Paint and save as **breakeven.jpg**

Activity 4: Splitting the bill (Portfolio task)

Project name: split_bill

Ensure you complete this task and add the source code split_bill.py and screenshot of the output to your week 1 folder (see Step 5 below the task)

Three friends Holly, Jack and Mike have been out for a curry to celebrate Mike's birthday. They all agree to split the bill equally, however they don't have any change. They decide to add a tip to bring their share up to the nearest pound.

The total bill came to £84.87.

In your program you should output the following (*must use formulas to calculate and outputs must have correct use of currency symbol and suitable label*):

- Amount each friend owes (as a float)
- Amount extra each friend needs to make up their share to the nearest pound (as a float)
- Amount each friend needs to pay including tip (as an integer)
- Total amount paid (as a float)
- Total amount tip paid (as a float)

Step 1: Analyse the problem

- Make sure that you have read through, and understand, the problem
- Decide the type of data being used

Step 2: Create a Python script file

- Create a new file called **split_bill.py**
- Make sure that you put it in your **week 1** folder

Step 3: Write source code

- Using your analysis from Step 1, write code to produce the required outputs
- Ensure that you use appropriate names for all your variables

Step 4: Run and test your application

- Run your application with the example data to ensure that you get the correct answer
- Identify and correct any errors in your source code if you don't get the output that you were expecting

Step 5: Take a screenshot of the output

- Press [ALT + Print Screen] for your output screen
- Paste your screenshot into Paint and save as **split_bill.jpg**

Activity 5: Introducing turtle

In this activity we're going to use the turtle library which is built into the Python environment we set up earlier.

To start using turtle we need to start our Python environment. Follow the steps below to get started:

Step 1: Open a terminal window

- Select Terminal > New Terminal from the top navigation
- A panel will open at the bottom of your screen
- Type **python3** and press enter to run the script and open a new Python environment

```
PS Microsoft.PowerShell.Core\FileSystem::\\Mac\Home\Documents\APC\Week 1> python3
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr  2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Step 2: Import turtle

- In the open terminal window, now enter **import turtle** and press enter

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS python3.11 + ▢ 🗑️ ⋮ ^ ✕

PS Microsoft.PowerShell.Core\FileSystem::\\Mac\Home\Documents\APC\Week 1> python3
Python 3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import turtle
>>> 
```

Step 3: Open turtle screen

- In the open terminal window, enter **`s = turtle.getscreen()`**
- The turtle screen will launch, and you'll see something like this:



Activity 6: Controlling turtle (Portfolio task)

Project name: turtle

Ensure you complete this task and add screenshots of the terminal window showing all your commands and screenshots of the turtle screen to your week 1 folder (see Step 5 below the task)

Using the lecture slides to help you, enter the commands which will get turtle to do the following actions:

Part 1: Controlling the turtle screen:

- Set the screen size to **0.5, 0.75**
- Set the background colour to **1, 1, 0**
- Set the title to **Activity Program**

Take a screenshot of the terminal window:

- Press [ALT + Print Screen] for your output screen
- Paste your screenshot into Paint and save as **part1_terminal.jpg**

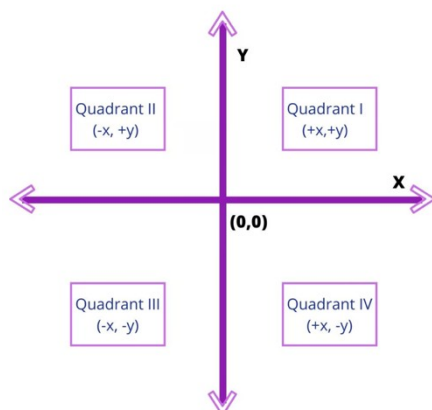
Take a screenshot of the turtle screen:

- Press [ALT + Print Screen] for your output screen
- Paste your screenshot into Paint and save as **part1_turtle.jpg**

Part 2: Controlling the turtle

- Initialise a turtle variable as t: **t = turtle.Turtle()**
- Move the turtle pointer from origin to 3rd quadrant
- Change pointer colour to blue
- Change the size of turtle
- Move the turtle pointer from the 3rd quadrant to the 1st quadrant (you may take any path you wish)

Hint:



Take a screenshot of the terminal window:

- Press [ALT + Print Screen] for your output screen
- Paste your screenshot into Paint and save as ***part2_terminal.jpg***

Take a screenshot of the turtle screen:

- Press [ALT + Print Screen] for your output screen
- Paste your screenshot into Paint and save as ***part2_turtle.jpg***

Appendix

Activity 1 – Calculate circle area

Complete code example:

```
#Calculate Circle Area
#import standard math module
import math

#Declare variables
radius = 30
pi = math.pi

#Check value of pi is set
print("The value of pi is", pi)

#Compute area
area = radius*radius*pi #Formula

#Display result
print("The area of circle is:", area)
```