Computer Programming Lab 2

2024/03/05 Tim Chen

Contents

Package Managers 4

Python Environment - `pip`, `pyenv` and `virtualenv` 6

Running Your First Python Program 14

Variables and Basic Data Types 17

Declare v.s. Define 21

The `input` Function 24

Simple BMI Calculator 26

Assignment 1 30

Appendix 35

Before writing in the Python programming language, you need to understand the environment first; just like you need to know the race track before you start racing.

Package Managers

Package Managers

MacOS

- Homebrew
- MacPorts

Windows

- WinGet
- Scoop
- Chocolatey

Python Environment - `pip`, `pyenv` and `virtualenv`

`pip` - Python Package Manager

Functionality:

- Installs and manages software packages found in the Python Package Index (PyPI).
- Handles package upgrades, installation of specific package versions, and requirements files.

`pip` - Python Package Manager (cont.)

Example:

- Installing a package: `pip install package_name`
- Upgrading a package: `pip install --upgrade package_name`
- Installing a specific version

```
pip install package_name==version_number
```

`pyenv` - Python Version Manager

Functionality:

- Enables the installation and management of multiple Python versions.
- Supports switching between Python versions on a global, per-user, or per-project basis.
- Facilitates specifying Python version through a `.python-version` file in project directories.

pyenv - Python Version Manager (cont.)

Example:

- List available versions: `pyenv install --list`
- Install a specifc version of python: `pyenv install version_number`
- Setting global version: `pyenv global version_number`
- Setting local version for project: `pyenv local version_number`

`virtualenv` - Python Environment Manager

Functionality:

- Allows for the creation of isolated environments, each with its own Python interpreter and set of libraries.
- Prevents different projects from having conflicting dependencies.
- Supports creating environments for any Python version, independent of the Python version used to create the environment.

`virtualenv` - Python Environment Manager (cont.)

Example:

- Creating a virtual environment: `virtualenv env_name`
- Activating a virtual environment:
  - Windows: `.\env_name\Scripts\activate`
  - UNIX or MacOS: `source env_name/bin/activate`
- Deactivating a virtual environment: `deactivate`

Reference:

- Stack Overflow: Difference between pyenv, virtualenv and Anaconda
- Python Packaging User Guide
- pip
- pyenv
- virtualenv
- Anaconda
- Miniconda

Running Your First Python Program

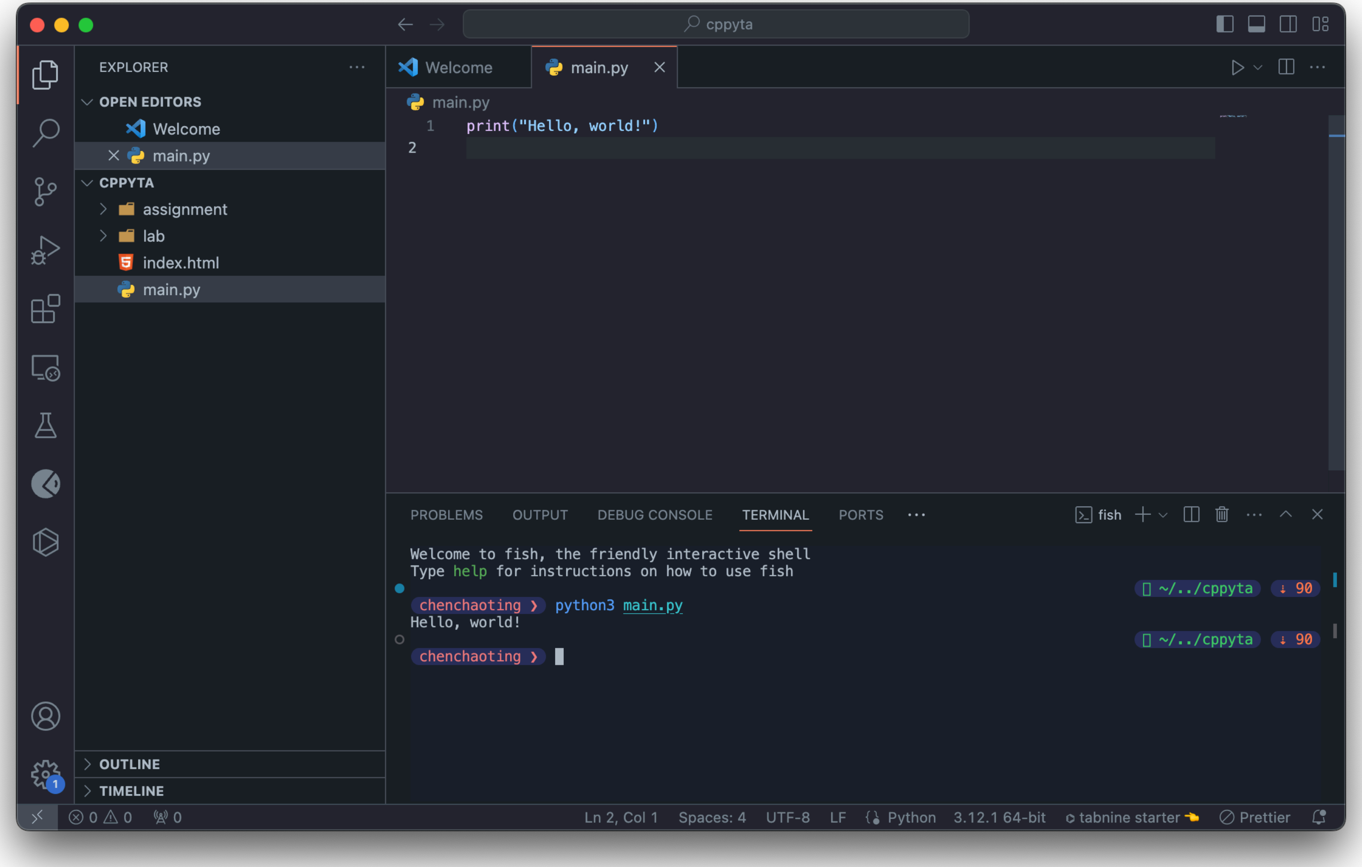
Running Your First Python Program

1. Open VSCode and open the target folder.
2. Create a new `main.py` file and edit.

    ```python
    print("Hello, world!")
    ```

3. Press ``Ctrl + ` `` to open new terminal.
4. Type `python main.py` or `python3 main.py` on MacOS.

Running Your First Python Program



```python
print("Hello, world!")
```

```
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
                                                          ~/../cppyta  ↓ 90
chenchaoting ❯ python3 main.py
Hello, world!
                                                          ~/../cppyta  ↓ 90
chenchaoting ❯
```

Variables and Basic Data Types

Variables

- Python has no command for declaring a variable.
- A variable is created the moment you first assign a value to it.
- e.g.

```
height = 1.65 # 165 cm
weight = 52   # 52 kg
```

Basic Data Types

- Numbers
  - Integer (int): Whole numbers. e.g. 5, -3.
  - Floating Point (float): Numbers with a decimal point. e.g. 2.9
- Text Type:
  - String (str): Text or characters. e.g. “Hello”, ‘World’.
- Boolean:
  - Boolean (bool): Represents True or False.
- Collection data types includes list(`list`), tuple(`tuple`), Dictionary(`dict`), Set(`set`) …

Basic Data Types (cont.)

Dynamic Typing

```
x = 5       # Here `x` is a integer
x = "Hello" # Here `x` has changed to a `str`
```

Type Conversion

```
x = 10        # Type `int`
x = float(10) # Change `x` in to type float
```

Notes: `print(type(variable))` can print the given variable’s type.

Declare v.s. Define

Declare v.s. Define

Declare

Definition: To announce its existence and type to the compiler or interpreter.

- Notes: Python has no command for declaring a variable.

```c
/* C Programming Language */
int age; // Declare a variable named age of type int
```

Declare v.s. Define

Define

Definition: To provide the actual implementation or value of something.

```python
# Example in Python
age = 25 # Define a variable named age with value of 25
```

The `input` Function

The input Function

Syntax: `input(prompt)`

Notes: The returned value is always a string i.e. `str`, even if the user enters number.

```
height = input("Height: (m) ") # Ask for user's height

height = float(height) # Convert the input to type `float`

print(f"Your height is {height}m") # Print the height
```

Simple BMI Calculator

Simple BMI Calculator

- First, get the input of the user. Please note that the input will be read as a `str`.

```python
height = input("Height: (m) ")
weight = input("Weight: (kg) ")
```

Simple BMI Calculator (cont.)

- Second, turn the input `str` to `float`.

```
height = float(height)
weight = float(weight)
```

- Third, calculate the BMI and round to one decimal.

```
bmi = round(weight / (height * height), 1)
```

Simple BMI Calculator (cont.)

- Last, print out the result according to the BMI.

```python
if bmi < 18.5: # Conditional Statements
  note = 'underweight!!!'
elif bmi >= 18.5 and bmi <= 25:
  note = 'good!!!'
else:
  note = 'tooooo heavy!!!'
print(f'Your BMI: {bmi}, {note}')
```

Assignment 1

Assignment 1

Description: Write a program that converts temperature from Fahrenheit to Celsius and vice versa based on the user’s choice.

Introductions: Perform the conversion using the appropriate formula.

- Celsius to Fahrenheit: $F = \left(\frac{9}{5}\right) * C + 32$
- Fahrenheit to Celsius: $C = \left(\frac{5}{9}\right) * (F - 32)$

Assignment 1 (cont.)

Input Format: A single float number to convert and a character to indicate whether the previous float number is Fahrenheit or Celsius.

Output Format:

$$\begin{cases} \text{"Invalid input"} \leftarrow \text{if the character is neither 'C' nor 'F'} \\ \text{"The temperature is \{temperature\} \{C or F\}"} \leftarrow \text{else} \end{cases}$$

Assignment 1 (cont.)

Notes:

- The second character will be valid only if it's either 'C' or 'F'.
- If the second character isn't 'C' nor 'F', it's invalid.
- If the given float number is Fahrenheit, please convert it to Celsius and vice versa based on the given character.
- Please round the output number to **1 decimal point**.
- The input function **do not** need a prompt. Please use it without prompt. e.g. `height = input()`

Assignment 1 (cont.)

Input Sample 1

```
32
C
```

Output Sample 1

```
The temperature is 89.6 F
```

Input Sample 2

```
54
A
```

Output Sample 2

```
Invalid input
```

Appendix

Text Editor

GUI-based

- Visual Studio Code - Smart, fast, customizable
- Zed - Code at the speed of thought

CLI-based

- Neovim - Hyperextensible Vim-based text editor
- Helix - A post-modern text editor

Contact Me

- My GitHub - https://github.com/gnitoahc/
- My personal email - chaotingchen10@gmail.com
- My website - https://chaoting.xyz

Thank you