

Writing Neural Network Code: Introduction to TensorFlow, Hands-On

Class 3: TensorFlow Hands-On Part 1: Hello World!

May 13, 2020

Charles J. Lord, PE
President, Consultant, Trainer
Blue Ridge Advanced Design and Automation

This Week's Agenda

- 5/11 A Brief History of Artificial Neural Networks
- 5/12 Neural Network Simulation and Programming
- 5/13 TensorFlow Hands-On Part 1: Hello World!
- 5/14 TensorFlow Hands-On Part 2: Defining and Building Your Network
- 5/15 TensorFlow Hands-On Part 3: Teaching and Testing and Conclusion

This Week's Agenda

5/11 A Brief History of Artificial Neural Networks

5/12 Neural Network Simulation and Programming

5/13 TensorFlow Hands-On Part 1: Hello World!

5/14 TensorFlow Hands-On Part 2: Defining and Building
Your Network

5/15 TensorFlow Hands-On Part 3: Teaching and Testing
and Conclusion

Lessons Learned – Version Updates



Life comes at you fast.

Most Literature on TensorFlow is Out of Date

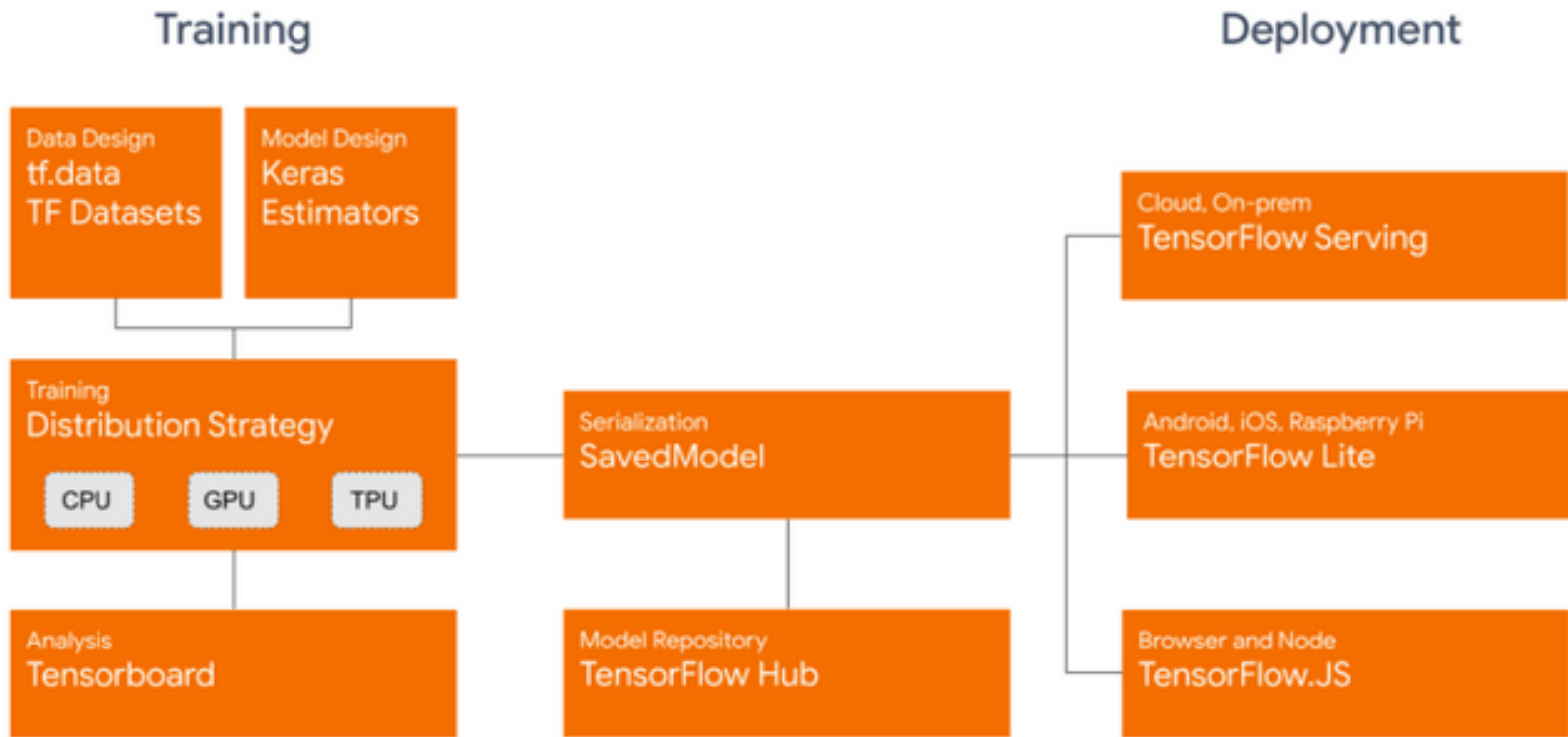
- TensorFlow made major changes from version 1.x to 2.x in late October 2019.
- This class was originally designed in 1.x
- All information here should be correct for v2
 - TRUST YET VERIFY
- If a book, article, or website mentions sessions, disregard it.

Question 1 – Experience with TensorFlow (including lite)?

Major Updates

- “tf.contrib” is gone and mostly integrated
- Easy model building with Keras (now fully integrated) and *eager execution* - no more build then run sessions.
- Robust model deployment in production on any platform.
- Standard databases are integral
- Powerful experimentation for research.
- API simplification by reducing duplication and removing deprecated endpoints.

TensorFlow 2.x Framework



Installing TensorFlow 2

- There are many ways to install (optional) and run TensorFlow
- Simplest for just testing out a few simple concepts – Google Colaboratory (Colab)
- Direct on a python installation (command line or IDE) on PC/Mac/Linux
- In a development environment, typically that runs in a virtual machine (Docker, PyCharm, etc)

<https://colab.research.google.com/notebooks/welcome.ipynb>

Welcome To Colaboratory
File Edit View Insert Runtime Tools Help

Share Settings

Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
- Machine Learning Examples
- Section

+ Code + Text Copy to Drive

Connect Editing

What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch [Introduction to Colab](#) to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
```

86400

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
seconds_in_a_week
```

604800

Click on the [] to run the code

▼ Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:


```
[ ] seconds_in_a_day = 24 * 60 * 60
    seconds_in_a_day
```

 86400

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Variables that you define in one cell can later be used in other cells:

```
[ ] seconds_in_a_week = 7 * seconds_in_a_day
    seconds_in_a_week
```

 604800

<https://www.python.org/downloads/>

The screenshot shows the Python.org website. At the top left is the Python logo. To its right is a 'Donate' button, a search bar with a magnifying glass icon, a 'GO' button, and a 'Socialize' button. Below this is a navigation menu with links for 'About', 'Downloads', 'Documentation', 'Community', 'Success Stories', 'News', and 'Events'. The main content area features the heading 'Download the latest version of Python' and a yellow button labeled 'Download Python 3.8.2'. Below the button are links for 'Looking for Python with a different OS?' (with sub-links for Windows, Linux/UNIX, Mac OS X, and Other) and 'Want to help test development versions of Python?' (with sub-links for Preleases and Docker images). At the bottom of this section is a link for 'Looking for Python 2.7? See below for specific releases'. To the right of the text is an illustration of two parachutes with cargo boxes hanging from them, set against a blue sky with clouds.

Active Python Releases

For more information visit the Python Developer's Guide.

Python version	Maintenance status	First released	End of support	Release schedule
3.8	bugfix	2019-10-14	2024-10	PEP 569
3.7	bugfix	2018-06-27	2023-06-27	PEP 537
3.6	security	2016-12-23	2021-12-23	PEP 494
3.5	security	2015-09-13	2020-09-13	PEP 478
2.7	end-of-life	2010-07-03	2020-01-01	PEP 373

<https://docs.anaconda.com/anaconda/>

- ▶ Home
- ▶ Anaconda Team Edition
- ▶ Anaconda Enterprise 5
- ▶ Anaconda Enterprise 4
- Anaconda Individual Edition**
 - Installation
 - User guide
 - Reference
 - End User License Agreement - Anaconda® Individual Edition
- ▶ Anaconda Cloud
- ▶ Archive

☰ Anaconda Individual Edition 🔗

The Most Trusted Distribution for Data Science

Anaconda® is a package manager, an environment manager, a Python/R data science distribution, and a collection of [over 7,500+ open-source packages](#). Anaconda is free and easy to install, and it offers [free community support](#).

Get the [Anaconda Cheat Sheet](#) and then [download Anaconda](#).

Want to install conda and use conda to install just the packages you need? Get [Miniconda](#).

Anaconda Navigator or conda?

After you install Anaconda or Miniconda, if you prefer a desktop graphical user interface (GUI) then use [Navigator](#). If you prefer to use Anaconda prompt (or terminal on Linux or macOS), then use that and conda. You can also switch between them.

You can install, remove, or update any Anaconda package with a few clicks in Navigator, or with a single conda command in Anaconda Prompt (terminal on Linux or macOS).

- **To try Navigator**, after installing Anaconda, click the Navigator icon on your operating system's program menu, or in Anaconda prompt (or terminal on Linux or macOS), run the command `anaconda-navigator`.
- **To try conda**, after installing Anaconda or Miniconda, take the 30-minute [conda test drive](#) and download a [conda cheat sheet](#).

Packages available in Anaconda

- Over 250 [packages](#) are automatically installed with Anaconda.
- Over 7,500 additional open-source packages (including R) can be individually installed from the Anaconda repository with the `conda install` command.
- Thousands of other packages are available from [Anaconda Cloud](#).
- You can download other packages using the `pip install` command that is installed with Anaconda. [Pip packages](#) provide many of the features of conda packages and in some cases they can work together. However, the preference should be to install the conda package if it is available.
- You can also make your own [custom packages](#) using the `conda build` command, and you can share them with others by uploading them to [Anaconda Cloud](#), PyPI, or other repositories.

Our Setup

- Python 3.8.3 from python.org
- PyCharm IDE and virtual environments
- Will add numpy and TensorFlow elements
- We will also look at Colab
- We will also set up the environment for the NVIDIA Jetson NANO

<https://www.jetbrains.com/pycharm/download/#section=windows>

PyCharm

[What's New](#) [Features](#) [Learning Center](#) [Buy](#)

[Download](#)



Version: 2020.1.1
Build: 201.7223.92
6 May 2020

[System requirements](#)

[Installation Instructions](#)

[Other versions](#)

Download PyCharm

[Windows](#) [Mac](#) [Linux](#)

Professional

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

[Download](#)

Free trial

Community

For pure Python development

[Download](#)

Free, open-source



Get the Toolbox App to download PyCharm and its future updates with ease

[X] Cookies and IP addresses allow us to deliver personalized web content and to provide you with a better experience. Our website uses cookies and your IP address for these purposes.

<https://www.tensorflow.org/install>

Install TensorFlow

Install TensorFlow 2

pip install tensorflow

TensorFlow is tested and supported on the following 64-bit systems:

- Python 3.5–3.7
- macOS 10.12.6 (Sierra) or later (no GPU support)
- Ubuntu 16.04 or later
- Raspbian 9.0 or later
- Windows 7 or later

Download a package

Install TensorFlow with Python's *pip* package manager.

★ TensorFlow 2 packages require a [pip version >19.0](#).

Official packages available for Ubuntu, Windows, macOS, and the Raspberry Pi.

See the [GPU guide](#) for CUDA®-enabled cards.

[Read the pip install guide](#)

```
# Requires the latest pip
$ pip install --upgrade pip

# Current stable release for CPU and GPU
$ pip install tensorflow

# Or try the preview build (unstable)
$ pip install tf-nightly
```

Run a TensorFlow container

The [TensorFlow Docker images](#) are already configured to run TensorFlow. A [Docker](#) container runs in a virtual environment and is the easiest way to set up [GPU support](#).

Contents

- Install TensorFlow 2
- Download a package
- Run a TensorFlow container
- Google Colab: An easy way to learn and use TensorFlow
- Build your first ML app
 - Web developers
 - Mobile developers

Question 2 – What does “PIP” stand for?

API Documentation

TensorFlow has APIs available in several languages both for constructing and executing a TensorFlow graph. The Python API is at present the most complete and the easiest to use, but other language APIs may be easier to integrate into projects and may offer some performance advantages in graph execution.

A word of caution: the APIs in languages other than Python are not yet covered by the [API stability promises](#).

- [Python](#)
- [JavaScript](#)
- [C++](#)
- [Java](#)
- [Go](#)
- [Swift \(Early Release\)](#)

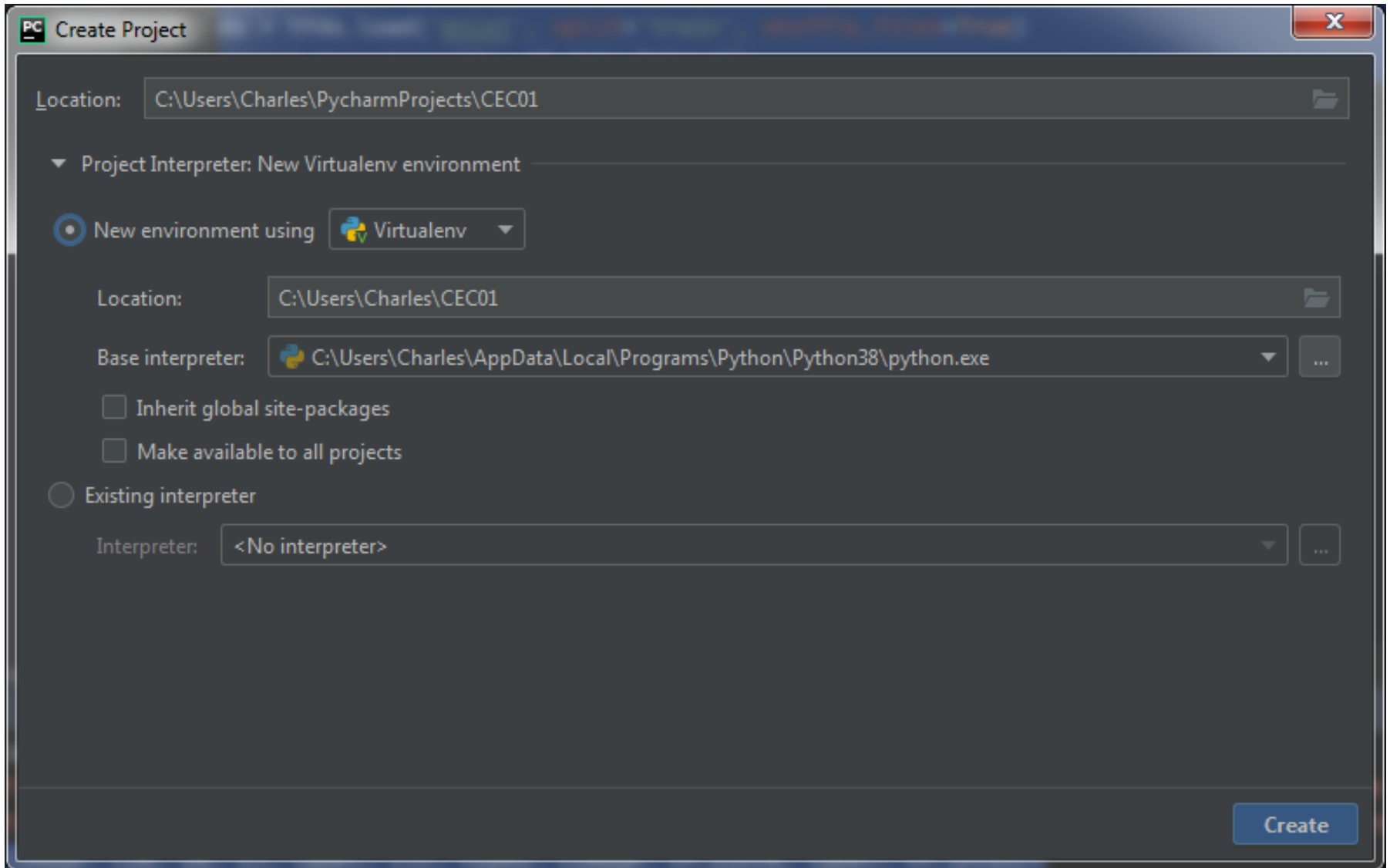
We encourage the community to develop and maintain support for other languages with the [approach recommended by the TensorFlow maintainers](#). For example, see the bindings for:

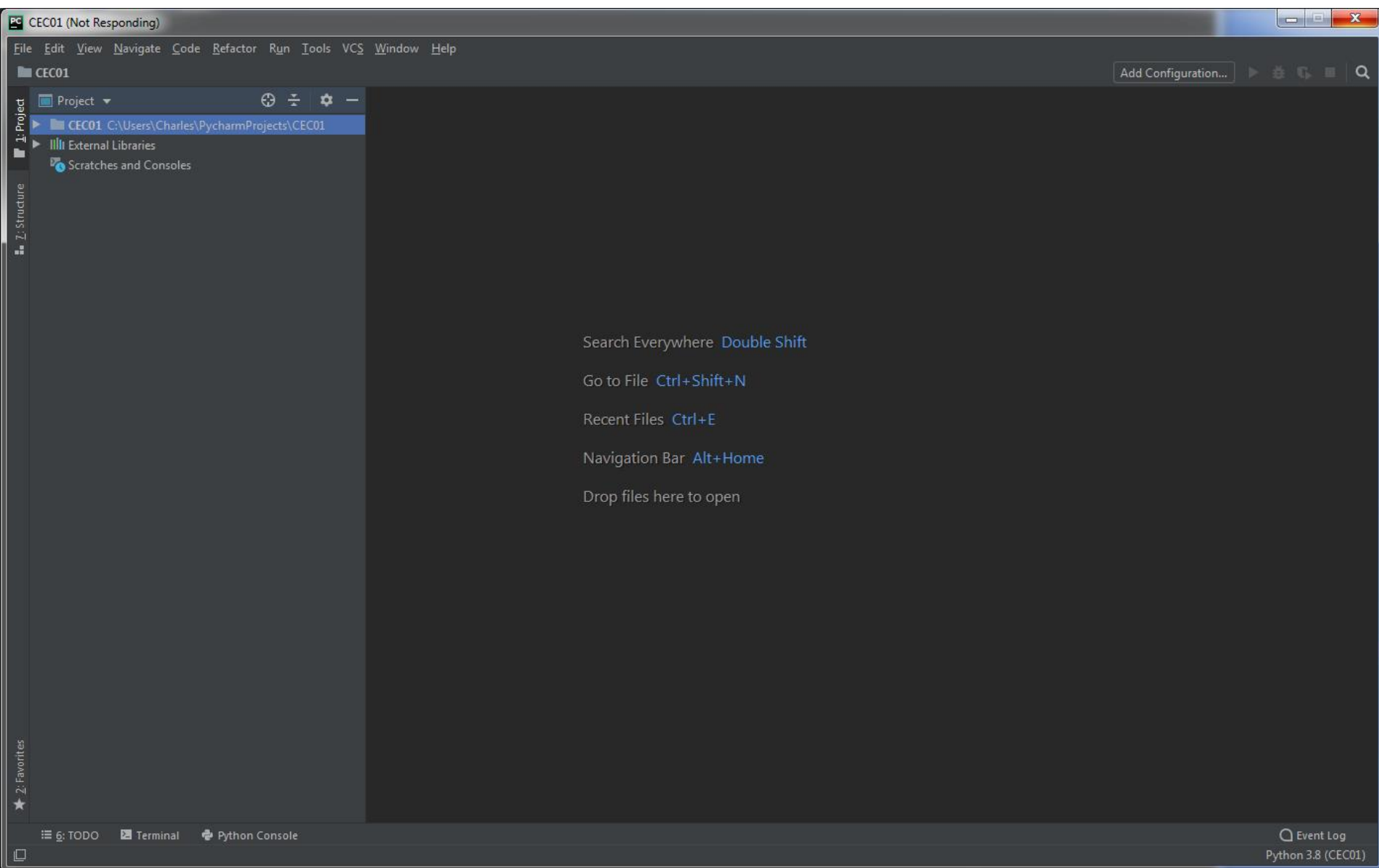
- C#: [TensorFlowSharp](#) and [TensorFlow.NET](#),
- [Haskell](#),
- [Julia](#),
- [MATLAB](#),
- [R](#),
- [Ruby](#),
- [Rust](#), and
- [Scala](#).

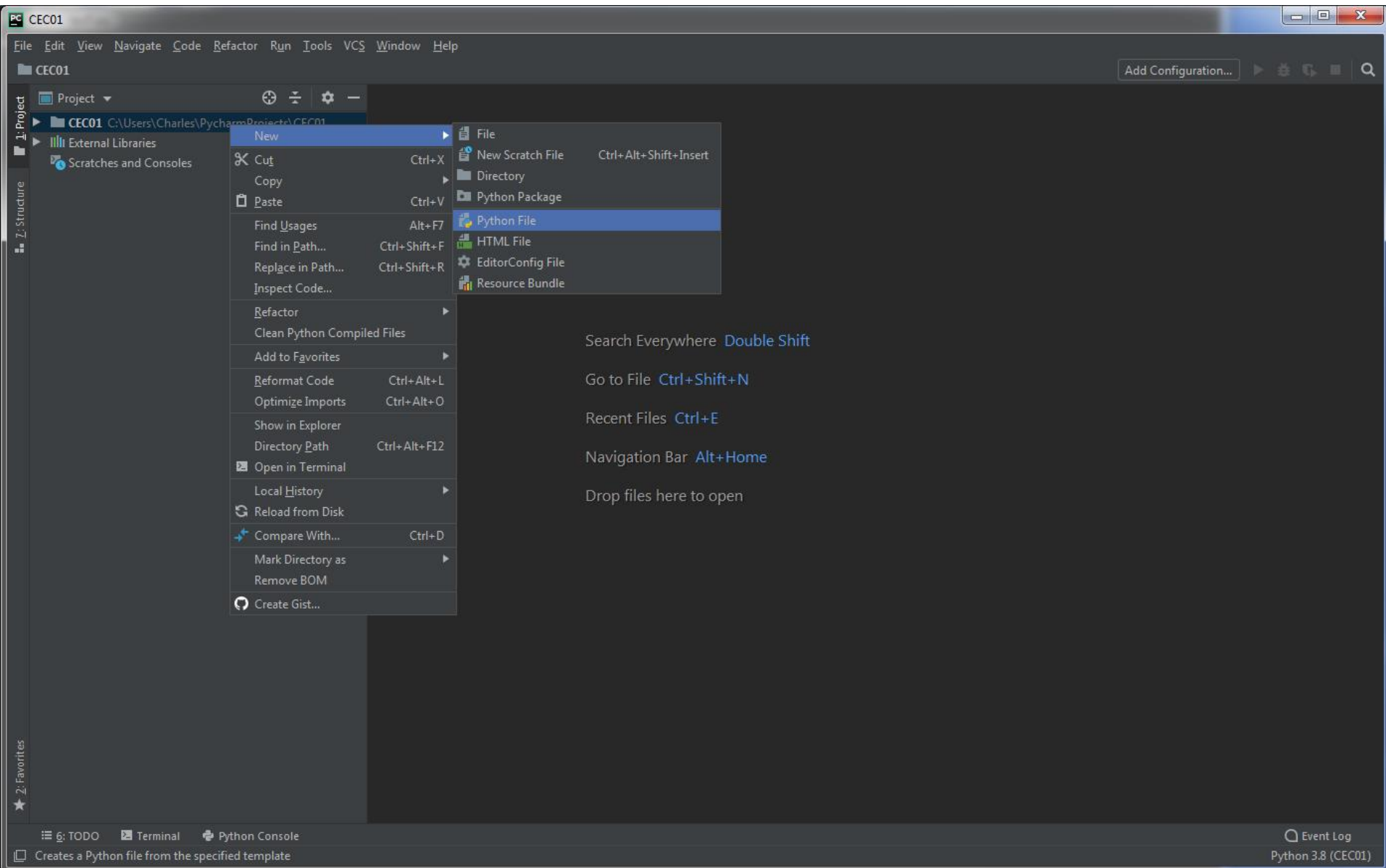
https://www.tensorflow.org/api_docs

We also provide the C++ API reference for TensorFlow Serving:

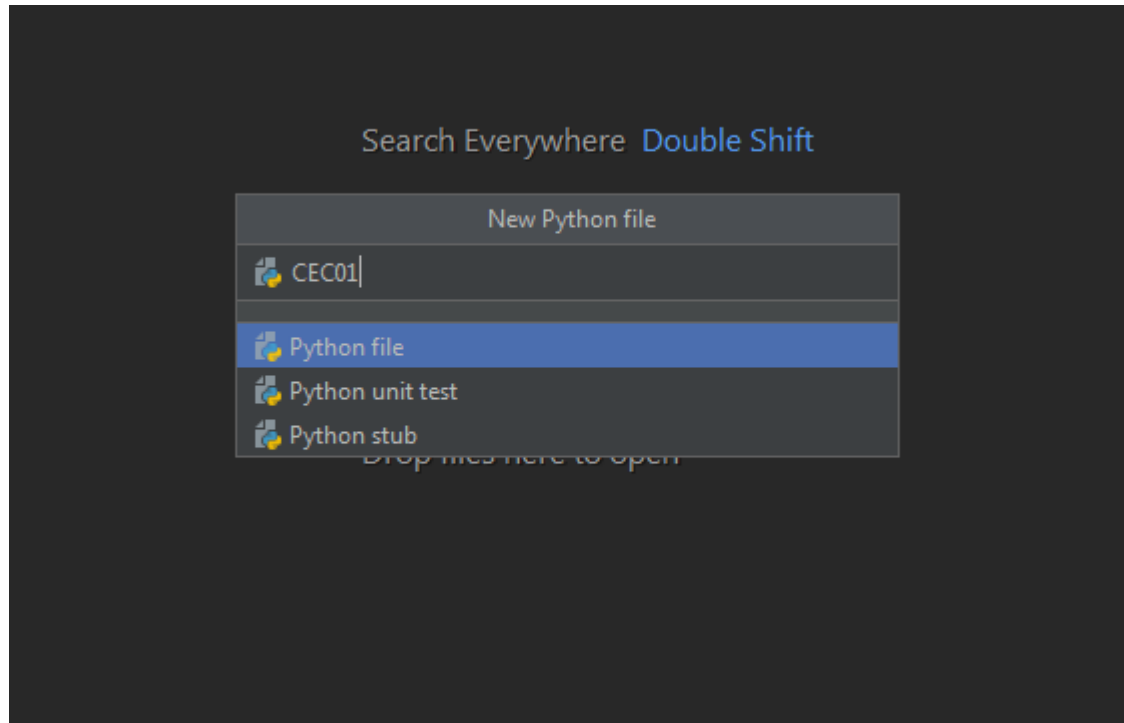
- [TensorFlow Serving](#)



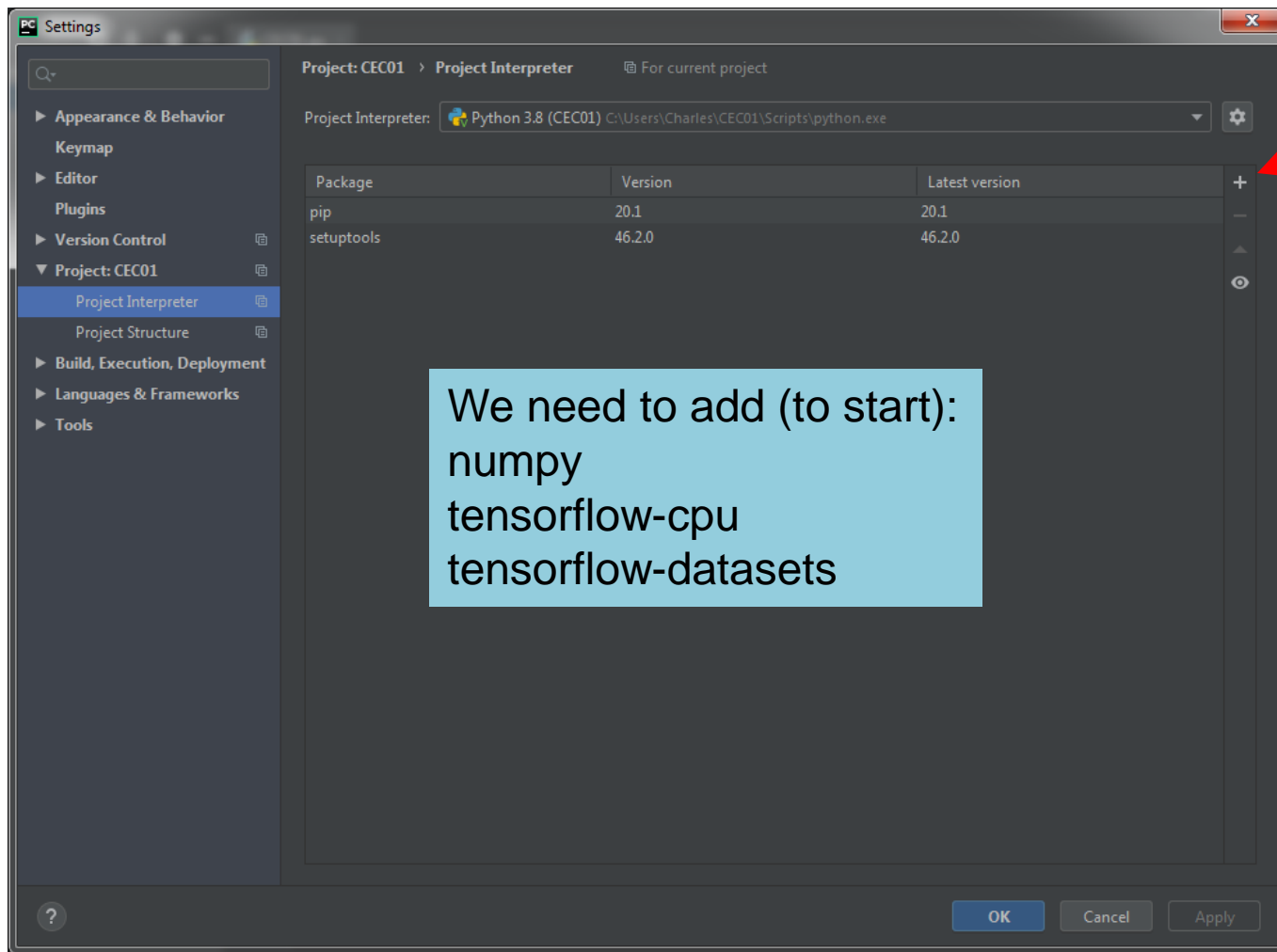


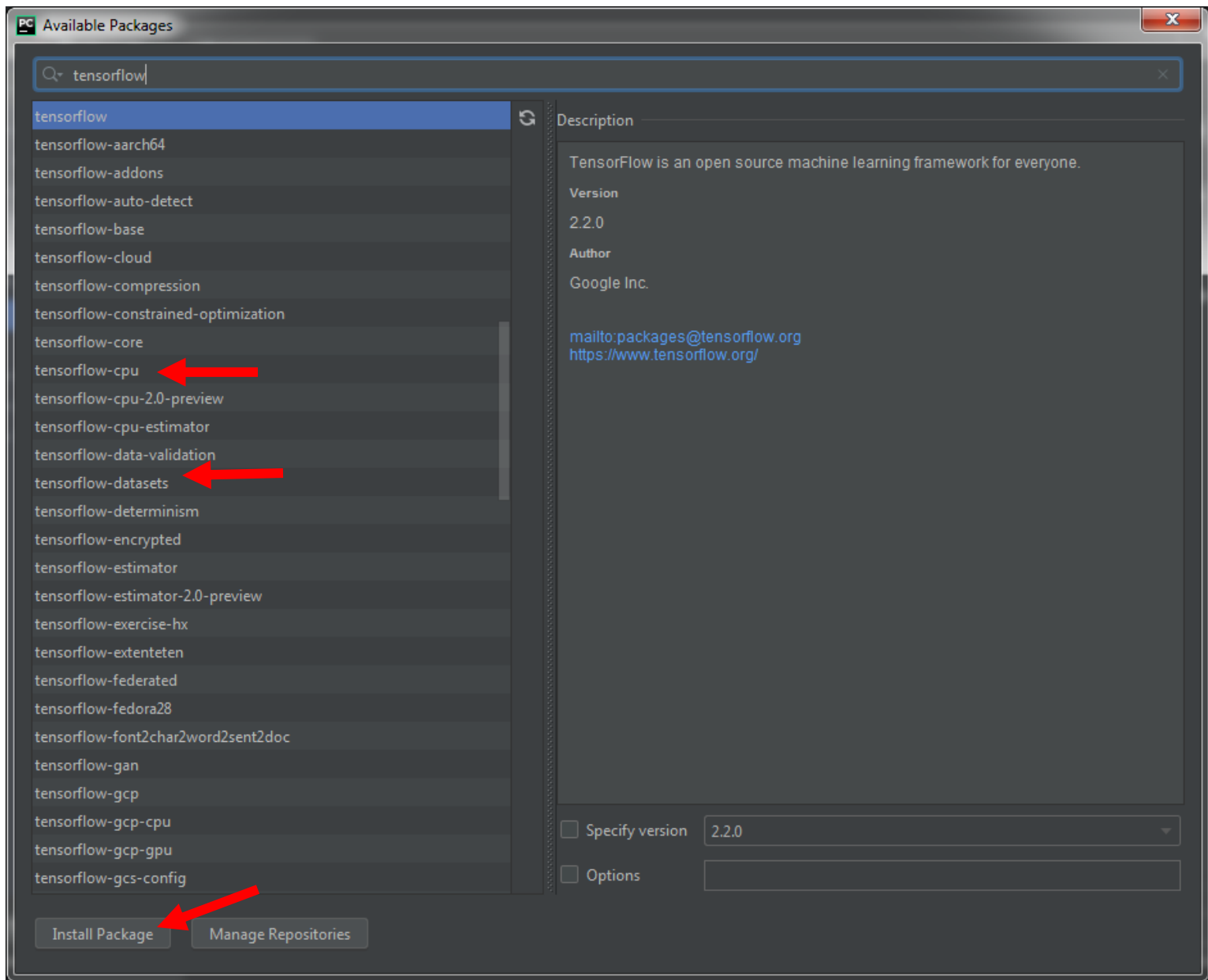


Name Our File (.py added automatically)



We Need Our Environment! Click on file, settings





PC Settings

Project: CEC01 > Project Interpreter For current project

Project Interpreter: Python 3.8 (CEC01) C:\Users\Charles\CEC01\Scripts\python.exe

Package	Version	Latest version
cachetools	4.1.0	4.1.0
certifi	2020.4.5.1	2020.4.5.1
chardet	3.0.4	3.0.4
dill	0.3.1.1	0.3.1.1
future	0.18.2	0.18.2
gast	0.3.3	0.3.3
google-auth	1.14.2	1.14.2
google-auth-oauthlib	0.4.1	0.4.1
google-pasta	0.2.0	0.2.0
googleapis-common-protos	1.51.0	1.51.0
grpcio	1.28.1	1.28.1
h5py	2.10.0	2.10.0
idna	2.9	2.9
numpy	1.18.4	1.18.4
oauthlib	3.1.0	3.1.0
opt-einsum	3.2.1	3.2.1
pip	20.1	20.1
promise	2.3	2.3
protobuf	3.11.3	3.11.3
pyasn1	0.4.8	0.4.8
pyasn1-modules	0.2.8	0.2.8
requests	2.23.0	2.23.0
requests-oauthlib	1.3.0	1.3.0
rsa	4.0	4.0
scipy	1.4.1	1.4.1
setuptools	46.2.0	46.2.0
six	1.14.0	1.14.0
tensorboard	2.2.1	2.2.1
tensorboard-plugin-wit	1.6.0.post3	1.6.0.post3
tensorflow-cpu	2.2.0	2.2.0
tensorflow-datasets	3.1.0	3.1.0
tensorflow-estimator	2.2.0	2.2.0
tensorflow-metadata	0.21.2	0.21.2
termcolor	1.1.0	1.1.0
tqdm	4.46.0	4.46.0
urllib3	1.25.9	1.25.9
wheel	0.34.2	0.34.2
wrapt	1.12.1	1.12.1

Package 'numpy' installed successfully

OK Cancel Apply

Short Sample Code

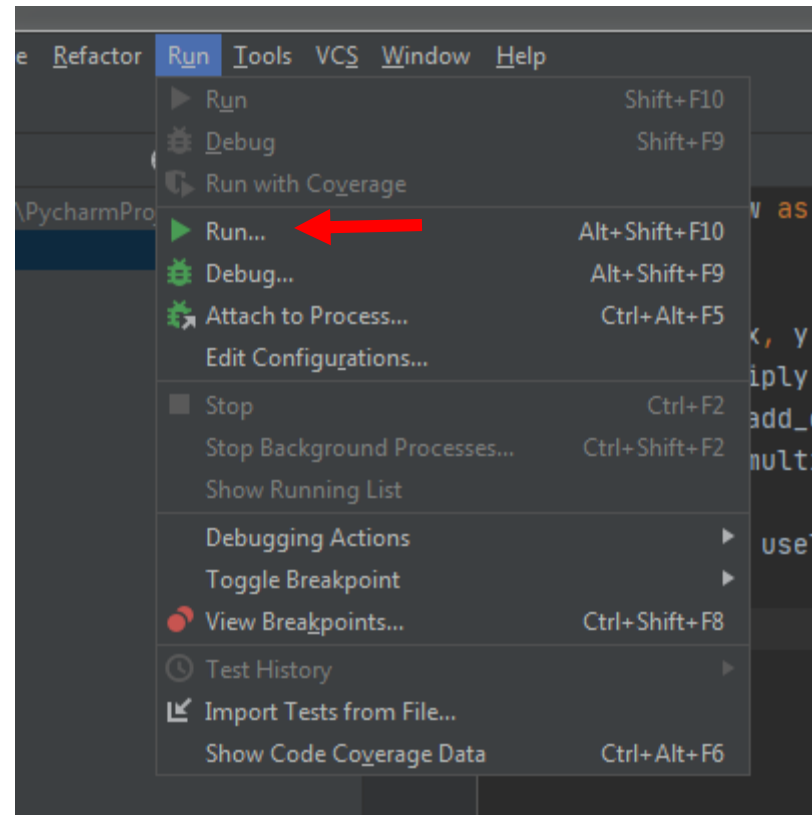
```
import tensorflow as tf
x = 2
y = 3
add_op = tf.add(x, y, name='Add')
mul_op = tf.multiply(x, y, name='Multiply')
pow_op = tf.pow(add_op, mul_op, name='Power')
useless_op = tf.multiply(x, add_op, name='Useless')

tens1 = [pow_op, useless_op]
print(tens1)
```

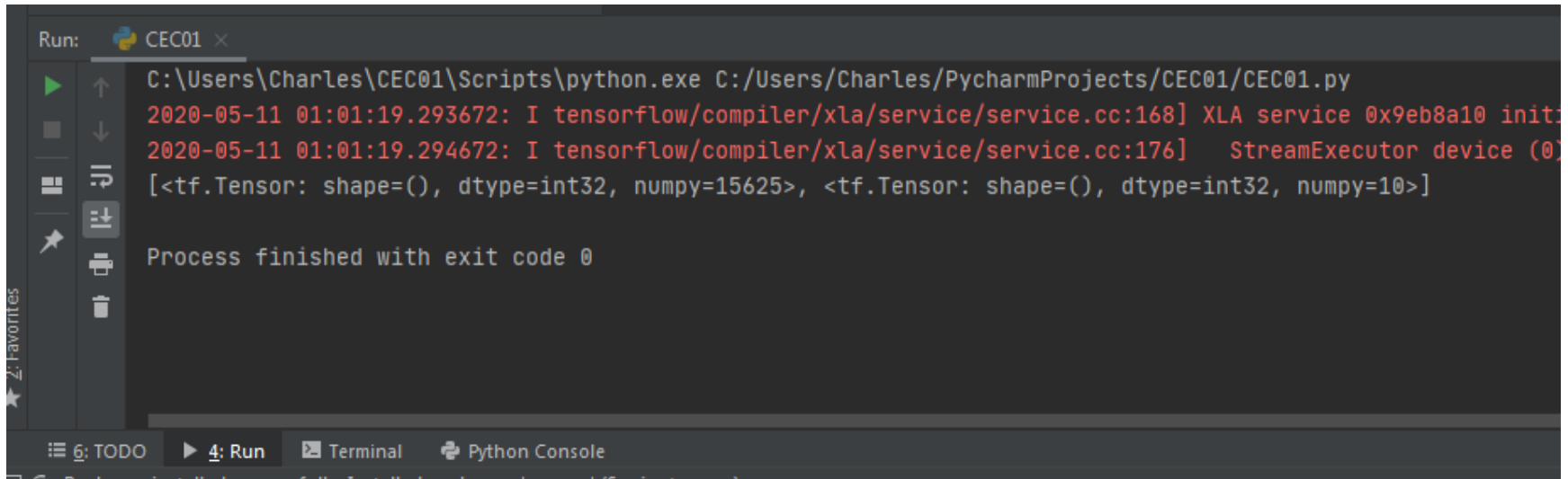

Color Coded!

```
CEC01.py x
1  import tensorflow as tf
2  x = 2
3  y = 3
4  add_op = tf.add(x, y, name='Add')
5  mul_op = tf.multiply(x, y, name='Multiply')
6  pow_op = tf.pow(add_op, mul_op, name='Power')
7  useless_op = tf.multiply(x, add_op, name='Useless')
8
9  tens1 = [pow_op, useless_op]
10 print(tens1)
11
```

Run , Run



Our Output



```
Run: CEC01 x
C:\Users\Charles\CEC01\Scripts\python.exe C:/Users/Charles/PycharmProjects/CEC01/CEC01.py
2020-05-11 01:01:19.293672: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x9eb8a10 initialized for platform cuda (0); no explicit device_id requested.
2020-05-11 01:01:19.294672: I tensorflow/compiler/xla/service/service.cc:176]   StreamExecutor device (0): cuda, 0
[<tf.Tensor: shape=(), dtype=int32, numpy=15625>, <tf.Tensor: shape=(), dtype=int32, numpy=10>]
Process finished with exit code 0
```

Question 3 – What is this output called and what is its shape?

Tomorrow!

- We will take what we learned on Tuesday about the mathematics of a CNN and apply it to TensorFlow
- We will also look at how the new enhancements to TF have sped up and simplified the definition of our network definition

This Week's Agenda

- 5/11 A Brief History of Artificial Neural Networks
- 5/12 Neural Network Simulation and Programming
- 5/13 TensorFlow Hands-On Part 1: Hello World!
- 5/14 TensorFlow Hands-On Part 2: Defining and Building Your Network
- 5/15 TensorFlow Hands-On Part 3: Teaching and Testing and Conclusion

Please stick around as I answer your questions!

- Please give me a moment to scroll back through the chat window to find your questions
- I will stay on chat as long as it takes to answer!
- I am available to answer simple questions or to consult (or offer in-house training for your company)

c.j.lord@ieee.org

<http://www.blueridgetechn.com>

<http://www.linkedin.com/in/charleslord>

Twitter: @charleslord

<https://www.github.com/bradatrainng>