COMP1811 Lab Sheet 1.0.a
Working with PyCharm IDE – Installation and Quick Start Guide

Aim

Today's lab is designed to get you familiar with using the PyCharm IDE (Integrated Development Environment for Python) and to start to write some simple Python commands. You will need PyCharm to implement the coding exercises given to you in the weekly lab sheets and you will need it to write the code needed for Python coursework project. It is important, therefore, that you make sure PyCharm is installed and working properly on your home machine during today's lab session.

The tasks walk you through how to check PyCharm is already installed on your machine, and how to install it, if it is not already installed. Its key features that you'll need to write and run Python code are then introduced. Appendix A, walks you through how to install PyCharm at home. Please make sure you install PyCharm on your home machine before next week.

Tasks

Complete this part of the lab tasks individually. This should take you 40-50 minutes to complete.

1. Installation

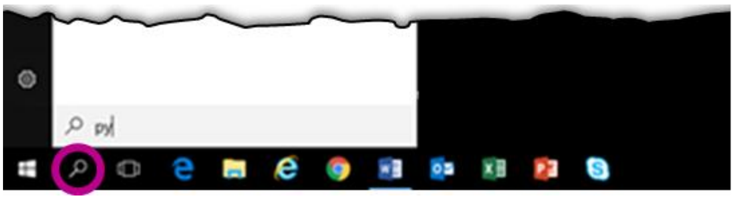
PyCharm is an editor to help you easily write and run Python code. It is an open-source and widely used IDE (Integrated Development Environment) created by JetBrains for both beginners and professional developers and includes features like code completion, project file management, debugging, refactoring, and much more. You will become familiar with these as we continue to use the IDE on this module. For PyCharm to work, it needs to use a Python Interpreter. The Python Interpreter is a piece of software that takes the code your write and converts it into code that the computer hardware can understand. Your code will not run without this software.

PyCharm and the Python Interpreter should be installed on all the machines in the ITLabs at Greenwich. If they are not available on the machine you use on campus, install it through the "Windows Software Center". The following are instructions on how to install PyCharm from the Software Center.

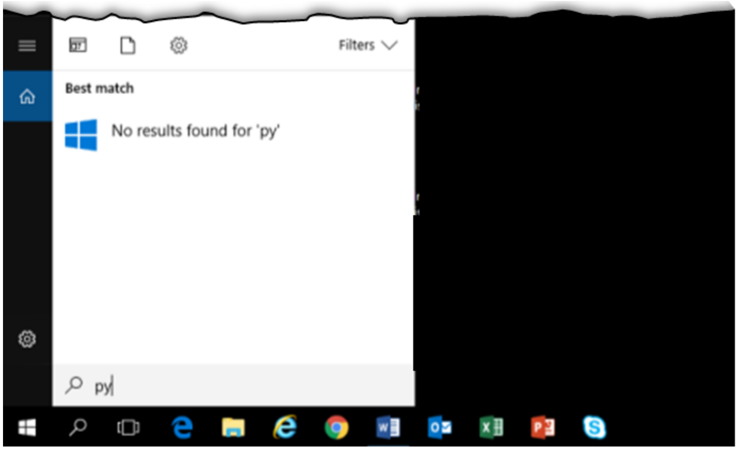
Step 1: Make sure PyCharm is installed on your machine

a. Check PyCharm is installed

Click the Search Windows icon ⌕ at the bottom left of your screen and type in "pycharm".



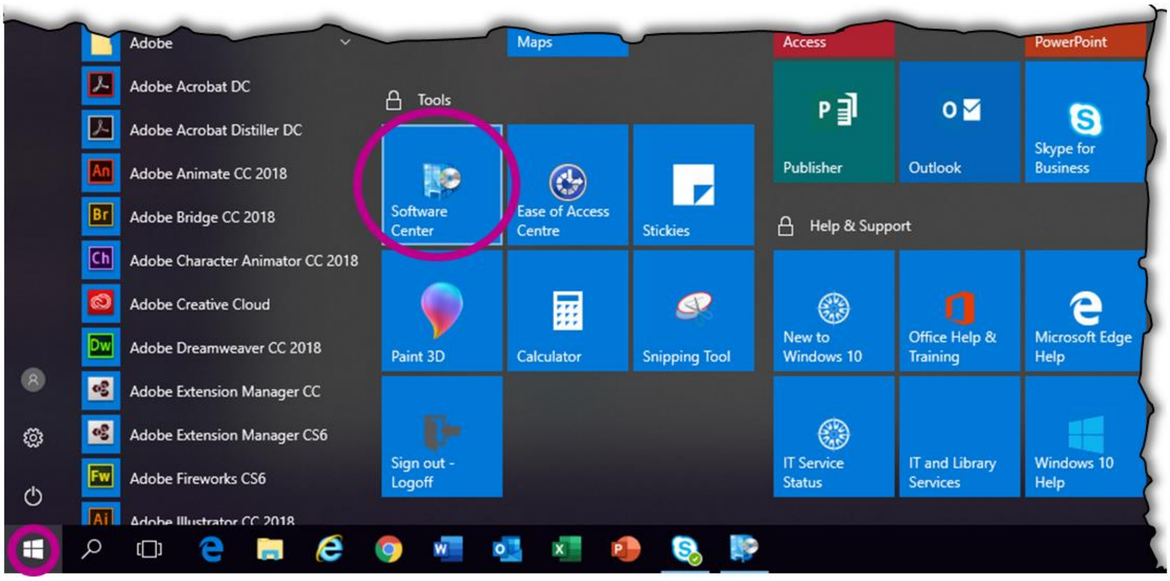
PyCharm is installed if it appears in the search results list. The following screenshot shows PyCharm is not installed. Jump to the **Quick Start** section, if it is installed.



No results found for 'py'

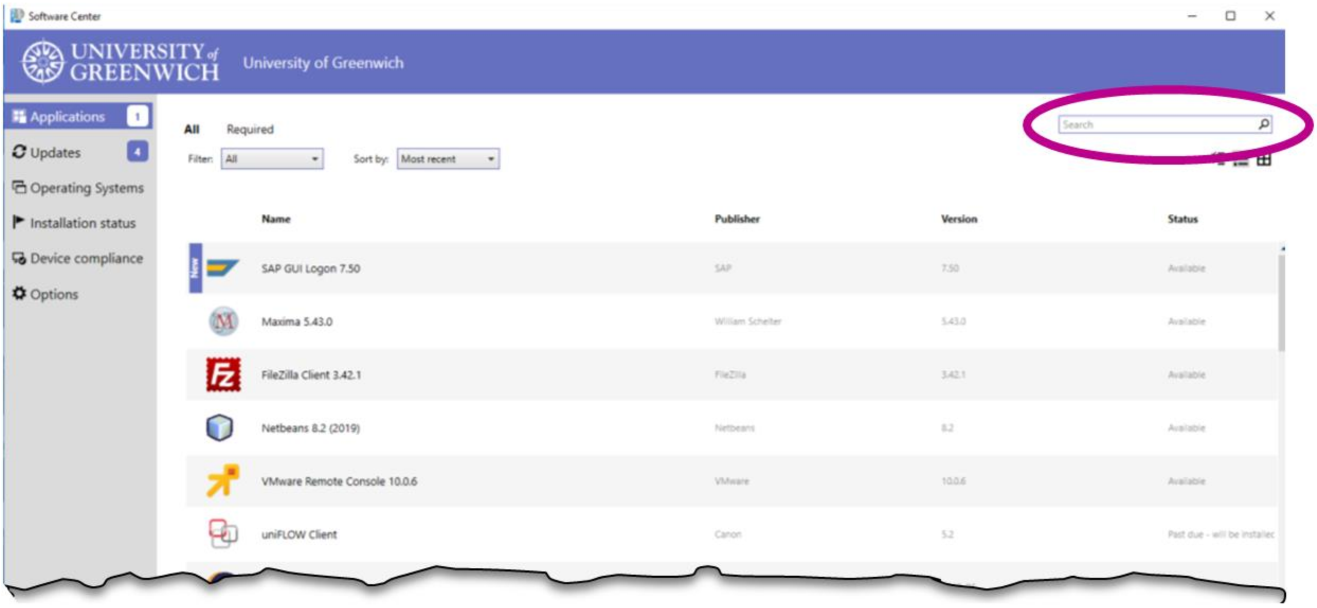
b. Launch Software Center

PyCharm can be installed from the Software Center. Click the Windows icon then click on Software Center.



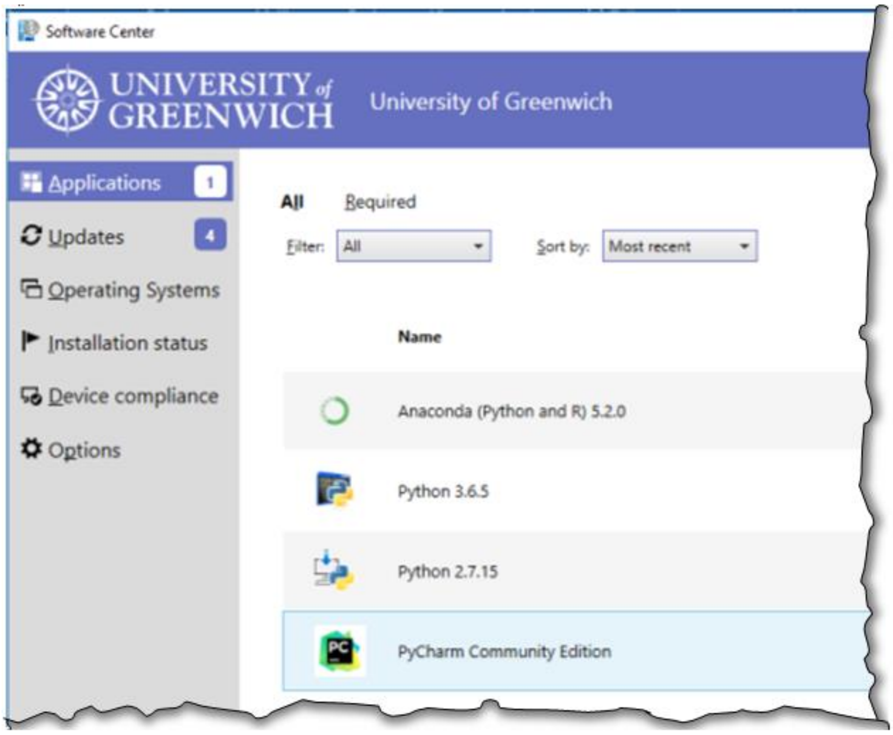
c. Install PyCharm

Type "PyCharm" in the search field.

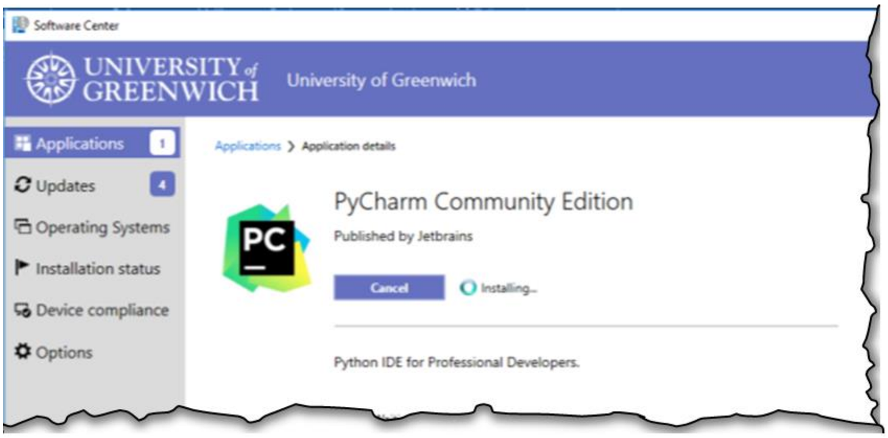


Name	Publisher	Version	Status
SAP GUI Logon 7.50	SAP	7.50	Available
Maxima 5.43.0	William Schelter	5.43.0	Available
FileZilla Client 3.42.1	FileZilla	3.42.1	Available
Netbeans 8.2 (2019)	Netbeans	8.2	Available
VMware Remote Console 10.0.6	VMware	10.0.6	Available
uniFLOW Client	Canon	5.2	Past due - will be installec

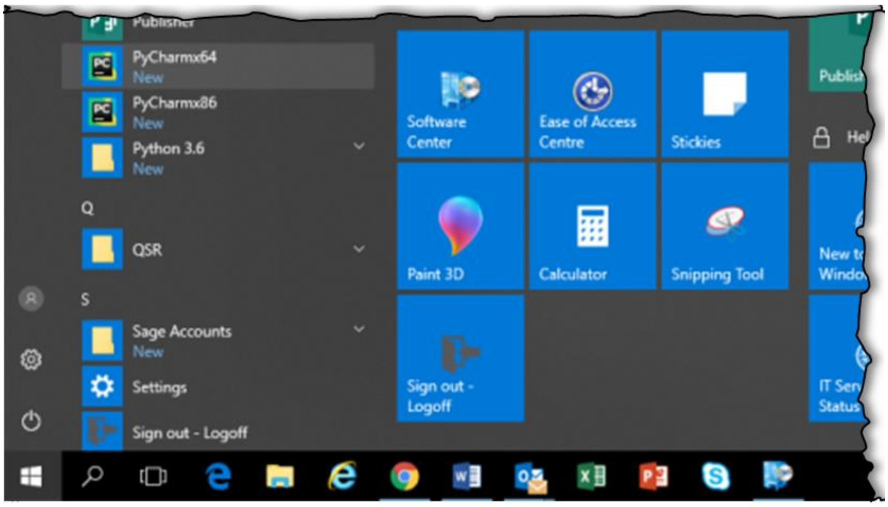
Double click PyCharm Community Edition on the results list as in the following screenshot, then click install on the prompted screen.



The following should now be displayed.



Once installed, check that it appears on your Windows Start menu.

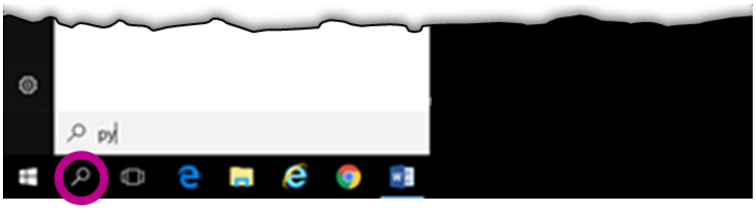


Step 2: Make sure the Python interpreter is installed.

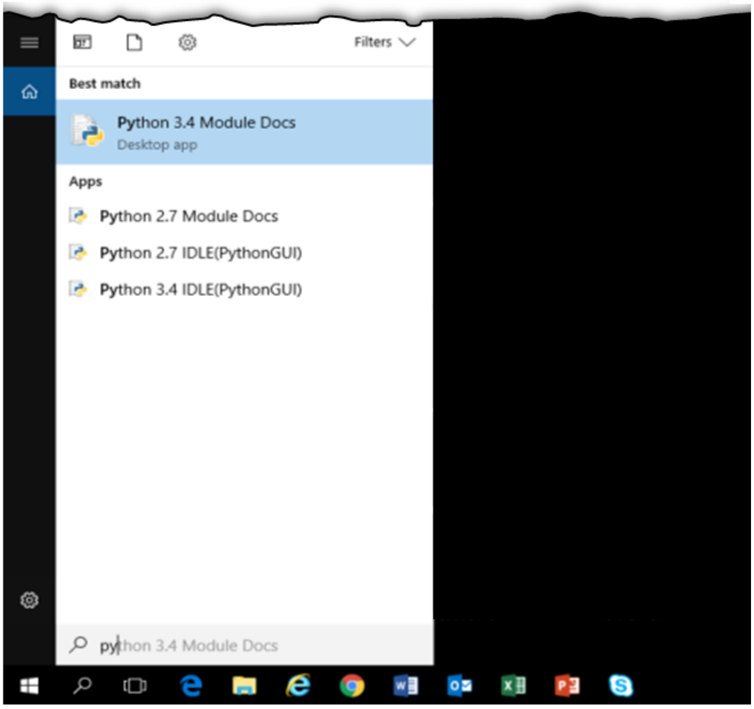
For PyCharm to work, Python (the Python interpreter) must be installed too. It comes with the PyCharm installer and should be installed on all the machines in the ITLabs at Greenwich, but let's check to be sure. We use Python 3.9.* on this module (Python 3.8.* works too). Python version 2.* is also installed on the lab machines. **DO NOT use Python 2** on the campus machines *or at home*. Python 2 uses a slightly different syntax to Python 3 and will hence cause many problems that can be avoided.

a. Check Python 3 is installed

Click the Search Windows icon ⌕ and type in "python".



Python is installed, if it appears in the search results list.



b. Install Python 3

If Python is not on the results list, install it in the same way you installed PyCharm (ie. via the Software Centre). The screenshot above shows Python is installed.

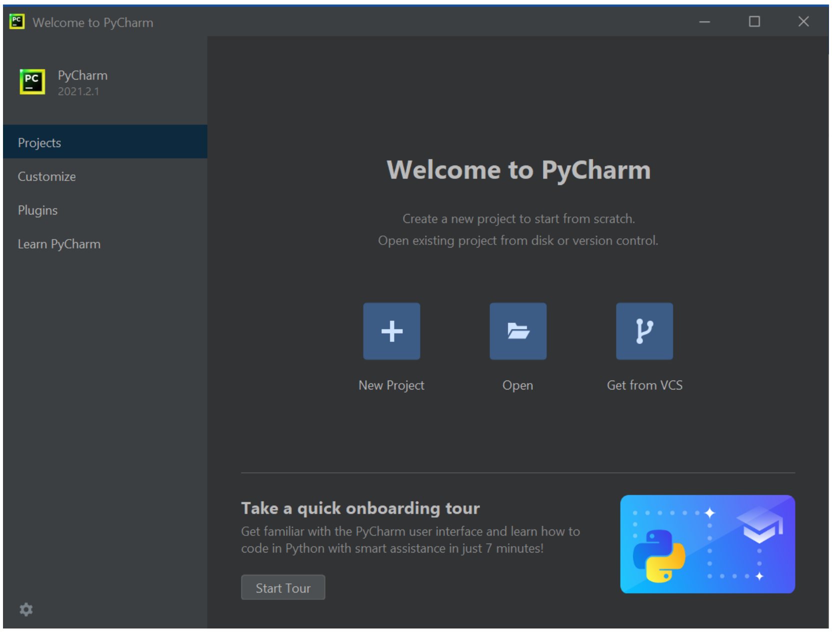
You're now all set to start using the PyCharm IDE.

2. Quick Start - Create a PyCharm project

You’re now all set to start using the PyCharm IDE.

a. Click Create New Project from the welcome screen

The following Welcome is prompted after the IDE has been launched. You see the Welcome screen when you launch PyCharm for the very first time, or when there are no open projects. It gives you the main entry points into the IDE: creating or opening a project,



Welcome to PyCharm

PyCharm 2021.2.1

Welcome to PyCharm

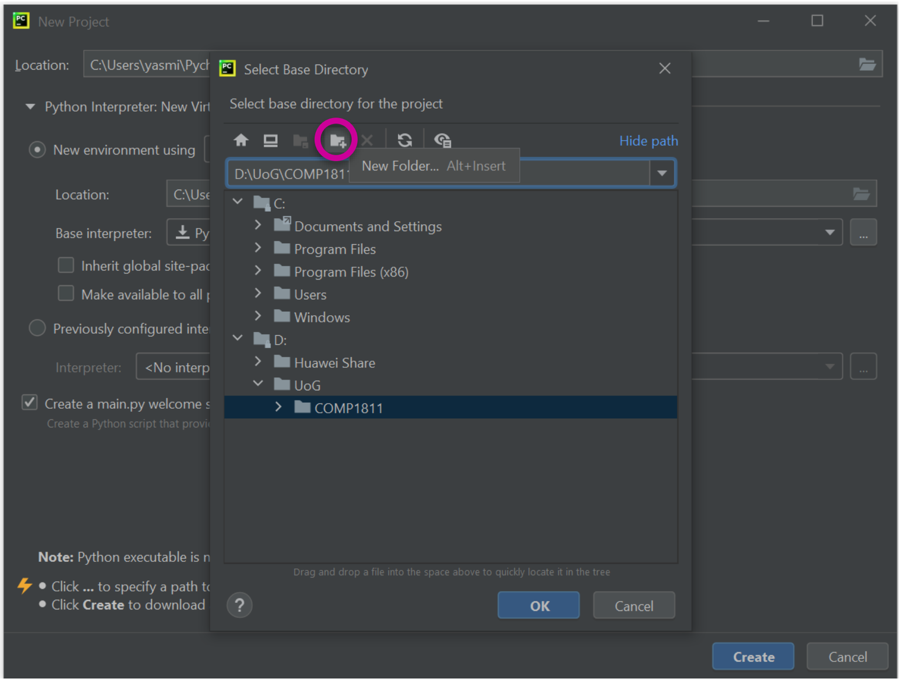
Create a new project to start from scratch.
Open existing project from disk or version control.

Take a quick onboarding tour

Get familiar with the PyCharm user interface and learn how to code in Python with smart assistance in just 7 minutes!

b. Name your project with a meaningful name that is indicative of the function within

You are prompted to store your project at a default location. To better organise your work, consider creating a separate folder for COMP1811 and in that folder create a directory tree (sub-folders) as follows: \LabWork\Lab1\. When working on one of the University machines on campus or are using remote desktop (virtual machine via VMWare), it is recommended that you use you University G drive to save your projects instead. This way you’ll always have access to your work wherever you are, even at home if you use a remote desktop.



New Project

Location: C:\Users\yasmi\Pych

Python Interpreter: New Virt

○ New environment using

Location: C:\Use

Base interpreter: Py

□ Inherit global site-pac

□ Make available to all p

○ Previously configured inte

Interpreter: <No interp

☑ Create a main.py welcome s

Create a Python script that provi

Note: Python executable is n

- Click **...** to specify a path to
- Click **Create** to download

Select Base Directory

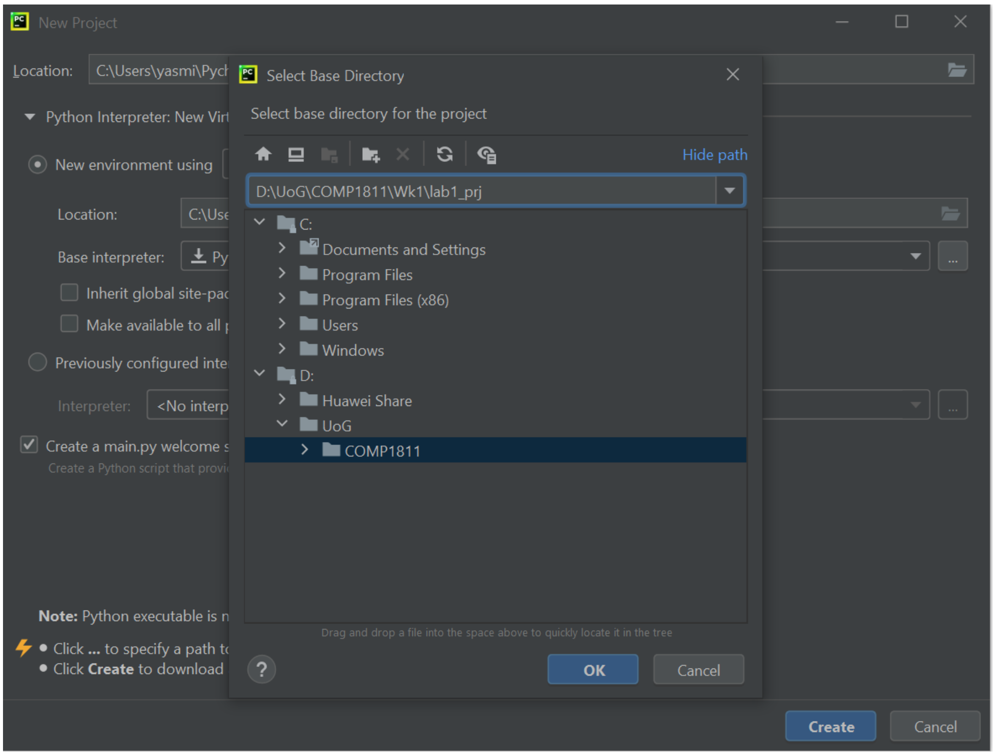
Select base directory for the project

D:\UoG\COMP1811

New Folder... Alt+Insert

Drag and drop a file into the space above to quickly locate it in the tree

Give a meaningful name to your project. "lab1_prj" is used here, which is not really meaningful! You can be more creative.



New Project

Location: C:\Users\yasmi\Pych

Select Base Directory

Select base directory for the project

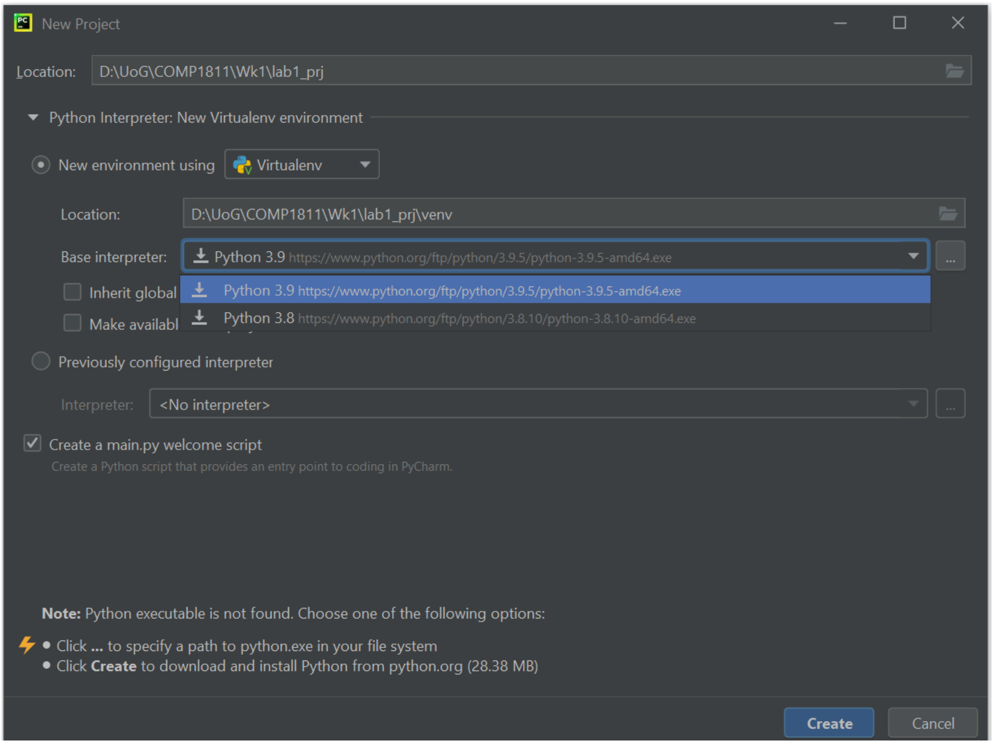
D:\UoG\COMP1811\Wk1\lab1_prj

Drag and drop a file into the space above to quickly locate it in the tree

Note: Python executable is n

- Click ... to specify a path to
- Click **Create** to download

Make sure the path to the Python interpreter is correct and shows in the **base interpreter box**. If nothing appears in the highlighted box, search for the location of the python-3.9.*-amd64.exe (or equivalent) on your local machine and enter it in the highlighted base interpreter box by clicking on the 3 horizontal dots to the right of the box.



New Project

Location: D:\UoG\COMP1811\Wk1\lab1_prj

Python Interpreter: New Virtualenv environment

● New environment using Virtualenv

Location: D:\UoG\COMP1811\Wk1\lab1_prj\venv

Base interpreter: Python 3.9 https://www.python.org/ftp/python/3.9.5/python-3.9.5-amd64.exe

- Python 3.9 https://www.python.org/ftp/python/3.9.5/python-3.9.5-amd64.exe
- Python 3.8 https://www.python.org/ftp/python/3.8.10/python-3.8.10-amd64.exe

□ Inherit global

□ Make availabl

○ Previously configured interpreter

Interpreter: <No interpreter>

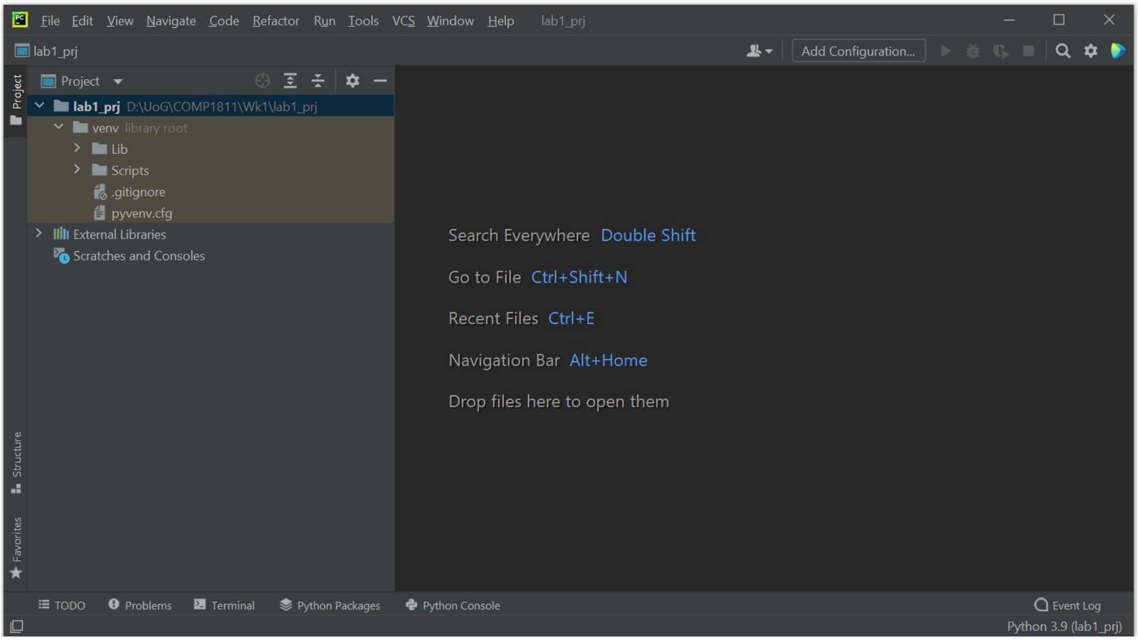
☑ Create a main.py welcome script

Create a Python script that provides an entry point to coding in PyCharm.

Note: Python executable is not found. Choose one of the following options:

- Click **...** to specify a path to python.exe in your file system
- Click **Create** to download and install Python from python.org (28.38 MB)

Once created, you are presented with an empty project in PyCharm as follows:

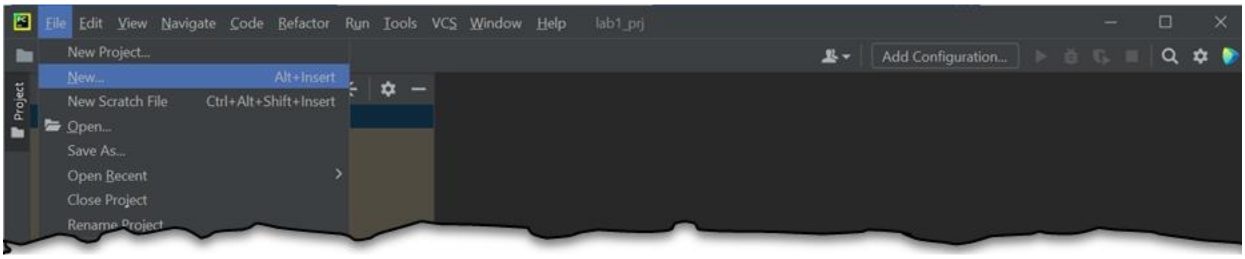


3. Write and run your first Python program

a. Open a New Python file

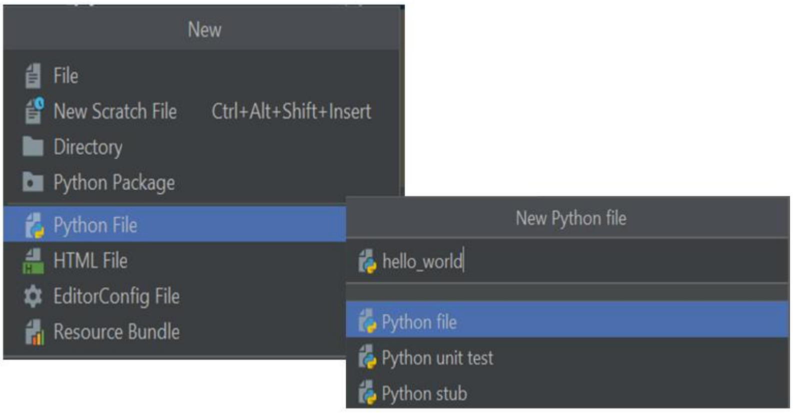
Now that we have created a Python project, it’s time to create a Python program file to write and run your first Python program.

To create a file, you either right click on the folder name > New > Python File or click File on the menu bar and select New...



b. Select Python File

You are then prompted with a menu of different file types. **Select Python File**, then name your new file as " hello_world" and click ok.

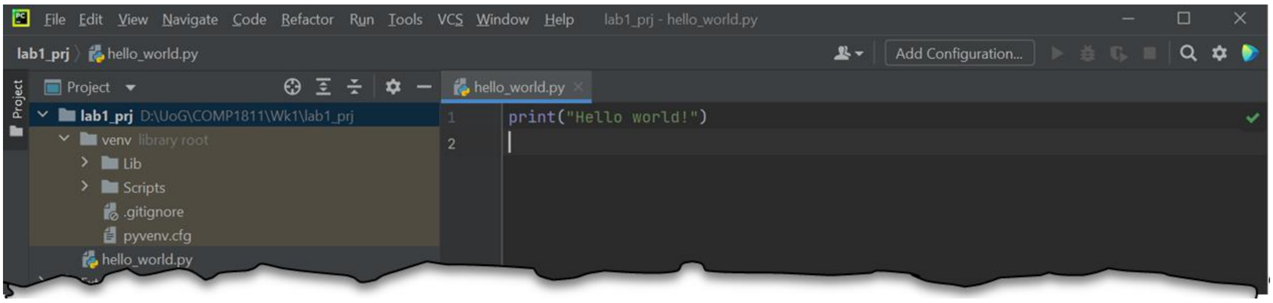


hello_world

c. Write your code in the editing pane.

Type or cut-and-paste the following code in the editing panel of the "HelloWorld" file you just created: print("Hello World")

This Python code prints Hello World on the console.

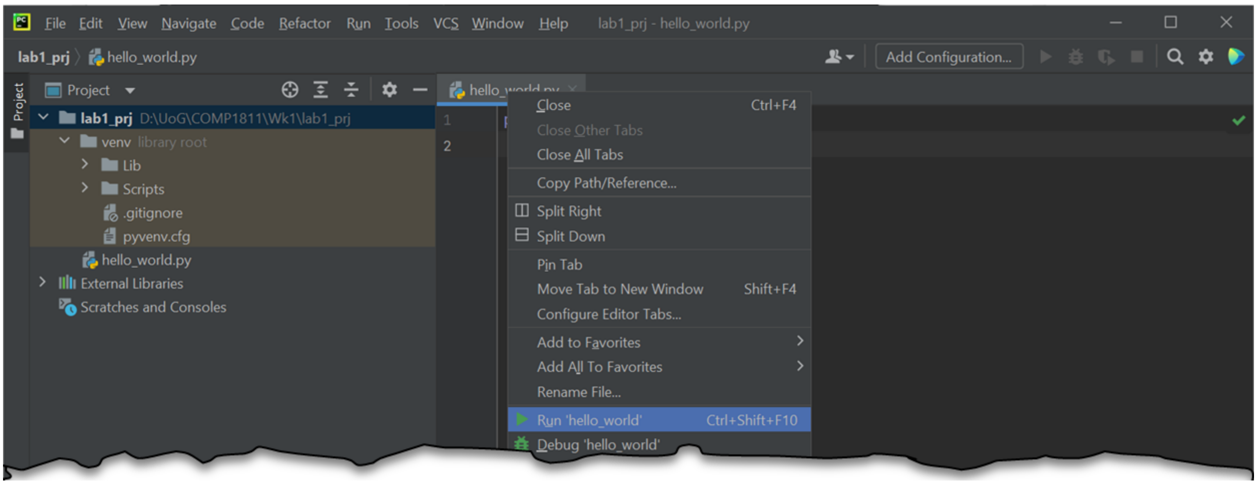


File Edit View Navigate Code Refactor Run Tools VCS Window Help lab1_prj - hello_world.py
lab1_prj 〉 hello_world.py
Add Configuration...
Project
lab1_prj D:\UoG\COMP1811\Wk1\lab1_prj
venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
hello_world.py

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

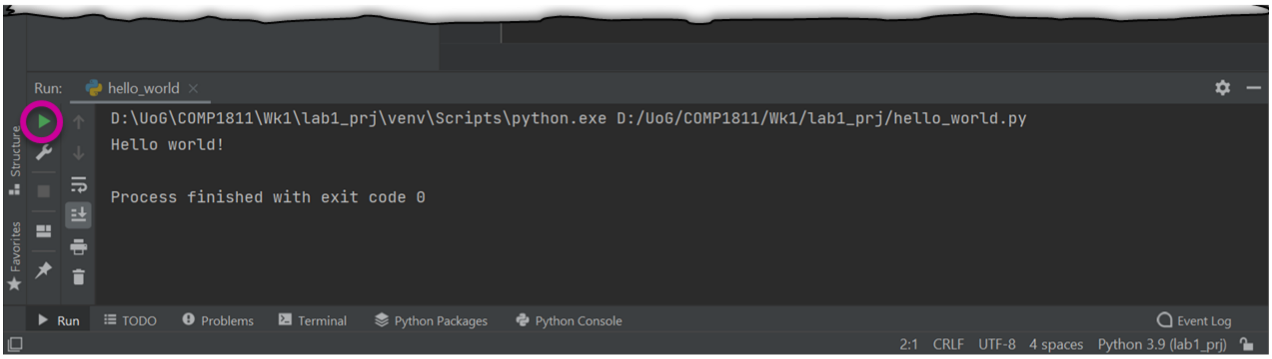
d. Run your code.

Right click on the hello_world.py file (or the name you gave to the Python file you just created) in the left sidebar or the tab in the editing pane and click on Run hello_world. You could also click Run on the menu bar and the select run from the drop-down menu. Alternatively, you can use the short-cut hot keys Alt+Shift+F10.



e. Check your program’s output

You can see the output of the program at the bottom of the screen. The area shown is called the console and this where the output of the "print" function is displayed.



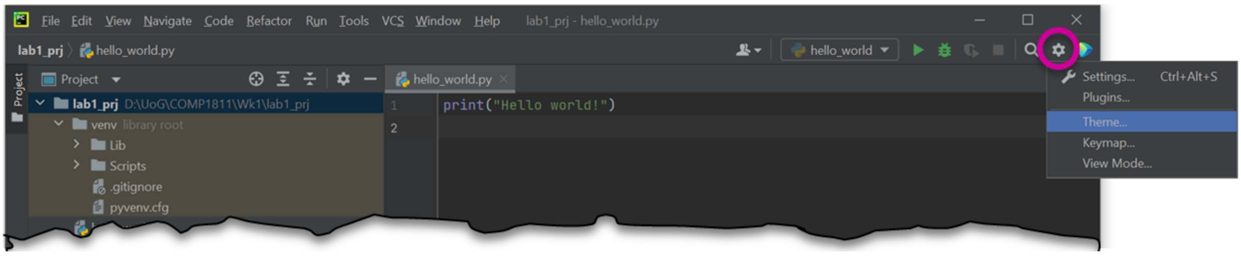
```
D:\UoG\COMP1811\Wk1\lab1_prj\venv\Scripts\python.exe D:/UoG/COMP1811/Wk1/lab1_prj/hello_world.py
Hello world!

Process finished with exit code 0
```

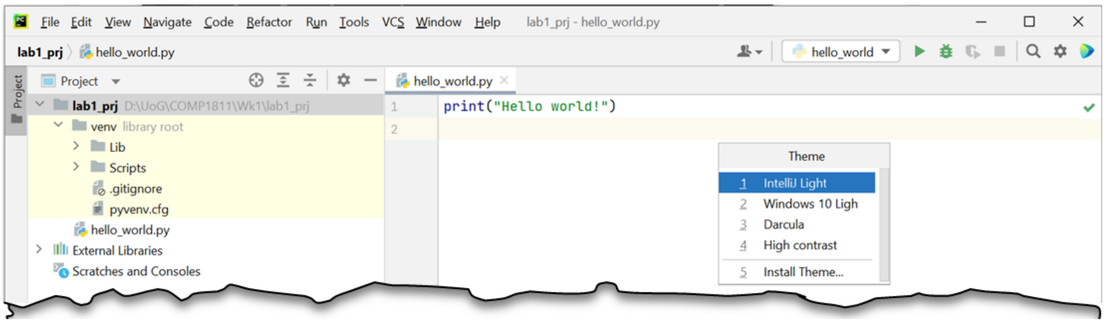
To run your code again, click on the Run (play) icon circled.

f. Change the editor theme

You can change the colour theme of PyCharm (background and foreground) by clicking on the cog icon circled ⚙, then selecting "theme".



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

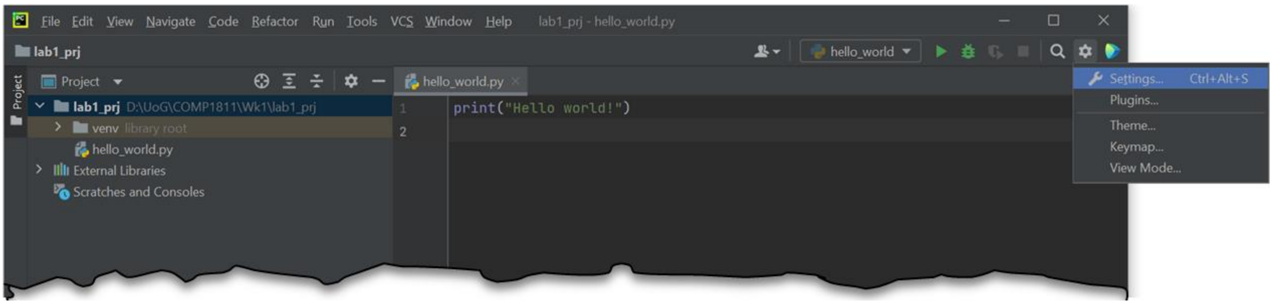


```
print("Hello world!")
```

g. "Interpreter not found" Error

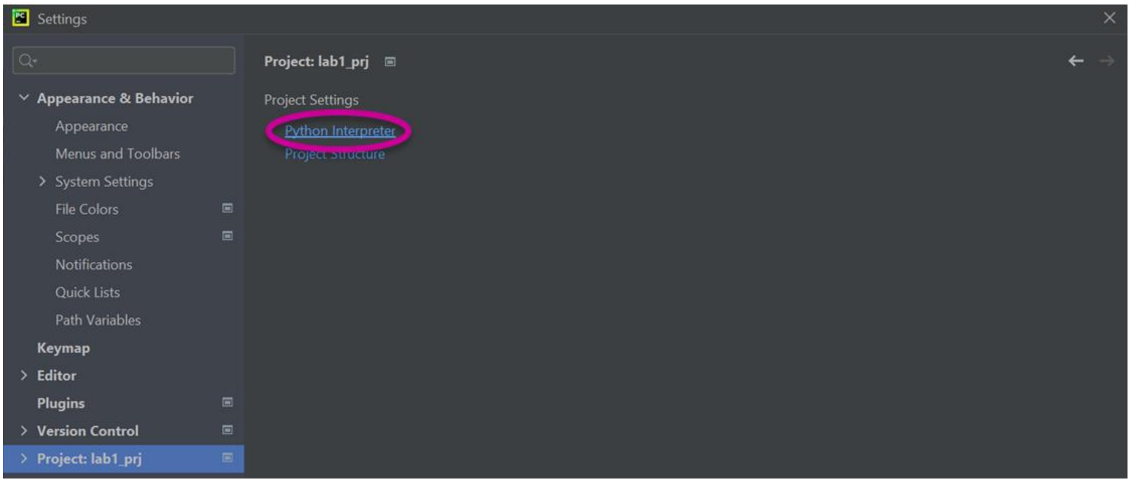
If the Python interpreter is installed, but you get the error "Interpreter not found…" when you try to run your program you, must manually tell PyCham where to find Python.

Press Ctrl+Alt+S (by clicking on the cog icon ⚙ then selecting "settings") to open the IDE settings.

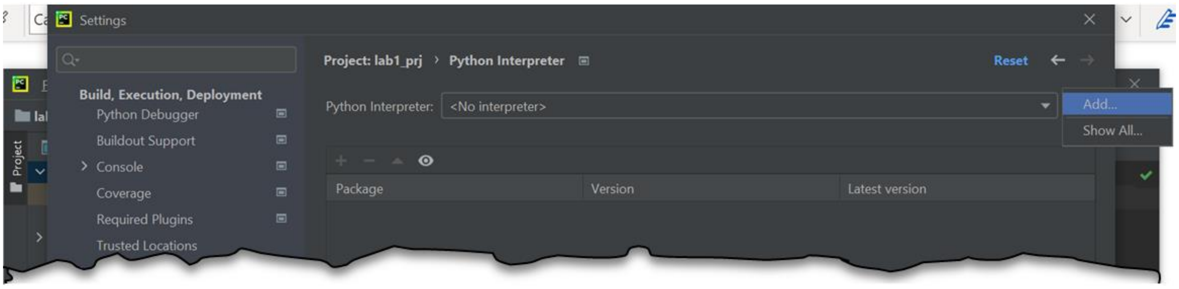


```
print("Hello world!")
```

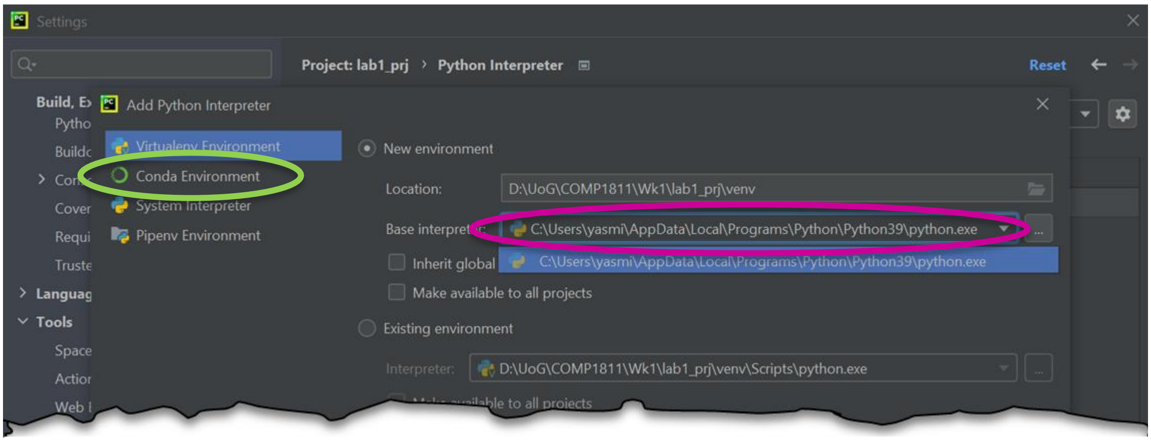
Select Project <project name> then Python Interpreter.



Select Show All.


Python Interpreter: <No interpreter>

Select the target interpreter. When PyCharm stops supporting any of the outdated Python versions, the corresponding Python interpreter is marked as unsupported.


Location: D:\UoG\COMP1811\Wk1\lab1_prj\venv

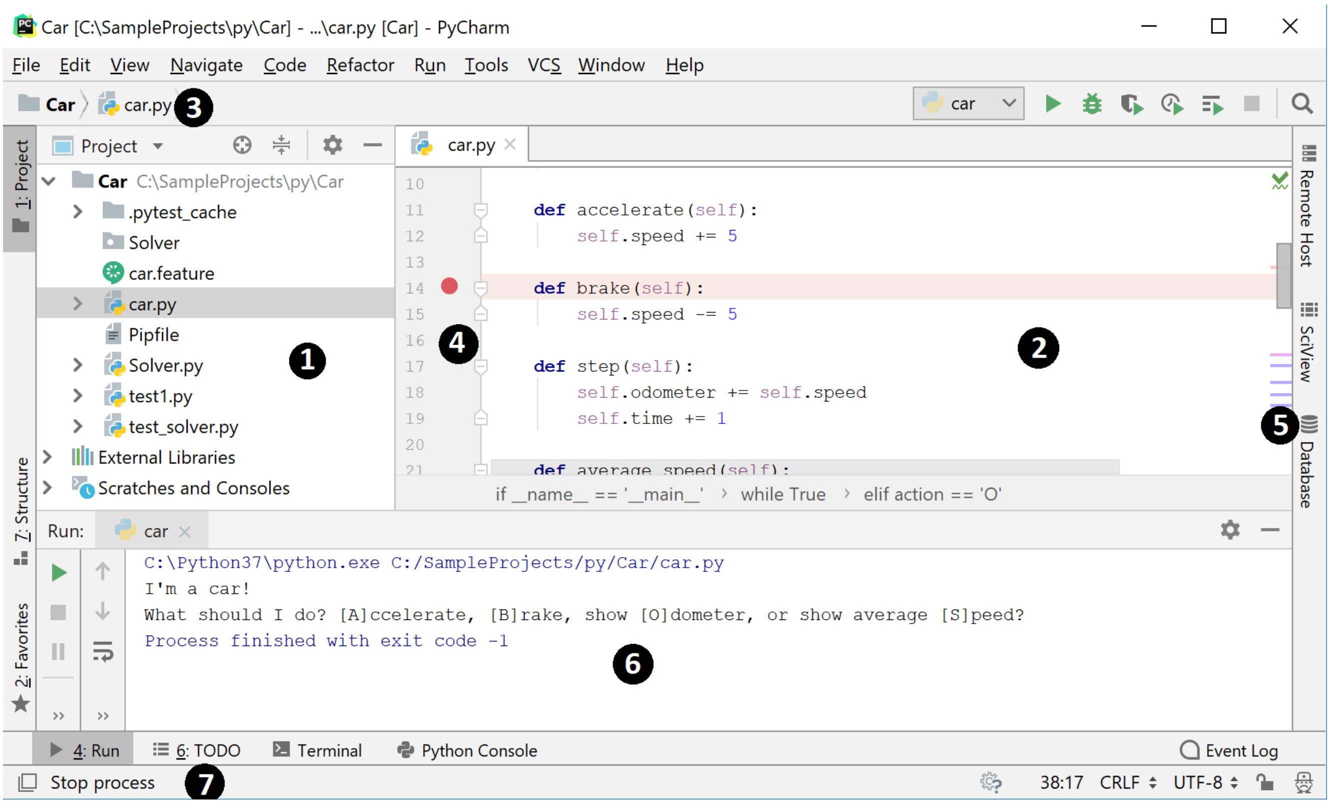
Base interpreter: C:\Users\yasmi\AppData\Local\Programs\Python\Python39\python.exe

Interpreter: D:\UoG\COMP1811\Wk1\lab1_prj\venv\Scripts\python.exe

If nothing appears in the highlighted box, choose "Conda Environment" circled from the list on the left of the image.

4. Take a look round PyCharm

When a project is opened, you see the main window divided into several logical areas. Let’s take a moment to see the key UI elements here:



```
    def accelerate(self):
        self.speed += 5

    def brake(self):
        self.speed -= 5

    def step(self):
        self.odometer += self.speed
        self.time += 1

    def average_speed(self):
```

```
C:\Python37\python.exe C:/SampleProjects/py/Car/car.py
I'm a car!
What should I do? [A]ccelerate, [B]rake, show [O]dometer, or show average [S]peed?
Process finished with exit code -1
```

1. Project view on the left side displays your project files.
2. Editor on the right side, where you actually write your code. It has tabs for easy navigation between open files.
3. Navigation bar above the editor additionally allows you to quickly run and debug your application as well as do the basic VCS actions.
4. **Left gutter**, the vertical stripe next to the editor, shows the breakpoints you have, and provides a convenient way to navigate through the code hierarchy like going to definition/declaration. It also shows line numbers and per-line VCS history.
5. **Right gutter**, on the right side of the editor. PyCharm constantly monitors the quality of your code and always shows the results of its code inspections in the right gutter: errors, warnings, etc. The indicator in the top right-hand corner shows the overall status of code inspections for the entire file.
6. Tool windows are specialized windows attached to the bottom and sides of the workspace and provide access to typical tasks such as project management, source code search and navigation, integration with version control systems, etc.
7. The status bar indicates the status of your project and the entire IDE, and shows various warnings and information messages like file encoding, line separator, inspection profile, etc.

Also, on the bottom-left corner of the PyCharm window, in the Status bar, you see the button □ or □. This button toggles the showing of the tool window bars. If you hover your mouse pointer over this button, the list of the currently available tool windows show up:

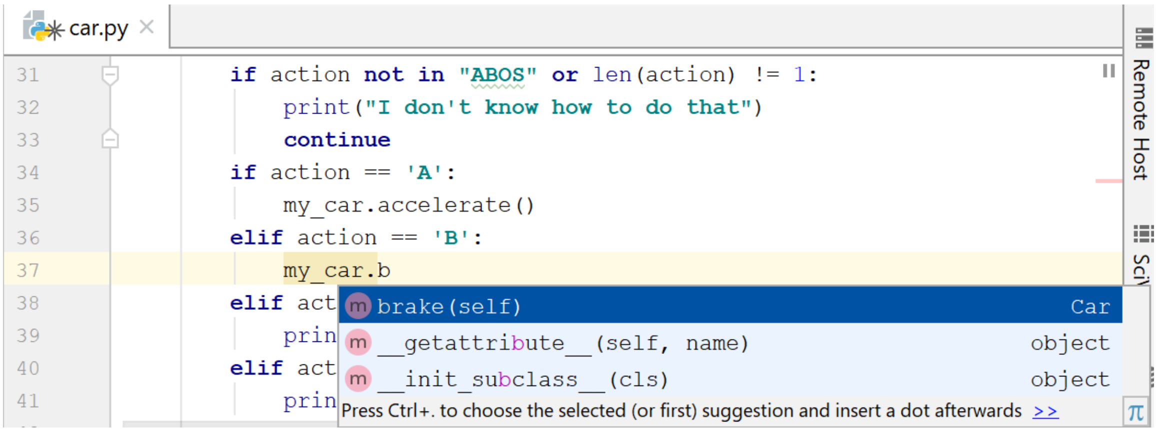
a. Code with smart assistance

PyCharm takes care of the routine so that you can focus on the important. Use the following coding capabilities to create error-free applications without wasting precious time.

i. Code completion

Code completion is a great time-saver, regardless of the type of file you're working with. Basic completion works as you type and completes any name instantly.

Smart type completion analyses the context you're currently working in and offers more accurate suggestions based on that analysis.



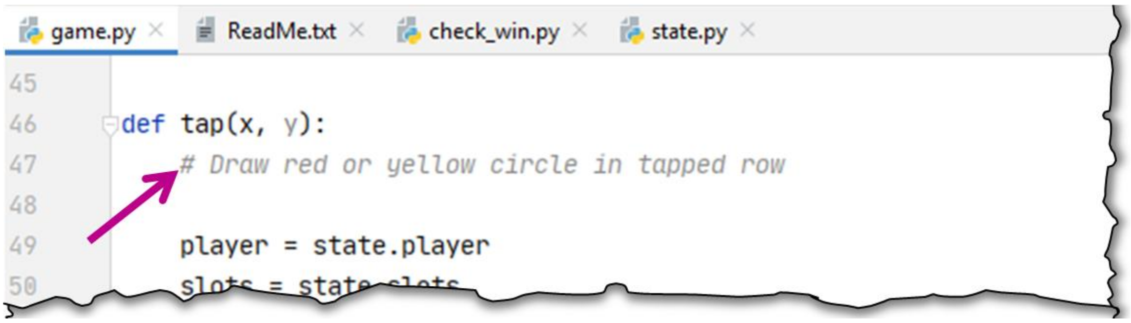
```
if action not in "ABOS" or len(action) != 1:
    print("I don't know how to do that")
    continue
if action == 'A':
    my_car.accelerate()
elif action == 'B':
    my_car.b
elif act
    prin
elif act
    prin
```

```
brake(self)                         Car
__getattribute__(self, name)        object
__init_subclass__(cls)              object
Press Ctrl+. to choose the selected (or first) suggestion and insert a dot afterwards >>
```

ii. Comments (multiple-line comments)

A Python comment is a line of text in a program that is not executed. Comments are used to explain the code that follows. This is useful for you, as well as other programmers that may use your code so to better understand the intent and functionality of the program or a block of code.

The Python interpreter knows that a particular line of code is a "comment" if it starts with a "#" as shown in the extract below:



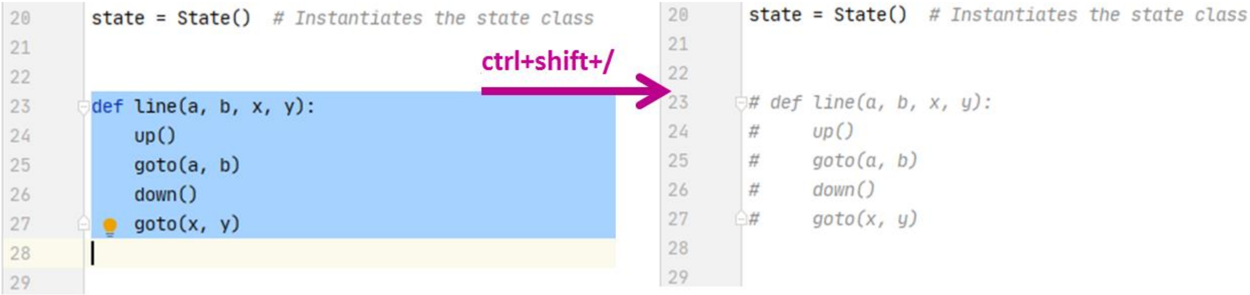
```

def tap(x, y):
    # Draw red or yellow circle in tapped row

    player = state.player
    slots = state.slots
```

Comments can also be used to tell the Python interpreter to ignore lines or chunks of your code. You would comment out code mainly because that part of the code may be incomplete or has bugs (issues) with it but you want the program to run without errors. So, often you will need to comment out multiple lines of code.

If you need to comment out multiple lines of code in PyCharm, select the lines to be commented out and then press **ctrl+shift+/** (or **ctrl+/** depending on your installation) . To uncomment the lines, we just have to select the lines and then again press **ctrl+shift+/** (or **ctrl+/**).



```
state = State()  # Instantiates the state class


def line(a, b, x, y):
    up()
    goto(a, b)
    down()
    goto(x, y)

```

```
state = State()  # Instantiates the state class


# def line(a, b, x, y):
#     up()
#     goto(a, b)
#     down()
#     goto(x, y)

```

b. Find your way through a program / PyCharm project

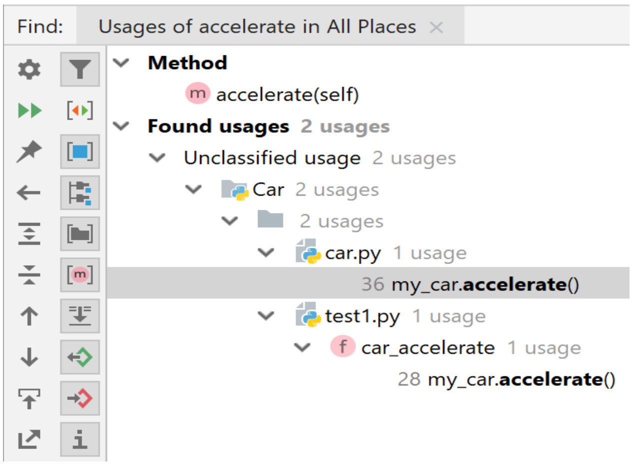
When your project is big, or when you need to work with someone else's code, it's vital to be able to quickly find what you are looking for and dig into the code. This is why PyCharm comes with a set of search features that help you find your way through any code no matter how tangled it is.

i. Basic search

With these search facilities, you can find and replace any fragment of code both in the currently opened file (**Ctrl+F**), or in an entire project (**Ctrl+Shift+F**).

ii. Search for usages

To find where a particular symbol is used, PyCharm suggests full-scale search via Find Usages (**Alt+F7**):



```
Method
    accelerate(self)
Found usages  2 usages
    Unclassified usage  2 usages
        Car  2 usages
            2 usages
                car.py  1 usage
                        36 my_car.accelerate()
                test1.py  1 usage
                    car_accelerate  1 usage
                        28 my_car.accelerate()
```

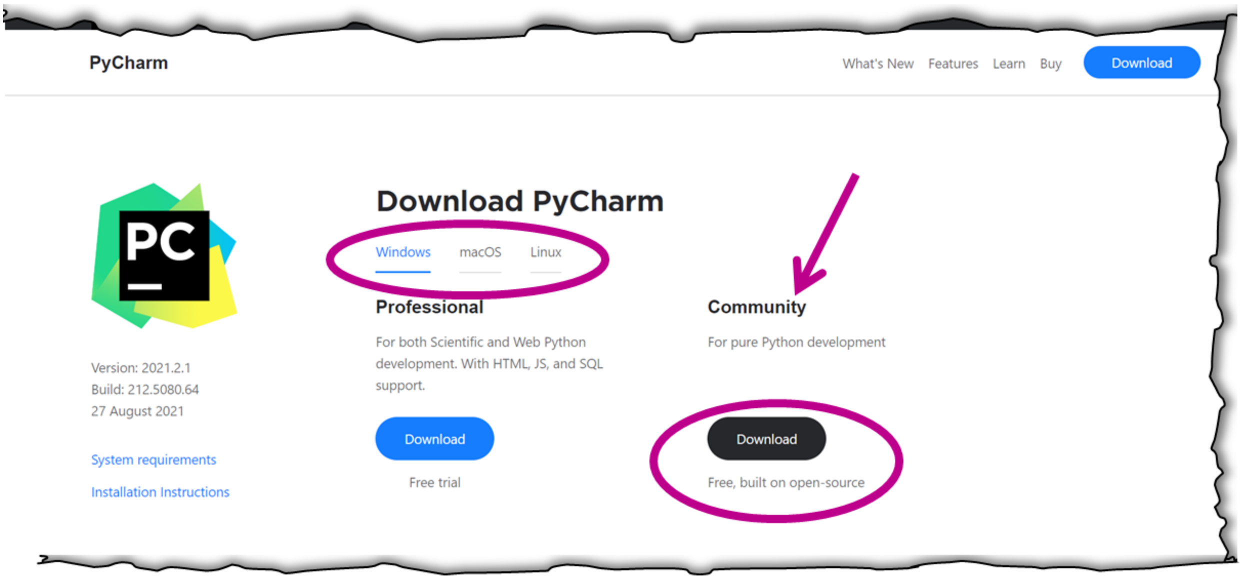
Appendix1: Install PyCharm at Home

Install Python on your home machine. This is useful as you will need the IDE for your coursework and to practice coding in Python at home. Alternatively, you can create a virtual machine that connects to the Greenwich machines using VMWare, but this is sometime slow. Instructions for a creating virtual machine are available on the Greenwich portal.

a. Launch PyCharm

For PyCharm to work, Python must be installed too. We use Python 3.9.* on this module. The latest PyCharm installer will also install an appropriate Python interpreter, so you do not need to install it separately.

Download PyCharm from the JetBrains website (https://www.jetbrains.com/pycharm/download/). Select an appropriate version for the platform you use (PC / Mac / etc).



Download PyCharm

Windows macOS Linux

Version: 2021.2.1
Build: 212.5080.64
27 August 2021

System requirements
Installation Instructions

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

Download

Free trial

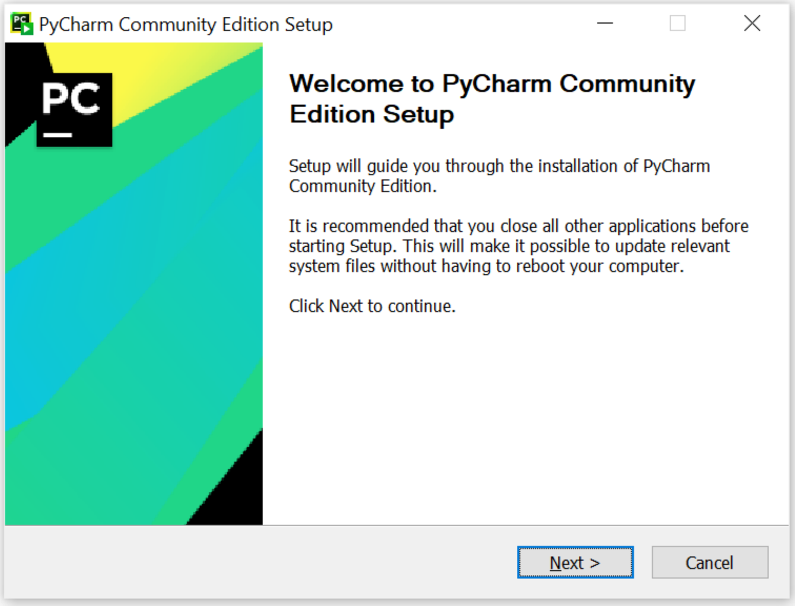
Community

For pure Python development

Download

Free, built on open-source

Make sure you choose the Community Edition of PyCharm (it is free and open-source). Once download is complete, double click on the downloaded file and you should see the following:



PyCharm Community Edition Setup

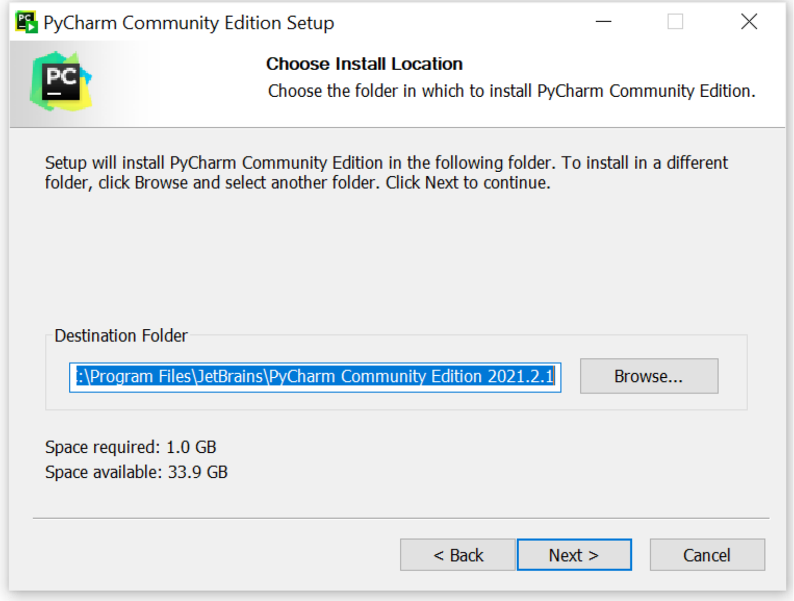
Welcome to PyCharm Community Edition Setup

Setup will guide you through the installation of PyCharm Community Edition.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Chose where to install PyCharm:



PyCharm Community Edition Setup

Choose Install Location

Choose the folder in which to install PyCharm Community Edition.

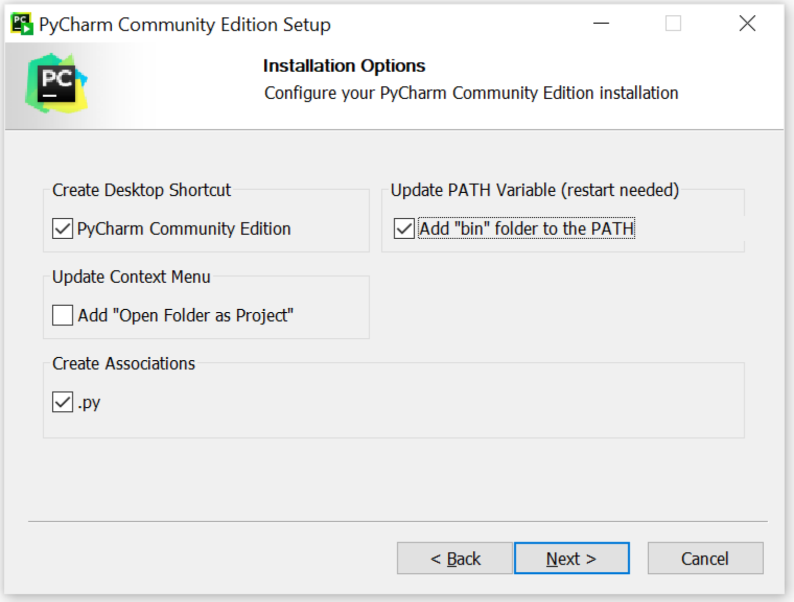
Setup will install PyCharm Community Edition in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder

:\Program Files\JetBrains\PyCharm Community Edition 2021.2.1

Space required: 1.0 GB
Space available: 33.9 GB

Select the following settings, and especially select the "update PATH variable" – this will add the path to the Python interpreter to the Windows variables so that it can be accessed from anywhere. Note that you will need to reboot your machine for this setting to take effect.



PyCharm Community Edition Setup

Installation Options

Configure your PyCharm Community Edition installation

Create Desktop Shortcut

☑ PyCharm Community Edition

Update PATH Variable (restart needed)

☑ Add "bin" folder to the PATH

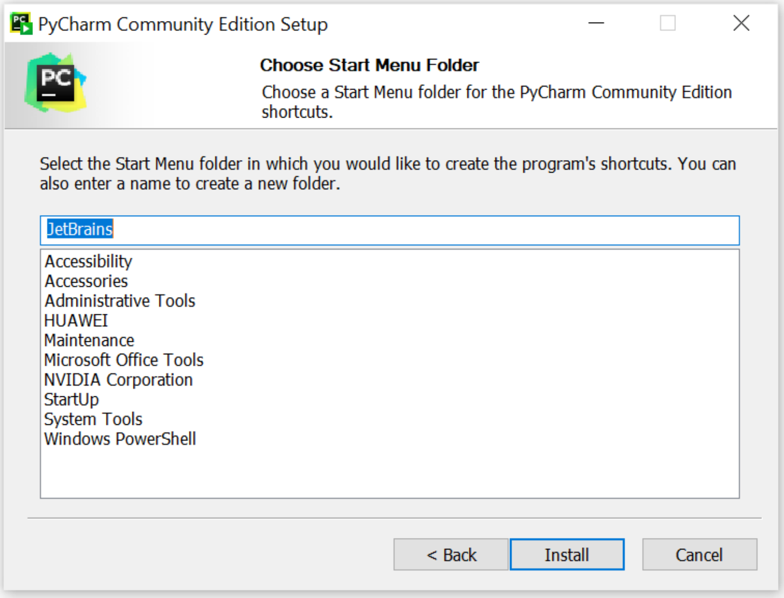
Update Context Menu

□ Add "Open Folder as Project"

Create Associations

☑ .py

Click next and select which Windows Start Menu folder PyCharm should go under:



PyCharm Community Edition Setup

Choose Start Menu Folder

Choose a Start Menu folder for the PyCharm Community Edition shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

JetBrains

Accessibility
Accessories
Administrative Tools
HUAWEI
Maintenance
Microsoft Office Tools
NVIDIA Corporation
StartUp
System Tools
Windows PowerShell

Press Install and follow remaining instructions to complete installation. You may need to reboot your machine once complete.

b. Launch PyCharm

Launch PyCharm from the Windows Start menu. Launching may take a short while and you should see the splash screen with a progress bar indicating an estimate of remaining launch time.

Once launch is complete for the first time, you will be prompted with JetBrains's community edition terms. Read and then accept. Once done, reboot your machine and you're ready to start working with PyCharm.