Python

Basics - Tooling, Functions, and Variables

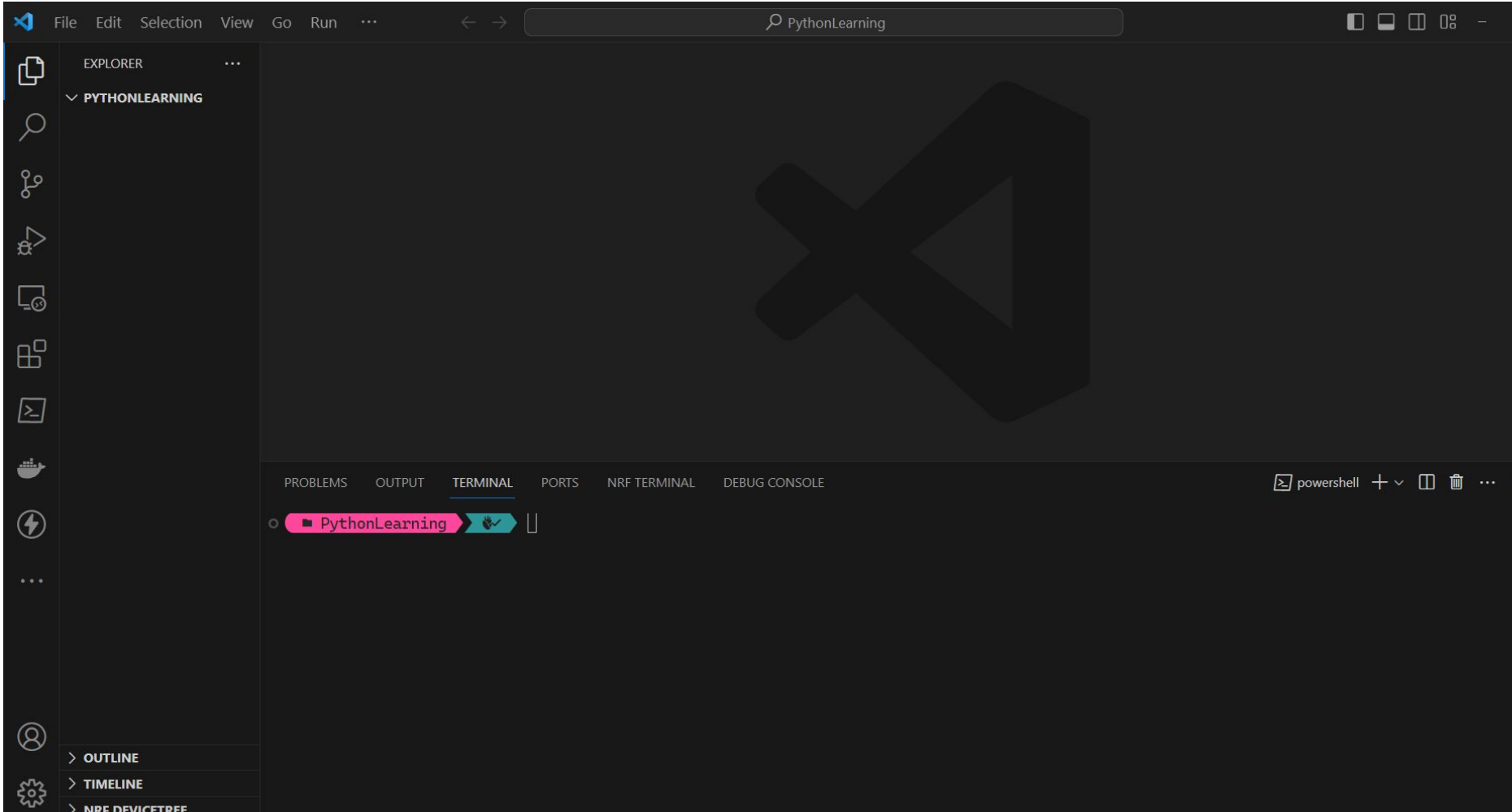
Requirements

Python installed - python interpreter

- Install to path to be able to use “python” keyword in system

VS Code - IDE

- Just a text editor with powerful features


File
Edit
Selection
View
Go
Run
PythonLearning
EXPLORER
PYTHONLEARNING
PROBLEMS
OUTPUT
TERMINAL
PORTS
NRF TERMINAL
DEBUG CONSOLE
powershell
PythonLearning
OUTLINE
TIMELINE

Writing first python code

- Write in a file with extension “.py”
  - Example “hello.py”
- Use print()
-
- ```
  print("Hello, world!")
  ```
- Run code with
  - “python hello.py” - in terminal
  ```

Functions

- “print()” is a function
  - Actions python knows how to perform
  - Print takes arguments - print(“these are arguments”)
- Lets make some bugs
  - print(“hello, world”
  - print(“hello” * “tim”)

Expanding first program

- Try “input()” function
- “input()” takes a prompt as the argument
- input(“What is your name”)

- Input returns the string of what the user types
  - String is text, characters

Variables

- Variable is container for a value
-
```
name = input("What is your name? ")
```
- = means assignment
  - Assign whats is on the right side of = to what is on the left

```
name = input("What's your name? ")
print("hello,")
print(name)
```

- Do not need to specify data type, - Dynamically typed language

Comments

- In python comment lines need that begin with a “#”
  - This tells python to ignore execution of these lines

```
# Ask user for name
name = input("What's your name? ")
print("hello,")
print(name)
```

- Can format string in different ways
  - `print("hello," + name)`
  - `print(f"hello, {name}")`

Other functions

- len()
  - Read the docs!

https://docs.python.org/3/library/functions.html

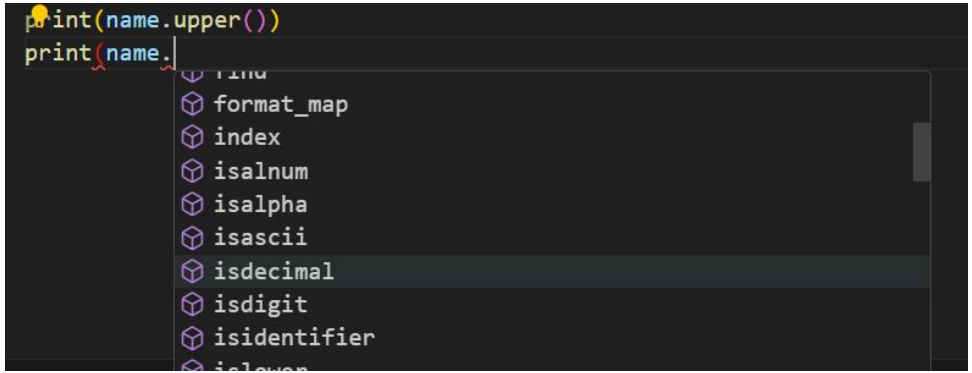
- Can change other parameters of print(“hello”, end=””)
  - Take out new line ending

```
print(*objects, sep=' ', end='\n', file=None, flush=False)
```

- **CHALLENGE PRINT OUT THE LENS OF THE USERS INPUT STRING IN YOUR PROGRAM**

Methods

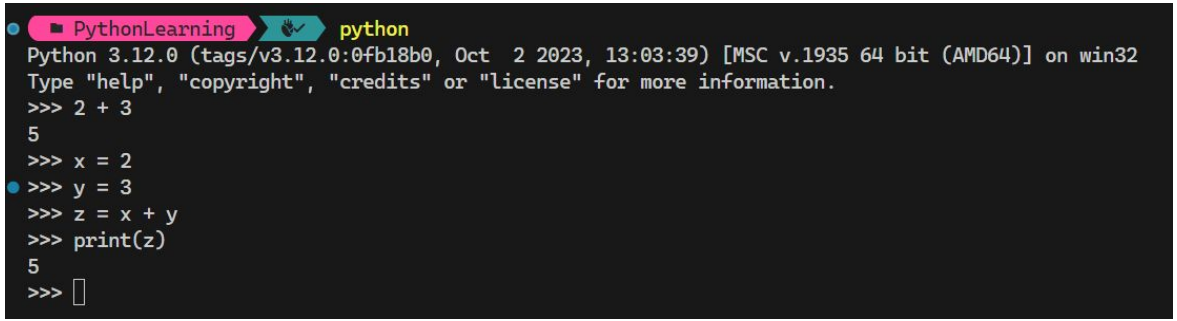
- Strings have some built-in methods that are used with the . (dot) operator
- https://docs.python.org/3/library/stdtypes.html#string-methods
- vs -code can autocomplete
- Can also string together methods
- `print(name.strip().upper())`



- **CHALLENGE Remove the white space around the name input and make lower case**

Integers

- Can test in interactive live python terminal
  - Type “python” in terminal then can write and run python live
- Integers or ints
  - Whole numbers
  - X = 2
  - Y = 3
  - Z = X + Y
- Try
  - `# Ask user for age`
  - `age = input("What's your age? ")`
  - `print(age)`



Ints with user input

- Now try:

```
# Ask user for age
age = input("What's your age? ")
modifier = input("What to add to age? ")

new_age = age + modifier
print(new_age)
```

- Does this work? What is output?

Need to cast to integer

- Cast means convert the data type
  - You can do this in python the surrounding with desired data type
    - Example int(“42”) - will turn the string “42” into an int 42 that you can do arithmetic on

- **`CHALLENGE: Fix the age calculator to correct the issue`**

- Next try the calculator with any number with a decimal point
  - What happens?

Floats

- Float is a data type of number with a decimal in it.
  - 2.5
  - 0.666666666
  - Can cast to float with float(x)
- Can format floats or round with built in “round()” function

Operators in python

Operator	Operation	Example	Evaluates to . . .
`**`	Exponent	`2 ** 3`	`8`
`%`	Modulus/remainder	`22 % 8`	`6`
`//`	Integer division/floored quotient	`22 // 8`	`2`
`/`	Division	`22 / 8`	`2.75`
`*`	Multiplication	`3 * 5`	`15`
`-`	Subtraction	`5 - 2`	`3`
`+`	Addition	`2 + 2`	`4`

Math Operators from Highest to Lowest Precedence

Sweigart, A. Automate the Boring Stuff with Python.
https://automatetheboringstuff.com/2e/chapter1/

Making Functions

- Creating functions with “def” keyword

```
def hello():
    print("hello")
```

- Indentation matters in python
- Can add parameter

```
def hello(name):
    print("hello,", name)

name = input("What's your name? ")
hello(name)
```

- Default parameter can be set

```
def hello(name="world"):
```

Order Matters

- The function needs to be defined before you call it

```
hello()

def hello(name="world"):
    print("hello,", name)

name = input("What's your name? ")
hello(name)
```

```
Traceback (most recent call last):
  File "C:\Users\ttrut\OneDrive\Documents\Dev\PythonLearning\functions.py", line 1, in <module>
    hello()
    ^^^^^
NameError: name 'hello' is not defined. Did you mean: 'help'?
```

- This can be annoying for clean code and ordering
  - Can use a main() function to hold the main running program that can help us make organized and readable code

Main

```
def main():

    # Output using our own function
    user_input = input("What's your name? ")
    hello(user_input)

    # Output without passing the expected arguments
    hello()


# Create our own function
def hello(name="world"):
    print("hello,", name)
```

- Remember to call main!

Function Return Values

- You probably want data out of some functions
- Return will do that

```
def add_ten(n):
    return n + 10
```

- **CHALLENGE: Make a function that squares a value and returns it**

Summary

- Setting up and running python and vs-code
- Functions
- Bugs
- Comments
- Variables
- Data Types
  - Strings
  - Ints
  - Floats
- Making functions
- Return values

Exercises:

1. Make a program that converts F to C
    a. Ask user for temp in F and output the conversion to C
2. Make a tip calculator program
    a. Asks user for two inputs
        i. Total Bill amount
        ii. Percent to tip
    b. Output the tip calculation