

<https://gpfai.net/sice/>

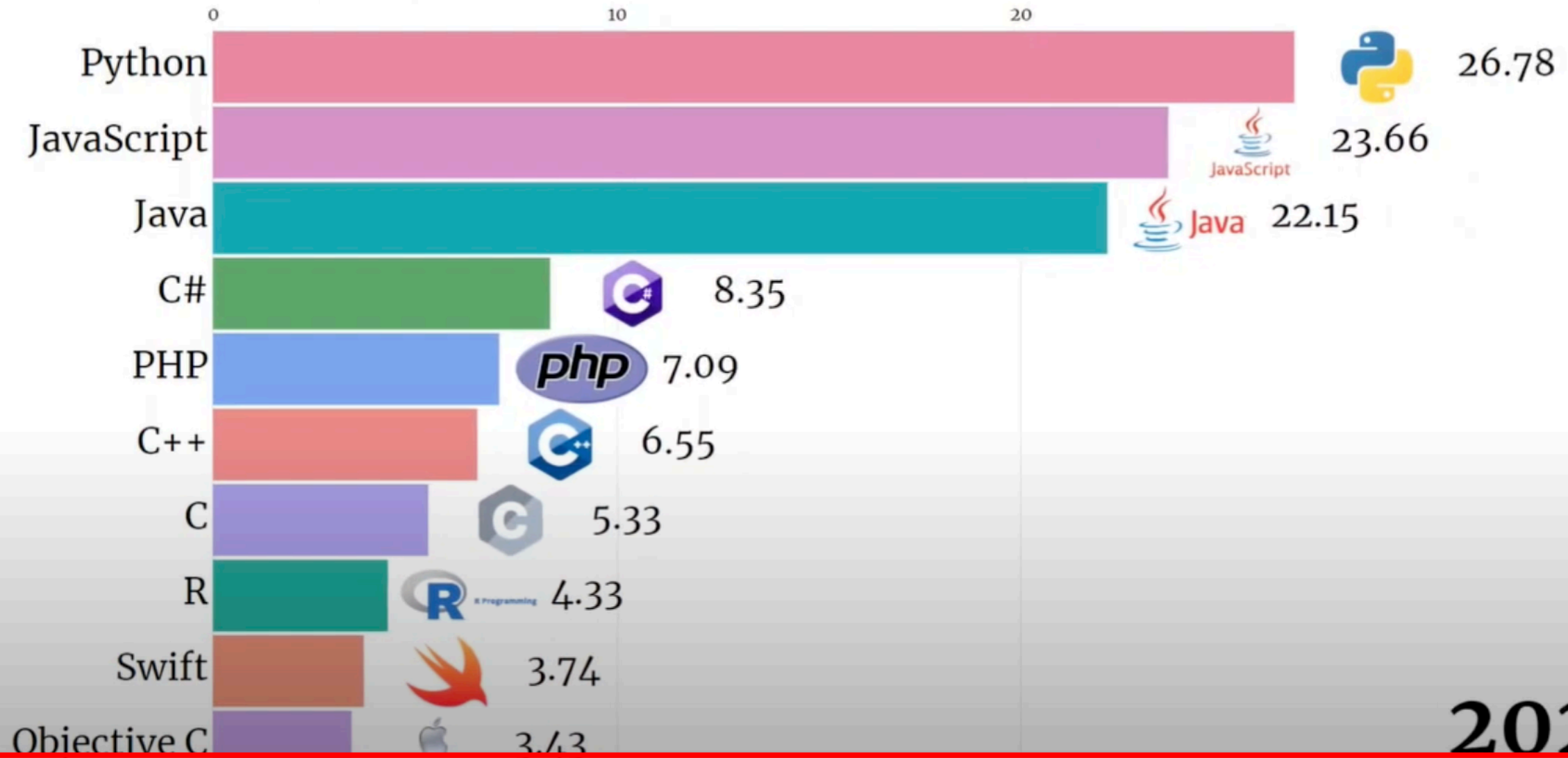
Software implementation and computation Python Programming

An introduction to Python and Python modules

Most Popular Programming Languages

1965-2021

Y axis is a relative value to define ranking popularity between all other items.



2021



開發者 JetBrains

初始版本 2010年7月，9年前

穩定版本 2018.3.4（2019年1月30日，12個月前）

預覽版本 2019.1 EAP build
191.4212.43（2019年1月24日，12個月前）

程式語言 Java, Python

作業系統 Windows, macOS, Linux

文件大小 174~270 MB

類型 集成開發環境

許可協議 Freemium

網站 jetbrains.com/pycharm/ 

PyCharm[Coming in 2020.1](#) [What's New](#) [Features](#) [Learning Center](#) [Buy](#)[Download](#)

PyCharm

The Python IDE
for Professional
Developers

[DOWNLOAD](#)

Full-fledged Professional or Free Community

[Python](#)[PSF](#)[Docs](#)[PyPI](#)[Jobs](#)[Community](#)[Donate](#)[GO](#)[Socialize](#)[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: Simple arithmetic
>>> 1 / 2
0.5
>>> 2 ** 3
8
>>> 17 / 3 # classic division returns a
float
5.666666666666667
>>> 17 // 3 # floor division
5
```

Intuitive Interpretation

Calculations are simple with Python, and expression syntax is straightforward: the operators `+`, `-`, `*` and `/` work as expected; parentheses `()` can be used for grouping. [More about simple math functions in Python 3.](#)

[1](#)[2](#)[3](#)[4](#)[5](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)

Welcome to PyCharm

py_code_2019
~/Desktop/py_code_2019

cifar10-GPU
~/Desktop/py_...9/cifar10-GPU

keras_ex1
~/Desktop/keras_ex1

cifar_10
~/Desktop/py_..._2019/cifar_10

keras_ex2
~/Desktop/py_...019/keras_ex2

test-GPU
~/Desktop/py..._2019/test-GPU

keras_ex3_matlab2keras
~/Desktop/py...3_matlab2keras



PyCharm

Version 2019.1.3

+ Create New Project

Open

Check out from Version Control ▾

⚙ Configure ▾ Get Help ▾

image3d.png



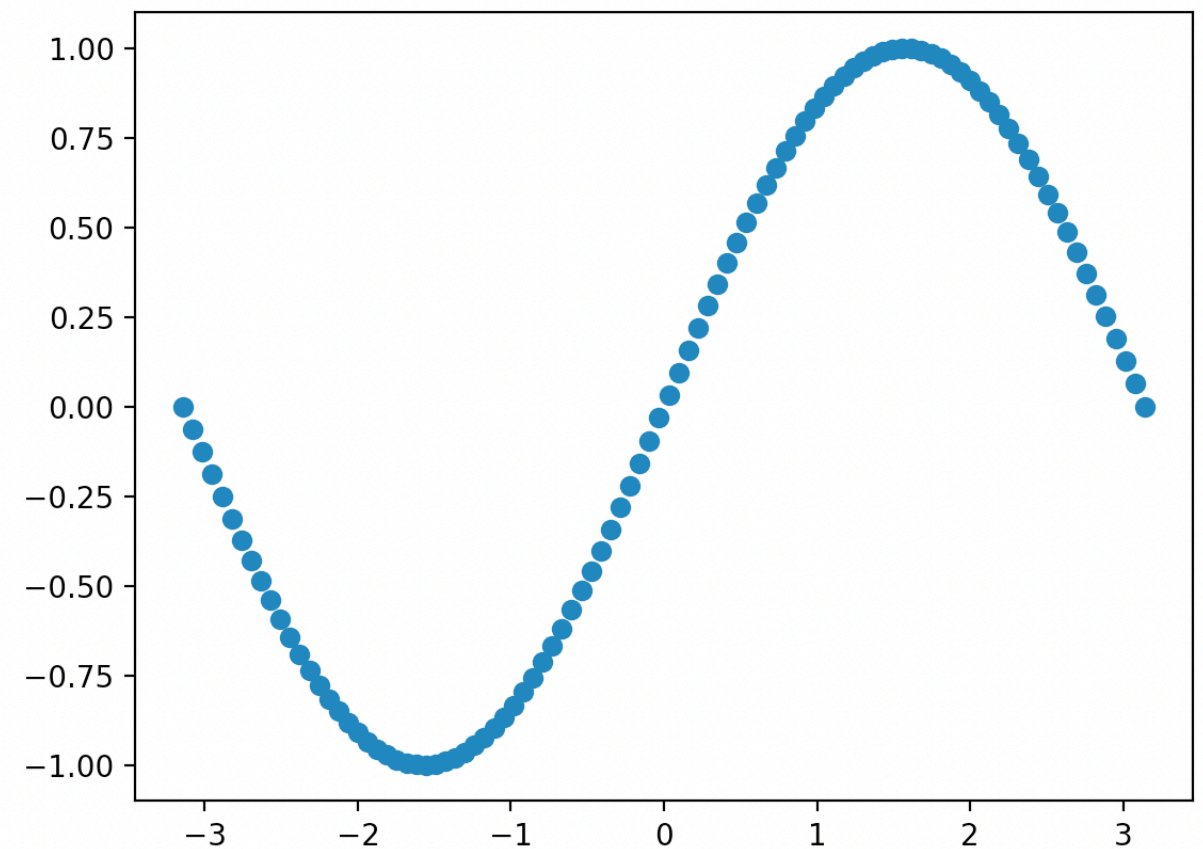
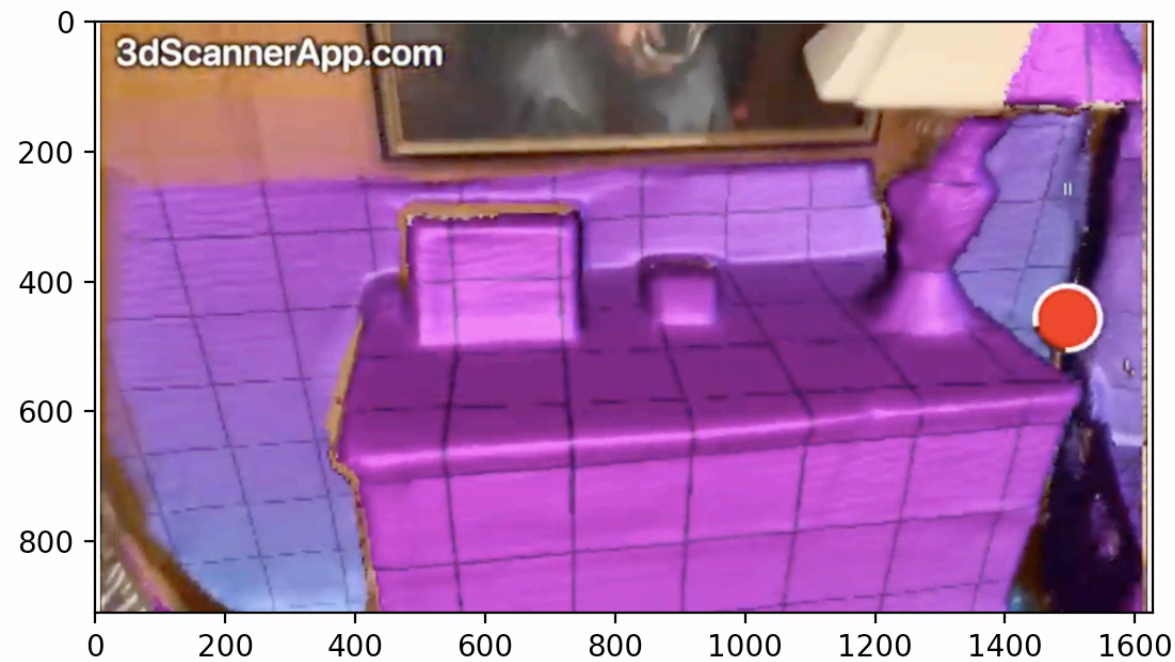
EX1: Read and Show an image

- Use PyCharm to new a project
- Place image3d.png to the project directory
- Install package matplotlib
- New a python file and write code to read and display an image

plt represents a
module

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
I = mpimg.imread('image3d.png')
plt.imshow(I)
plt.show()
```


Ex2: show an image and a sin plot



Ex2: show an image and a sin plot

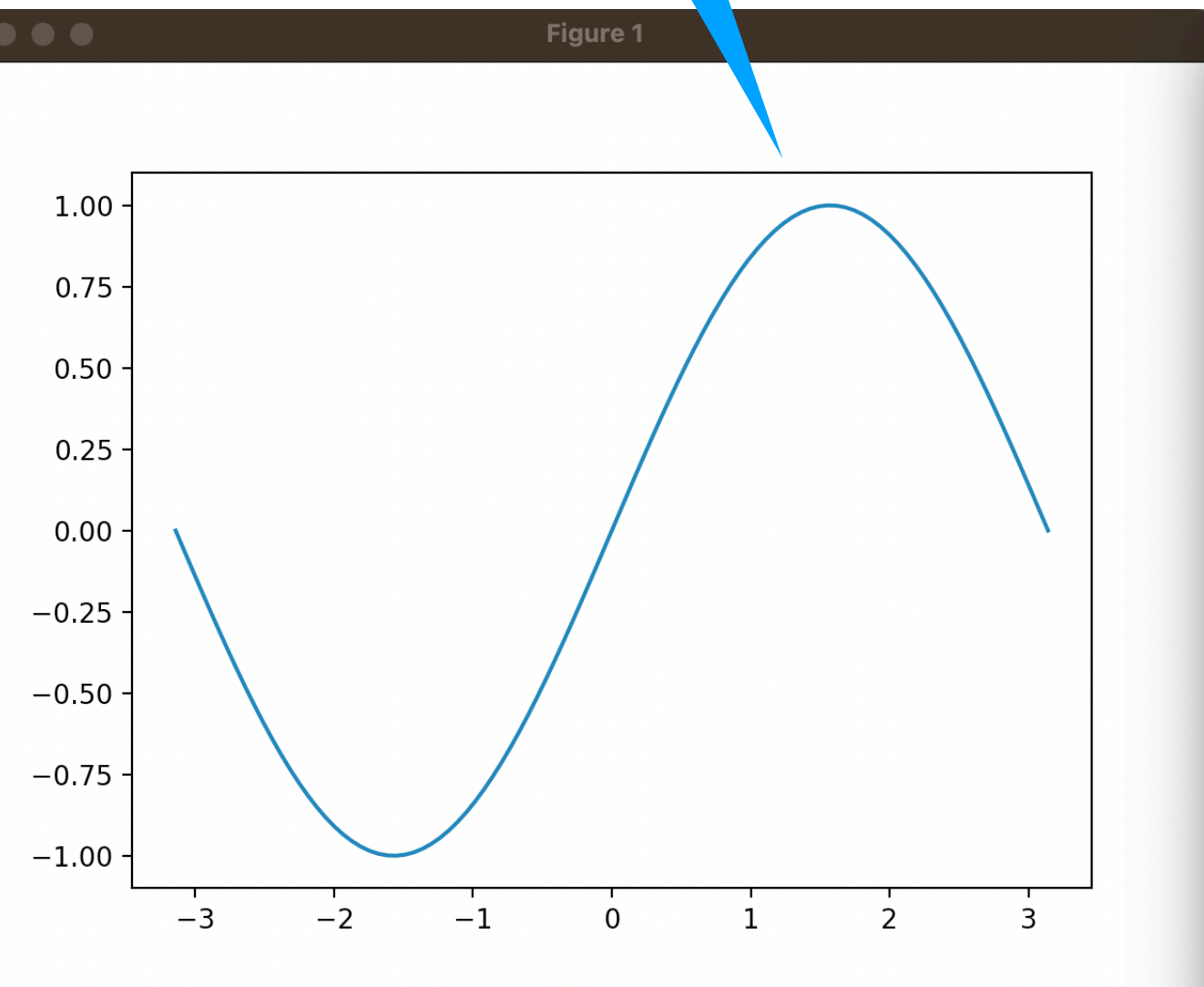
- Steps in Ex1
- Install numpy
- Partition $[-\pi, \pi]$ to 100 points and store them to x
- Set y to sin of x
- Plot points defined by x and y


```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
plt.figure(1)
plt.imshow(mpimg.imread('image3d.png'))
```

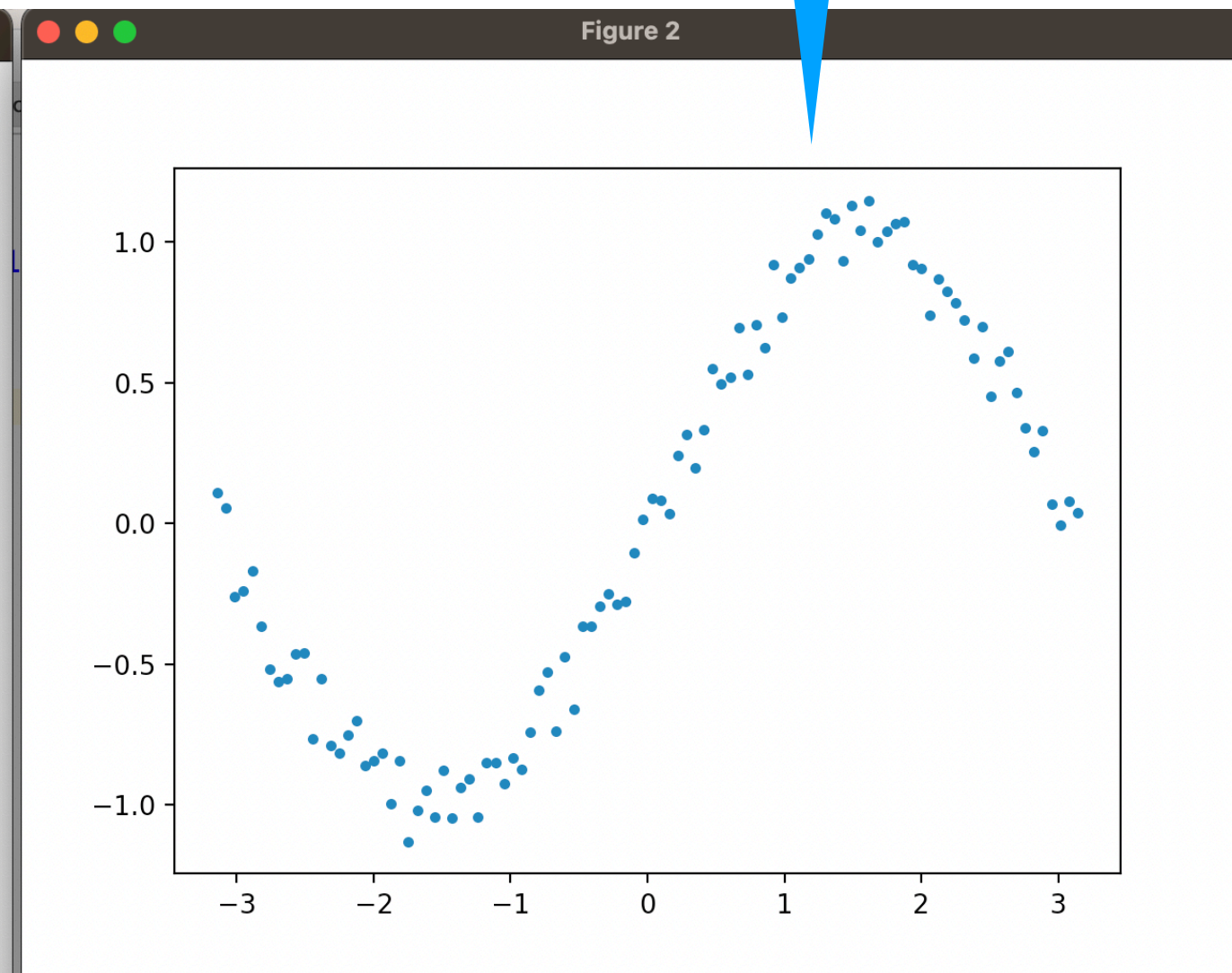
```
import numpy as np
x = np.linspace(-np.pi, np.pi, 100)
y = np.sin(x)
plt.figure(2)
plt.scatter(x, y, marker = 'o')
plt.show()
```

Ex2A

Use `plt.plot` to draw
sin function



Try to generate
 $y = \sin(x) + n$, where $n \in [-0.15, 0.15]$

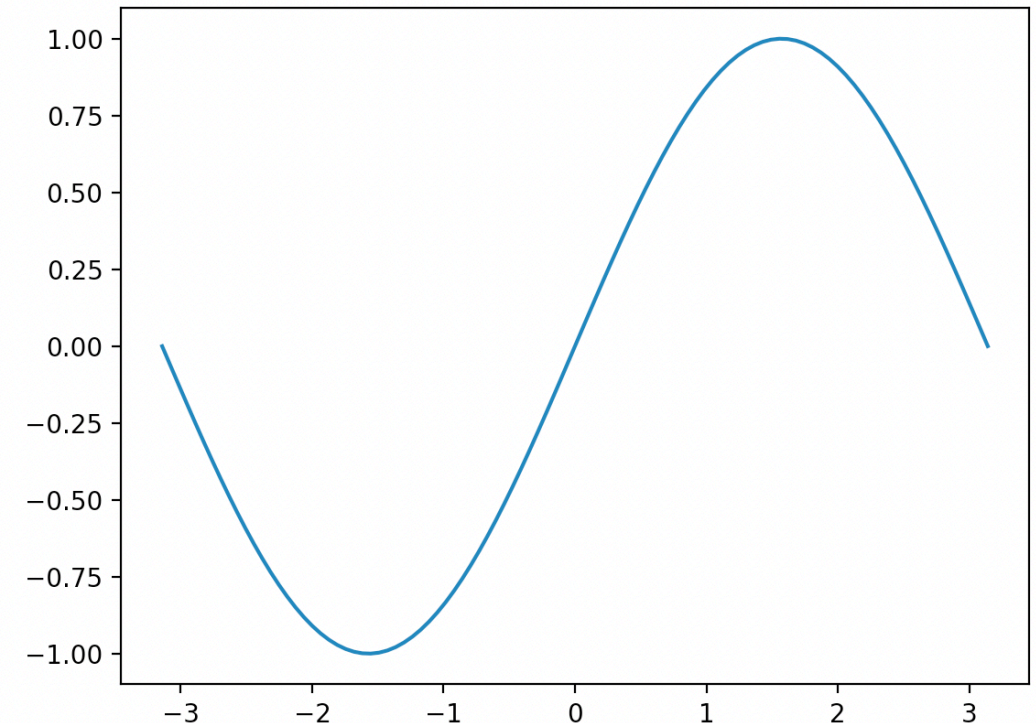
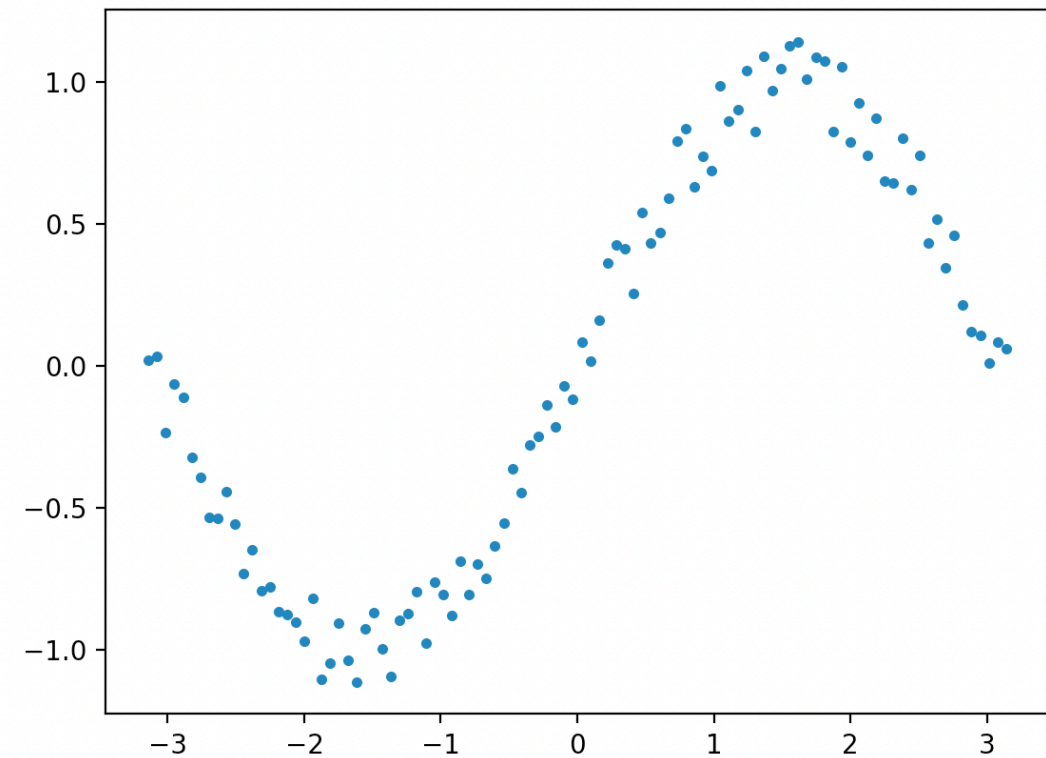


Project ▾
ex2023_ex1 [ex2022_ex1] ~/Desktop/py_...
venv
dice.py
ex1.py
ex1_new.py
ex1and2_new.py
ex2.py
ex2A.py
image2d.png
image3d.png
lion.png
External Libraries
Scratches and Consoles

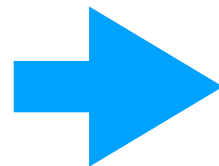
```
2 import matplotlib.pyplot as plt
3
4 x = np.linspace(-np.pi, np.pi, 100)
5 y = np.sin(x)
6 plt.figure(1)
7 plt.plot(x, y)
8
9 noise = np.random.rand(1, 100)*0.3-0.15
10 y = y + noise
11 plt.figure(2)
12 plt.scatter(x, y, marker='.')
13 plt.show()
14
```

[-0.15,0.15]

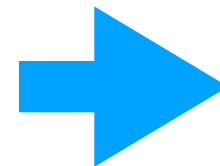
Challenge



Input : 100 points



Function
reconstruction



Output: a smooth
function

Figure 1

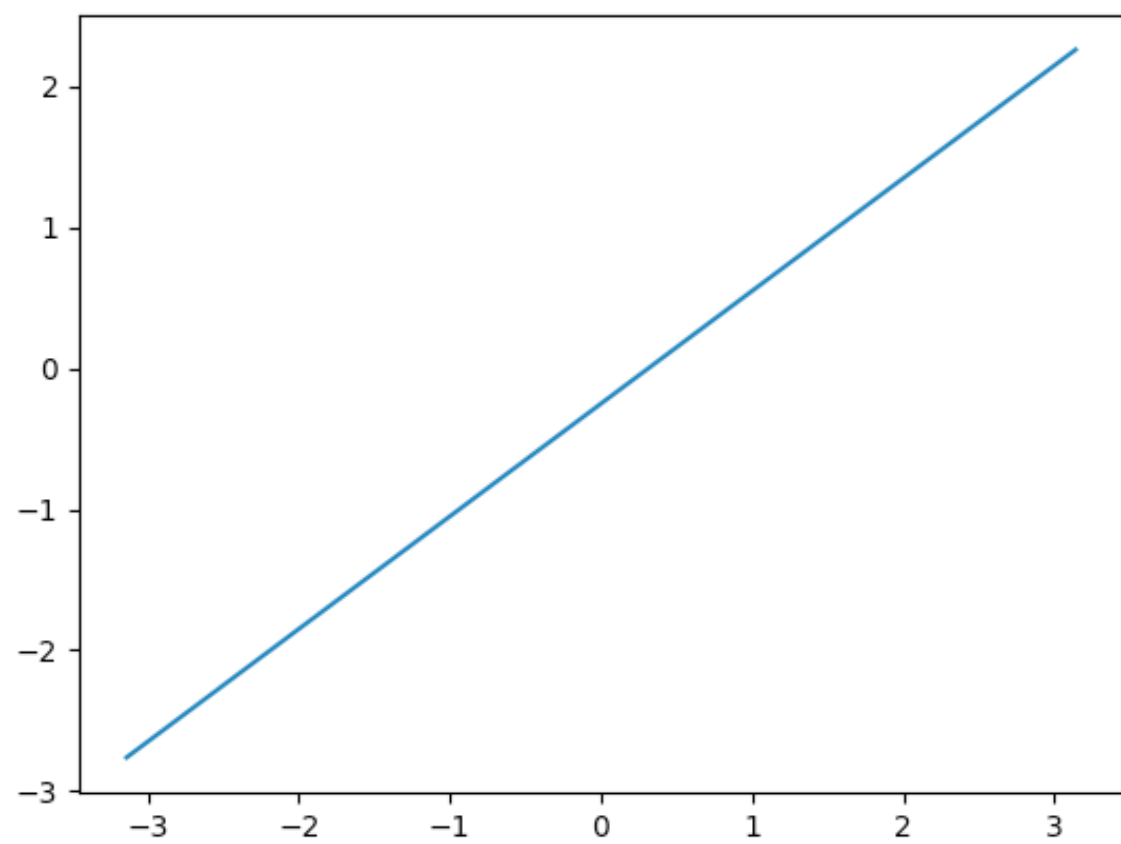
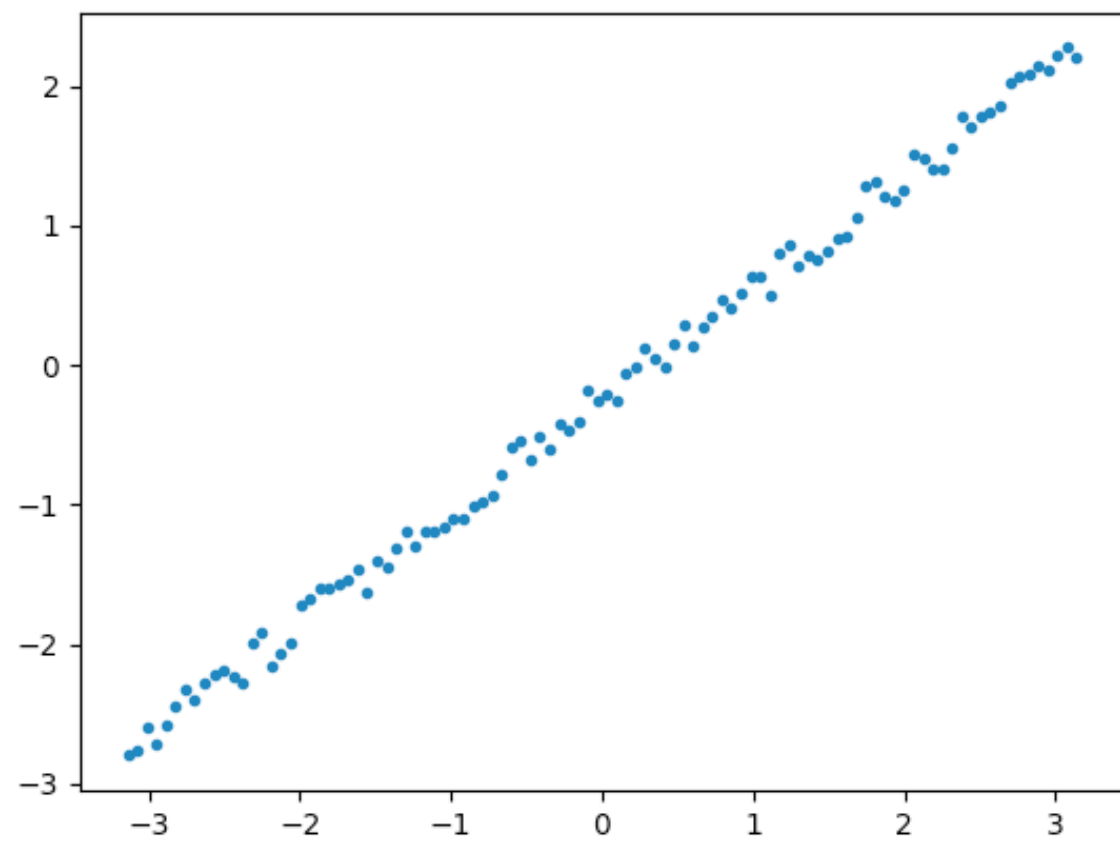


Figure 2



🏠 OpenCV-Python Tutorials



OpenCV

latest

`cv2.imread()`, `cv2.imshow()`

- Install opencv-python
- Import cv2
- Use `cv2.imread` to read an image
- Use `cv2.imshow` to show an image

```
import numpy as np  
import cv2
```

```
# Load an color image
```

```
img = cv2.imread('image3d.png')  
cv2.imshow('image',img)  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

demo_opencv [~/Desktop/py_code_2020/demo_opencv] - .../main.py [demo_opencv]

demo_opencv > main.py

Project

demo_opencv ~/Desktop/py_code

venv

demo_cam.py

image3d.png

main.py

External Libraries

Scratches and Consoles

main.py x demo_cam.py x

1

2

3

4

5

6

7

8

9

10

import numpy as np

import cv2

Load an color image

img = cv2.imread('image3d.png')

cv2.imshow('image',img)

cv2.waitKey(0)

cv2.destroyAllWindows()

Run: main x

4: Run

6: TODO

Terminal

Python Console

Event Log

image

3dScannerApp.com



```
cap = cv2.VideoCapture(0)  
cap.read()
```

- Set cap to cv2.VideoCapture(0)
- while(True):
 - Capture frame-by-frame
 - Display the resulting frame

```
import numpy as np
import cv2
```

```
cap = cv2.VideoCapture(0)
```

```
while(True):
```

```
    # Capture frame-by-frame
```

```
    ret, frame = cap.read()
```

```
    # Our operations on the frame come here
```

```
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
    # Display the resulting frame
```

```
    cv2.imshow('frame',gray)
```

```
    if cv2.waitKey(1) & 0xFF == ord('q'):
```

```
        break
```

```
    # When everything done, release the capture
```

```
    cap.release()
```

```
    cv2.destroyAllWindows()
```

demo_opencv [~/Desktop/py_code_2020/demo_opencv] - .../demo_cam.py [demo_opencv]

demo_opencv > demo_cam.py

Project ▾

- demo_opencv ~/Desktop/py_code
 - venv
 - demo_cam.py
 - image3d.png
 - main.py
- External Libraries
- Scratches and Consoles

```
1 import numpy as np
2 import cv2
3
4 cap = cv2.VideoCapture(0)
5
6 while(True):
7     # Capture frame-by-frame
8     ret, frame = cap.read()
9
10    # Our operations on the frame come here
11    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
12
13    # Display the
14    cv2.imshow('frame', gray)
15    if cv2.waitKey(1) && ord('q') == ord('q'):
16        break
17
18 # When everything done, release the camera
19 cap.release()
20 cv2.destroyAllWindows()
```

while (True)

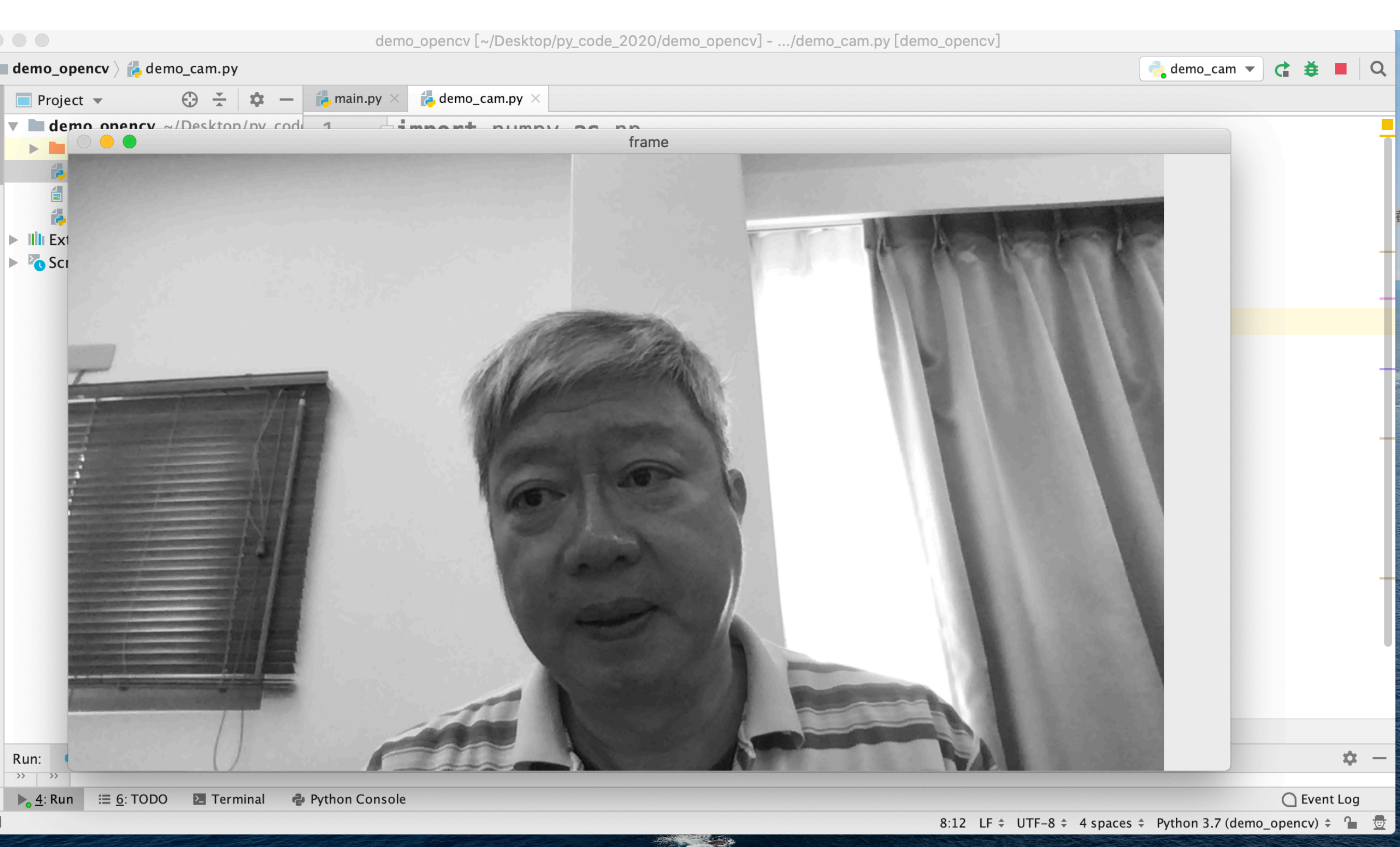
frame



Run: demo_cam ×

4: Run 6: TODO Terminal Python Console Event Log

8:12 15 UTF-8 4 spaces Python 3.7 (demo_opencv)



Programming Language I

- 1951 - Regional Assembly Language
- 1952 - Autocode
- 1954 - FORTRAN **
- 1954 - IPL (LISP的先驅)
- 1955 - FLOW-MATIC (COBOL的先驅)
- 1957 - COMTRAN (COBOL的先驅)
- 1958 - LISP **
- 1958 - ALGOL 58
- 1959 - FACT (COBOL的先驅)
- 1959 - COBOL **
- 1962 - APL
- 1962 - Simula
- 1962 - SNOBOL
- 1963 - CPL (C的先驅)
- 1964 - BASIC
- 1964 - PL/I
- 1967 - BCPL (C的先驅)
-

**有三個現代程式語言於1950年代被設計出來

這三者所衍生的語言直到今日仍舊廣泛地被採用

Programming Language II

確立了基礎範式

- 1968 - Logo
- 1970 - Pascal
- 1970 - Forth
- 1972 - C語言
- 1972 - Smalltalk
- 1972 - Prolog
- 1973 - ML
- 1975 - Scheme
- 1978 - SQL (起先只是一種查詢語言，擴充之後也具備了程式結構)
-

Programming Language III

1980年代：增強、模組、效能

- 1980 - Ada
 - 1983 - C++ (就像有類別的C)
 - 1984 - Common Lisp
 - 1985 - Eiffel
 - 1986 - Erlang
 - 1987 - Perl
 - 1988 - Tcl
 - 1989 - FL (Backus)
 -
- C++合併了物件導向以及系統程式設計

Programming Languages for Internet

- 1990 - Haskell
- 1991 - Python
- 1991 - Visual Basic
- 1993 - Ruby
- 1993 - Lua
- 1994 - CLOS (part of ANSI Common Lisp)
- 1995 - Java
- 1995 - Delphi (Object Pascal)
- 1995 - JavaScript
- 1995 - PHP
- 1997 - REBOL
- 1999 - D
-

提升程式設計師的生產力

現今的趨勢

- 元件導向(component-oriented)軟體開發
- 更重視分散式及移動式的應用

- 2001 - C#
- 2001 - Visual Basic .NET
- 2002 - F#
- 2003 - Scala
- 2003 - Factor
- 2006 - Windows PowerShell
- 2007 - Clojure
- 2009 - Go
- 2014 - Swift (程式語言)
-



Python



Guido van Rossum at the [Dropbox](#) headquarters in 2014

Born	31 January 1956 (age 62) ^[1] Haarlem, Netherlands ^{[2][3]}
Residence	Belmont, California, U.S.
Nationality	Dutch
Alma mater	University of Amsterdam
Occupation	Computer programmer, author
Employer	Dropbox ^[4]
Known for	Creating the Python programming language
Spouse(s)	Kim Knapp (m. 2000)
Children	Orlijn Michiel Knapp–van Rossum ^[5]
Awards	Award for the Advancement of Free Software (2001)
Website	gvanrossum.github.io ^[6]

Computer Programming for Everybody [\[edit \]](#)

In 1999, Van Rossum submitted a funding proposal to [DARPA](#) called "[Computer Programming for Everybody](#)[↗](#)," in which he further defined his goals for Python:

- An easy and intuitive language just as powerful as major competitors
- [Open source](#), so anyone can contribute to its development
- Code that is as understandable as plain English
- Suitability for everyday tasks, allowing for short development times

Python has grown to become a popular programming language. As of October 2017, it was the second most popular language on [GitHub](#), a social coding website, behind [JavaScript](#) and ahead of [Java](#).^[21] According to a programming language popularity survey^[22] it is consistently amongst the top 10 most mentioned languages in job postings. Furthermore, Python is consistently^[*clarification needed*] in the top 10 most popular languages according to the [TIOBE Programming Community Index](#).^[23]

Python [\[edit \]](#)

In December 1989, Van Rossum had been looking for a 'hobby' programming project that would keep [him] occupied during the week around Christmas" as his office was closed when he decided to write an [interpreter](#) for a "new scripting language [he] had been thinking about lately: a descendant of [ABC](#) that would appeal to [Unix/C](#) hackers". He attributes choosing the name "Python" to "being in a slightly irreverent mood (and [being] a big fan of *[Monty Python's Flying Circus](#)*)".^[18]

He has explained that Python's predecessor, [ABC](#), was inspired by [SETL](#), noting that ABC co-developer [Lambert Meertens](#) had "spent a year with the SETL group at [NYU](#) before coming up with the final ABC design".^[19]

In July 2018, Van Rossum announced that he would be stepping down from the position of [BDFL](#) of the Python programming language.^[20]

Python Tutorial

Release 3.7.0

**Guido van Rossum
and the Python development team**

Python modules for Web development

Web development

Requests: HTTP for Humans™



Star

44,653

Requests is an elegant and simple HTTP library for Python, built for human beings.

Sponsored by **CERT**
Gouvernemental -
GOVCERT.LU.



The Hitchhiker's Guide to Python is now available in book form!

ADS VIA CARBON

Release v2.25.1. ([Installation](#))

downloads

2G

license

Apache 2.0

wheel

yes

python

2.7 | 3.5 | 3.6 | 3.7 | 3.8 | 3.9

Requests is an elegant and simple HTTP library for Python, built for human beings.

Behold, the power of Requests:

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
'{"type":"User"... '
>>> r.json()
{'private_gists': 419, 'total_private_repos': 77, ...}
```

```
1 import requests
2
3 r = requests.get("https://api.github.com/repos/psf/requests")
4
5 print(r.json()['description'])
6
```


demo_requests [~/Desktop/py_code_2020/demo_requests] - .../main.py [demo_requests]

demo_requests

main

Project

demo_requests ~/Desktop/py_co

venv

main.py

External Libraries

Scratches and Consoles

main.py

1 import requests

2

3 r = requests.get("https://api.github.com/repos/psf/requests")

4

5 print(r.json()['description'])

6 print(r.json()['url'])

7

Run: main

↑

↓

↶

↷

⌂

🗑

/Users/apple/Desktop/py_code_2020/demo_requests/venv/bin/python /Users/apple/Desktop/py_code_2020/demo_requests/main.py
A simple, yet elegant HTTP library.
<https://api.github.com/repos/psf/requests>
Process finished with exit code 0

4: Run

6: TODO

Terminal

Python Console

Event Log

7:1 LF UTF-8 4 spaces Python 3.7 (demo_requests)



The web framework for
perfectionists with deadlines.

[OVERVIEW](#)[DOWNLOAD](#)[DOCUMENTATION](#)[NEWS](#)[COMMUNITY](#)[CODE](#)[ISSUES](#)[ABOUT](#)[♥ DONATE](#)

Django makes it easier to build better Web
apps more quickly and with less code.

[Get started with Django](#)

Meet Django

[Download latest release: 3.1.7](#)

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

website [~/Desktop/py_code_2020/website] - .../gettextvalues2_diff2.py [website]

website > gettextvalues2_diff2.py

gettextvalue

Project

website ~/Desktop/py_code_2020/website

templates

from_ex.html

post_submit.html

venv

app.py

gettextvalue.py

gettextvalues2.py

gettextvalues2_diff.py

gettextvalues2_diff2.py

External Libraries

Scratches and Consoles

gettextvalues2.py

gettextvalues2_diff2.py

gettextvalues2_diff.py

gettextvalue.py

post_submit.html

app.py

1

from flask import Flask, request, render_template

2

import sympy as sp

3

from sympy import *

4

app = Flask(__name__)

5

@app.route("/", methods=['GET', 'POST'])

6

def submit():

7

8

9

10

11

12

13

14

15

16

17

18

19

20

if request.method == 'POST':

x = sp.symbols('x')

ss = request.values.get('username')

tt = "diff("+ss+")"

ans = eval(tt)

return 'Ans is '+str(ans)

return render_template('post_submit.html')

if __name__ == '__main__':

app.debug = True

app.run()

submit() > if request.method == 'POST'

Run: gettextvalue

* Debug mode: on

* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

* Restarting with stat

* Debugger is active!

* Debugger PIN: 634-232-453

127.0.0.1 - - [23/Sep/2021 22:24:47] "GET / HTTP/1.1" 200 -

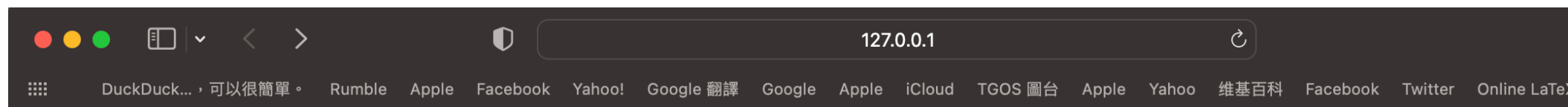
4: Run

6: TODO

Terminal

Python Console

Event Log



本網站會作微分

微分運算，請輸入函數 $f(x)$

範例，輸入函數： x^2+2x-1 ，答案： $2x+2$

範例，輸入函數： $\cos(x)$ ，答案： $-\sin(x)$



Ans is $2x + 2$

Data Science

[Install](#) [Documentation](#) [Learn](#) [Community](#) [About Us](#) [Contribute](#)



The fundamental package for scientific computing with Python

GET STARTED

NumPy v1.20.0 Type annotation support - Performance improvements through multi-platform SIMD

POWERFUL N-DIMENSIONAL ARRAYS

Fast and versatile, the NumPy vectorization, indexing, and broadcasting concepts are the de-facto standards of array computing today.

NUMERICAL COMPUTING TOOLS

NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.

INTEROPERABLE

NumPy supports a wide range of hardware and computing platforms, and plays well with distributed, GPU, and sparse array libraries.

NUMERICAL COMPUTING TOOLS

NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.

POWERFUL N-DIMENSIONAL ARRAYS

Fast and versatile, the NumPy vectorization, indexing, and broadcasting concepts are the de-facto standards of array computing today.

INTEROPERABLE

NumPy supports a wide range of hardware and computing platforms, and plays well with distributed, GPU, and sparse array libraries.

myExp8 [~/Desktop/py_code_2020/myExp8] - .../sumSeries.py [myExp8]

sumSeries

Project

myExp8 ~/Desktop/py_code_2020/myExp8

venv

sumSeries.py

External Libraries

Scratches and Consoles

1: Project

sumSeries.py

1 import numpy as np

2

3 def sumCeilFloor(N):

4 result = 0

5 for n in range(1,N+1):

6 result += np.floor(n**2/5) + np.ceil(2*n/3)

7

8 return result

9

10

11 for n in range(1,6):

12 print(n, sumCeilFloor(n))

13

Run: sumSeries

1 1.0

2 3.0

3 6.0

4 12.0

5 21.0

Process finished with exit code 0

4: Run

6: TODO

Terminal

Python Console

Event Log

LF UTF-8 4 spaces Python 3.7 (myExp8)

$$\sum_{n=1}^6 \left\lfloor \frac{n^2}{5} \right\rfloor + \left\lceil \frac{2n}{3} \right\rceil$$

pandas

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

[Install pandas now!](#)

Latest version: 1.2.2

- [What's new in 1.2.2](#)
- Release date:
Feb 09, 2021
- [Documentation \(web\)](#)
- [Documentation \(pdf\)](#)
- [Download source code](#)

Follow us

[Follow @pandas_dev](#)

Get the book



Getting started

- [Install pandas](#)
- [Getting started](#)

Documentation

- [User guide](#)
- [API reference](#)
- [Contributing to pandas](#)
- [Release notes](#)

Community

- [About pandas](#)
- [Ask a question](#)
- [Ecosystem](#)

With the support of:



Previous versions

```
1 import pandas as pd
2
3 df = pd.read_csv("data.csv")
4
5 print(df)
```

readCSV2023 > main.py

main.py x A.csv x B.csv x readA.py x

Project

readCSV2023 ~/Desktop

- A.csv
- A.numbers
- B.csv
- B.numbers
- main.py
- readA.py
- External Libraries
- Scratches and Consoles

```
1 import csv
2
3 with open('A.csv') as f:
4     reader = csv.reader(f)
5     a = []
6     for row in reader:
7         # print(row[0])
8         a.append(row[0])
9     print(len(a))
10
```

for s in a

Run: main x

```
/Users/apple/Desktop/py_code_2020/readcsvs/venv/bin/python
/Users/apple/Desktop/py_code_2020/readCSV2023/main.py
```

4

4

1 numbers in a not in b

3 numbers in a also in b

Data Visualization



Fork me on GitHub

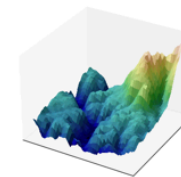
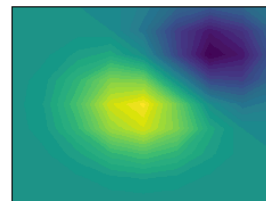
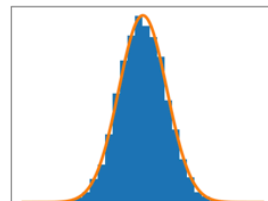
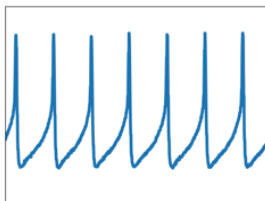
[Installation](#) [Documentation](#) [Examples](#) [Tutorials](#) [Contributing](#)

[home](#) | [contents](#) » [Matplotlib: Python plotting](#)

[modules](#) | [index](#)

Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.



Latest stable release
3.3.4: [docs](#) | [changelog](#)

Last release for Python 2
2.2.5: [docs](#) | [changelog](#)

Development version
[docs](#)

Support Matplotlib

Create

- Develop [publication quality plots](#) with just a few lines of code
- Use [interactive figures](#) that can zoom, pan, update...

Customize

- [Take full control](#) of line styles, font properties, axes properties...
- [Export and embed](#) to a number of file formats and interactive environments

Extend

- Explore tailored functionality provided by [third party packages](#)
- Learn more about Matplotlib through the many [external learning resources](#)

projectEx2025demo_my3Dsin.py

shape2D2.pydemo_tf_ex3.pydemo_tf_ex2.pydemo_3pointPlane.pydemo_parfor.pydemo_my3Dsin.py

demo_my3Dsin.py

14

15

16

17

18

19

[0.4992, -3.2731, -0.8272, 2.2502, 0.]

b = np.array

r = np.array

z = a0 @ x[0

h0 = np.res

v = np.tanh(

mysin()

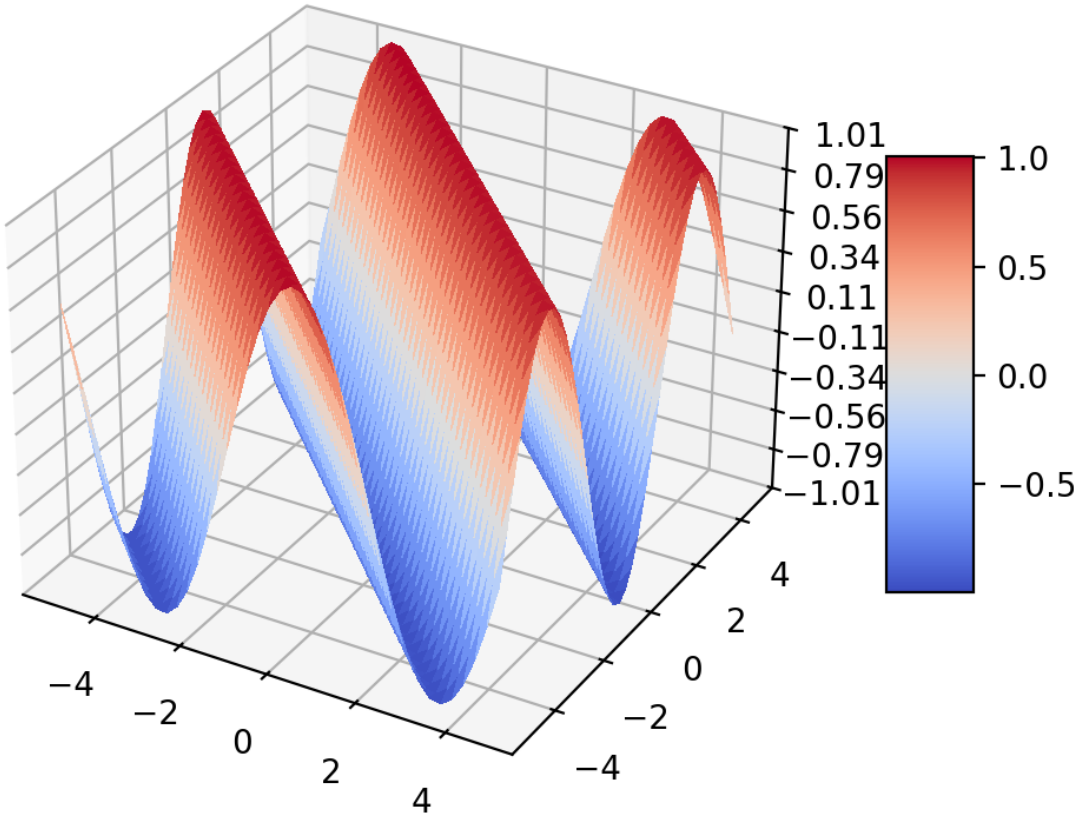
array before performing th

1.25.)

z[i,j] = mysin(v)

RunPython PackagesTODOPython ConsoleProblems

Figure 1



parfor 2026.5.0

A package to mimic the use of parfor as done in Matlab.

```
pip install parfor
```



 Description

 下載檔案

 發行歷史

pytest no status

Parfor

Used to parallelize for-loops using parfor in Matlab? This package allows you to do the same in python. Take any normal serial but parallelizable for-loop and execute it in parallel using easy syntax. Don't worry about the technical details of using the multiprocessing module, race conditions, queues, parfor handles all that. Now powered by [ray](#).

Tested on linux, Windows and OSX with python 3.10 and 3.12.

專案連結

 [repository](#)

Key dates

[PyPI data](#) ›

Released: 2026年5月12日

LATEST RELEASE

pythonProjectEx2025demo_parfor.pydemo_parfor

shape2D2.pydemo_tf_ex3.pydemo_tf_ex2.pydemo_3pointPlane.pydemo_parfor.pydemo_my3Dsin.py

demo_classDesdemo_complex.pdemo_conjugatedemo_fac.pydemo_fib.pydemo_grading.pdemo_isPrime.pydemo_linearSystdemo_markov2.pdemo_matrixMuldemo_min.pydemo_multiply.pdemo_my3Dsin.pdemo_myconjugdemo_parfor.pydemo_parfor_loade demo_permu.pydemo_planeFittirdemo_prefix_eva demo_primes.pydemo_scope.pydemo_sudokuGedemo_tf.ex5.py

2import numpy as np

3from time import sleep

4from parfor import parfor

5if __name__ == '__main__':

6p = [1,2]

7usage

8@parfor(*args: range(10000), np.ones((1,1)))

9def fun(i, a):

sleep(1)

if __name__ == '__main__'

NLTK 3.5 documentation

[NEXT](#) | [MODULES](#) | [INDEX](#)

Natural Language Toolkit

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to [over 50 corpora and lexical resources](#) such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active [discussion forum](#).

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

NLTK has been called “a wonderful tool for teaching, and working in, computational linguistics using Python,” and “an amazing library to play with natural language.”

[Natural Language Processing with Python](#) provides a practical introduction to programming for language processing. Written by the creators of NLTK, it guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analyzing linguistic structure, and more. The online version of the book has been updated for Python 3 and NLTK 3. (The original Python 2 version is still available at http://nltk.org/book_1ed.)

TABLE OF CONTENTS

[NLTK News](#)[Installing NLTK](#)[Installing NLTK Data](#)[Contribute to NLTK](#)[FAQ](#)[Wiki](#)[API](#)[HOWTO](#)

SEARCH

```
1 import nltk
2
3 sentence = "hello my name is tim and this is cool"
4
5 words = nltk.word_tokenize(sentence)
6
7 print(words)
```

[Help](#)[Sponsor](#)[Log in](#)[Register](#)

opencv-python 4.5.1.48

[Latest version](#)`pip install opencv-python`

Released: Jan 2, 2021

Wrapper package for OpenCV python bindings.

Navigation

[Project description](#)[Release history](#)[Download files](#)

Project links

Project description

downloads 93M

OpenCV on Wheels

Unofficial pre-built CPU-only OpenCV packages for Python.

Check the manual build section if you wish to compile the bindings from source to enable additional modules such as CUDA.

Machine Learning and AI

導覽列

≡ 專案敘述

🕒 發行歷史

📄 下載檔案

Verified details *(What is this?)*

These details have been verified by PyPI

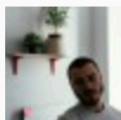
維護者



amysartran



ArthurZucker



lysandre



Thomwolf

專案敘述



Transformers

build passing

license Apache-2.0

website online

release v4.44.2

Contributor Covenant v2.0 adopted

DOI 10.5281/zenodo.7391177

English | [简体中文](#) | [繁體中文](#) | [한국어](#) | [Español](#) | [日本語](#) | [हिन्दी](#) | [Русский](#) | [Português](#) | [తెలుగు](#) | [Français](#)
| [Deutsch](#) | [Tiếng Việt](#) |

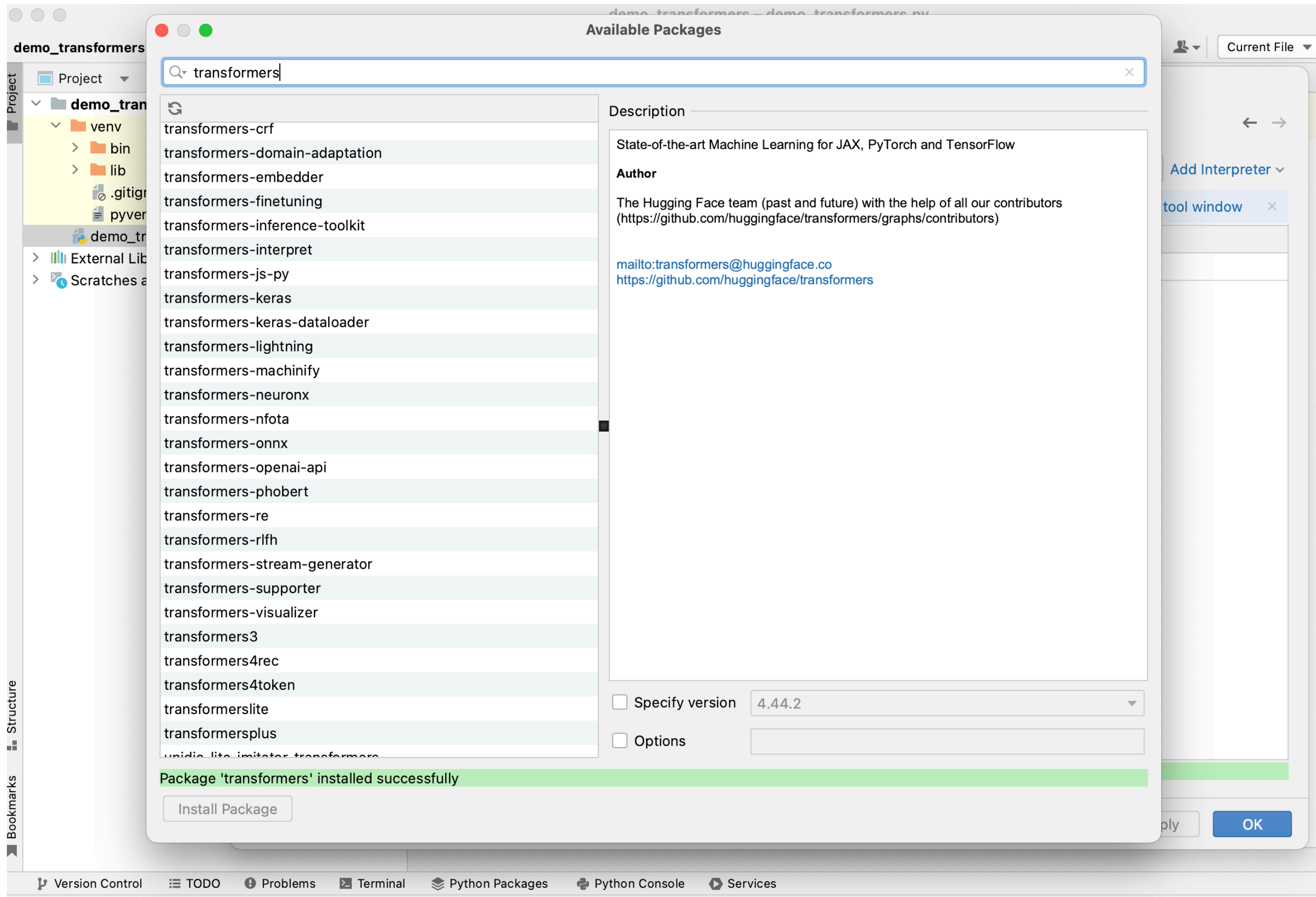
State-of-the-art Machine Learning for JAX, PyTorch and TensorFlow



Part of the Hugging Face course!



Transformers provides thousands of pretrained models to perform tasks on different modalities such as text



exCoreml demo_transformers2.py

Project

- buildBridgeEx3A_conv1D.py
- buildBridgeEx3A_transpose.py
- buildBridgeEx3AwithTfinal.py
- buildBridgeEx4.py
- cat2.png
- cats.png
- Copy_of_buildBridgeEx4.py
- demo_A_frontTest.py
- demo_B_bridgeTest.py
- demo_C_PostTest.py
- demo_D_entireTest.py
- demo_E_entireLoad&Test.py
- demo_F_saveAsCoremlModel.py
- demo_FrontBridgePostEx1.py
- demo_transformers.py
- demo_transformers2.py
- dog.png
- ex1_editCoreml.py
- ex2_exploreVggmDag.py
- ex3_exploreVggmDagFeature.py
- form_T_final.m

```
1 from transformers import DetrImageProcessor, DetrForObjectDetection
2 import torch
3 from PIL import Image
4 import requests
5
6 url = "http://images.cocodataset.org/val2017/000000039769.jpg"
7 image = Image.open(requests.get(url, stream=True).raw)
8
9 # you can specify the revision tag if you don't want the timm dependency
10 processor = DetrImageProcessor.from_pretrained("facebook/detr-resnet-50", revision="no_timm")
11 model = DetrForObjectDetection.from_pretrained("facebook/detr-resnet-50", revision="no_timm")
12
13 inputs = processor(images=image, return_tensors="pt")
14 outputs = model(**inputs)
15
16 for score, label, box in zip(re...
```

Run: demo_transformers2 x

```
/Users/a326/venv/bin/python /Users/a326/Desktop/python/exCoreml/demo_transformers2.py
2024-09-15 22:31:28.009740: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use available
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
Detected remote with confidence 0.998 at location [40.16, 70.81, 175.55, 117.98]
Detected remote with confidence 0.996 at location [333.24, 72.55, 368.33, 187.66]
Detected couch with confidence 0.995 at location [-0.02, 1.15, 639.73, 473.76]
Detected cat with confidence 0.999 at location [13.24, 52.05, 314.02, 470.93]
Detected cat with confidence 0.999 at location [345.4, 23.85, 640.37, 368.72]
```

FROM RESEARCH TO PRODUCTION

An open source machine learning framework that accelerates the path from research prototyping to production deployment.

[Install](#) >

Prototype Features Now Available - APIs for Hardware Accelerated Mobile and ARM64 Builds





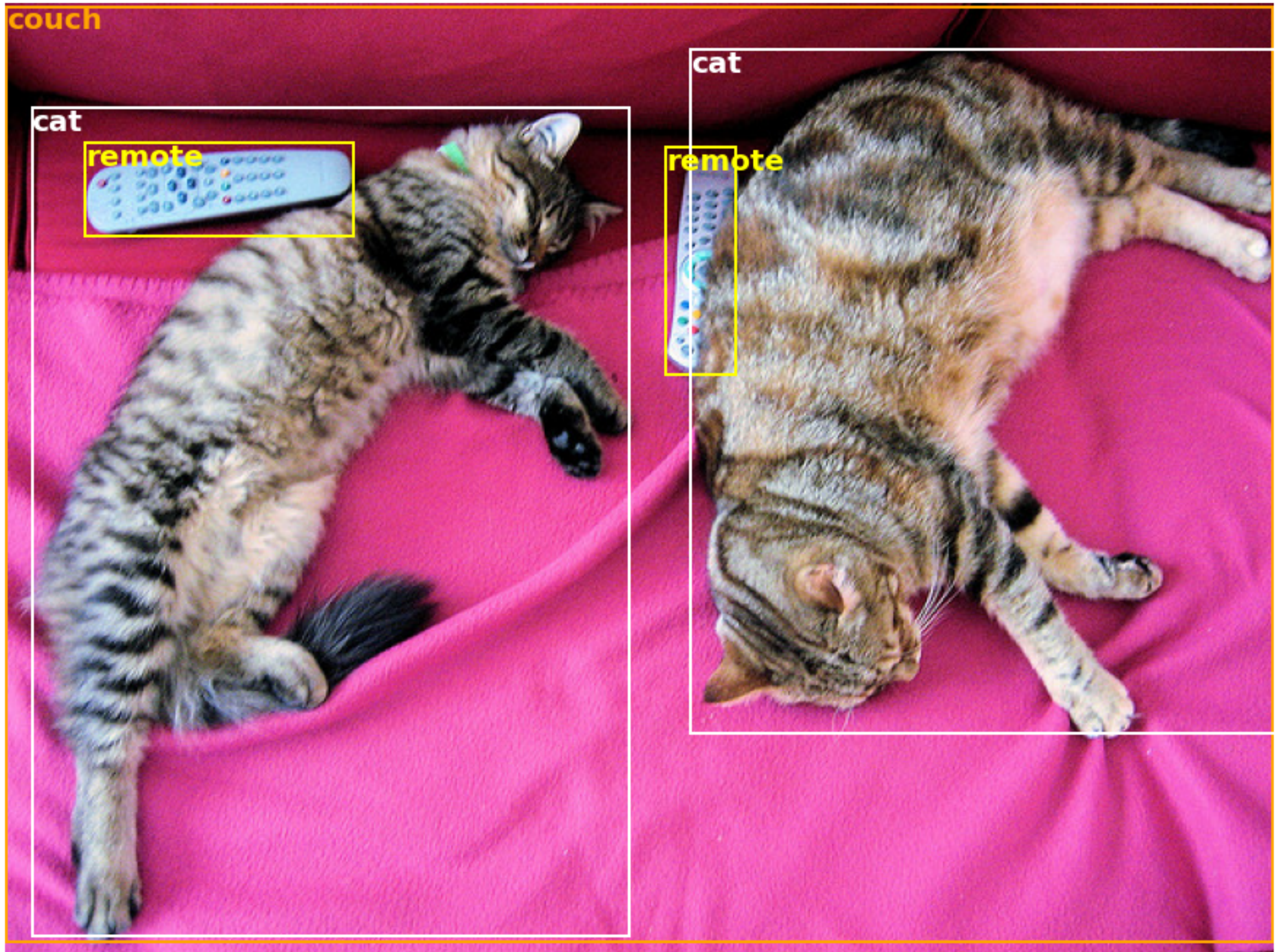
couch

cat

remote

cat

remote



Tensorflow

[Help](#)[Sponsor](#)[Log in](#)[Register](#)

tensorflow 2.4.1

```
pip install tensorflow
```

[Latest version](#)

Released: Jan 22, 2021

TensorFlow is an open source machine learning framework for everyone.

Navigation

[Project description](#)[Release history](#)[Download files](#)

Project description

python 3.6 | 3.7 | 3.8 pypi package 2.4.1

TensorFlow is an open source software library for high performance numerical computation. Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices.

Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is

MLP_neupy > cnn_mnist.py

Project

MLP_neupy ~/Desktop/py_code

venv

cnn_mnist.py

demo_transformers.py

load_data_show.py

machine_translator.py

mnist_mlp.py

External Libraries

Scratches and Consoles

cnn_mnist.py

demo_transformers.py

load_data_show.py

machine_translator.py

mnist_mlp.py

epochs = 15

model.compile(loss="categorical_crossentropy", optimizer="adam",

model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs,

score = model.evaluate(x_test, y_test, verbose=0)

print("Test loss:", score[0])

print("Test accuracy:" score[1])

Run: cnn_mnist

422/422 [=====] - 24s 57ms/step - loss: 0.0632 - accuracy: 0.9801 - val_loss: 0.0379 - val_accuracy: 0.9887

Epoch 6/15

422/422 [=====] - 25s 58ms/step - loss: 0.0554 - accuracy: 0.9823 - val_loss: 0.0338 - val_accuracy: 0.9912

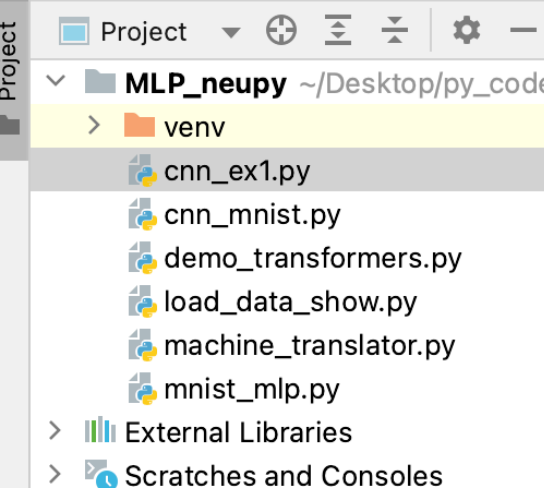
Epoch 7/15

63/422 [==>.....] - ETA: 18s - loss: 0.0569 - accuracy: 0.9818

MLP_neupy > cnn_ex1.py



cnn_ex1



```
2 print("TensorFlow version:", tf.__version__)
3
4 mnist = tf.keras.datasets.mnist
5
6 (x_train, y_train), (x_test, y_test) = mnist.load_data()
7 x_train, x_test = x_train / 255.0, x_test / 255.0
8
9 model = tf.keras.models.Sequential([
10     tf.keras.layers.Flatten(input_shape=(28, 28)),
11     tf.keras.layers.Dense(units=128, activation='relu'),
12     tf.keras.layers.Dropout(0.2),
13     tf.keras.layers.Dense(10)
```

Run: cnn_ex1

313/313 - 0s - loss: 0.0795 - accuracy: 0.9747 - 362ms/epoch - 1ms/step

Process finished with exit code 0

RSVP for your your local TensorFlow Everywhere event today!

[Find an event](#)

端對端的開放原始碼機器學習平台

[TensorFlow](#)
[適用於 JavaScript](#)
[適用於行動裝置及 IoT](#)
[適用於生產環境](#)

核心的開放原始碼程式庫可協助你開發及訓練機器學習模型。直接在瀏覽器中執行 Colab 筆記本，即可快速開始使用。

[開始使用 TensorFlow](#)


Keras



Simple. Flexible. Powerful.

[Get started](#)

[Guides](#)

[API docs](#)

```
from tensorflow import keras
from tensorflow.keras import layers

# Instantiate a trained vision model
vision_model = keras.applications.ResNet50()

# This is our video encoding branch using the trained vision_model
video_input = keras.Input(shape=(100, None, None, 3))
encoded_frame_sequence = layers.TimeDistributed(vision_model)(video_input)
encoded_video = layers.LSTM(256)(encoded_frame_sequence)

# This is our text-processing branch for the question input
question_input = keras.Input(shape=(100,), dtype='int32')
```

Deep learning for humans.

Keras is an API designed for human beings, not machines.
Keras follows best practices for reducing cognitive load: it

exCoreml > demo_E_entireLoad&Test.py

Project

- demo_C_PostTest.py
- demo_D_entireTest.py
- demo_E_entireLoad&Test.py
- demo_F_saveAsCoremlModel.py
- demo_FrontBridgePostEx1.py
- demo_transformers.py
- demo_transformers2.py
- dog.png
- ex1_editCoreml.py
- ex2_exploreVggmDag.py
- ex3_exploreVggmDagFeature.py
- form_T_final.m
- imagenetVggmDag.mlmodel
- imagenetVggmDagFeature.mlmodel
- imagesMean.mat
- inputAverageImage.mat
- inputImage.mat
- main.py
- MNISTClassifier.mlmodel
- model.keras
- model_keras_mnist.h5
- post_net_meta_classes_name.mat
- T_final.mat
- testBridgeData.mat
- tiger.jpg

```
6
7 select_image = 5
8 if select_image == 1:
9     img_path = 'tiger.jpg'
10 elif select_image == 2:
11     img_path = 'cats.png'
12 elif select_image == 3:
13     img_path = 'dog.png'
14 elif select_image == 4:
15     img_path = 'turtle.png'
16 elif select_image == 5:
17     img_path = 'tiger2.png'
18 elif select_image == 6:
19     img_path = 'cat2.png'
20 elif select_image == 7:
21     img_path = 'tiger3.png'
```

elif select_image == 5



Run: demo_E_entireLoad&Test

WARNING:tensorflow:No training configuration found in save file, so the model

1/1 [=====] - 0s 342ms/step

['老虎']

score: 1.0 argmax: 52

1.0

exCoreml > demo_E_entireLoad&Test.py

Project

Structure

```
4 import scipy.io
5 import numpy as np
6
7 select_image = 4
8 if select_image == 1:
9     img_path = 'tiger.jpg'
10 elif select_image == 2:
11     img_path = 'cats.png'
12 elif select_image == 3:
13     img_path = 'dog.png'
14 elif select_image == 4:
15     img_path = 'turtle.png'
16 elif select_image == 5:
17     img_path = 'tiger2.png'
18 elif select_image == 6:
19     img_path = 'cat2.png'
20 elif select_image == 7:
```

> If select_image == 1

Run: demo_E_entireLoad&Test ×

WARNING:tensorflow:No training configuration found in save fi

1/1 [=====] - 0s 234ms/step

['龜']

score: 0.95934254 argmax: 88

1.0



exCoreml > demo_E_entireLoad&Test.py

buildBridgeEx3A_transpose.py × demo_E_entireLoad&Test.py × turtle.png × tiger2.png × demo

```
4 import scipy.io
5 import numpy as np
6
7 select_image = 3
8 if select_image == 1:
9     img_path = 'tiger.jpg'
10 elif select_image == 2:
11     img_path = 'cats.png'
12 elif select_image == 3:
13     img_path = 'dog.png'
14 elif select_image == 4:
15     img_path = 'turtle.png'
16 elif select_image == 5:
17     img_path = 'tiger2.png'
18 elif select_image == 6:
19     img_path = 'cat2.png'
20 elif select_image == 7:
```



Run: demo_E_entireLoad&Test ×

```
WARNING:tensorflow:No training configuration found in save file
1/1 [=====] - 0s 231ms/step
['狗']
score: 0.72645026 argmax: 42
0.9999999
```


scikit-learn

Machine Learning in Python

[Getting Started](#)
[Release Highlights for 0.24](#)
[GitHub](#)

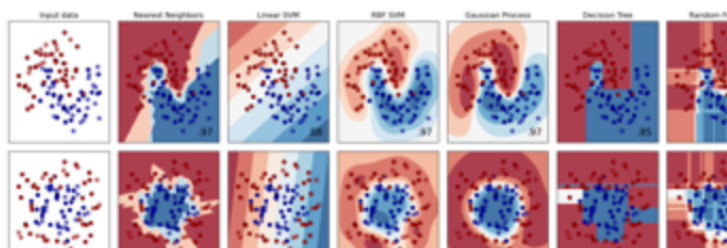
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: [SVM](#), [nearest neighbors](#), [random forest](#), and [more...](#)

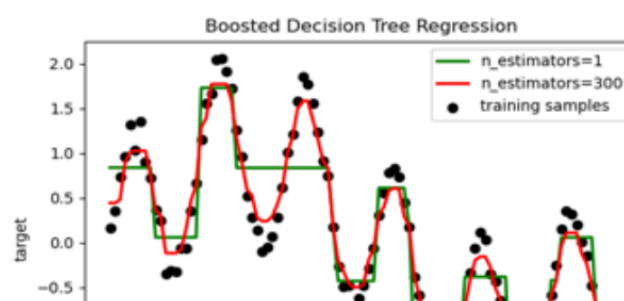


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: [SVR](#), [nearest neighbors](#), [random forest](#), and [more...](#)

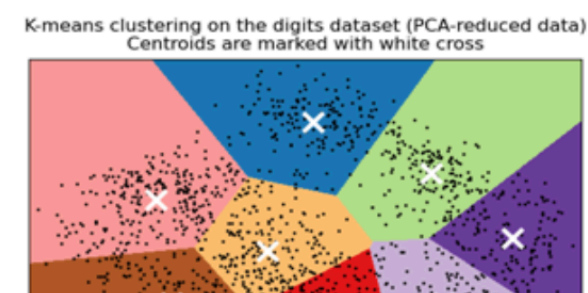


Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: [k-Means](#), [spectral clustering](#), [mean-shift](#), and [more...](#)



GUI

[Guides](#)[Garden](#)[API Reference](#)[PDF](#)[Wiki](#)

Version

stable

Quick search

Getting Started

Gallery of Examples

Getting Started

Kivy Project

Programming Guide

Kivy Basics

Controlling the environment

Configure Kivy

Architectural Overview

Programming Guide » Kivy Basics

Installation of the Kivy environment

Kivy depends on many Python libraries, such as pygame, gstreamer, PIL, Cairo, and more. They are not all required, but depending on the platform you're working on, they can be a pain to install. For Windows and MacOS X, we provide a portable package that you can just unzip and use.

Have a look at one of these pages for detailed installation instructions:

- [Installation on Windows](#)
- [Installation on OS X](#)
- [Installation on Linux](#)
- [Installation on Raspberry Pi](#)

Alternatively, instructions for the development version can be found here:

- [installation](#)

Create an application

Table Of Contents

[Kivy Basics](#)[Installation of the Kivy environment](#)[Create an application](#)[Kivy App Life Cycle](#)[Running the application](#)[Customize the application](#)

[Help](#)[Sponsor](#)[Log in](#)[Register](#)

PyQt5 5.15.3

```
pip install PyQt5
```

[Latest version](#)

Released: 2 minutes ago

Python bindings for the Qt cross platform application toolkit

Navigation

[Project description](#)[Release history](#)[Download files](#)

Project description

Qt is set of cross-platform C++ libraries that implement high-level APIs for accessing many aspects of modern desktop and mobile systems. These include location and positioning services, multimedia, NFC and Bluetooth connectivity, a Chromium based web browser, as well as traditional UI development.

PyQt5 is a comprehensive set of Python bindings for Qt v5. It is implemented as more than 35 extension modules and enables Python to be used as an alternative application development language to C++ on all supported platforms including iOS and Android.

```
import sys
from PyQt5.QtCore import *
from PyQt5.QtWidgets import *
from PyQt5.QtWebEngineWidgets import *

class MainWindow(QMainWindow):
    def __init__(self):
        super(MainWindow, self).__init__()
        self.brower = QWebEngineView()
        self.brower.setUrl(QUrl('http://google.com'))
        # self.brower.setUrl(QUrl('http://134.208.26.59'))
        self.setCentralWidget(self.brower)
        self.showMaximized()

        navbar = QToolBar()
        self.addToolBar(navbar)
        forward_btn = QAction('My Cool Brower forward', self)
        # forward_btn.triggered.connect(self.brower.forward)
        navbar.addAction(forward_btn)

app = QApplication(sys.argv)
QApplication.setApplicationName('My Cool Brower')
window = MainWindow()
app.exec_()
```

web_view [~/Desktop/py_code_2020/web_view] - .../main.py [web_view]

web_view

main.py

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

import sys

from PyQt5.QtCore import *

from PyQt5.QtWidgets import *

from PyQt5.QtWebEngineWidgets import *

class MainWindow(QMainWindow):

def __init__(self):

super(MainWindow, self).__init__()

self.brower = QWebEngineView()

self.brower.setUrl(QUrl('http://google.com'))

self.brower.setUrl(QUrl('http://134.208.26.59'))

self.setCentralWidget(self.brower)

self.showMaximized()

navbar = QToolBar()

self.addToolBar(navbar)

forward_btn = QAction('My Cool Brower forward', self)

forward_btn.triggered.connect(self.brower.forward)

navbar.addAction(forward_btn)

app = QApplication(sys.argv)

QApplication.setApplicationName('My Cool Brower')

window = MainWindow()

app.exec_()

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

MainWindow > __init__()

Run: main

>>

>>

/Users/apple/Desktop/py_code_2020/web_view/venv/bin/python /Users/apple/Desktop/py_code_2020/web

4: Run

6: TODO

Terminal

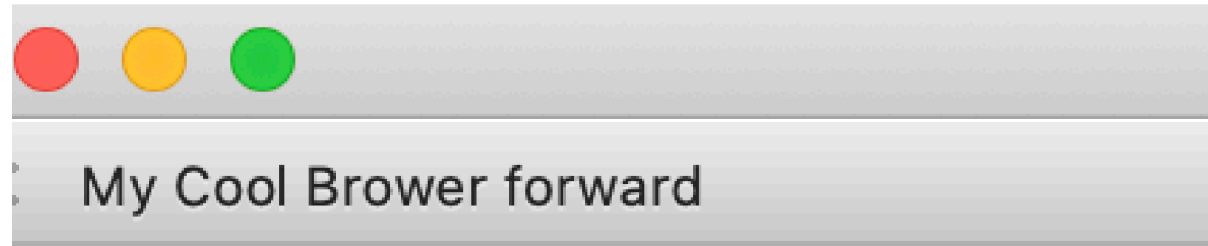
Python Console

Type hints are not installed

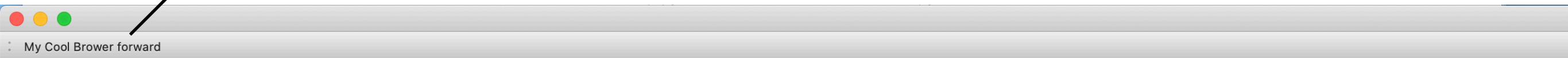
They are needed for better code insight.

[Install 'PyQt5-stubs'](#) [Ignore](#) [Settings](#)

4:1 LF UTF-8 4 spaces Python 3.7 (web_view)



關於 Google Google 商店



關於 Google Google 商店

Gmail 圖片 

Google

Google 搜尋

好手氣

Table of Contents

tkinter — Python interface to Tcl/Tk

- Tkinter Modules
- Tkinter Life Preserver
 - How To Use This Section
 - A Simple Hello World Program
- A (Very) Quick Look at Tcl/Tk
- Mapping Basic Tk into Tkinter
- How Tk and Tkinter are Related
- Handy Reference
 - Setting Options
 - The Packer
 - Packer Options
 - Coupling Widget Variables
 - The Window Manager
 - Tk Option Data Types
 - Bindings and Events
 - The index
 - Parameter

«

tkinter — Python interface to Tcl/Tk

Source code: [Lib/tkinter/__init__.py](#)

The **tkinter** package (“Tk interface”) is the standard Python interface to the Tk GUI toolkit. Both Tk and **tkinter** are available on most Unix platforms, as well as on Windows systems. (Tk itself is not part of Python; it is maintained at ActiveState.)

Running `python -m tkinter` from the command line should open a window demonstrating a simple Tk interface, letting you know that **tkinter** is properly installed on your system, and also showing what version of Tcl/Tk is installed, so you can read the Tcl/Tk documentation specific to that version.

See also: Tkinter documentation:

Python Tkinter Resources

The Python Tkinter Topic Guide provides a great deal of information on using Tk from Python and links to other sources of information on Tk.

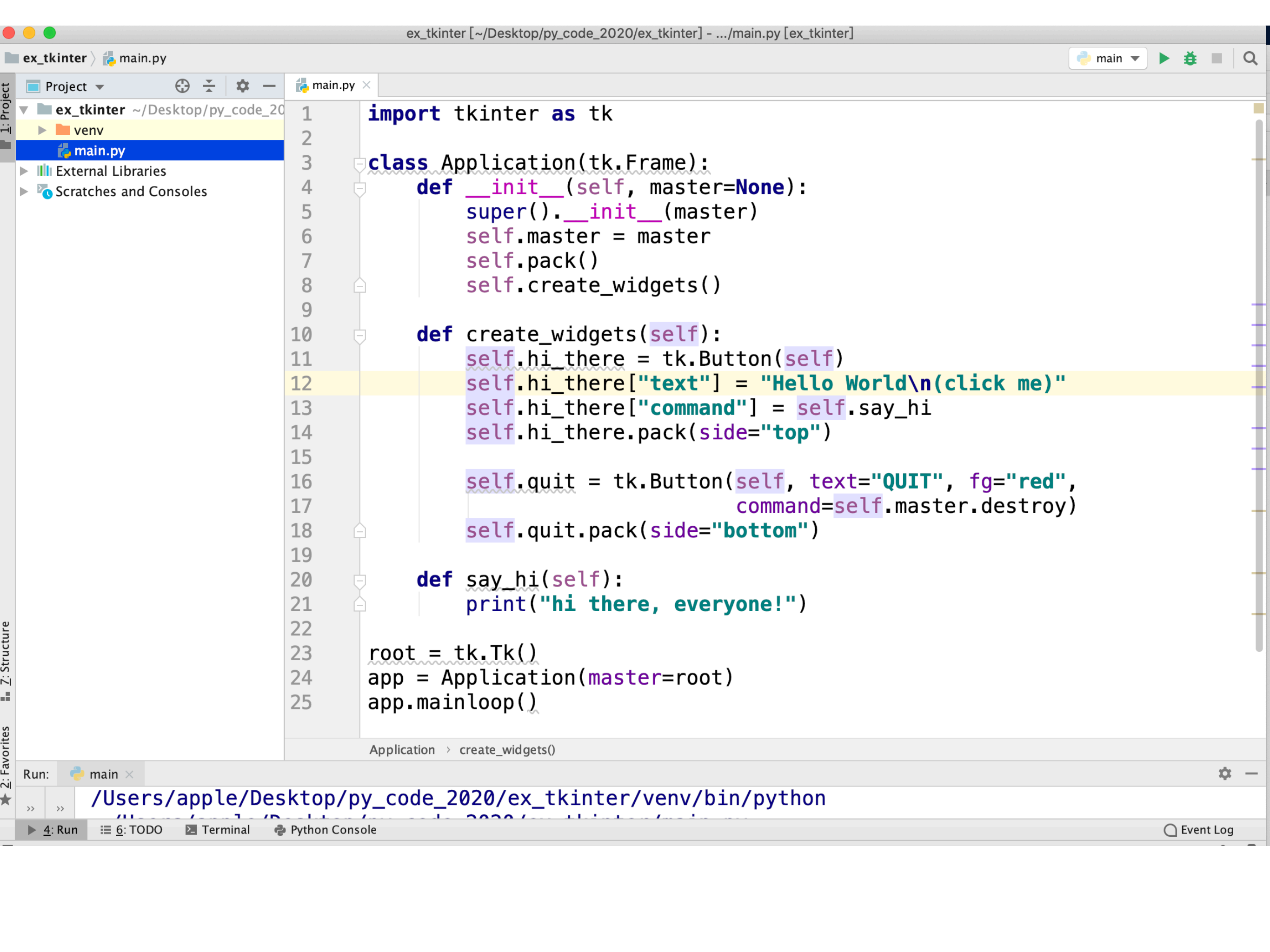
TKDocs

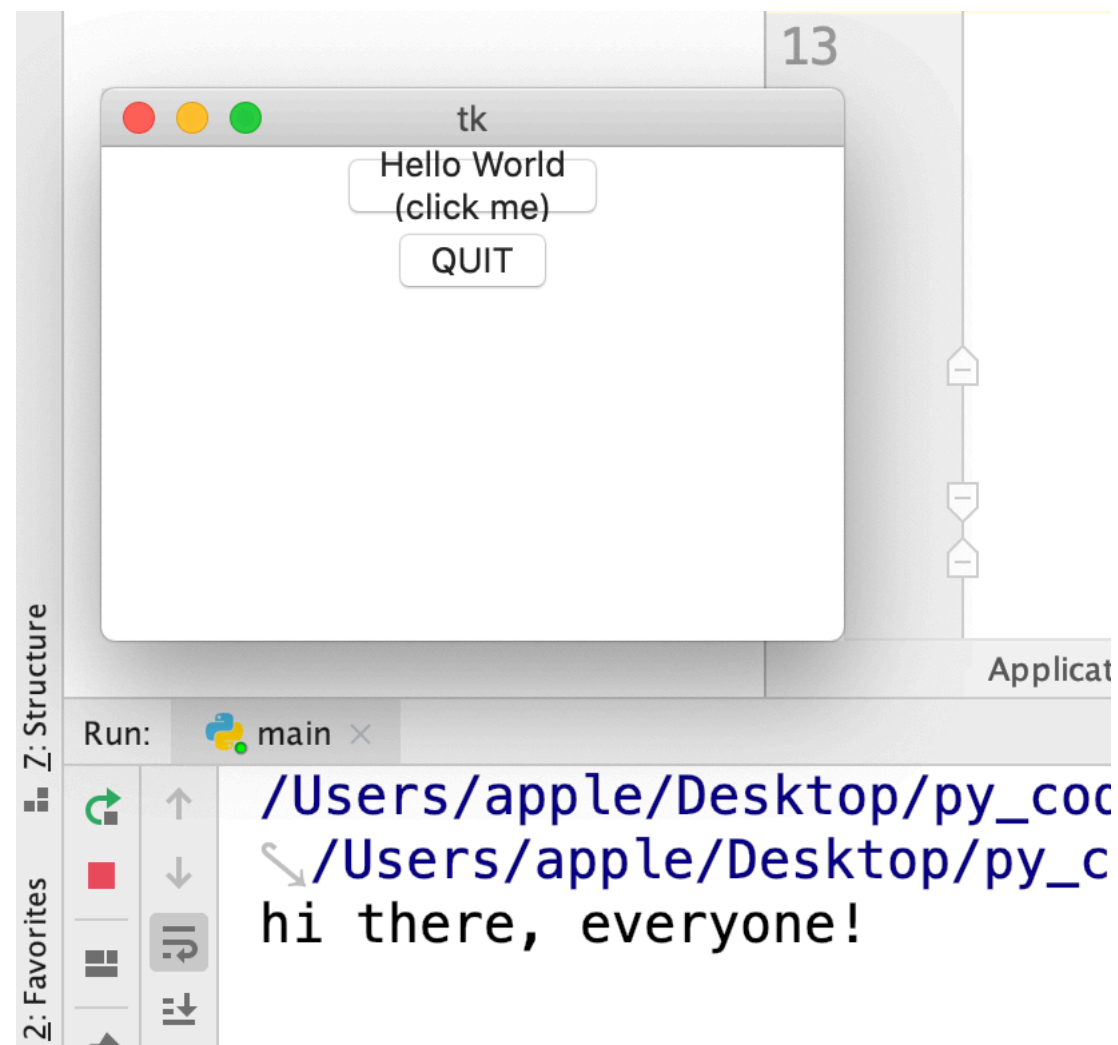
Extensive tutorial plus friendlier widget pages for some of the widgets.

Tkinter 8.5 reference: a GUI for Python

On-line reference material.

Tkinter docs from effbot





News

[New here?](#)

pygame 2 — 28 Oct, 2020

It's happy dance time.

[pygame 2](#) is out.

pygame 20th birthday — 28 Oct, 2020

It looks like we're 20 years old.



[New members signup](#)

[Log In](#)

Recent Releases



16 Feb, 2021

