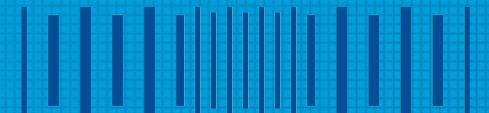


# bit.ly/pystarters

**SWC PyStarters - Introduction to  
scientific programming in Python.**

May 6<sup>th</sup> & 13<sup>th</sup>, 10am-4pm



**Sainsbury Wellcome Centre**

# Schedule

- May 6<sup>th</sup> AM – Installing Python (Adam)
- May 6<sup>th</sup> PM – Introduction to Python (Steve)
- May 13<sup>th</sup> AM & PM– Advanced Python (Joaquin)
- May 13<sup>th</sup> PM– Wrap up & questions (Adam & Steve)

# Structure

- Main zoom room for tutorial
- Breakout rooms for group problems & one to one help
- SWC slack (#pystarters) for questions
  - Zoom chat if you haven't got access yet
- TA will answer your question, or move to breakout room.

TAs:

- Steve, Joaquin & me (when not teaching)
- Federico Claudi, Spencer Wilson, & Lars Rollik

(For admin/zoom, otherwise non-programming questions, ask me)

# Why learn Python?

## Popularity

- Most popular language for data science <sup>a</sup>
- Fastest-growing major language <sup>a</sup>

# Why learn Python?

## Free and open source

- Unlike MATLAB, IGOR etc
- Saves your lab money
- More importantly, anyone can use your code (4.3M<sup>2</sup> open Python repositories on GitHub)

# Why learn Python?

## Versatility

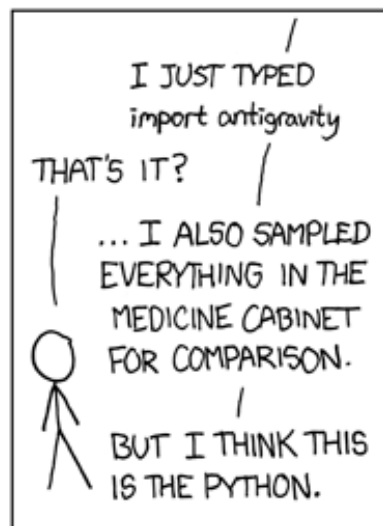
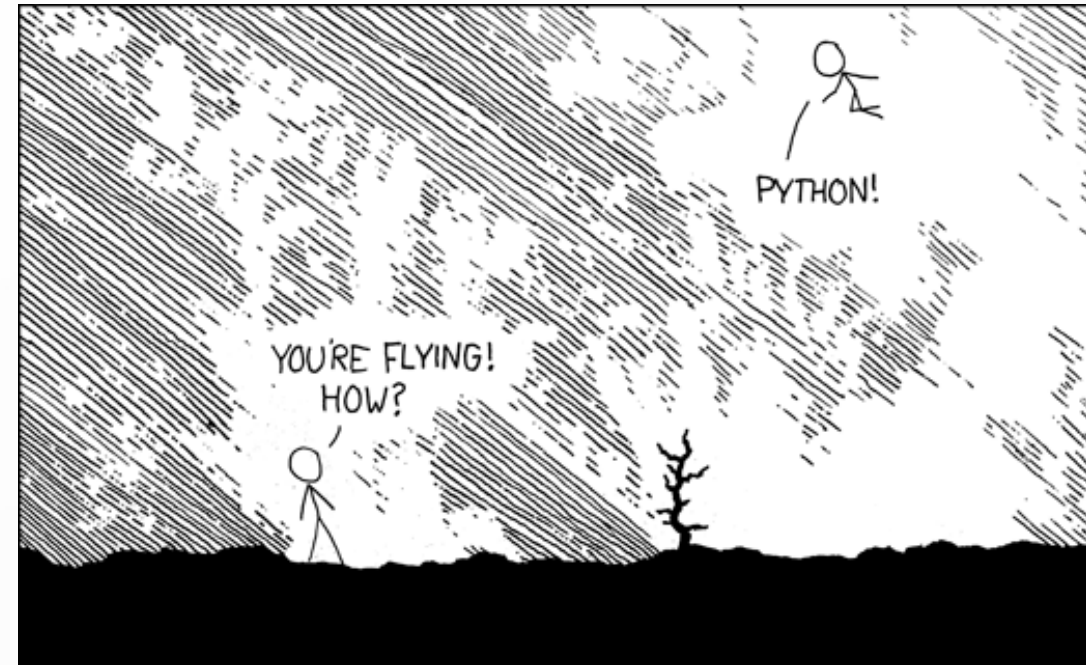
- Not just data analysis / machine learning
  - Visualisation
  - Data acquisition
  - Web development
  - Etc etc ...



# Why learn Python?

## Packages for everything

- Numpy – numerical computing
- Scipy – scientific computing
- Scikit-image – image analysis
- TensorFlow – deep learning
- Matplotlib – plotting
- Etc etc ...



# Downsides

## Installation

- Not a single program to download
- Potentially complicated ecosystem (if used to e.g. MATLAB)
  - Python versions
  - Installing packages
  - Virtual environments



# Installing Python

- Download miniconda installer (link on website)
- Install miniconda (varies with OS)

## Windows

- Double click on installer
- Accept all defaults, BUT
- Check “**Add Anaconda to my PATH environment variable.**”

## MacOS

- Double click on installer
- Accept all default options

## Linux

- cd Downloads
- bash Miniconda3-latest-Linux-x86\_64.sh
- Accept defaults
- Press enter to prepend miniconda to PATH (will be your default Python)

# Conda environments

- Separate from each other (and system python)
- Allow for incompatible dependencies
- Reproducible workflows, share with:
  - Collaborators
  - Readers of your paper
  - Yourself (HPC etc.)

# Creating an environment

- Open a terminal (anaconda prompt on Windows)
- `conda create -name pystarters python=3.9`
- Accept “y” for yes
- `conda activate pystarters`

You can make as many different environments as you like, but remember to “activate”

# Installing packages

`pip install numpy`

Arrays & maths etc

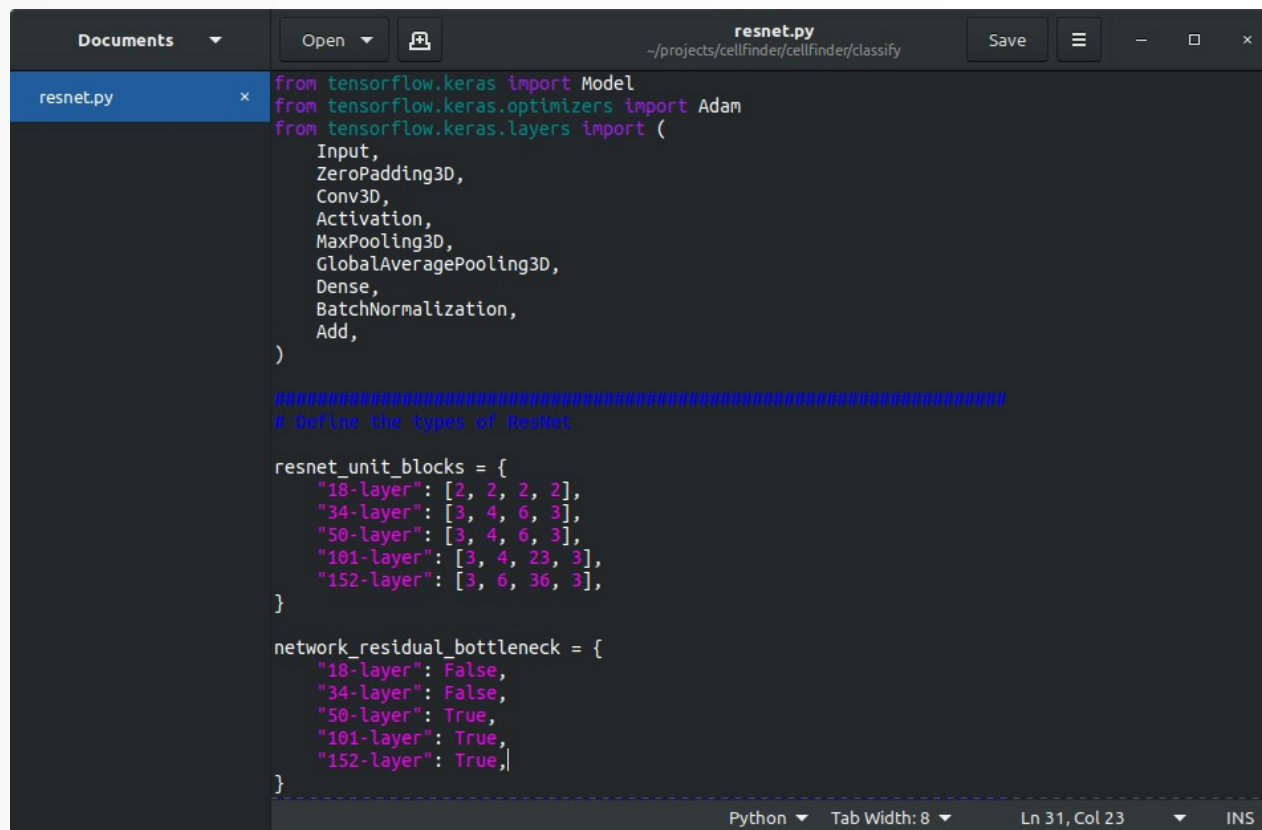


`pip install matplotlib`

Plotting



# Editing files



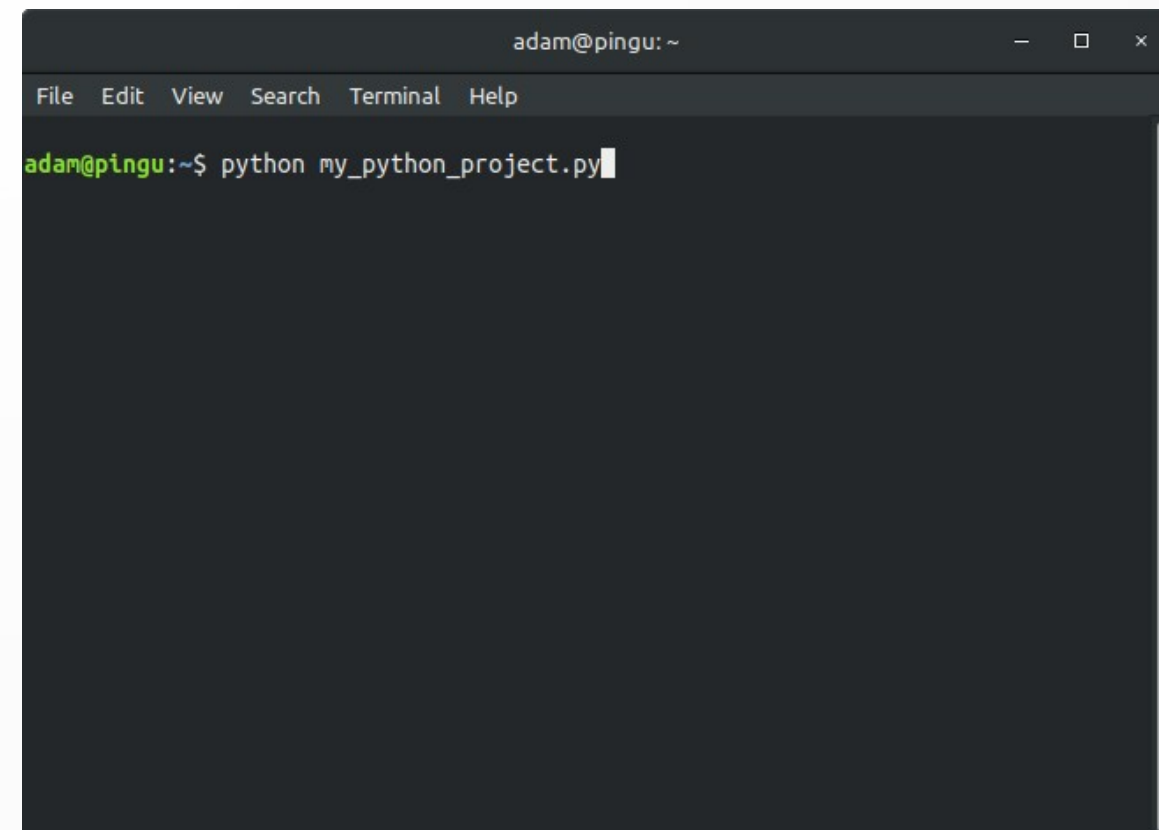
A screenshot of a code editor window titled "resnet.py" with a file path of "~/projects/cellfinder/cellfinder/classify". The editor shows Python code for a ResNet model. The code includes imports for tensorflow.keras and tensorflow.keras.layers, followed by a list of layers to import. It then defines a dictionary for ResNet unit blocks and a dictionary for network residual bottlenecks. The status bar at the bottom indicates "Python", "Tab Width: 8", "Ln 31, Col 23", and "INS" mode.

```
from tensorflow.keras import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import (
    Input,
    ZeroPadding3D,
    Conv3D,
    Activation,
    MaxPooling3D,
    GlobalAveragePooling3D,
    Dense,
    BatchNormalization,
    Add,
)

#####
# Define the types of ResNet

resnet_unit_blocks = {
    "18-layer": [2, 2, 2, 2],
    "34-layer": [3, 4, 6, 3],
    "50-layer": [3, 4, 6, 3],
    "101-layer": [3, 4, 23, 3],
    "152-layer": [3, 6, 36, 3],
}

network_residual_bottleneck = {
    "18-layer": False,
    "34-layer": False,
    "50-layer": True,
    "101-layer": True,
    "152-layer": True,
}
```



A screenshot of a terminal window titled "adam@pingu: ~". The terminal shows the command "python my\_python\_project.py" being entered at the prompt "adam@pingu:~\$".

```
adam@pingu:~$ python my_python_project.py
```

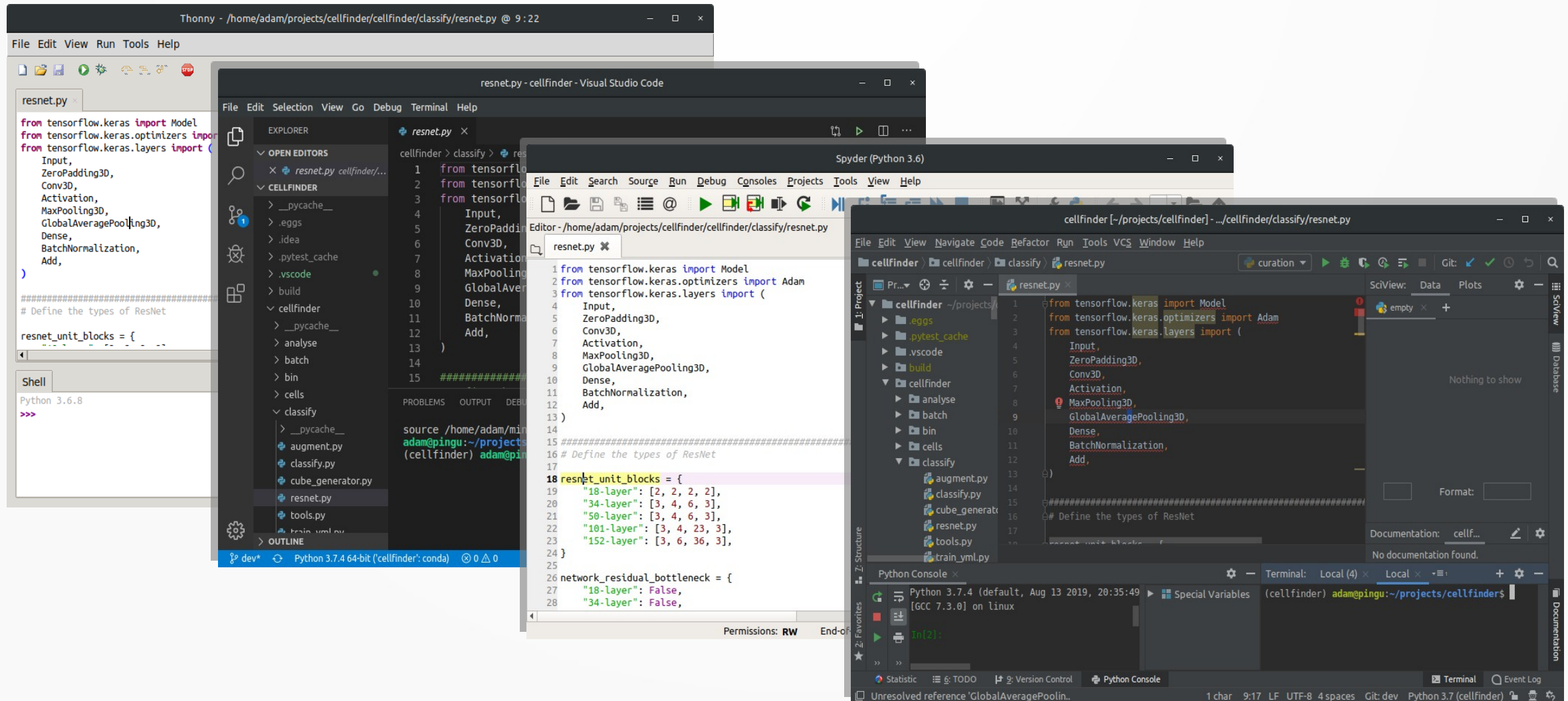
# IDEs (Integrated Development Environments)

- Friendly to beginners, and used by most developers <sup>a</sup>
- Edit, organise and run your code in one program
- Configure once for each project.
- But ... can be unnecessarily complicated

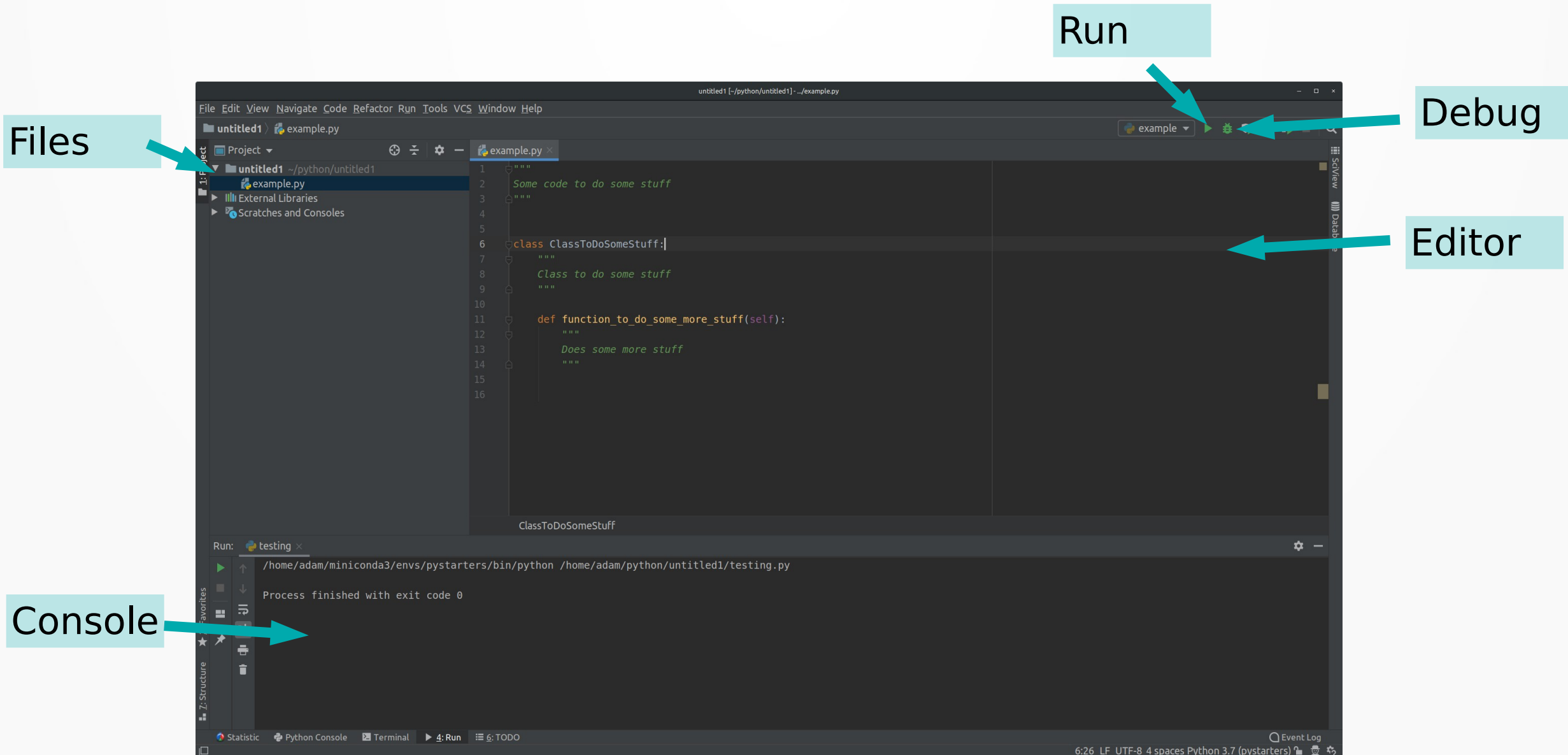


# IDEs

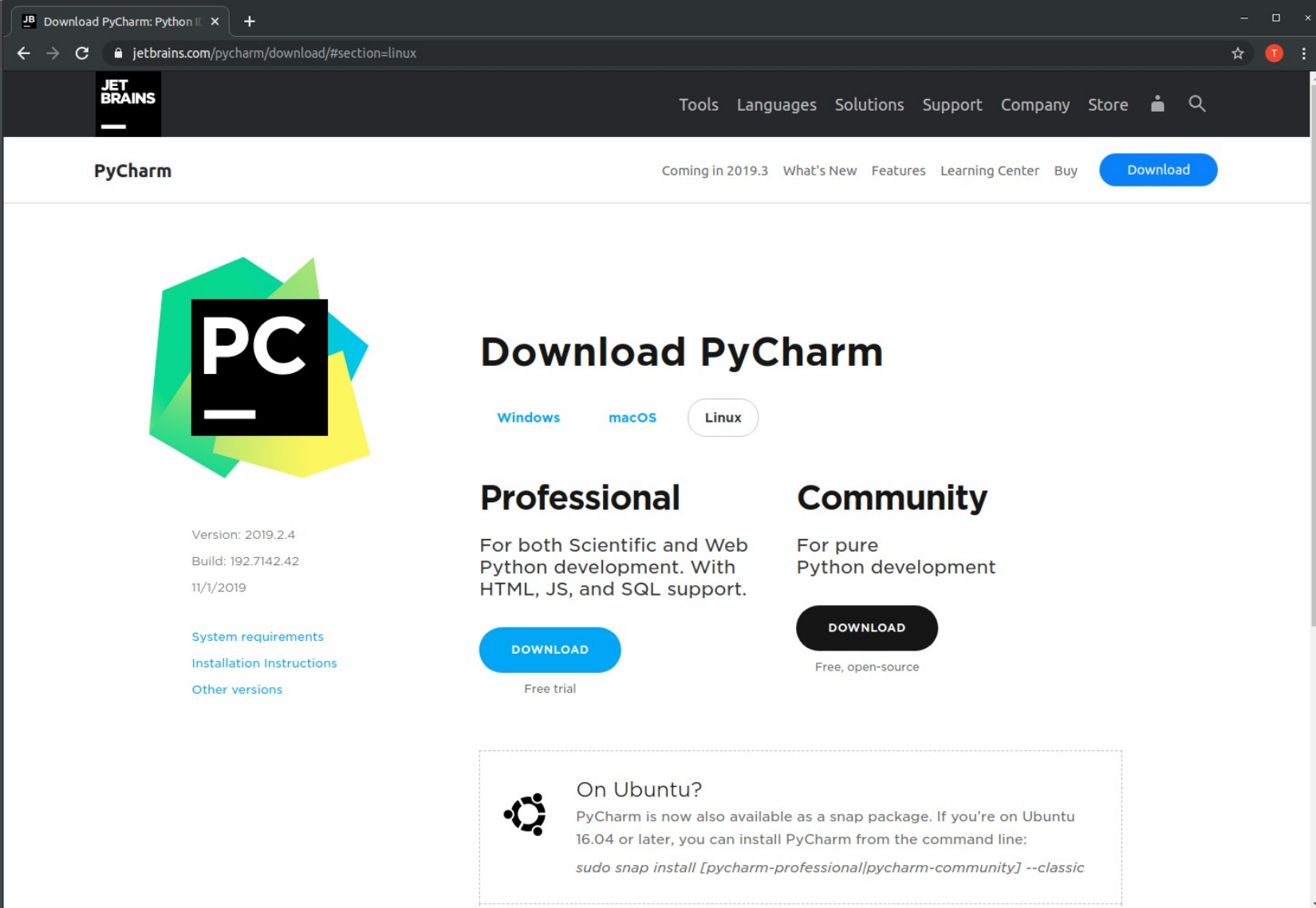
Everyone has a preference (but essentially all the same).



# PyCharm



# Download



The screenshot shows the JetBrains PyCharm download page in a web browser. The browser's address bar shows the URL `jetbrains.com/pycharm/download/#section=linux`. The page features the JetBrains logo and a navigation menu with links to Tools, Languages, Solutions, Support, Company, and Store. The main heading is "PyCharm", followed by a "Download" button. Below this, the PyCharm logo is displayed, along with version information: "Version: 2019.2.4", "Build: 192.7142.42", and "11/1/2019". Links for "System requirements", "Installation Instructions", and "Other versions" are provided. The page is divided into two columns for "Professional" and "Community" editions. The Professional edition is described as "For both Scientific and Web Python development. With HTML, JS, and SQL support." and includes a "DOWNLOAD" button with a "Free trial" note. The Community edition is described as "For pure Python development" and includes a "DOWNLOAD" button with a "Free, open-source" note. At the bottom, a section titled "On Ubuntu?" provides information about installing PyCharm as a snap package, including the command `sudo snap install [pycharm-professional/pycharm-community] --classic`.

Download PyCharm: Python II x


jetbrains.com/pycharm/download/#section=linux

JETBRAINS

Tools Languages Solutions Support Company Store

PyCharm

Coming in 2019.3 What's New Features Learning Center Buy [Download](#)



Version: 2019.2.4  
Build: 192.7142.42  
11/1/2019

[System requirements](#)  
[Installation Instructions](#)  
[Other versions](#)

## Download PyCharm

[Windows](#) [macOS](#) [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

#### On Ubuntu?

PyCharm is now also available as a snap package. If you're on Ubuntu 16.04 or later, you can install PyCharm from the command line:

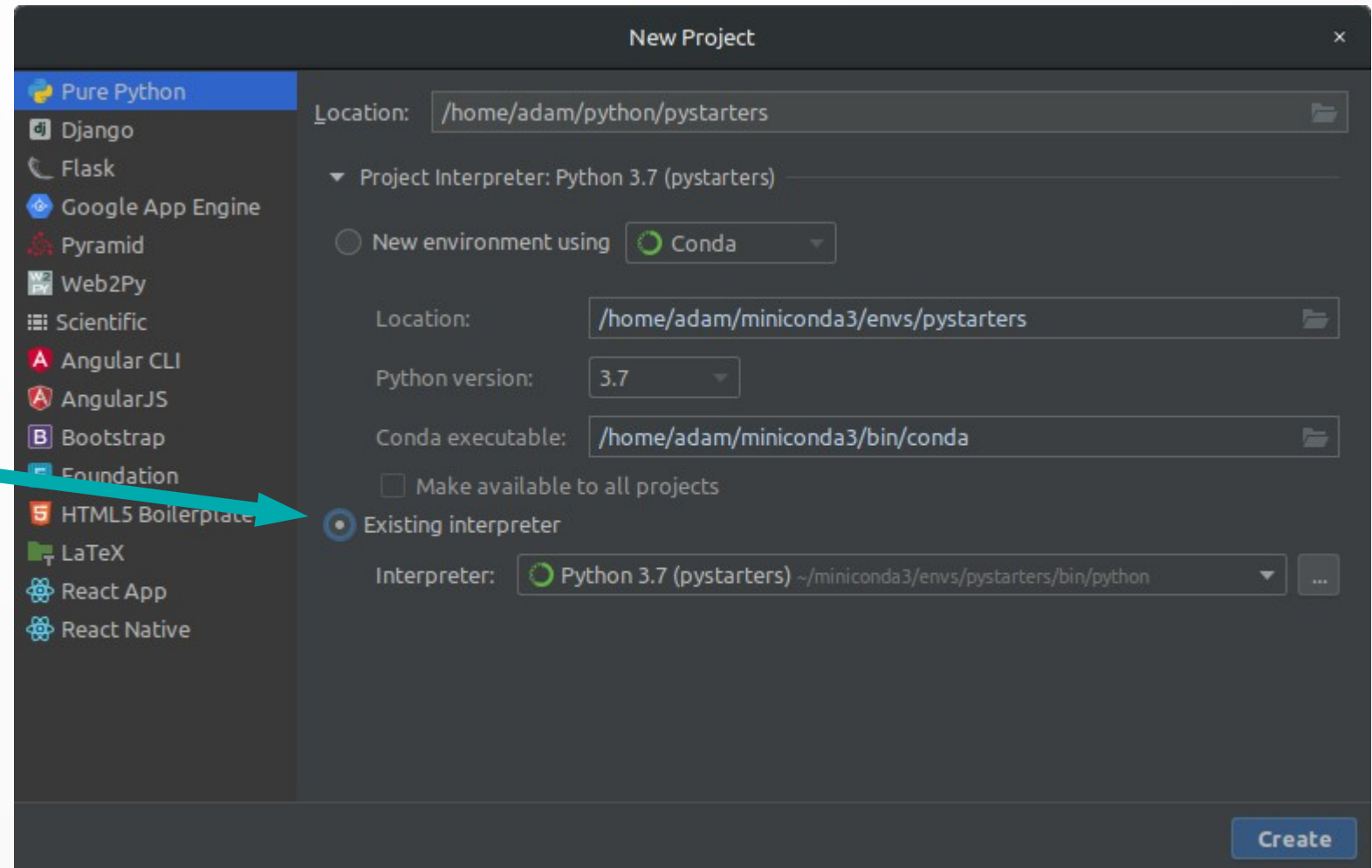
```
sudo snap install [pycharm-professional/pycharm-community] --classic
```



# New project

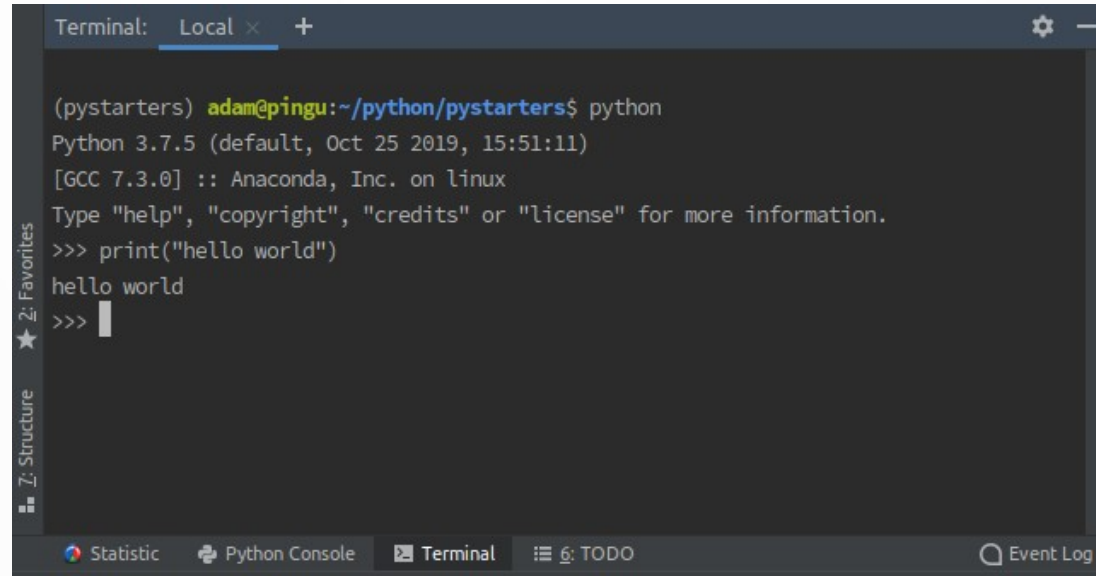
File → New Project

Important to set  
interpreter



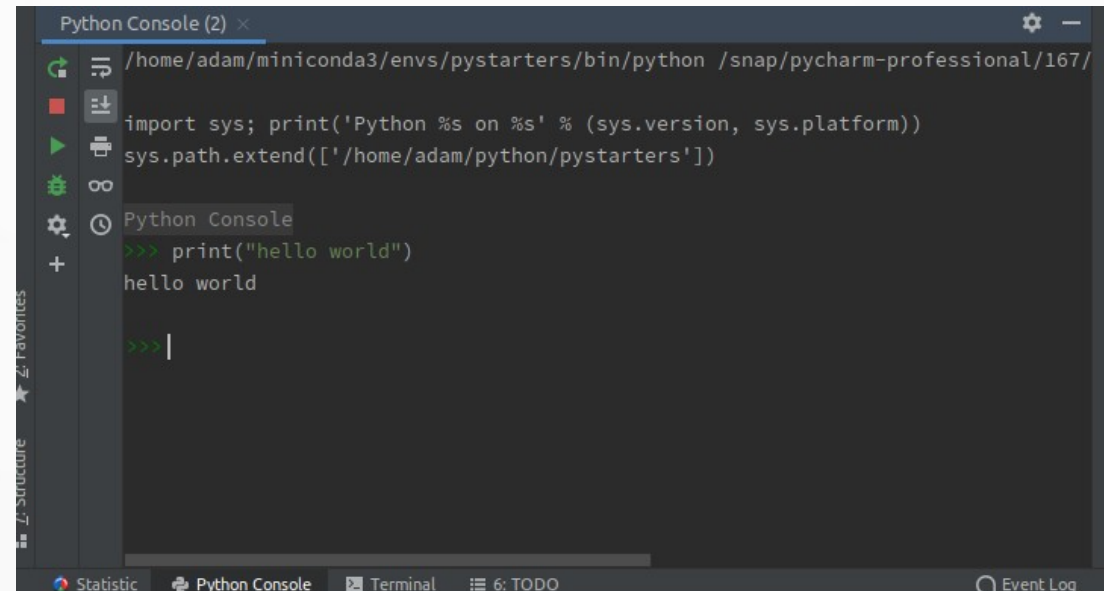
# Hello world

## Terminal



```
Terminal: Local x +
(pystarters) adam@pingu:~/python/pystarters$ python
Python 3.7.5 (default, Oct 25 2019, 15:51:11)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello world")
hello world
>>> 
```

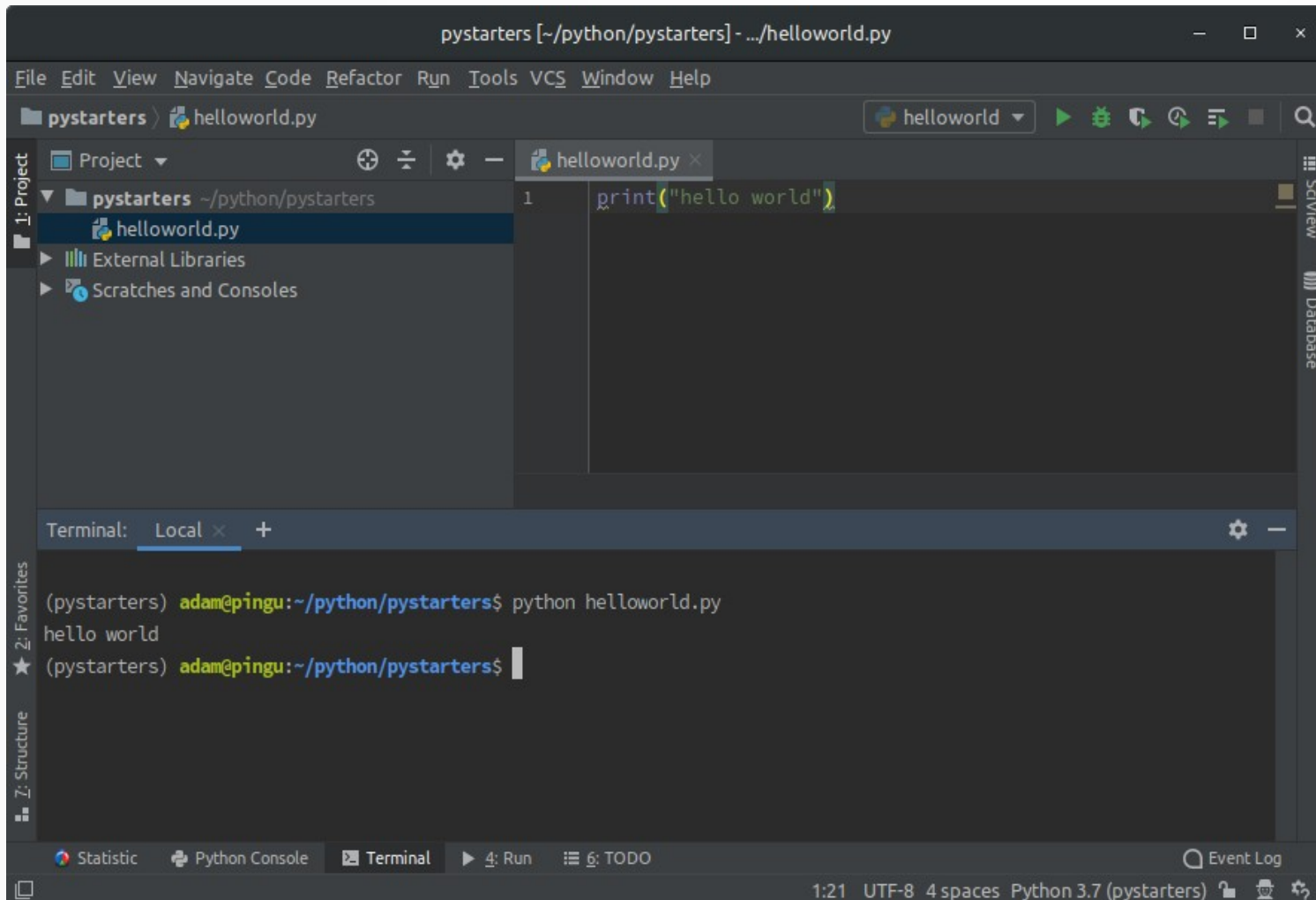
## Python console



```
Python Console (2) x
/home/adam/miniconda3/envs/pystarters/bin/python /snap/pycharm-professional/167/
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/home/adam/python/pystarters'])
Python Console
>>> print("hello world")
hello world
>>> 
```

# Hello world

File → New file → “helloworld.py”



Terminal:  
`python helloworld.py`

Or:  
Right click →  
Run 'helloworld'