# INTRODUCTION TO PYTHON

In this session, you will set up your Python programming environment

Multiple TAs will be available today to help you set up your Python programming environment

# Reading and Watching

- Real Python – excellent videos (some free, all videos for $20/month)
  https://realpython.com/

# References

- Conda https://conda.io/en/latest/

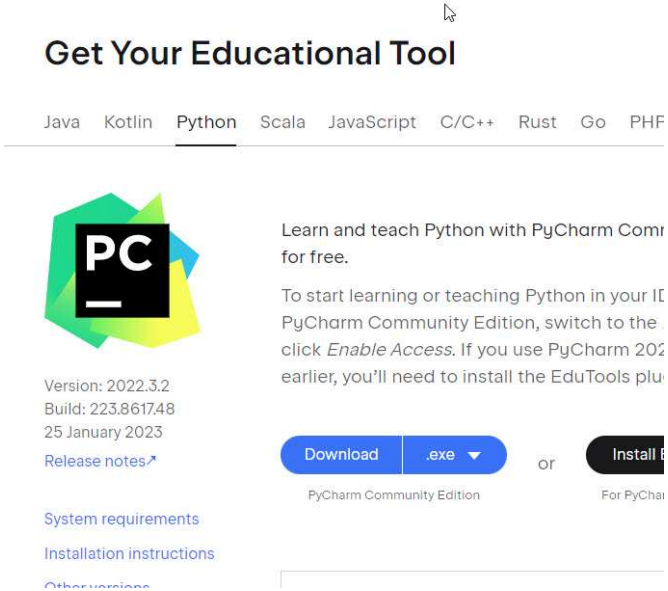
# Class Assumptions

- Everyone has had some (maybe limited) programming experience
- Not everyone has used Python
- Not everyone has used an Integrated Development Environment
- Class approach
  - Introduce use of Python slowly, working on real data and real political informatics problems
  - Monitor class progress with many “Are We on Track?” exercises
  - Adjust rate of material delivery based on class feedback

# PyCharm

- You should use the Community Edition of PyCharm for your work, which you can access at https://www.jetbrains.com/pycharm-edu/

If you have not yet installed PyCharm, you can do it now

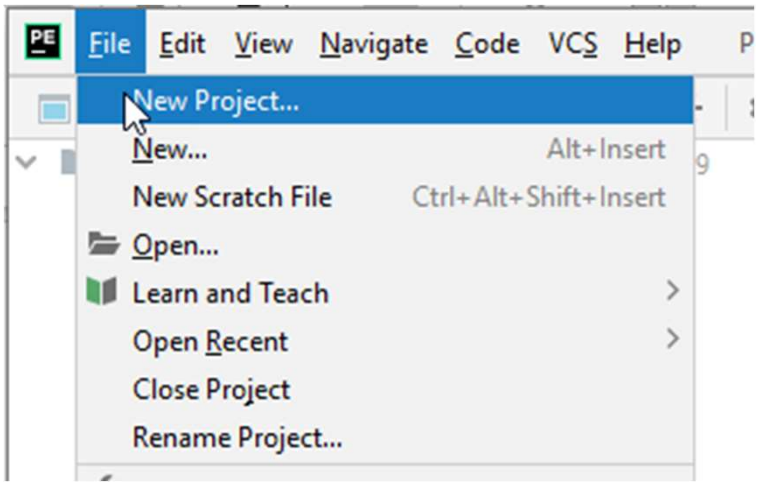


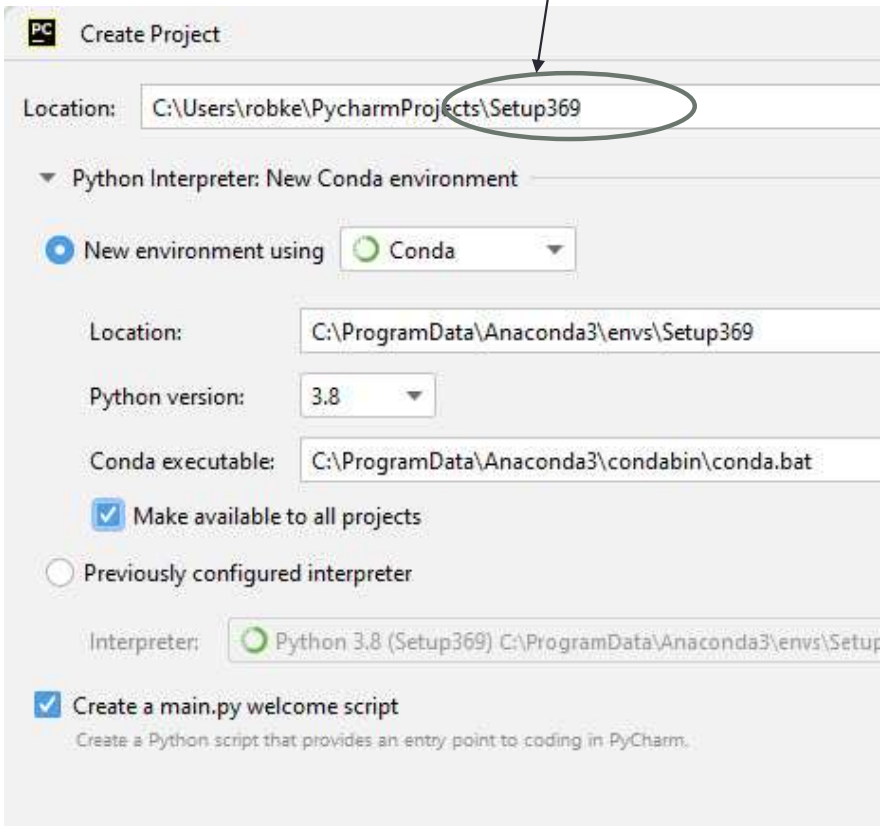
# PyCharm Setup

- You can use any Python Interactive Development Environment (IDE) you are comfortable with, but the course-supported IDE is PyCharm
- If you are using PyCharm,
  - Set up at least one PyCharm project for the course
  - Set your Python interpreter for that project as Python 3.8
  - Set Conda as your configuration manager
- If you have any difficulty in setting up PyCharm, we will provide TA support to help you complete it

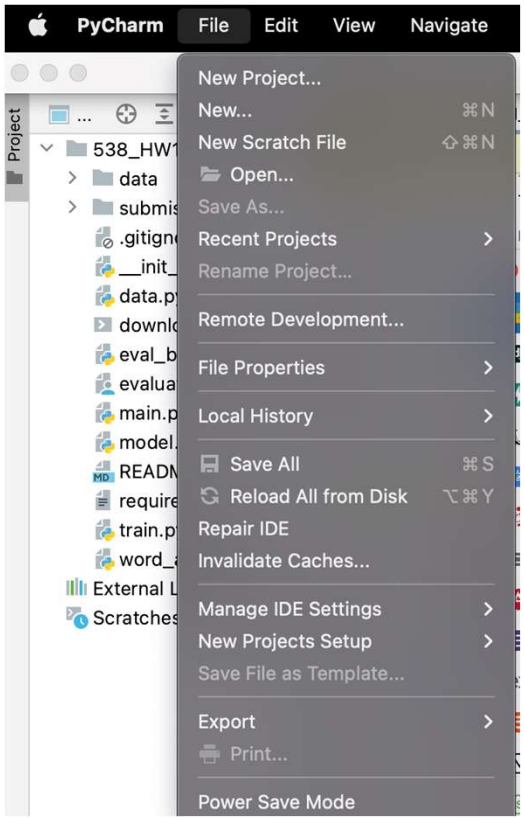
  You may need to install Conda from https://conda.io/projects/conda/en/latest/user-guide/install/index.html#regular-installation

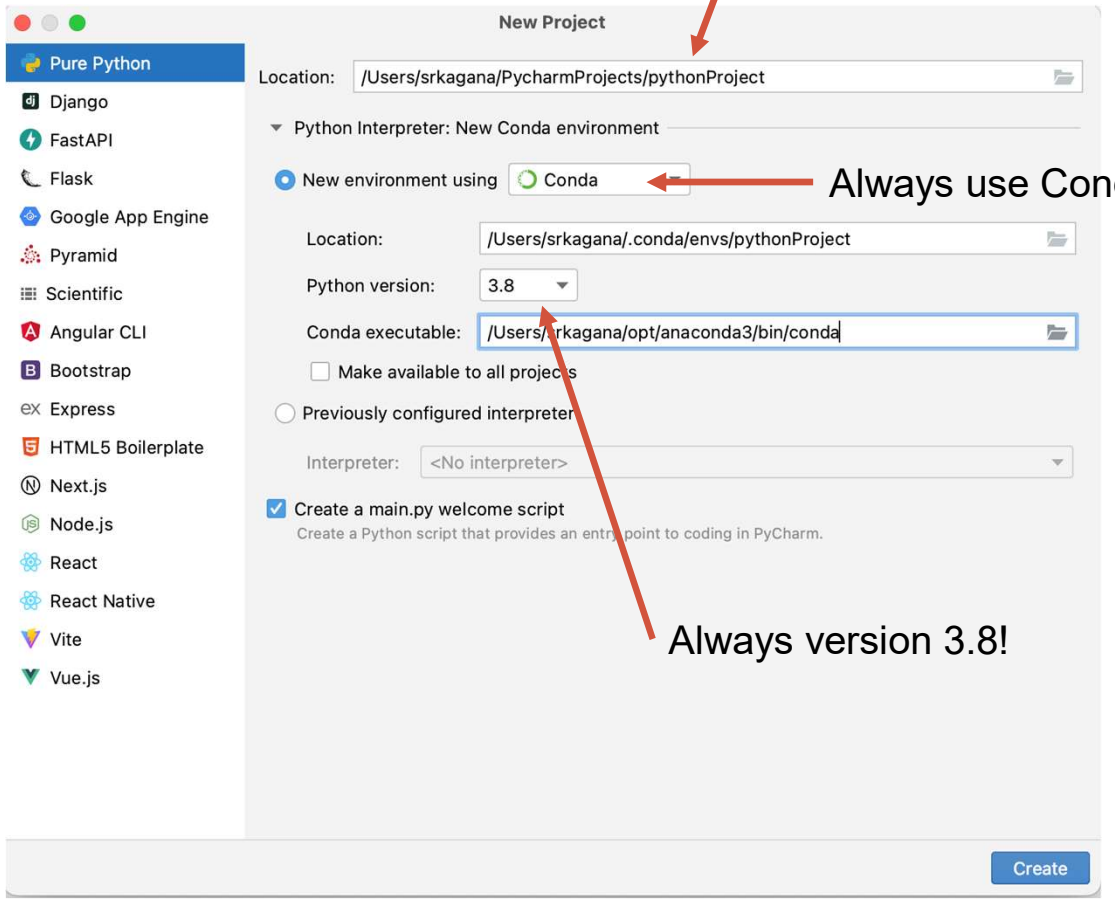
# Setting a Windows PyCharm Project



# Setting a Mac OS PyCharm Project




Enter your project name

Always use Conda!

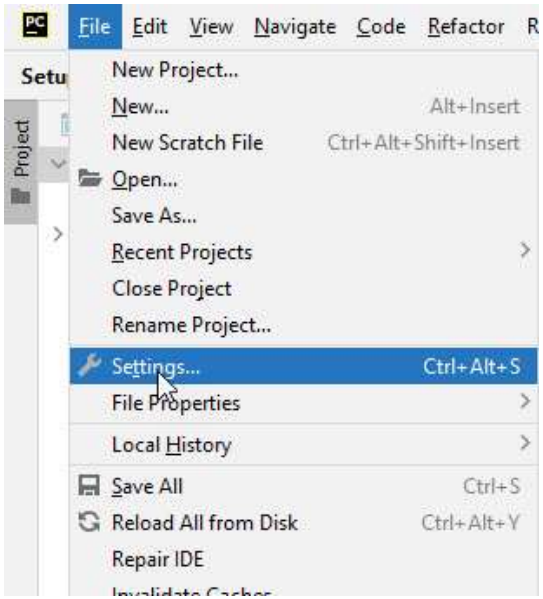
Always version 3.8!

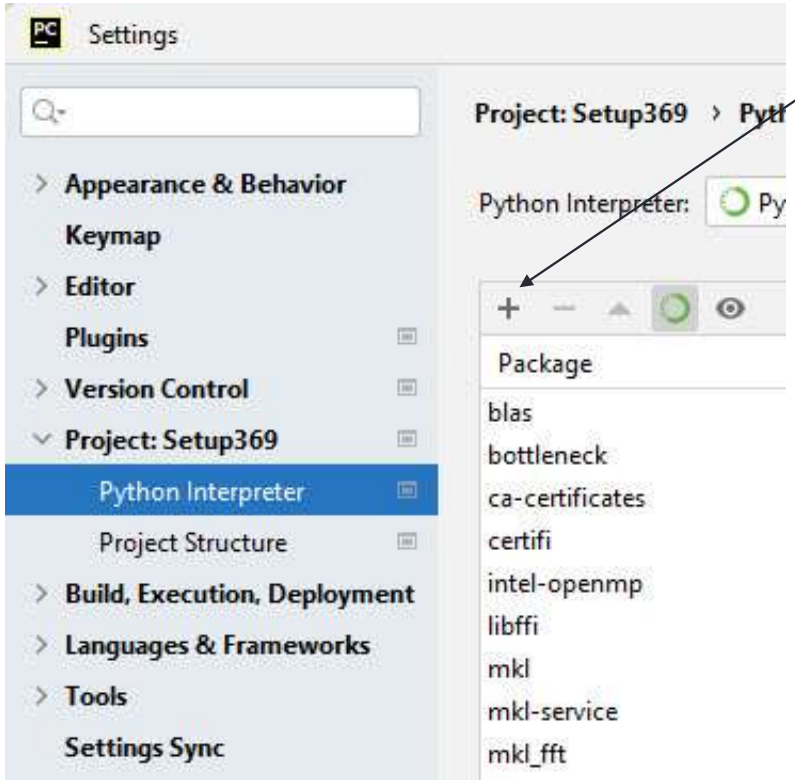
# Installing Python Libraries

- You will need to install Python libraries into your project during the semester
- Most (but not all) libraries will already be available in PyCharm
- You need to explicitly install most libraries in PyCharm (using Conda)
- Operates differently on Windows and on Mac OS

Think of a PyCharm project as a combination of Python interpreter, Python libraries, and your code

# Installing pandas on Windows …

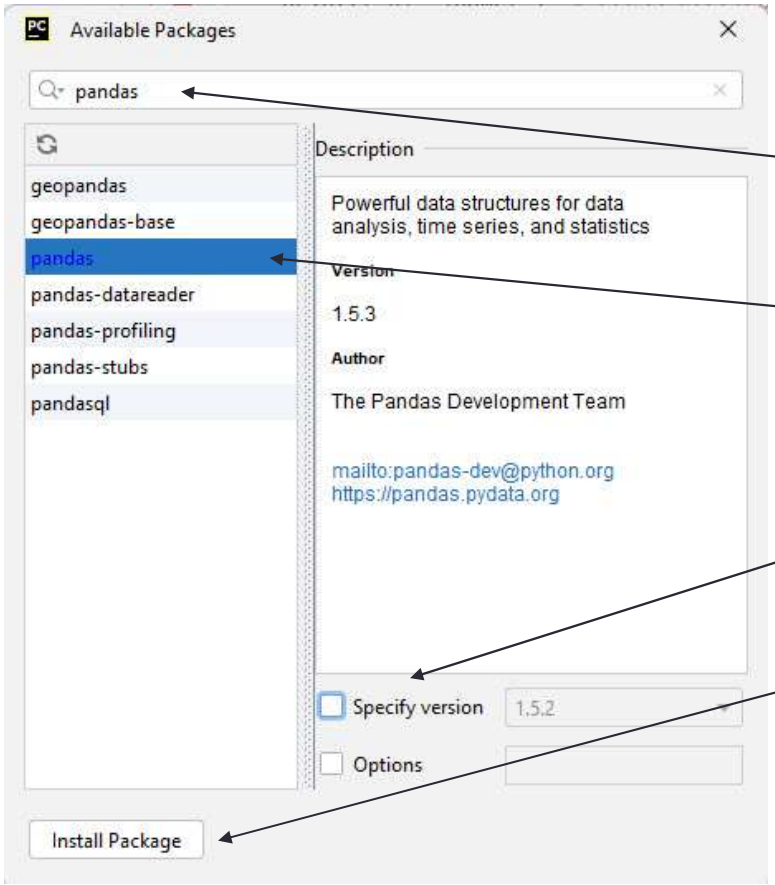


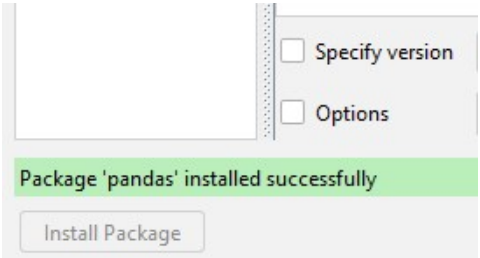

Click the + button to search for the library to install

You will use the pandas library starting next week

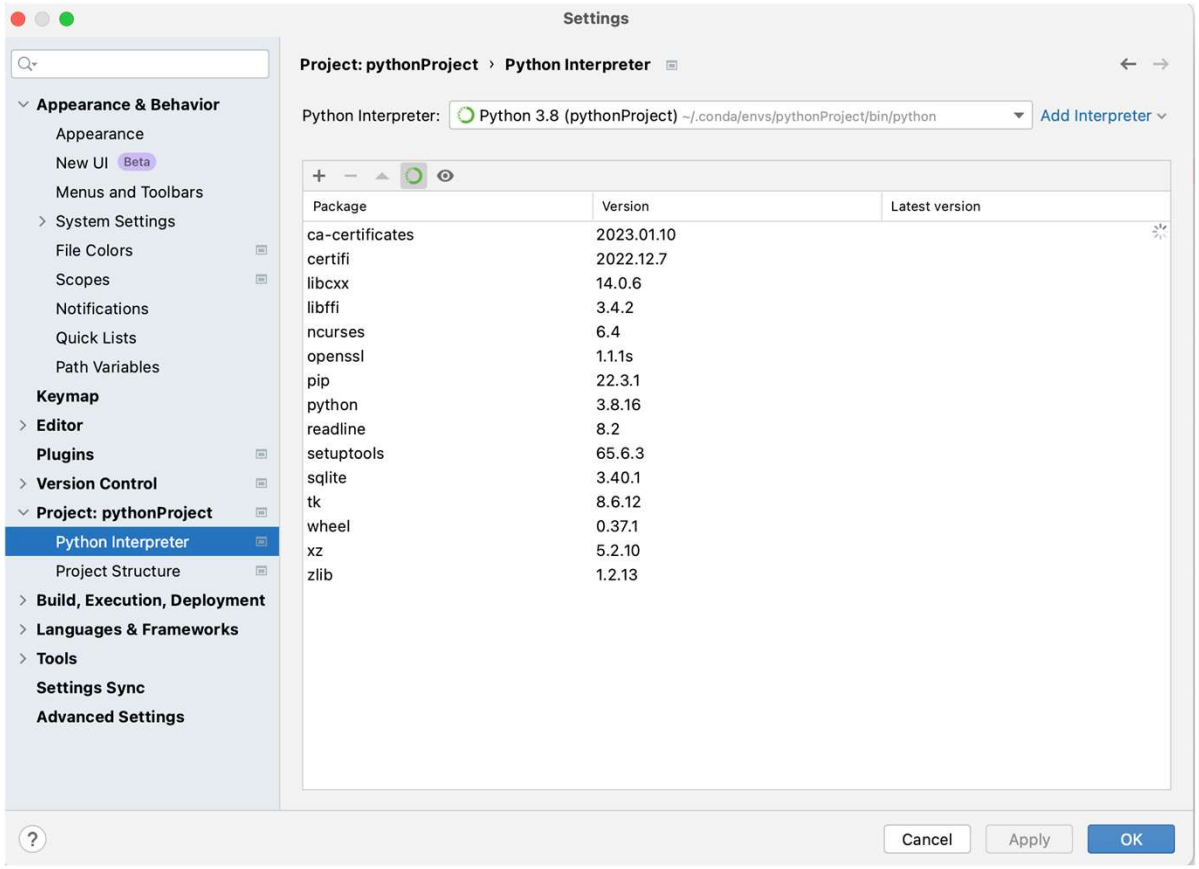
# … Installing pandas on Windows




- Steps
  - Enter the library name in the search bar
  - If needed, select the library from the search list
  - If needed, specify the version of the library
  - Click “Install Package”



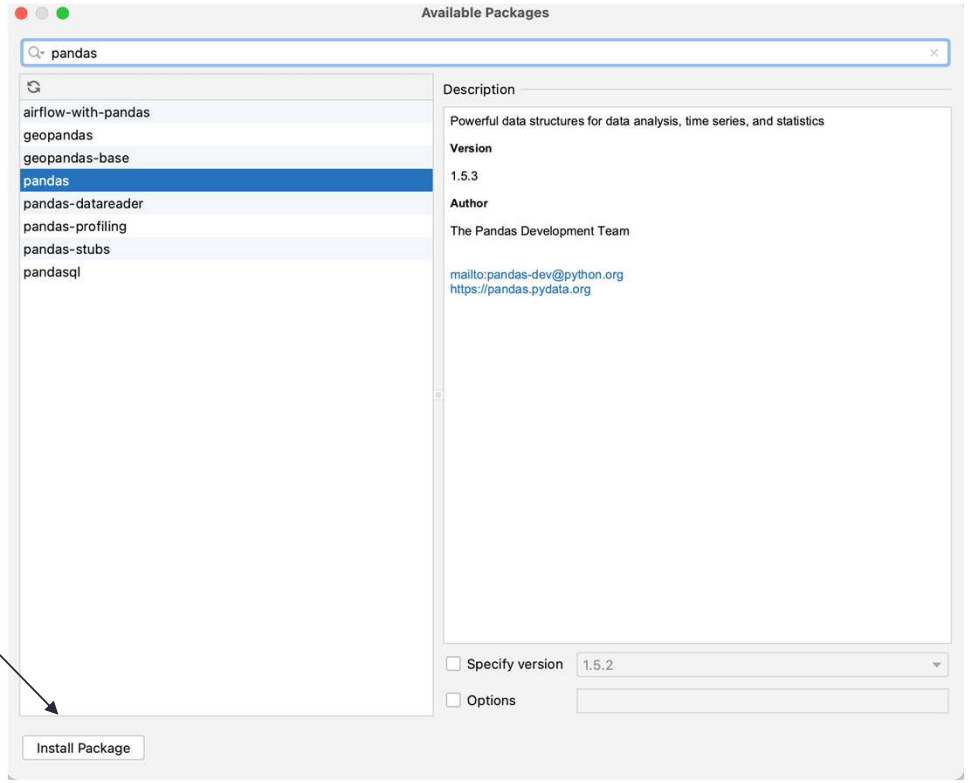
# Installing Python libraries on Mac OS …



# … Installing Python libraries on Mac OS

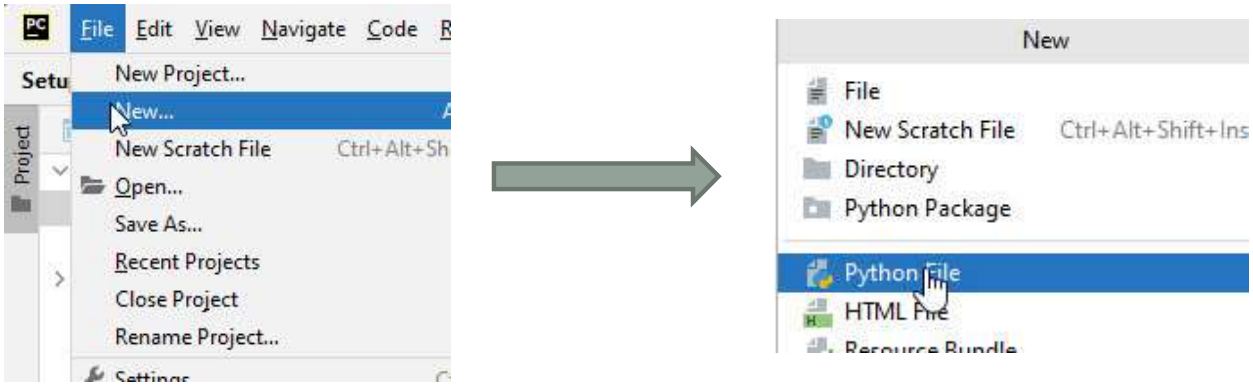
- Enter the desired library name in the search bar

Click Install Package



# Your First Python Program

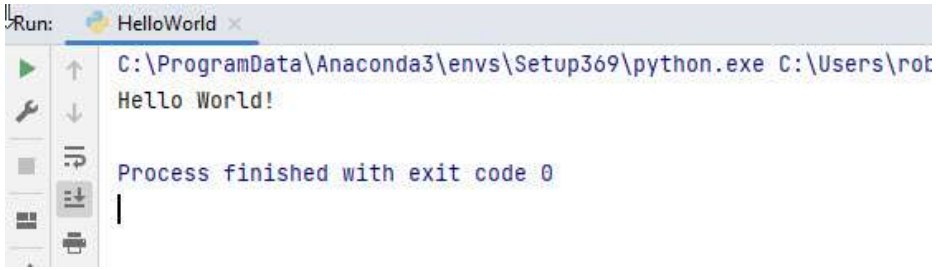
- Print “Hello World!” to your console




Enter your program and click on Run (selecting the program)

Python code

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



# Python Libraries

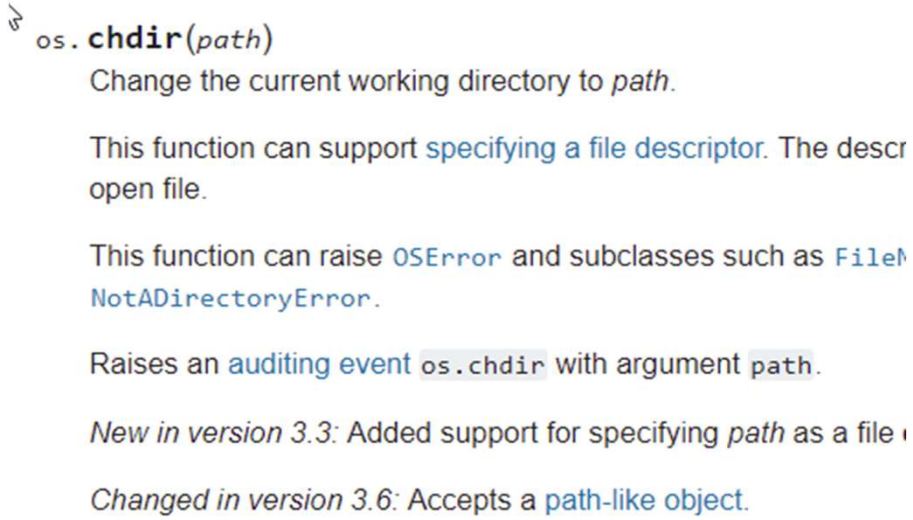
- The next Python track uses the os library
- Libraries you need are often pre-loaded into PyCharm, but you need to add them to your PyCharm project

```
import os
```

# Library Documentation

- Python library documentation is available for all the libraries you use
- os library documentations is available at https://docs.python.org/3/library/os.html

Link to the os library documentation and get familiar with the style. If needed, library extracts will be given to you in your exam API



os.**chdir**(*path*)

Change the current working directory to *path*.

This function can support specifying a file descriptor. The descr
open file.

This function can raise `OSError` and subclasses such as `FileN`
`NotADirectoryError`.

Raises an auditing event `os.chdir` with argument `path`.

*New in version 3.3:* Added support for specifying *path* as a file

*Changed in version 3.6:* Accepts a path-like object.