



Python Programming



Acknowledgement

www.hbpatel.in

Course Name: **Python Tutorial - Python Full Course for Beginners**

Platform: YouTube

Instructor: Mosh Hamedani

Channel: Programming with Mosh

Video Link: https://youtu.be/_uQrJ0TkZlc

Channel Link: <https://www.youtube.com/@programmingwithmosh>

Course Name: **100 Days of Code: The Complete Python Pro Bootcamp for 2023**

Platform: Udemy

Instructor: Dr. Angela Yu

Video Link: <https://www.udemy.com/course/100-days-of-code/>

Platform Link: [udemy.com](https://www.udemy.com)

Applications of Python

- Automation
- AI
- Applications
- Websites



Various Dimensions using Python

- Python Scripting
- Web Scraping
- Data Science
- Web Development / Designing
- Game Development
- GUI Development
- API
- Automation
- OOP





Installing Python

www.hbpatel.in

The screenshot shows the Python.org homepage. A red box highlights the URL bar in the browser, which displays "python.org". The page features the Python logo and navigation links for About, Downloads, Documentation, Community, Success Stories, News, and Events. A code snippet demonstrates Python list comprehensions and the enumerate function. To the right, a section titled "Compound Data Types" explains lists and provides a link to "More about lists in Python 3". A navigation menu at the bottom includes links for 1, 2, 3, 4, and 5.

Python 3: List comprehensions
>>> fruits = ['Banana', 'Apple', 'Lime']
>>> loud_fruits = [fruit.upper() for fruit in
fruits]
>>> print(loud_fruits)
['BANANA', 'APPLE', 'LIME']

List and the enumerate function
>>> list(enumerate(fruits))
[(0, 'Banana'), (1, 'Apple'), (2, 'Lime')]

Compound Data Types

Lists (known as arrays in other languages) are one of the compound data types that Python understands. Lists can be indexed, sliced and manipulated with other built-in functions. [More about lists 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](#)

Get Started

Whether you're new to programming or an experienced

Download

Python source code and installers are available for download for all

Docs

Documentation for Python's standard library, along with tutorials

Jobs

Looking for work or have a Python related position that you're trying to



Installing Python

www.hbpatel.in

The screenshot shows the Python.org homepage. The navigation bar includes links for Python, PSF, Docs, PyPI, Jobs, and Community. The main content area features the Python logo and a search bar. A red box highlights the 'Downloads' tab, which is currently selected. Another red box highlights the 'Download for Windows' button, which is also labeled 'Python 3.10.6'. A note below the button states: 'Note that Python 3.9+ cannot be used on Windows 7 or earlier.' Below this, it says: 'Not the OS you are looking for? Python can be used on many operating systems and environments.' At the bottom, there's a quote: 'Python is a programming language that lets you work quickly and integrate systems more effectively. [Learn More](#)'.

Get Started
Whether you're new to

Download
Python source code and installers

Docs
Documentation for Python's

Jobs
Looking for work or have a Python



Installing Python

www.hbpatel.in

Python Releases for Windows | Python

python.org/downloads/windows/

Python PSF Docs PyPI Jobs Community

python™

Donate Search GO Socialize

About Downloads Documentation Community Success Stories News Events

Python »» Downloads »» Windows

Python Releases for Windows

- Latest Python 3 Release - Python 3.10.6
- Latest Python 2 Release - Python 2.7.18

Stable Releases

- Python 3.10.6 - Aug. 2, 2022
 - Note that Python 3.10.6 cannot be used on Windows 7 or earlier.
 - Download Windows embeddable package (32-bit)
 - Download Windows embeddable package (64-bit)
 - Download Windows help file
 - Download Windows installer (32-bit)
 - Download Windows installer (64-bit)

Pre-releases

- Python 3.11.0rc1 - Aug. 8, 2022
 - Download Windows embeddable package (32-bit)
 - Download Windows embeddable package (64-bit)
 - Download Windows embeddable package (ARM64)
 - Download Windows installer (32-bit)
 - Download Windows installer (64-bit)
 - Download Windows installer (ARM64)
- Python 3.11.0b5 - July 26, 2022



Installing Python (Download Python Executable Installer)

www.hbpatel.in

Python Release Python 3.10.6 | [python.org/downloads/release/python-3106/](#)

create them. If quarks have a baryon number of (+1/3), and antiquarks of (-1/3), the pentaquark would have a total baryon number of 1 and thus would be a baryon. Further, because it has five quarks instead of the usual three found in regular baryons (a.k.a. 'triquarks'), it is classified as an exotic baryon. The name pentaquark was coined by Claude Gignoux et al. (1987) and Harry J. Lipkin in 1987; however, the possibility of five-quark particles was identified as early as 1964 when Murray Gell-Mann first postulated the existence of quarks. Although predicted for decades, pentaquarks proved surprisingly tricky to discover and some physicists were beginning to suspect that an unknown law of nature prevented their production.

[Full Changelog](#)

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		d76638ca8bf57e44ef0841d2cde557a0	25986768	SIG
XZ compressed source tarball	Source release		afc7e14f7118d10d1ba95ae8e2134bf0	19600672	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	2ce68dc6cb870ed3beea8a20b0de71fc	40826114	SIG
Windows embeddable package (32-bit)	Windows		a62cca7ea561a037e54b4c0d120c2b0a	7608928	SIG
Windows embeddable package (64-bit)	Windows		37303f03e19563fa87722d9df11d0fa0	8585728	SIG
Windows help file	Windows		0aee63c8fb87dc71bf2bcc1f62231389	9329034	SIG
Windows installer (32-bit)	Windows		c4aa2cd7d62304c804e45a51696f2a88	27750096	SIG
Windows installer (64-bit)	Windows	Recommended	8f46453e68ef38e5544a76d84df3994c	28916488	SIG

python-3.10.6-am....exe

python-3.10.6-emb....zip

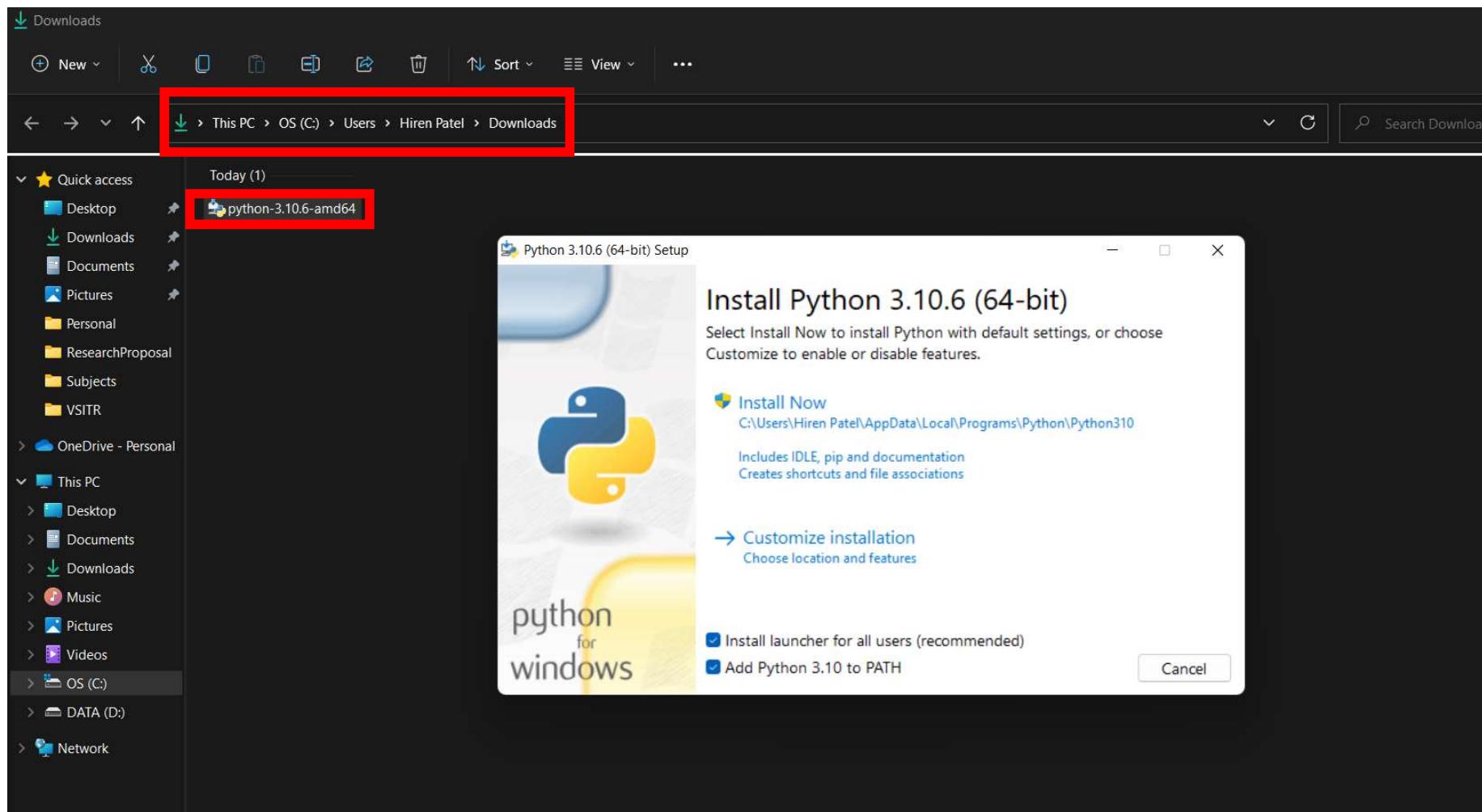
Show all X



Installing Python

(Unzip / Extract the downloaded file and double click to install it)

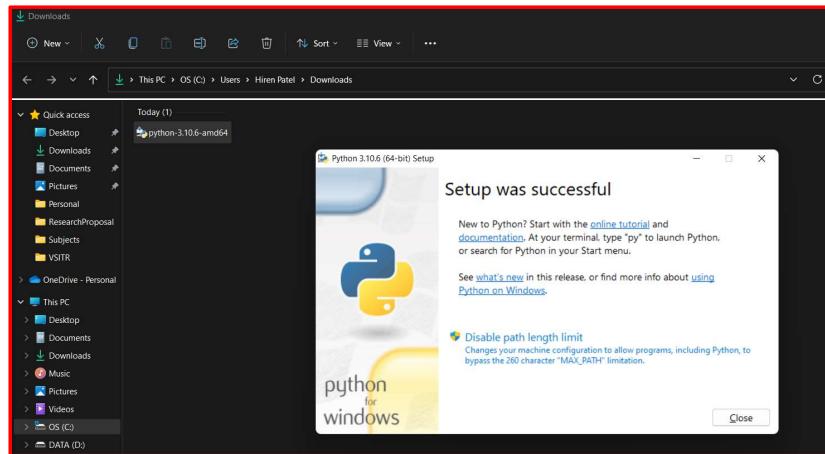
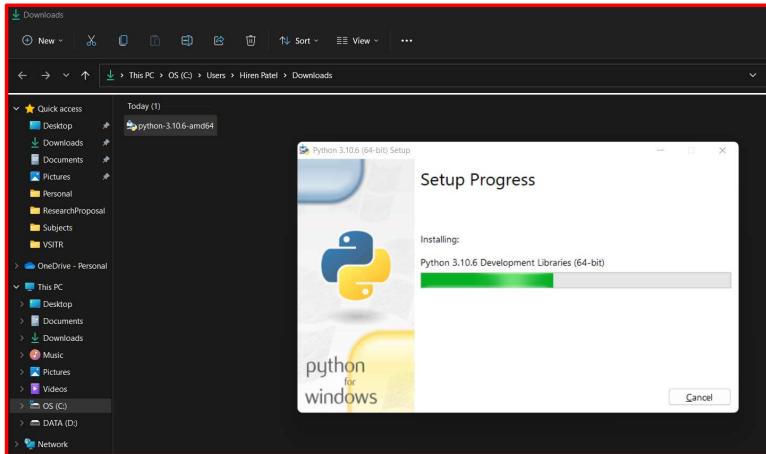
www.hbpatel.in





Installing Python

www.hbpatel.in



```
Downloads
New Sort View ...
This PC > OS (C) > Users > Hiren Patel > Downloads

python-3.10.6-amd64

Command Prompt - python
Microsoft Windows [Version 10.0.22000.708]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hiren Patel>python
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Verify Installation

```
Downloads
New Sort View ...
This PC > OS (C) > Users > Hiren Patel > Downloads

python-3.10.6-amd64

Command Prompt
C:\Users\Hiren Patel>pip -V
pip 22.2.1 from C:\Users\Hiren Patel\AppData\Local\Programs\Python\Python310\lib\site-packages\pip (python 3.10)
C:\Users\Hiren Patel>
```

Verify pip



Installing Code Editor (PyCharm)

(<https://www.jetbrains.com/pycharm/download/>)

www.hbpatel.in

The screenshot shows the PyCharm download page. At the top, there are links for Developer Tools, Team Tools, Learning Tools, Solutions, Support, Store, Pricing, and Download. Below this, there's a large "PyCharm" logo and a "Download PyCharm" section. It offers three editions: Professional (Windows, macOS, Linux), Community (Windows, macOS, Linux), and Educational (Windows, macOS, Linux). The Community edition is highlighted with a red box. A note says it's "Free, built on open-source". Below the editions, there's a "System requirements" section and a "Installation instructions" link. A purple banner at the bottom encourages using the JetBrains Startup discount.

The screenshot shows the "Thank you for downloading PyCharm!" page. It features a "Getting Started" section and a "New to PyCharm?" section. A purple banner at the bottom encourages using the JetBrains Startup discount. At the bottom, there's a download progress bar for "pycharm-community.exe" which is 15.0/377 MB, with 2 mins left.



Installing Code Editor (PyCharm)

(<https://www.jetbrains.com/pycharm/download/>)

www.hbpatel.in

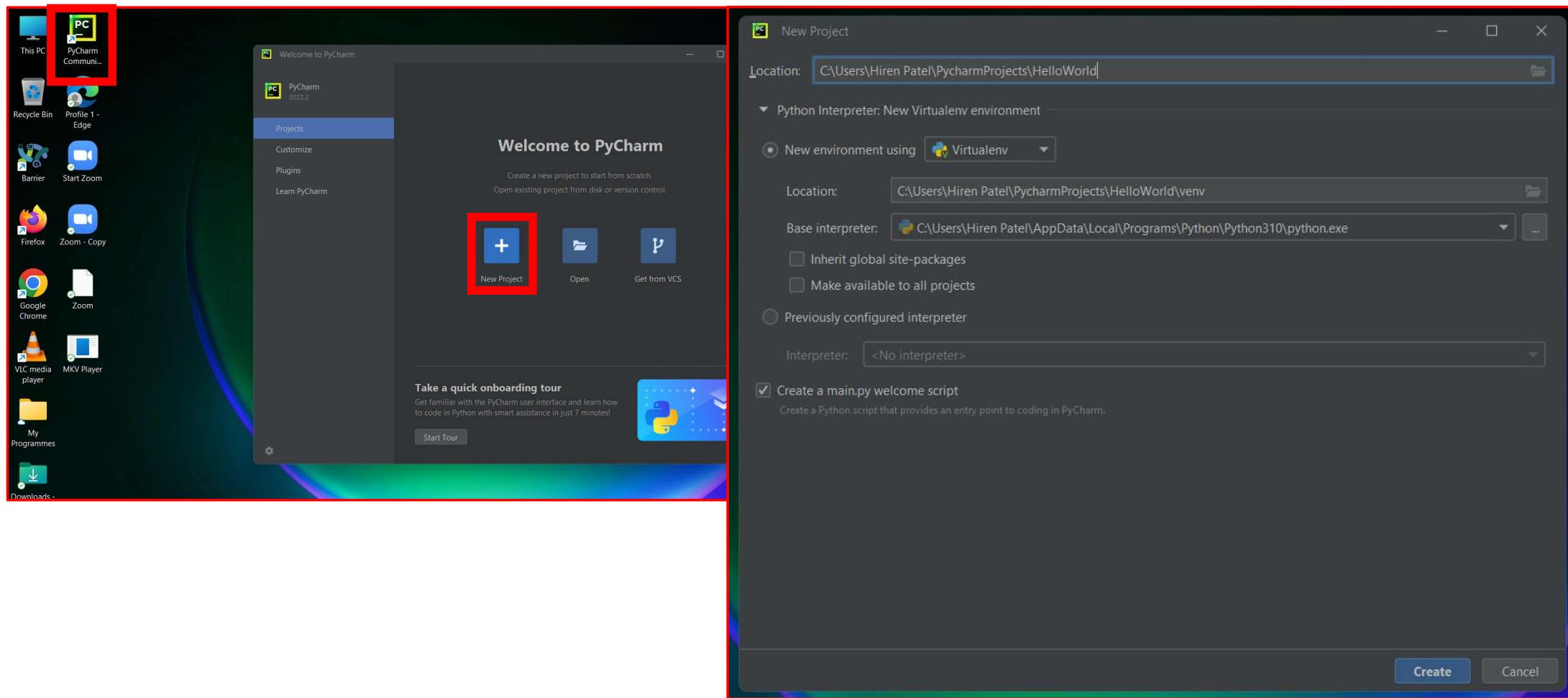
The image displays a sequence of six screenshots from the PyCharm Community Edition Setup process:

- Welcome to PyCharm Community Edition Setup:** A welcome screen with a "Next >" button.
- Choose Install Location:** A window asking to choose a destination folder, showing the default path `C:\Program Files\JetBrains\PyCharm Community Edition 2022.2`. It also shows available space: 1.4 GB required and 70.4 GB available.
- Installation Options:** A window for configuring the installation, with options like "Create Desktop Shortcut" (checked), "Update PATH Variable (restart needed)", and "Add 'bin' folder to the PATH".
- Choose Start Menu Folder:** A window for selecting a start menu folder, listing categories like JetBrains, Accessibility, Accessories, etc. An "Install" button is at the bottom.
- Installing:** A progress window showing the extraction of `cwm-lobby-common.jar`.
- Completing PyCharm Community Edition Setup:** A final confirmation screen stating the setup is complete, with a "Run PyCharm Community Edition" checkbox and a "Finish" button.
- JETBRAINS COMMUNITY EDITION TERMS:** A user agreement window containing the "IMPORTANT! READ CAREFULLY!" section, the Apache 2.0 license text, and a checkbox for accepting the terms.



Getting familiar with PyCharm

www.hbpatel.in





Getting familiar with PyCharm

www.hbpatel.in

PC File Edit View Navigate Code Refactor Run Tools VCS Window Help HelloWorld - main.py

HelloWorld > main.py

Project

HelloWorld C:\Users\Hiren Patel\PycharmProjects\HelloWorld

main.py

External Libraries

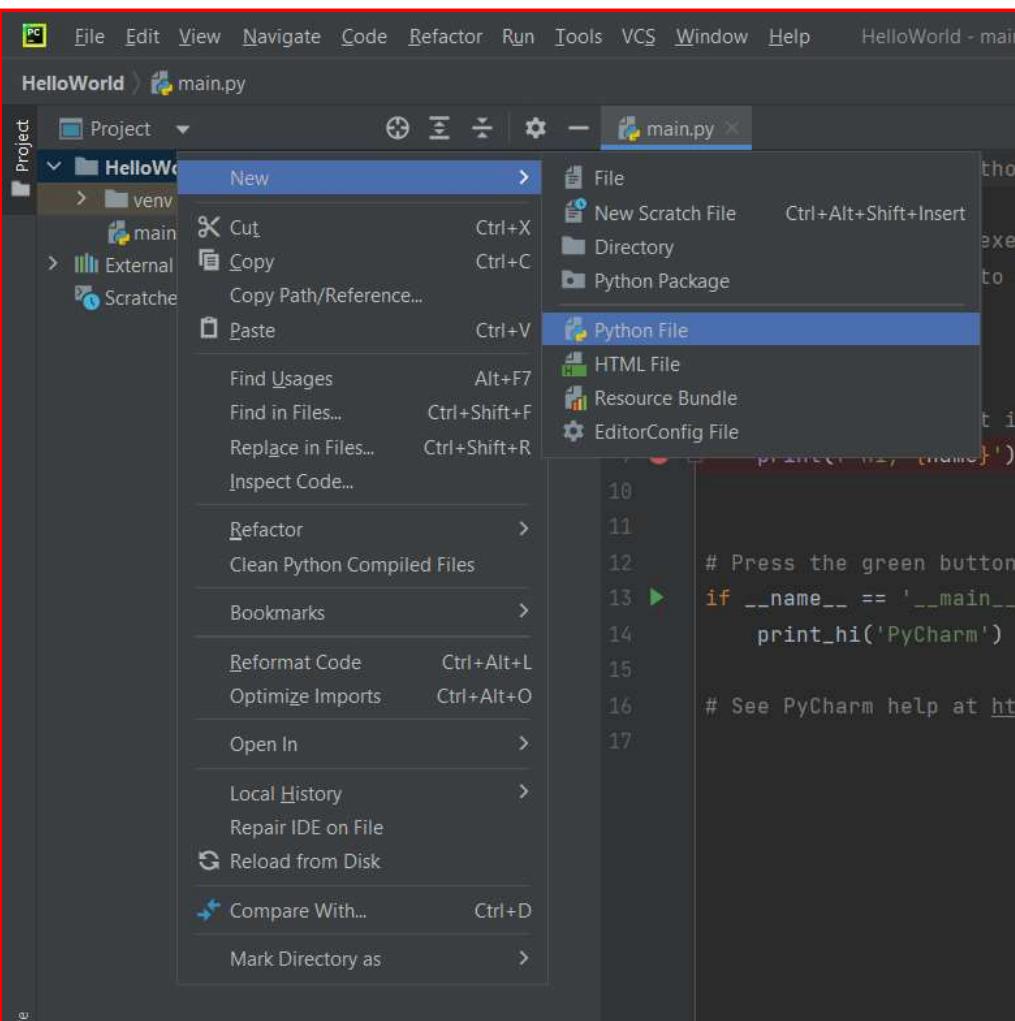
Scratches and Consoles

```
1 # This is a sample Python script.
2
3 # Press Shift+F10 to execute it or replace it with your code.
4 # Press Double Shift to search everywhere for classes, files, tool windows, actions, and settings.
5
6
7 def print_hi(name):
8     # Use a breakpoint in the code line below to debug your script.
9     print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.
10
11
12 # Press the green button in the gutter to run the script.
13 if __name__ == '__main__':
14     print_hi('PyCharm')
15
16 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
17
```



Getting familiar with PyCharm

www.hbpatel.in



The screenshot shows the 'New' submenu expanded, displaying options for creating new files. The 'Python File' option is selected. Other options shown are 'New Python file', 'app.py', 'Python file' (selected), 'Python unit test', and 'Python stub'. The main editor window shows the code for 'main.py'.

```
world - main.py

# Python script.

# to execute it or replace it with your code.

# If you want to search everywhere for classes, files, tool windows, actions, and
# settings, use the search in this window.

# Press the green button below to debug your script.

# Press Ctrl+F8 to toggle the breakpoint.

# Press the green button in the gutter to run the script.

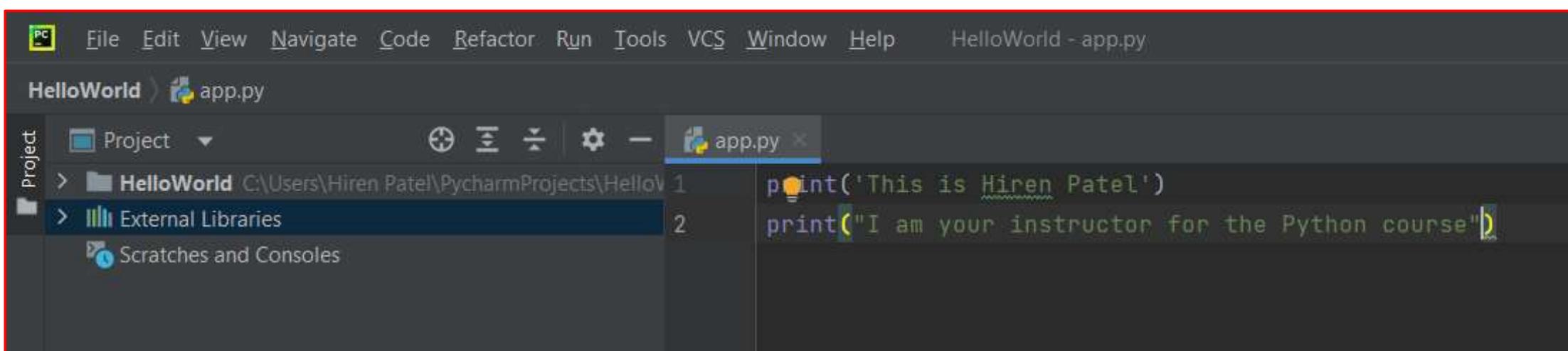
if __name__ == '__main__':
    print_hi('PyCharm')

# See PyCharm help at ht
```



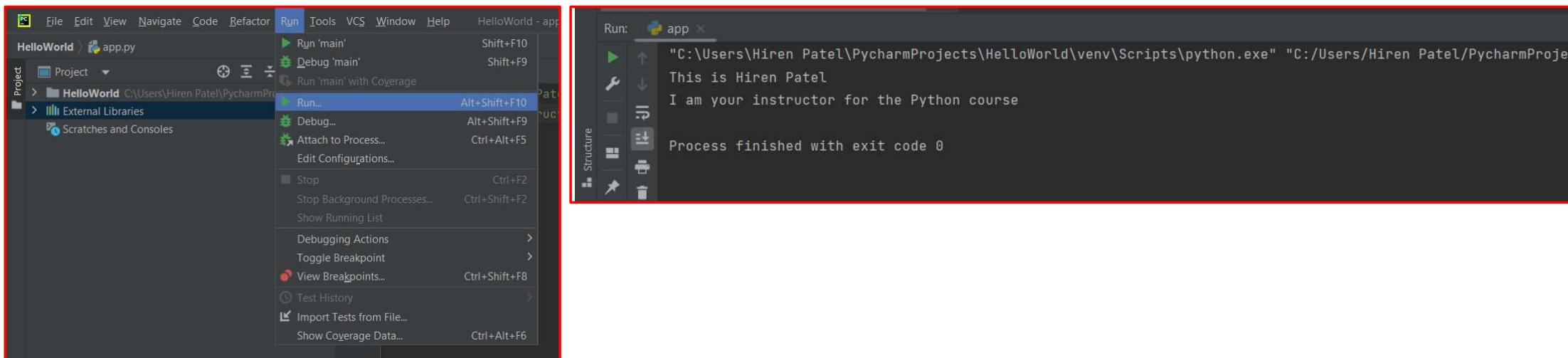
Writing small programs in PyCharm

www.hbpatel.in



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The current file is "HelloWorld - app.py". The code editor contains the following Python code:

```
print('This is Hiren Patel')
print("I am your instructor for the Python course")
```



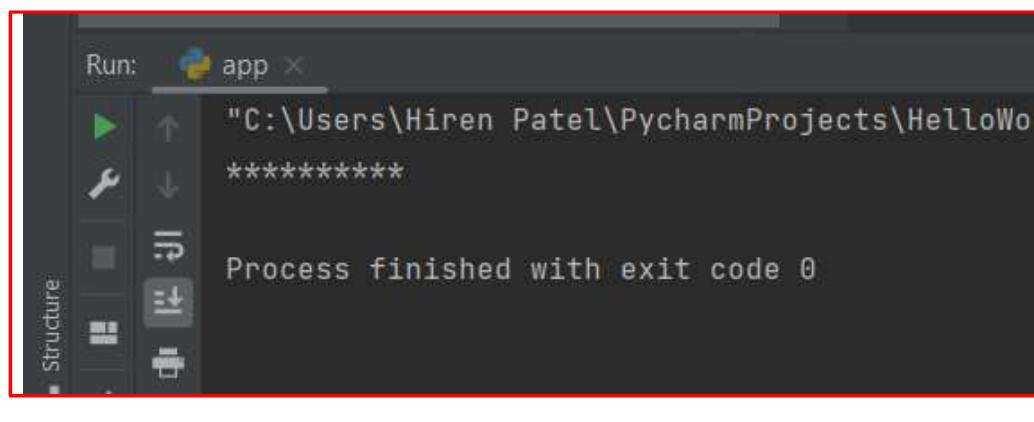
The screenshot shows the PyCharm IDE interface with the "Run" tool window open. The "Run" menu is visible, with "Run..." highlighted. The tool window displays the output of the program execution:

```
Run: app x
"C:\Users\Hiren Patel\PycharmProjects\HelloWorld\venv\Scripts\python.exe" "C:/Users/Hiren Patel/PycharmProjects/HelloWorld/app.py"
This is Hiren Patel
I am your instructor for the Python course
Process finished with exit code 0
```



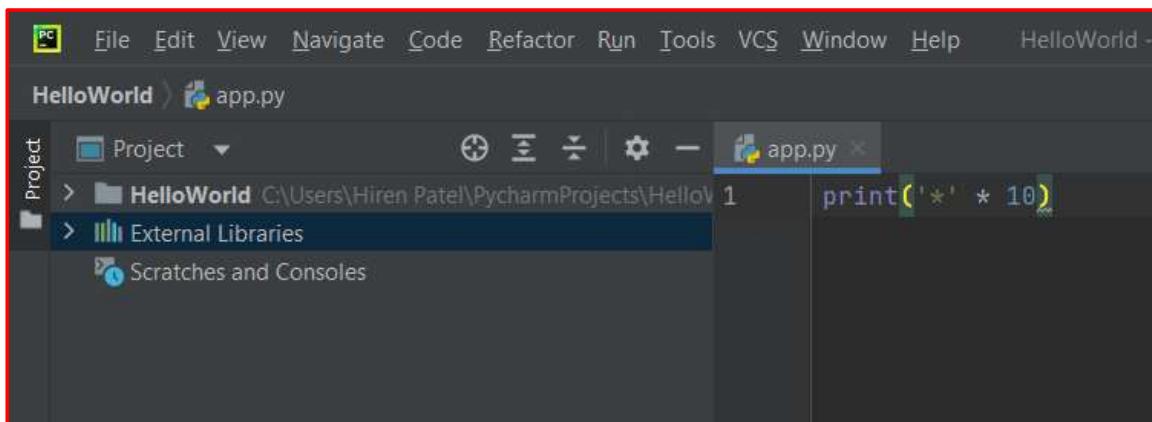
Writing small programs in PyCharm

www.hbpatel.in



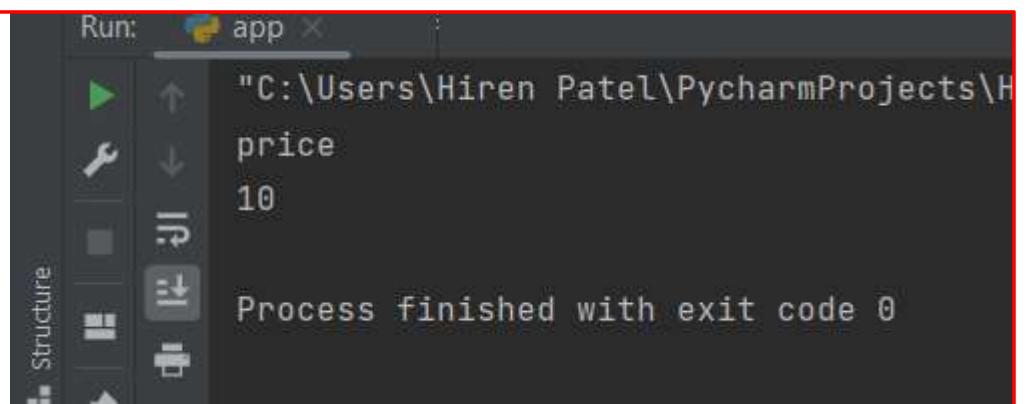
PyCharm interface showing the code `print('*' * 10)` in the app.py file. The terminal output shows the printed string of ten asterisks.

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help HelloWorld -> app.py
HelloWorld > app.py
Project > HelloWorld C:\Users\Hiren Patel\PycharmProjects\HelloWorld 1
External Libraries
Scratches and Consoles
print('*' * 10)
```



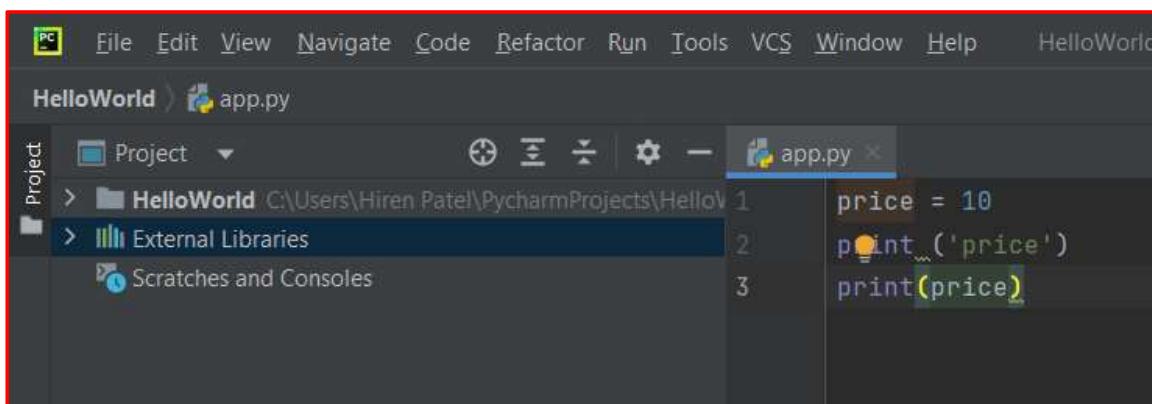
PyCharm interface showing the code `print('*' * 10)` in the app.py file. The terminal output shows the printed string of ten asterisks.

```
Run: app
" C:\Users\Hiren Patel\PycharmProjects\HelloWorld 1
*****
Process finished with exit code 0
```



PyCharm interface showing the code `price = 10
print('price')
print(price)` in the app.py file. The terminal output shows the variable `price` and its value `10`.

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help HelloWorld -> app.py
HelloWorld > app.py
Project > HelloWorld C:\Users\Hiren Patel\PycharmProjects\HelloWorld 1
External Libraries
Scratches and Consoles
1 price = 10
2 print('price')
3 print(price)
```



PyCharm interface showing the code `price = 10
print('price')
print(price)` in the app.py file. The terminal output shows the variable `price` and its value `10`.

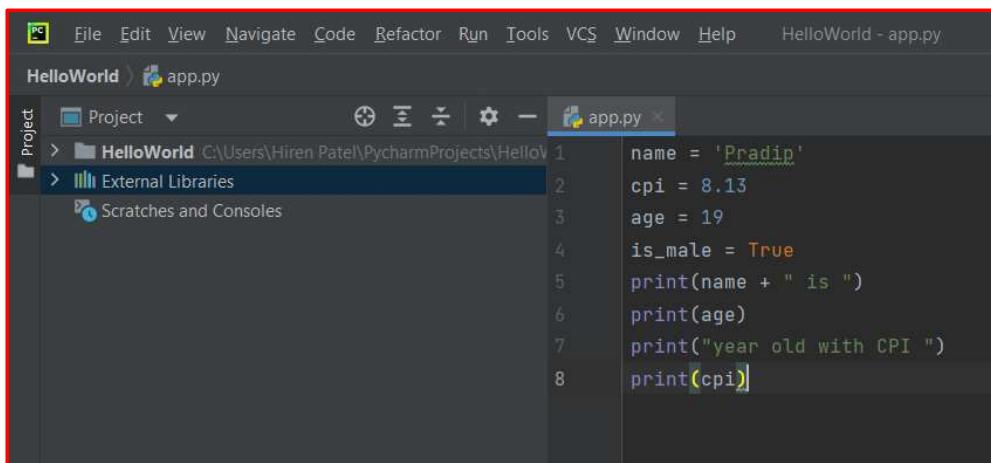
```
Run: app
" C:\Users\Hiren Patel\PycharmProjects\HelloWorld 1
price
10
Process finished with exit code 0
```

Variables



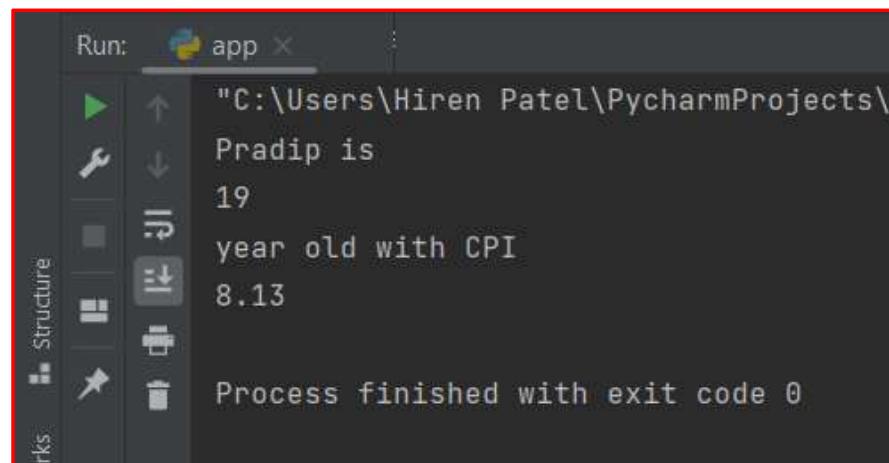
Variables

www.hbpatel.in



PyCharm IDE interface showing the code in `app.py`:

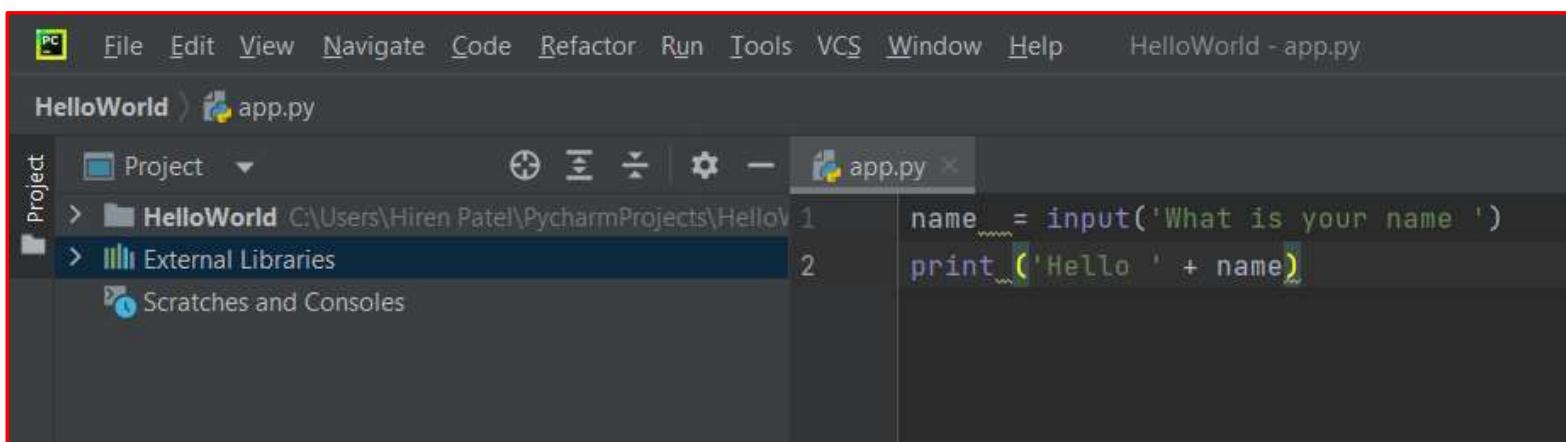
```
name = 'Pradip'
cpi = 8.13
age = 19
is_male = True
print(name + " is ")
print(age)
print("year old with CPI ")
print(cpi)
```



PyCharm Run tab output window showing the execution of `app.py`:

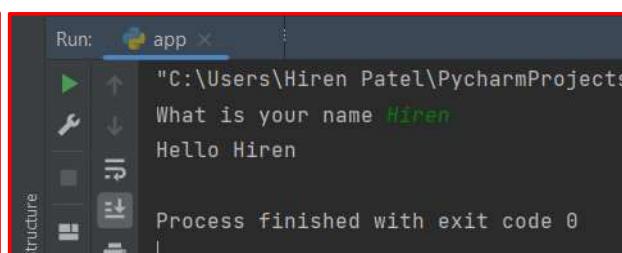
```
Pradip is
19
year old with CPI
8.13

Process finished with exit code 0
```



PyCharm IDE interface showing the code in `app.py`:

```
name = input('What is your name ')
print('Hello ' + name)
```



PyCharm Run tab output window showing the execution of `app.py`:

```
What is your name Hiren
Hello Hiren

Process finished with exit code 0
```

Getting input from the keyboard



Input from Keyboard and Type Casting

www.hbpatel.in

```
app.py
1 name = input('What is your name ')
2 food = input('What is your favorite food ')
3 print(name + ' loves ' + food)
```

```
app x
" C:\Users\Hiren Patel\PycharmProjects\HelloWorld\app.py "
What is your name Hiren
What is your favorite food Gulab Jamun
Hiren loves Gulab Jamun
Process finished with exit code 0
```

Getting input from the keyboard

```
app.py
1 birth_year = input('Enter your birth year ')
2 age = 2022 - birth_year
3 print(age)
```

```
app x
" C:\Users\Hiren Patel\PycharmProjects\HelloWorld\app.py "
Enter your birth year 1976
Traceback (most recent call last):
  File "C:\Users\Hiren Patel\PycharmProjects\HelloWorld\app.py", line 2, in <module>
    age = 2022 - birth_year
TypeError: unsupported operand type(s) for -: 'int' and 'str'

Process finished with exit code 1
```

Type casting / conversion

```
app.py
1 birth_year = input('Enter your birth year ')
2 age = 2022 - int(birth_year)
3 print(age)
```

```
app x
" C:\Users\Hiren Patel\PycharmProjects\HelloWorld\app.py "
Enter your birth year 1976
<class 'str'>
<class 'int'>
46
Process finished with exit code 0
```



Type Casting and Strings

The screenshot shows two panels of PyCharm. The left panel contains the Python code:

```
1 weight_pound = int(input('Enter weight in lbs : '))
2 weight_kg = weight_pound * 0.45
3 print(weight_kg)
```

The right panel shows the terminal output:

```
"C:\Users\Hiren Patel\PycharmProjects\HelloWorld\app.py" >
Enter weight in lbs : 100
45.0
Process finished with exit code 0
```

The screenshot shows two panels of PyCharm. The left panel contains the Python code:

```
1 subject1 = 'Python Programming'
2 subject2 = "Python Programming"
3 subject3 = "Python's Programming by Hiren Patel"
4 subject4 = '"Python" Programming by Hiren Patel'
5 subject5='''This is a multiline message.
Multiple lines are enclosed in three quotes.
The three quotes could be single or double.'''
6 print(subject1)
7 print(subject2)
8 print(subject3)
9 print(subject4)
10 print(subject5)
```

The right panel shows the terminal output:

```
"C:\Users\Hiren Patel\PycharmProjects\HelloWorld\app.py" >
Python Programming
Python Programming
Python's Programming by Hiren Patel
"Python" Programming by Hiren Patel
This is a multiline email.
Multiple lines are enclosed in three quotes.
The three quotes could be single or double.
Process finished with exit code 0
```



Playing with strings

www.hbpatel.in

```
app.py x
1 first_name = 'Hiren'
2 last_name = 'Patel'
3 msg1 = first_name + ' ' + last_name + ' is an instructor'
4 #formatted string
5 msg2 = f'{first_name} [{last_name}] is an instructor'
6 print_(msg1)
7 print_(msg2)
```

```
app x
1 "C:\Users\Hiren Patel\PycharmProject\HelloWorld\app.py:1: DeprecationWarning: the 'print' function is deprecated, use 'print_'
2 Hiren [Patel] is an instructor
3 Hiren [Patel] is an instructor
4
5 Process finished with exit code 0
```

```
app.py x
1 subject = 'Python programming'
2 print(len(subject))
3 print(subject.upper())
4 print(subject.lower())
5 print(subject.title())
6 print(subject[0].islower())
7 print(subject[0].isupper())
8 print(subject.find('P'))
9 print(subject.find('o'))
10 print(subject.find('p'))
11 print(subject.find('program'))
12 print(subject.replace('programming', 'coding'))
13 print('program' in subject)
```

```
app x
18
PYTHON PROGRAMMING
python programming
Python Programming
False
True
0
4
7
7
Python coding
True
```

PROGRAM

```
s = "hello"
print(s.capitalize())
print(s.upper())
print(s.rjust(7))
print(s.center(7))
print(s.replace('l', '(ell)'))
print(' world '.strip())
```

OUTPUT

```
Hello
HELLO
    hello
    hello
he(ell)(ell)o
world
```



Arithmetic Operators

```
app.py ×  
1 print_(10 + 3)  
2 print_(10 - 3)  
3 print_(10 * 3)  
4 print(10 / 3)  
5 print_(10 // 3)  
6 print_(10 % 3)  
7 print_(10 ** 3)
```

```
app ×  
C:\Users\Hiren Patel  
13  
7  
30  
3.333333333333335  
3  
1  
1000
```

```
app.py ×  
1 x = 10  
2 x = x+3  
3 print(x)  
4 x += 3  
5 print(x)  
6 x -= 3  
7 print(x)  
8 x *= 3  
9 print(x)  
10 x /= 3  
11 print(x)
```

```
app ×  
C:\Users\  
13  
16  
13  
39  
13.0
```



Arithmetic Operations

(Precedence and In-built functions)

```
app.py ×  
1 x = 10 + 3 * 2 ** 4  
2 print(x)  
3 x = (10 + 3) * 2 ** 4  
4 print(x)
```

```
app ×  
"C:\Users\Hi...  
58  
208
```

```
app.py ×  
1 x = 3.6  
2 print(x)  
3 print(round(x))  
4 x = -3.6  
5 print(abs(x))
```

```
app ×  
"C:\Users\Hi...  
3.6  
4  
3.6
```



Arithmetic Operations

(In-built mathematical functions)

```
import math  
  
x = 3.6  
  
print(x)  
  
print(math.ceil(x))  
print(math.floor(x))  
print(math.factorial(5))  
print(math.isfinite(100))  
print(math.sqrt(100))  
print(math.lcm(50, 40))  
print(math.gcd(50, 40))  
print(math.exp(10))  
print(math.log(100, 2))
```

```
app  
"C:\Users\Hiren Patel  
3.6  
4  
3  
120  
True  
10.0  
200  
10  
22026.465794806718  
6.643856189774725
```



Bitwise Operators

www.hbpatel.in

OPERATOR	DESCRIPTION	SYNTAX
&	Bitwise AND	x & y
	Bitwise OR	x y
~	Bitwise NOT	~x
^	Bitwise XOR	x ^ y
>>	Bitwise right shift	x>>
<<	Bitwise left shift	x<<

PROGRAM

```
a = 10
```

```
b = 4
```

```
# Print bitwise AND operation  
print("a & b =", a & b)
```

```
# Print bitwise OR operation  
print("a | b =", a | b)
```

```
# Print bitwise NOT operation  
print("~a =", ~a)
```

```
# print bitwise XOR operation  
print("a ^ b =", a ^ b)
```

OUTPUT

```
a & b = 0
```

```
a | b = 14
```

```
~a = -11
```

```
a ^ b = 14
```

```
=====
```

Explanation

a = 1010 (10)

b = 0100 (04)

a & b = 0000 (00)

a | b = 1110 (14)

~a = 0101 (-11, 2's complement)

a ^ b = 1110 (14)



Bitwise Operators

www.hbpatel.in

OPERATOR	DESCRIPTION	SYNTAX
&	Bitwise AND	x & y
	Bitwise OR	x y
~	Bitwise NOT	~x
^	Bitwise XOR	x ^ y
>>	Bitwise right shift	x>>
<<	Bitwise left shift	x<<

PROGRAM

```
a = 10
```

```
# print bitwise right shift  
operator
```

```
print("a >> 1 =", a >> 1)
```

```
a = 9
```

```
# print bitwise left shift  
operator
```

```
print("a << 1 =", a << 1)
```

OUTPUT

```
a >> 1 = 5
```

```
a << 1 = 18
```

```
=====
```

Explanation

```
a = 00001010 (10)
```

```
a >> 1 = 00000101 (5)
```

```
a = 00001001(5)
```

```
a << 1 = 00010010 (18)
```



Conditional Statements (if)

www.hbpatel.in

```
app.py ×  
marks = 49  
  
if marks >= 35:  
    print("Congratulation: You are PASS")  
print("Have a good day !")  
  
C:\Users\Hiren Patel\PycharmProjects  
Congratulation: You are PASS  
Have a good day !  
  
Process finished with exit code 0
```

```
app ×  
marks = 29  
if marks >= 35:  
    print("Congratulation: You are PASS")  
print("Have a good day !")  
  
C:\Users\Hiren Patel\PycharmProjects  
Have a good day !  
  
Process finished with exit code 0
```



Conditional Statements (if-else)

www.hbpatel.in

```
app.py x
marks = 59
if marks >= 35:
    print("Congratulation: You are PASS")
else:
    print("Sorry: Better luck next time")
print("Have a good day !")
```

```
app x
"C:\Users\Hiren Patel\PycharmProject"
Congratulation: You are PASS
Have a good day !

Process finished with exit code 0
```

```
app.py x
marks = 29
if marks >= 35:
    print("Congratulation: You are PASS")
else:
    print("Sorry: Better luck next time")
print("Have a good day !")
```

```
"C:\Users\Hiren Patel\PycharmProject"
Sorry: Better luck next time
Have a good day !

Process finished with exit code 0
```



Conditional Statements (if-elif-else)

www.hbpatel.in

```
marks = 80
if marks >= 70:
    print("Congratulation: You are PASS with Distinction")
elif marks >= 35:
    print("Congratulation: You are PASS")
else:
    print("Sorry: Better luck next time")
print("Have a good day !")
```

```
"C:\Users\Hiren Patel\PycharmProjects\HelloWorl
Congratulation: You are PASS with Distinction
Have a good day !
|
Process finished with exit code 0
```



Conditional Statements (if-elif-else)

www.hbpatel.in

```
app.py
# Logical Operators: AND (All conditions), OR (At least, one condition)
# Relational Operators: <, >, =
# Home loans are offered based on CIBIL score of the candidate
# if CIBIL score is more than or equal to 700 then loan is offered 7.90%
# if CIBIL score is within 600 and 699 then loan is offered at 8.00%
# if CIBIL score is within 500 and 599 loan is offered at 8.25%
# for CIBIL score less than 500, loan shall not be offered
# read the CIBIL score and display the loan option available
CIBIL = int(input('Enter the CIBIL Score : '))
print(CIBIL)
if (CIBIL >= 700) and (CIBIL <= 1000):
    print('You are eligible for the home loan at the rate of 7.90%')
elif (CIBIL >= 600) and (CIBIL <= 699):
    print('You are eligible for the home loan at the rate of 8.00%')
elif (CIBIL >= 500) and (CIBIL <= 599):
    print('You are eligible for the home loan at the rate of 8.25%')
elif (CIBIL >= 0) and (CIBIL <= 499):
    print('Sorry: You are not qualified for the loan')
else:
    print('Invalid CIBIL Score')
print('Thank you for using the loan rate calculator')
```

```
app.py
"C:\Users\Hiren Patel\PycharmProjects\HelloWorld\venv\Script"
Enter the CIBIL Score : 750
750
You are eligible for the home loan at the rate of 7.90%
Thank you for using the loan rate calculator

Process finished with exit code 0
```



Simple Programs

(Temperature Converter)

www.hbpatel.in

```
# Fahrenheit to Celsius Conversion: (F - 32) x .5556 = C
# Celsius to Fahrenheit Conversion: (C x 1.8) + 32 = F
temp = int(input('Enter Temperature : '))
unit = input('(F)ahrenheit to (C)celsius : ')
if unit.upper()=='F':
    converted = (temp - 32) * 0.5556
    print(f"Temperature in Celsius = {converted}")
elif unit.upper()=='C':
    converted = (temp * 1.8) + 32
    print(f"Temperature in Fahrenheit = {converted}")
else:
    print('Invalid Choice')
```

```
"C:\Users\Hiren Patel\PycharmProjects\HelloWorld"
Enter Temperature : 102
(F)ahrenheit to (C)celsius : F
Temperature in Celsius = 38.891999999999996
```



Loops (while)

```
i = 1
while i <= 5:
    print(i)
    i = i + 1
print('Have a good day')
```

```
i = 1
while i <= 5:
    print(i * '*')
    i = i + 1
print('Have a good day')
```

```
"C:\Users\Hiren Pa
1
2
3
4
5
Have a good day
```

```
"C:\Users\Hiren Pa
*
**
***
****
*****
Have a good day
```



Loops (while – simple program)

www.hbpatel.in

```
# Let us ask the user to make a guess (between 1 to 10) and s/he makes right guess, appropriate message should be printed ▲2 ✅
# We shall offer three attempts for the guess and s/he can't make the right guess, another appropriate message should be printed

private_number = 7
guessing_attempts = 3
guessing_counter = 0

while guessing_counter < guessing_attempts:
    my_guess = int(input('Guess a number : '))
    guessing_counter += 1
    if my_guess == private_number:
        print('Congratulations: You have made a right guess ! ')
        break
    else:
        print('Sorry: You are run out of all the attempts...better luck next time')
```

```
"C:\Users\Hiren Patel\PycharmProjects\HelloWorld\venv\Scripts\python
Guess a number : 1
Guess a number : 2
Guess a number : 3
Sorry: You are run out of all the attempts...better luck next time"
```

```
"C:\Users\Hiren Patel\PycharmProjects\HelloWorld\venv\Scripts\python
Guess a number : 1
Guess a number : 7
Congratulations: You have made a right guess !"
```



Loops

(while – simple program)

```
py >>>
# Write a menu driven program to turn a bulb ON/OFF
action = ''
bulb_status = False
while True:
    action = input(" : ").upper()
    if action == "ON":
        if bulb_status:
            print('Bulb is already ON')
        else:
            print('Turning ON the bulb....Bulb is ON now')
            bulb_status = True
    elif action == "OFF":
        if not bulb_status:
            print('Bulb is already OFF')
        else:
            print('Turning OFF the bulb....Bulb is OFF now')
            bulb_status = False
    elif action == "HELP":
        print('Valid Commands: ON, OFF, HELP, EXIT')
    elif action == "EXIT":
        break
print('Have a great day')
```

```
app >>> bulb >>>
C:\Users\Hiren Patel\PycharmProjects\HelloWorld>
: help
Valid Commands: ON, OFF, HELP, EXIT
: on
Turning ON the bulb....Bulb is ON now
: on
Bulb is already ON
: off
Turning OFF the bulb....Bulb is OFF now
: off
Bulb is already OFF
: exit
Have a great day

Process finished with exit code 0
```



Loops (for)

www.hbpatel.in

ir.py

```
for item in 'Python Coding':  
    print(item)
```

py

```
for item in ['Hiren', 'Sanjay', 'Vijay', 'Pradip', 'Parimal']:  
    print(item)
```

app

```
for item in [1, 2, 3, 4]:  
    print(item)
```

"C:\Us

"C:\Users\H

0

Hiren

1

Sanjay

2

Vijay

3

Pradip

4

Parimal

"C:\Us
1
2
3
4

```
for item in range(10):  
    print(item)
```

5

6

7

8

9



Loops (for)

www.hbpatel.in

```
for item in range(5, 10):  
    print(item)
```

```
app ✘  
"C:\\Us  
5  
6  
7  
8  
9
```

```
for.py ✘  
for item in range(5, 10, 2):  
    print(item)
```

```
app ✘  
"C:\\Us  
5  
7  
9
```



Four collection data types

- **List** is a collection which is ordered and changeable. Allows duplicate members.
- **Tuple** is a collection which is ordered and unchangeable. Allows duplicate members.
- **Set** is a collection which is unordered, unchangeable, and unindexed. No duplicate members.
- **Dictionary** is a collection which is ordered and changeable. No duplicate members.



Four collection data types

Program

```
myList = [10, 20, 30]
```

Output

```
[10, 20, 30]
```

```
print (myList)
```

```
3
```

```
print (len(myList))
```

```
(11, 21, 31)
```

```
myTuple = (11, 21, 31)
```

```
3
```

```
print (myTuple)
```

```
{40, 50, 60}
```

```
print (len(myTuple))
```

```
3
```

```
{'name': 'Hiren', 'age': 47}
```

```
mySet = {40, 50, 60}
```

```
print (mySet)
```

```
print (len(mySet))
```

```
myDictionary = {"name": "Hiren", "age": 47}
```

```
print (myDictionary)
```



List

```
items = [10, 22, 34, 49, 56, 66]
print(items)
items.append(71)
print(items)
items.insert(2, 25)
print(items)
items.remove(49)
print(items)
items.pop()
print(items)
print(items.index(56))
print(56 in items)
print(items.count(22))
items.sort()
print(items)
items.reverse()
print(items)
dup = items.copy()
dup.append(99);
print(dup)
items.clear()
print(items)
```

```
[10, 22, 34, 49, 56, 66]
[10, 22, 34, 49, 56, 66, 71]
[10, 22, 25, 34, 49, 56, 66, 71]
[10, 22, 25, 34, 56, 66, 71]
[10, 22, 25, 34, 56, 66]
4
True
1

[10, 22, 25, 34, 56, 66]
[66, 56, 34, 25, 22, 10]

[66, 56, 34, 25, 22, 10, 99]
[]
```



List

```
for rows in range(5):
    for columns in range(4):
        print(f"({rows}, {columns})")
```

(0,0)
(0,1)
(0,2)
(0,3)
(1,0)
(1,1)
(1,2)
(1,3)
(2,0)
(2,1)
(2,2)
(2,3)
(3,0)
(3,1)
(3,2)
(3,3)
(4,0)
(4,1)
(4,2)
(4,3)

```
[[10, 20, 80, 90], [22, 92, 72, 12], [33, 83, 73, 13]]
20
[[10, 70, 80, 90], [22, 92, 72, 12], [33, 83, 73, 13]]
10
70
80
90
22
92
72
12
33
83
73
13
```

```
matrix = [[10, 20, 80, 90], [22, 92, 72, 12], [33, 83, 73, 13]]
print(matrix)
print(matrix[0][1])
matrix[0][1] = 70
print(matrix)
for rows in matrix:
    for cols in rows:
        print(cols)
```



List

www.hbpatel.in

```
listOperation.py

# Remove duplicate elements from the given list
original_list = [10, 20, 10, 30, 40, 50, 30, 60, 30, 70, 80, 40]
unique_list = []
for element in original_list:
    if element not in unique_list:
        unique_list.append(element)
print(original_list)
print(unique_list)
```

```
listOperation.py
"C:\Users\Hiren Patel\PycharmProjects\HelloWorld\ve
[10, 20, 10, 30, 40, 50, 30, 60, 30, 70, 80, 40]
[10, 20, 30, 40, 50, 60, 70, 80]
```



List

```
y
items = ['Apple', 'Rice', 'Almond']
quantity = [2, 3, 1]
price = [50, 90, 800]
cost = 0
index = 0
for product in items:
    temp_cost = quantity[index] * price[index]
    print(f"{product}, {quantity[index]} x {price[index]} = Rs. {temp_cost}")
    cost += temp_cost
    index += 1
print(f"Total Cost of the Card = Rs. ", {cost})
```

Apple, 2 x 50 = Rs. 100
Rice, 3 x 90 = Rs. 270
Almond, 1 x 800 = Rs. 800
Total Cost of the Card = Rs. {1170}



Tuples

```
# Tuples - immutable
items = (1, 2, 3)
print(items.index(1))
print(items.count(3))
print(len(items))
```

```
"C:\\Users\\H
0
1
3
```

```
coordinates = (11, 22, 33)
x = coordinates[0]
y = coordinates[1]
z = coordinates[2]
print(x, y, z)

p, q, r = coordinates
print(p, q, r)

# unpacking is also available with List
List = [77, 88, 99]
a, b, c = List
print(a, b, c)
```

```
"C:\\Users\\H
11 22 33
11 22 33
77 88 99
```



Dictionaries

www.hbpatel.in

```
student = {  
    "name": "Pradip",  
    "age": 19,  
    "is_pass": True  
}  
  
print(student["name"])  
print(student.get("DoB"))  
print(student.get("DoB", "Feb 14 1976"))  
  
student["name"] = "Parimal"  
print(student["name"])  
  
student["DoB"] = "Jan 1 1976"  
print(student["DoB"])  
  
print(student)  
#print(student["DoB"])
```

```
"C:\Users\Hiren Patel\PycharmProjects\HelloWorld\venv\Scripts\python.exe"  
Pradip  
None  
Feb 14 1976  
Parimal  
Jan 1 1976  
{'name': 'Parimal', 'age': 19, 'is_pass': True, 'DoB': 'Jan 1 1976'}
```



Dictionaries

www.hbpatel.in

```
number = input("Enter a number: ")  
mapping = {  
    "0": "Zero",  
    "1": "One",  
    "2": "Two",  
    "3": "Three",  
    "4": "Four",  
    "5": "Five",  
    "6": "Six",  
    "7": "Seven",  
    "8": "Eight",  
    "9": "Nine"  
}  
  
output = ""  
for digit in number:  
    output += mapping.get(digit, "Default") + " "  
print(output)
```

```
"C:\Users\Hiren Patel\Py  
Enter a number: 6293  
Six Two Nine Three
```



Dictionaries

www.hbpatel.in

```
student_marks = {"Hiren": 80, "Pradip": 90, "Parimal": 75, "Sanjay": 65, "Vijay": 50}
student_grades = {}

for student in student_marks:
    if(student_marks[student]>90):
        student_grades[student]="Outstanding"
    elif(student_marks[student]>80):
        student_grades[student]="Exceeds Expectations"
    elif(student_marks[student]>70):
        student_grades[student]="Acceptable"
    else:
        student_grades[student]="Fail"
    print(student, student_marks[student], student_grades[student])

print(student_marks)
print(student_grades)
```

```
Hiren 80 Acceptable
Pradip 90 Exceeds Expectations
Parimal 75 Acceptable
Sanjay 65 Fail
Vijay 50 Fail
{'Hiren': 80, 'Pradip': 90, 'Parimal': 75, 'Sanjay': 65, 'Vijay': 50}
{'Hiren': 'Acceptable', 'Pradip': 'Exceeds Expectations', 'Parimal': 'Acceptable', 'Sanjay': 'Fail', 'Vijay': 'Fail'}
```



Dictionaries

www.hbpatel.in

```
States = [
    {"name": "Uttar Pradesh",
     "cities": ["Agra", "Aligarh", "Allahabad"],
     "population": 199812341
    },
    {"name": "Maharashtra",
     "cities": ["Mumbai", "Pune", "Solapur"],
     "population": 112374333
    },
    {"name": "Gujarat",
     "cities": ["Ahmedabad", "Baroda", "Rajkot"],
     "population": 60439692
    },
]
def add_new_state(state_name, state_cities, state_population):
    new_state = {}
    new_state["name"] = state_name
    new_state["city"] = state_cities
    new_state["population"] = state_population
    States.append(new_state)

add_new_state("Rajasthan", ["Udaipur", "Jesalmer", "Kumbhalgarh"], 68548437)
print(States)
print(States[2])
```



Dictionaries

www.hbpatel.in

OUTPUT

```
[{'name': 'Uttar Pradesh', 'cities': ['Agra', 'Aligarh', 'Allahabad'], 'population': 199812341}, {'name': 'Maharashtra', 'cities': ['Mumbai', 'Pune', 'Solapur'], 'population': 112374333}, {'name': 'Gujarat', 'cities': ['Ahmedabad', 'Baroda', 'Rajkot'], 'population': 60439692}, {'name': 'Rajasthan', 'city': ['Udaipur', 'Jesalmer', 'Kumbhalgarh'], 'population': 68548437}]\n{'name': 'Gujarat', 'cities': ['Ahmedabad', 'Baroda', 'Rajkot'], 'population': 60439692}
```



Functions

www.hbpatel.in

```
function.py
def welcome_function():
    print('Hello There...')
    print('Welcome to the function of Python')

print('Program starts here....')
welcome_function()
print('Program ends here....')
```

```
"C:\Users\Hiren Patel\PycharmProjects'
Program starts here...
Hello There...
Welcome to the function of Python
Program ends here....
```



Functions

```
def welcome_function(name):  
    print('Hello ' + name)  
    print('Welcome to the function of Python')  
  
print('Program starts here....')  
welcome_function('Hiren')  
print('Program ends here....')
```

Program starts here....
Hello Hiren
Welcome to the function of Python
Program ends here....

```
def welcome_function(name):  
    print('Hello ' + name)  
    print('Welcome to the function of Python')  
  
print('Program starts here....')  
welcome_function('Hiren')  
welcome_function('Pradip')  
print('Program ends here....')
```

Program starts here....
Hello Hiren
Welcome to the function of Python
Hello Pradip
Welcome to the function of Python
Program ends here....



Functions

```
def welcome_function(first_name, last_name):  
    print(f'Hello {first_name} {last_name}')  
    print('Welcome to the function of Python')  
  
print('Program starts here....')  
welcome_function('Hiren', 'Patel')  
print('Program ends here....')
```

Program starts here....
Hello Hiren Patel
Welcome to the function of Python
Program ends here....

```
def welcome_function(first_name, last_name):  
    print(f'Hello {first_name} {last_name}')  
    print('Welcome to the function of Python')  
  
print('Program starts here....')  
welcome_function('Patel', 'Hiren')  
print('Program ends here....')
```

Program starts here....
Hello Patel Hiren
Welcome to the function of Python
Program ends here....

```
def welcome_function(first_name, last_name):  
    print(f'Hello {first_name} {last_name}')  
    print('Welcome to the function of Python')  
  
print('Program starts here....')  
welcome_function(last_name='Patel', first_name='Hiren')  
print('Program ends here....')
```

Program starts here....
Hello Hiren Patel
Welcome to the function of Python
Program ends here....



Functions

www.hbpatel.in

```
def emoji_converter (msg):  
    statement = msg.split(" ")  
    emojis = {  
        ":)": "\ud83d\udcbb",  
        ":(": "\ud83d\udcbe"  
    }  
    result = ""  
    for word in statement:  
        result += emojis.get(word, word) + " "  
    return result
```

```
message = input("Enter a string : ")  
print(emoji_converter(message))
```

```
Enter a string : I am good :)  
I am good \ud83d\udcbb
```

```
Enter a string : I am sad :(  
I am sad \ud83d\udcbe
```

```
def square (num):  
    return num * num  
  
print(square(3))
```

```
9
```

```
def simple_interest(principal_amount, rate_of_interest, number_of_years):  
    return (principal_amount * rate_of_interest * number_of_years) / 100
```

```
print(f"Simple Interest is Rs. {simple_interest(1000, 7.5, 5)}")
```

```
Simple Interest is Rs. 375.0
```



Functions

www.hbpatel.in

```
num = int(input('Enter a number : '))
print(num)
```

```
Enter a number : 10
10
```

```
Enter a number : abc
Traceback (most recent call last):
  File "C:\Users\Hiren Patel\PycharmProjects\HelloWorld\exception.py", line 1, in <module>
    num = int(input('Enter a number : '))
ValueError: invalid literal for int() with base 10: 'abc'
```

```
try:
    num = int(input('Enter a number : '))
    print(num)
except ValueError:
    print('Invalid Number')
```

```
Enter a number : abc
Invalid Number
```



Class/Object (OOP)

www.hbpatel.in

sses.py

```
class Point:  
    def __init__(self, f, s):  
        self.x = f  
        self.y = s  
  
    def move(self):  
        print('move')  
  
    def draw(self):  
        print('draw')  
  
  
obj = Point(1000, 2000)  
obj.x = 1500  
print(obj.x, obj.y)  
obj.move()  
obj.draw()
```

"C:\Users\Hirer

1500 2000

move

draw

class Student:

parameterized constructor

```
def __init__(self, n):
```

```
    self.name = n
```

```
def talk(self):
```

```
    print("Hi, My Name is " + self.name)
```

```
student1 = Student('Pradip')
```

```
student1.talk()
```

```
student2 = Student('Sanjay')
```

```
student2.talk()
```



Class/Object (OOP)

www.hbpatel.in

```
entance.py  classes.py

class Person:
    def displayperson(self):
        print('This is a person class')

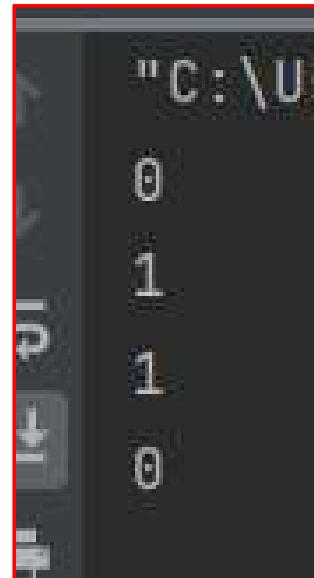
class Employee(Person):
    def displayemployee(self):
        print('This is an Employee class')

class Student(Person):
    def displaystudent(self):
        print('This is a student class')

emp = Employee()
emp.displayperson()
emp.displayemployee()

stu = Student()
stu.displayperson()
stu.displaystudent()
```

```
EXCEPTION   INSTANCE
This is a person class
This is an Employee class
This is a person class
This is a student class
```



```
class User:
    def __init__(self, userID, userName):
        self.id = userID
        self.username = userName
        self.following = 0
        self.followers = 0

    def follow(self, user):
        user.followers += 1
        self.following += 1

user1 = User("1", "Hiren")
user2 = User("1", "Pradip")

user1.follow(user2)

print(user1.followers)
print(user1.following)
print(user2.followers)
print(user2.following)
```



Polymorphism (OOP)

Program (In-built Polymorphism)

```
print (len([10, 20, 30, 40]))  
print (len((11, 22, 33)))  
print (len({40, 50, 60, 70, 80}))  
print (len({"Hiren": 47, "Pradip": 32}))
```

Output

4
3
5
2

Program (Polymorphism with user-defined function)

```
def add(x, y, z = 0):  
    return x + y + z  
  
print(add(2, 3))  
print(add(2, 3, 4))
```

Output

5
9



Polymorphism (OOP)

www.hbpatel.in

Program (Polymorphism with class methods)

```
class India():
    def capital(self):
        print("New Delhi is the capital of India")
```

```
class USA():
    def capital(self):
        print("Washington, D.C. is the capital of USA")
```

```
class UAE():
    def capital(self):
        print("Abu Dhabi is the capital of UAE")
```

```
obj_ind = India()
obj_usa = USA()
obj_uae = UAE()
for country in (obj_ind, obj_usa, obj_uae):
    country.capital()
```

Output

```
New Delhi is the capital of India
Washington, D.C. is the capital of USA
Abu Dhabi is the capital of UAE
```



Polymorphism (OOP)

www.hbpatel.in

Program (Polymorphism with Inheritance (along with Method Overriding))

```
class Bird:  
    def about(self):  
        print("This is a bird class -  
Parent class")  
  
    def canFly(self):  
        print("Not all birds can fly")  
  
class sparrow(Bird):  
    def canFly(self):  
        print("Sparrows can fly.")  
  
class ostrich(Bird):  
    def canFly(self):  
        print("Ostriches cannot fly.")
```

```
obj_bird = Bird()  
obj_spr = sparrow()  
obj_ost = ostrich()  
obj_bird.about()  
obj_bird.canFly()  
obj_spr.about()  
obj_spr.canFly()  
obj_ost.about()  
obj_ost.canFly()
```

Output

```
This is a bird class - Parent class  
Not all birds can fly  
  
This is a bird class - Parent class  
Sparrows can fly.  
  
This is a bird class - Parent class  
Ostriches cannot fly.
```



Polymorphism (OOP)

Program (Polymorphism with a function and objects)

```
class India():
    def capital(self):
        print("New Delhi is the capital of India")

class USA():
    def capital(self):
        print("Washington, D.C. is the capital of USA")

class UAE():
    def capital(self):
        print("Abu Dhabi is the capital of UAE")

for country in (obj_ind, obj_usa, obj_uae):
    country.capital()

def func(obj):
    obj.capital()
```

```
obj_ind = India()
obj_usa = USA()
obj_uae = UAE()

func(obj_ind)
func(obj_usa)
func(obj_uae)
```

Output

```
New Delhi is the capital of India
Washington, D.C. is the capital of USA
Abu Dhabi is the capital of UAE
New Delhi is the capital of India
Washington, D.C. is the capital of USA
Abu Dhabi is the capital of UAE
```



Sorting (Bubble Sort)

```
def bubbleSort (A) :  
    n = len(A)  
    flag = False  
    for i in range(n-1) :  
        for j in range(0, n-i-1) :  
            if A[j] > A[j + 1] :  
                flag = True  
                A[j], A[j + 1] = A[j + 1], A[j]  
            if not flag:  
                return  
A = [42, 23, 74, 11, 65, 58]  
bubbleSort(A)  
print(A)
```



Sorting (Selection Sort)

```
def selectionSort(numbers, size):
    for I in range(size):
        iMin = I
        for j in range(I + 1, size):
            if numbers[j] < numbers[iMin]:
                iMin = j
        (numbers[I], numbers[iMin]) = (numbers[iMin], numbers[I])
numbers = [42, 23, 74, 11, 65, 58]
size = len(numbers)
selectionSort(numbers, size)
print(numbers)
```



Sorting (Insertion Sort)

```
def insertionSort(numbers):
    for i in range(1, len(numbers)):
        value = numbers[i]
        hole = i-1
        while hole >=0 and value < numbers[hole] :
            numbers[hole+1] = numbers[hole]
            hole -= 1
        numbers[hole+1] = value
numbers = [42,23,74,11,65,58]
insertionSort(numbers)
print(numbers)
```



Sorting (Quick Sort)

```
def partition(A, start, end):
    pivot = A[end]
    pIndex = start - 1
    for i in range(start, end):
        if A[i] <= pivot:
            pIndex = pIndex + 1
            (A[pIndex], A[i]) = (A[i], A[pIndex])
    (A[pIndex + 1], A[end]) = (A[end], A[pIndex + 1])
    return pIndex + 1

def quickSort(A, start, end):
    if start < end:
        pIndex = partition(A, start, end)
        quickSort(A, start, pIndex - 1)
        quickSort(A, pIndex + 1, end)

numbers = [5,7,10,5,2,9,1,8,6,3]
size = len(numbers)
quickSort(numbers, 0, size - 1)
print(numbers)
```



Searching (Linear)

```
def searchLinear(A, n, x):
    for i in range(0, n):
        if (A[i] == x):
            return i
    return -1

A = [77, 12, 8, 39, 27, 21, 44, 18, 6, 47, 11, 37, 60, 56]
x = 39

searchResult = searchLinear(A, len(A), x)
if searchResult == -1:
    print ("Not Found")
else:
    print (x, "Found at Location ", searchResult)
```



Searching (Binary)

```
def searchBinary(A, x, start, end):
    while start <= end:
        middle = start + (end - start)//2
        if A[middle] == x:
            return middle
        elif A[middle] < x:
            start = middle + 1
        else:
            end = middle - 1
    return -1

A = [6,8,11,12,18,21,27,37,39,44,47,58,60,77]
x = 39
searchResult = searchBinary(A, x, 0, len(A))
if searchResult == -1:
    print ("Not Found")
else:
    print (x, "Found at Location ", searchResult)
```



Functions and Modules (Standard Modules: sys)

www.hbpatel.in

Program

```
import sys  
print(sys.version)
```

Output

```
3.7.15 (default, Oct 12 2022, 19:14:55)  
[GCC 7.5.0]
```

Program

```
import sys  
sys.stdout.write('Python By Hiren')
```

Output

```
Python By Hiren
```

Program

```
import sys  
age = 17  
if age < 18:  
    sys.exit("Age less than 18")  
else:  
    print("Age is not less than 18")
```

Output

```
An exception has occurred, use %tb to see  
the full traceback.  
SystemExit: Age less than 18
```



Functions and Modules (Standard Modules: sys)

www.hbpatel.in

Program

```
import sys
age = 20
if age < 18:
    sys.exit("Age less than 18")
else:
    print("Age is not less than 18")
```

Output

Age is not less than 18

Program

```
import sys
print(sys.path)
```

Output

```
['/content', '/env/python', '/usr/lib/python37.zip',
 '/usr/lib/python3.7', '/usr/lib/python3.7/lib-dynload', '',
 '/usr/local/lib/python3.7/dist-packages',
 '/usr/lib/python3/dist-packages',
 '/usr/local/lib/python3.7/dist-packages/IPython/extensions',
 '/root/.ipython']
```

Program

```
import sys
print(sys.modules)
```

Output

```
{'sys': <module 'sys' (built-in)>, 'builtins': <module
'builtins' (built-in)>, '_frozen_importlib': <module
'importlib._bootstrap' (frozen)>, '_imp': <module '_imp'
(built-in)>, '_thread': <module '_thread' (built-in)>,
'_warnings': <module '_warnings' (built-in)>, '_weakref':
<module '_weakref' (built-in)>, 'zipimport': <module
'zipimport' (built-in)>, '_frozen_importlib_external': <module
'importlib._bootstrap_external' (frozen)>, '_io': <module 'io'
(built-
```



Functions and Modules (Standard Modules: math)

Program

```
import math  
print (math.tau)  
print (math.e)  
print (math.pi)  
print (math.inf)  
print (-math.inf)  
print (math.inf > 10e108)  
print (-math.inf < -10e108)  
print (math.nan)  
  
a = 2.3  
print (math.ceil(a))  
print (math.floor(a))  
  
a = 5  
print (math.factorial(a))
```

```
a = 15  
b = 5  
print (math.gcd(b, a))  
  
a = -10  
print (math.fabs(a))  
  
test_int = 4  
test_neg_int = -3  
test_float = 0.00  
  
print (math.exp(test_int))  
print (math.exp(test_neg_int))  
print (math.exp(test_float))
```

Output

```
6.283185307179586  
2.718281828459045  
3.141592653589793  
inf  
-inf  
True  
True  
nan  
3  
2  
120  
5  
10.0  
54.598150033144236  
0.049787068367863944  
1.0
```



Error Handling

www.hbpatel.in

Program

```
print(x)
```

Output

```
Traceback (most recent call last):
```

```
  File "C:\Users\VSITR\PycharmProjects\pythonProject\main.py", line 1, in <module>
    print(x)
```

```
NameError: name 'x' is not defined
```

Program

```
try:
    print(x)
except:
    print("An exception occurred")
```

Output

```
An exception occurred
```

Program

```
a = [1, 2, 3]
```

try:

```
    print ("Second element = %d" %(a[1]))
```

```
    print ("Fourth element = %d" %(a[3]))
```

except:

```
    print ("An error occurred")
```

Output / Error

```
Second element = 2
```

```
An error occurred
```



Error Handling

www.hbpatel.in

Program

```
a = 10  
b = 0  
print (a/b)
```

Output / Error

```
ZeroDivisionError Traceback (most recent call last)  
<ipython-input-3-57a7363b4947> in <module>
```

```
    1 a = 10  
    2 b = 0  
----> 3 print (a/b)  
ZeroDivisionError: division by zero
```

Program

```
a = 10  
b = 0  
try:  
    print (a/b)  
except ZeroDivisionError:  
    print("Can't divide by zero")
```

Output / Error

```
Can't divide by zero
```

Program

```
a = 10  
b = 0  
try:  
    print (a/b)  
except ZeroDivisionError:  
    print("Can't divide by zero")  
finally:  
    print('This always gets executed')
```

Output / Error

```
Can't divide by zero  
This always gets executed
```



Error Handling

www.hbpatel.in

Program

```
try:  
    a = int(input("Enter a positive integer: "))  
    if a <= 0:  
        raise ValueError("That is NOT a positive number!")  
    else:  
        print("This is a positive number!")  
except ValueError as ve:  
    print(ve)
```

Output

Enter a positive integer: 10

This is a positive number!

Enter a positive integer: -10

That is NOT a positive number!



I/O Handling

www.hbpatel.in

Program

```
file = open('sample.txt', 'r')
print('The file name is: ', file.name)
```

Output / Error

```
Traceback (most recent call last):
  File "C:\Users\VSITR\PycharmProjects\pythonProject\main.py", line 1, in <module>
    file = open('sample.txt', 'r')
FileNotFoundError: [Errno 2] No such file or directory: 'sample.txt'
```

Program

```
try:
    file = open('sample.txt', 'r')
    print('File found!!!!')

except IOError:
    print('File not found!!!!')
```

Output / Error

```
File not found!!!
```



I/O Handling

www.hbpatel.in

Program

```
num = [3, 4, 5, 7]
try:
    if len(num) > 3:
        raise Exception(f"Length of the given list must be less than or equal to 3 but is {len(num)}")
except Exception as ex:
    print(ex)
```

Output

Length of the given list must be less than or equal to 3 but is 4
File Management

Program

```
file1 = open("myfile.txt", "w")
L = ["VSITR Kadi \n", "KSV Gandhinagar \n", "SVKM Kadi \n"]
# Writing data to a file
file1.write("Hello \n")
file1.writelines(L)
file1.close()
```

Output [Contents of myfile.txt (in the same folder)]

Hello
VSITR Kadi
KSV Gandhinagar
SVKM Kadi



I/O Handling

Program

```
file1 = open("myfile.txt", "r+")
print("Output of Read function is ")
print(file1.read())
print()
```

Output

```
Hello
VSITR Kadi
KSV Gandhinagar
SVKM Kadi
```

Program

```
file1 = open("myfile.txt", "r+")
file1.seek(0)
print("Output of Readline function is ")
print(file1.readline())
print()
```

Output

```
Hello
```

Program

```
file1 = open("myfile.txt", "r+")
file1.seek(0)
print("Output of Read(9) function is ")
print(file1.read(9))
print()
```

Output

```
Hello
VS
```

Program

```
file1 = open("myfile.txt", "r+")
file1.seek(0)
print("Output of Readlines function is ")
print(file1.readlines())
print()
```

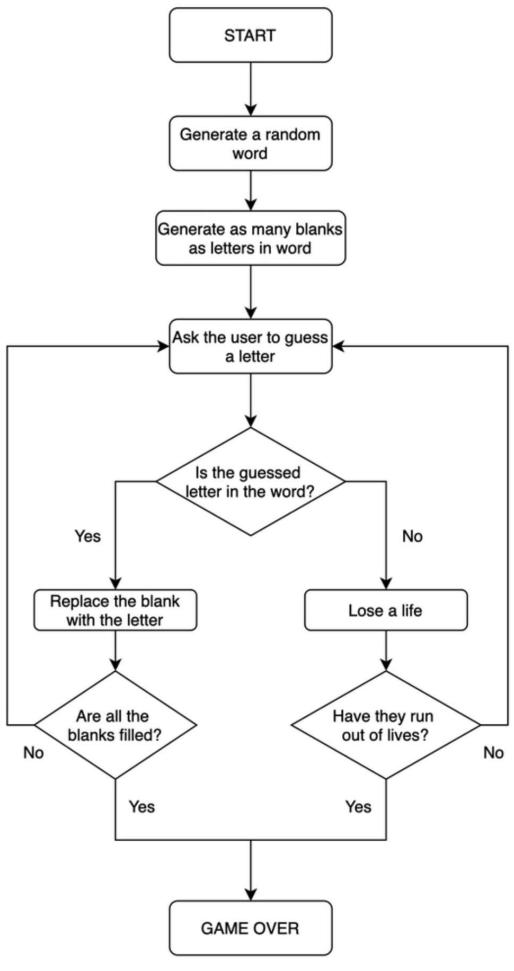
Output

```
['Hello \n', 'VSITR Kadi \n', 'KSV
Gandhinagar \n', 'SVKM Kadi \n']
```



Projects - Hangman Problem

www.hbpatel.in



```
stages = [
    +---+
    |   |
    O   |
    / \  |
    / \  |
    |
=====
    ' ', ' '
    +---+
    |   |
    O   |
    / \  |
    /   |
    |
=====
    ' ', ' '
    +---+
    |   |
    O   |
    / \  |
    |
    |
=====
    ' ', ' '
    +---+
    |   |
    O   |
    /   |
    |
    |
=====
```

```

    +---+
    |   |
    O   |
    |   |
    |   |
    =====
    ' ', ' '
    +---+
    |   |
    O   |
    |   |
    |   |
    =====
    ' ', ' '
    +---+
    |   |
    |   |
    |   |
    |   |
    |   |
    =====
    ' '']

```



Projects - Hangman Problem

www.hbpatel.in

```
game_over = False
word_list = ["hiren", "patel"]
chosen_word =
random.choice(word_list)
word_length = len(chosen_word)
lives = 6
display = []

for letter in chosen_word:
    display.append("_")

#list that stores prior gussed
letters
guessed_letters = []

print(logo)
```

```
while not game_over:
    already_guessed = ""
    guess = input("Guess a letter: ").lower()
    wrong_text = f"Letter {guess} is not in the word. Try again!"

    #Check if letter in chosen word, reveal if yes
    for position in range(word_length):
        letter = chosen_word[position]
        if letter == guess:
            display[position] = guess

    #Check if letter has already been guessed before by
    #checking guessed_letters
    if guess not in guessed_letters:
        guessed_letters.append(guess)
        already_guessed = False
    elif guess in guessed_letters:
        print("You've already guessed that letter. \n_ _")
        already_guessed = True
```



Projects - Hangman Problem

www.hbpatel.in

```
#Reduce lives if wrong and game over conditions
if guess not in chosen_word and already_guessed == False:
    lives -= 1
    print(wrong_text)
    already_guessed = True
elif guess not in chosen_word and already_guessed == True:
    print(wrong_text)

hangman = stages[lives]
print(hangman)

#Game over condition
if lives <= 0:
    game_over = True
    print(lose)
    print(f"The word was {chosen_word}")

print(f"{' '.join(display)}")

if "_" not in display:
    game_over = True
    print(win)
```



Projects – Caesar Cipher

www.hbpatel.in

```
alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
symbols_and_spaces = [' ', '!', '?', '.', ',', ':', ';', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0']

def caesar(plain_text, shift_amount, cipher_direction):
    output = ''
    if cipher_direction == 'decode':
        shift_amount *= -1
    for letter in plain_text:
        if letter in alphabet:
            letter_int = alphabet.index(letter)
            processed_letter = alphabet[(letter_int + shift_amount) % len(alphabet)]
            output += processed_letter
        if letter in symbols_and_spaces:
            output += letter
    print(f'The {cipher_direction}d text is {output}')
restart = True
while restart == True:
    direction = input("Type 'encode' to encrypt, type 'decode' to decrypt:\n")
    text = input("Type your message:\n").lower()
    shift = int(input("Type the shift number:\n"))

    caesar(plain_text=text, shift_amount=shift, cipher_direction=direction)

    restart_program = input("Restart the program? [y/n] ").lower()
    if restart_program == 'n':
        restart = False
    print("Good bye!")
```



Projects – Calculator

www.hbpatel.in

```
def add(n1, n2):
    return n1 + n2

def subtract(n1, n2):
    return n1 - n2

def multiply(n1, n2):
    return n1 * n2

def divide(n1, n2):
    return n1 / n2

operations = {
    "+": add,
    "-": subtract,
    "*": multiply,
    "/": divide
}
```

```
def calculator():

    num1 = float(input("What's the first number?: "))
    for symbol in operations:
        print(symbol)
    should_continue = True

    while should_continue:
        operation_symbol = input("Pick an operation: ")
        num2 = float(input("What's the next number?: "))
        calculation_function = operations[operation_symbol]
        answer = calculation_function(num1, num2)
        print(f"{num1} {operation_symbol} {num2} = {answer}")

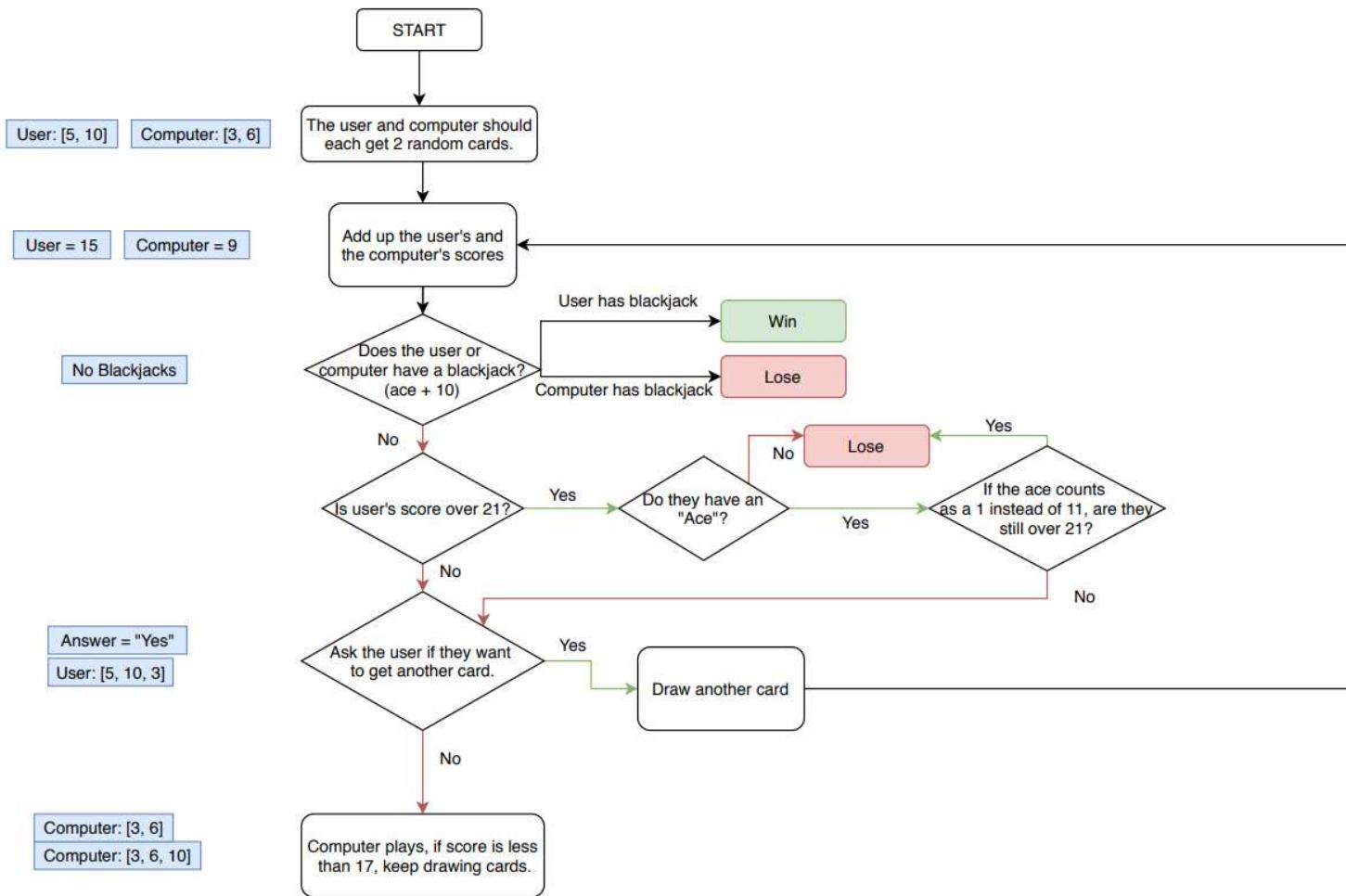
        if input(f>Type 'y' to continue calculating with {answer}, or type 'n' to
start a new calculation: ") == 'y':
            num1 = answer
        else:
            should_continue = False
            clear()
            calculator()

calculator()
```



Projects – Black Jack

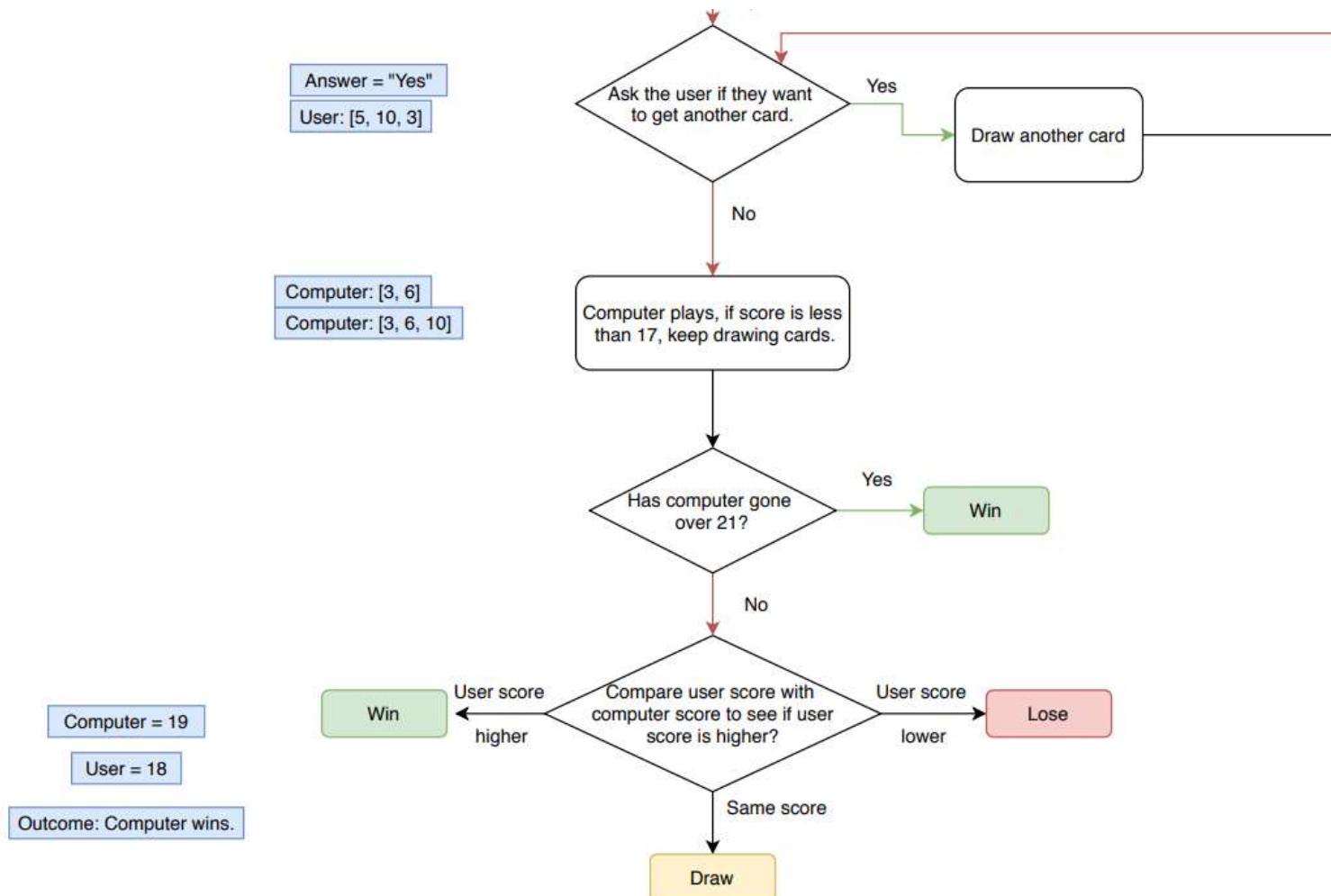
www.hbpatel.in





Projects – Black Jack

www.hbpatel.in





Projects – Black Jack

www.hbpatel.in

```
##### Blackjack Project #####
##### Our Blackjack House Rules #####
## The deck is unlimited in size.
## There are no jokers.
## The Jack/Queen/King all count as 10.
## The Ace can count as 11 or 1.
## Use the following list as the deck of cards:
## cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10]
## The cards in the list have equal probability of being drawn.
## Cards are not removed from the deck as they are drawn.

##### Hints #####
#Hint 1: Go to this website and try out the Blackjack game:
#   https://games.washingtonpost.com/games/blackjack/
#Then try out the completed Blackjack project here:
#   http://blackjack-final.appbrewery.repl.run

#Hint 2: Read this breakdown of program requirements:
#   http://listmoz.com/view/6h34DJpvJBFVRlZfJvxF
#Then try to create your own flowchart for the program.

#Hint 3: Download and read this flow chart I've created:
#   https://drive.google.com/uc?export=download&id=1rDkiHCrhaf9eX7u7yjM1qwSuyEk-rPnt

#Hint 4: Create a deal_card() function that uses the List below to *return* a random card.
#11 is the Ace.
```

```
import random

def deal_card():
    """Returns a random card from the deck."""
    cards = [11, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10]
    card = random.choice(cards)
    return card
```



Projects – Black Jack

www.hbpatel.in

```
#Hint 6: Create a function called calculate_score() that takes a List of cards as input
#and returns the score.
#Look up the sum() function to help you do this.
def calculate_score(cards):
    """Take a list of cards and return the score calculated from the cards"""

    #Hint 7: Inside calculate_score() check for a blackjack (a hand with only 2 cards: ace + 10) and return 0 instead of the actual
    score. 0 will represent a blackjack in our game.
    if sum(cards) == 21 and len(cards) == 2:
        return 0
    #Hint 8: Inside calculate_score() check for an 11 (ace). If the score is already over 21, remove the 11 and replace it with a 1.
    You might need to look up append() and remove().
    if 11 in cards and sum(cards) > 21:
        cards.remove(11)
        cards.append(1)
    return sum(cards)

#Hint 13: Create a function called compare() and pass in the user_score and computer_score. If the computer and user both have the
same score, then it's a draw. If the computer has a blackjack (0), then the user loses. If the user has a blackjack (0), then the
user wins. If the user_score is over 21, then the user loses. If the computer_score is over 21, then the computer loses. If none
of the above, then the player with the highest score wins.
def compare(user_score, computer_score):
    #Bug fix. If you and the computer are both over, you lose.
    if user_score > 21 and computer_score > 21:
        return "You went over. You lose 😤"
```



Projects – Black Jack

www.hbpatel.in

```
if user_score == computer_score:  
    return "Draw 😐"  
elif computer_score == 0:  
    return "Lose, opponent has Blackjack 😨"  
elif user_score == 0:  
    return "Win with a Blackjack 😍"  
elif user_score > 21:  
    return "You went over. You lose 😭"  
elif computer_score > 21:  
    return "Opponent went over. You win 😃"  
elif user_score > computer_score:  
    return "You win 😃"  
else:  
    return "You lose 😤"
```



Projects – Black Jack

www.hbpatel.in

```
def play_game():
    #Hint 5: Deal the user and computer 2 cards each using deal_card()
    user_cards = []
    computer_cards = []
    is_game_over = False

    for _ in range(2):
        user_cards.append(deal_card())
        computer_cards.append(deal_card())

    #Hint 11: The score will need to be rechecked with every new card drawn and the checks in Hint 9 need to be repeated until
    #the game ends.
    while not is_game_over:
        #Hint 9: Call calculate_score(). If the computer or the user has a blackjack (0) or if the user's score is over 21, then
        #the game ends.
        user_score = calculate_score(user_cards)
        computer_score = calculate_score(computer_cards)
        print(f"  Your cards: {user_cards}, current score: {user_score}")
        print(f"  Computer's first card: {computer_cards[0]}")

        if user_score == 0 or computer_score == 0 or user_score > 21:
            is_game_over = True
        else:
            #Hint 10: If the game has not ended, ask the user if they want to draw another card. If yes, then use the deal_card()
            #function to add another card to the user_cards List. If no, then the game has ended.
            user_should_deal = input("Type 'y' to get another card, type 'n' to pass: ")
            if user_should_deal == "y":
                user_cards.append(deal_card())
            else:
                is_game_over = True
```



Projects – Black Jack

www.hbpatel.in

#Hint 12: Once the user is done, it's time to let the computer play. The computer should keep drawing cards as long as it has a score less than 17.

```
while computer_score != 0 and computer_score < 17:  
    computer_cards.append(deal_card())  
    computer_score = calculate_score(computer_cards)  
  
print(f"  Your final hand: {user_cards}, final score: {user_score}")  
print(f"  Computer's final hand: {computer_cards}, final score: {computer_score}")  
print(compare(user_score, computer_score))
```

#Hint 14: Ask the user if they want to restart the game. If they answer yes, clear the console and start a new game of blackjack and show the logo from art.py.

```
while input("Do you want to play a game of Blackjack? Type 'y' or 'n': ") == "y":  
  
    play_game()
```



Projects – Black Jack

www.hbpatel.in

```
Do you want to play a game of Blackjack? Type 'y' or 'n': y
Your cards: [9, 8], current score: 17
Computer's first card: 7
Type 'y' to get another card, type 'n' to pass: y
Your cards: [9, 8, 3], current score: 20
Computer's first card: 7
Type 'y' to get another card, type 'n' to pass: y
Your cards: [9, 8, 3, 2], current score: 22
Computer's first card: 7
Your final hand: [9, 8, 3, 2], final score: 22
Computer's final hand: [7, 7, 3], final score: 17
You went over. You lose 😞

Do you want to play a game of Blackjack? Type 'y' or 'n': y
Your cards: [10, 9], current score: 19
Computer's first card: 2
Type 'y' to get another card, type 'n' to pass: y
Your cards: [10, 9, 4], current score: 23
Computer's first card: 2
Your final hand: [10, 9, 4], final score: 23
Computer's final hand: [2, 2, 2, 10, 9], final score: 25
You went over. You lose 😞

Do you want to play a game of Blackjack? Type 'y' or 'n': y
Your cards: [9, 5], current score: 14
Computer's first card: 10
Type 'y' to get another card, type 'n' to pass: y
Your cards: [9, 5, 6], current score: 20
Computer's first card: 10
Type 'y' to get another card, type 'n' to pass: n
Your final hand: [9, 5, 6], final score: 20
Computer's final hand: [10, 2, 10], final score: 22
Opponent went over. You win 😊
Do you want to play a game of Blackjack? Type 'y' or 'n': n
```

Module

converters.py

```
def lbs_to_kg(weight):
    return weight * 0.45

def kg_to_lbs(weight):
    return weight / 0.45
```

module.py

```
import converters
from converters import kg_to_lbs

print(converters.lbs_to_kg(100))
print(kg_to_lbs(140))
```

OUTPUT

```
45.0
311.1111111111111
```

converters.py

```
def find_max(number):
    max = number[0]
    for n in number:
        if n > max:
            max = n
    return max
```

module.py

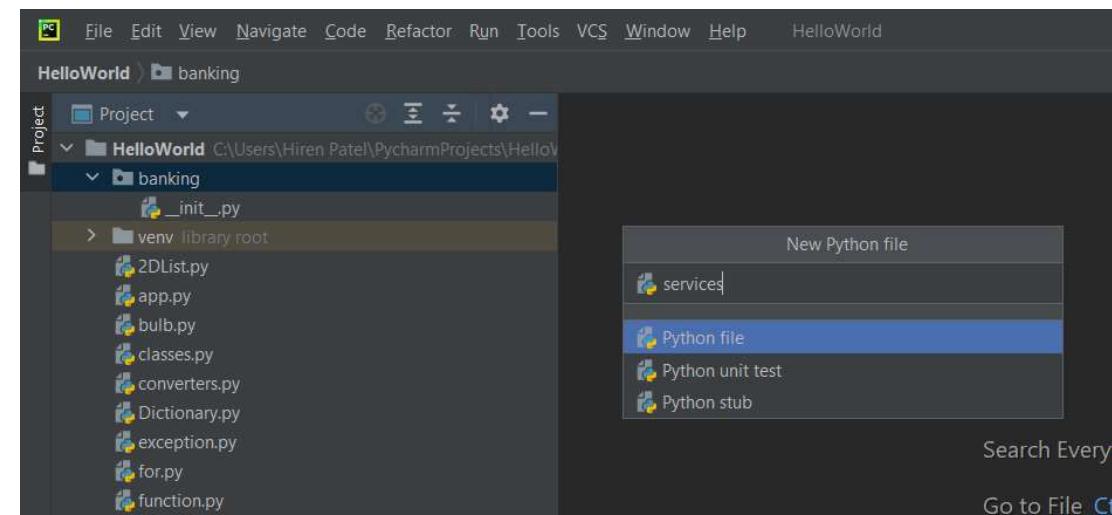
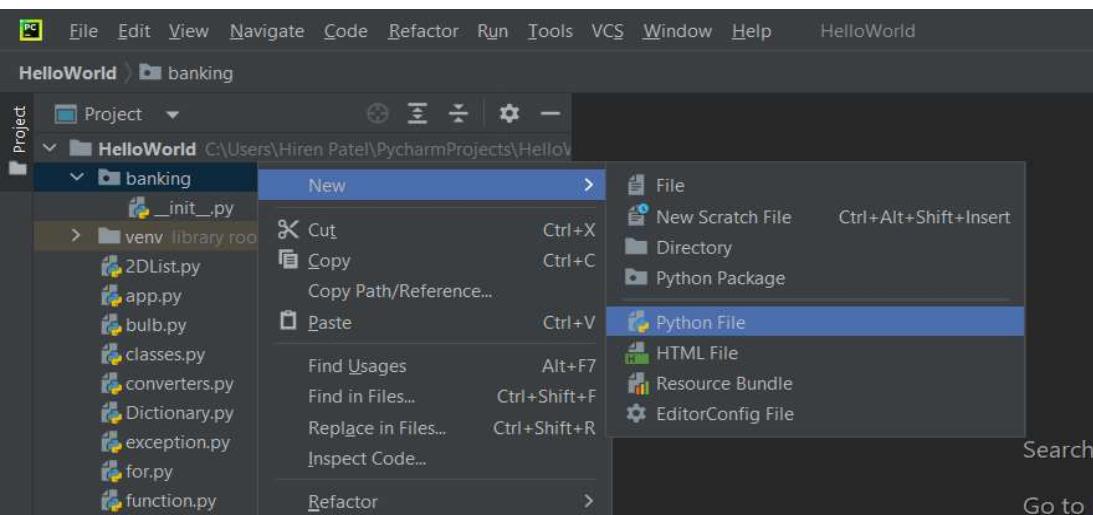
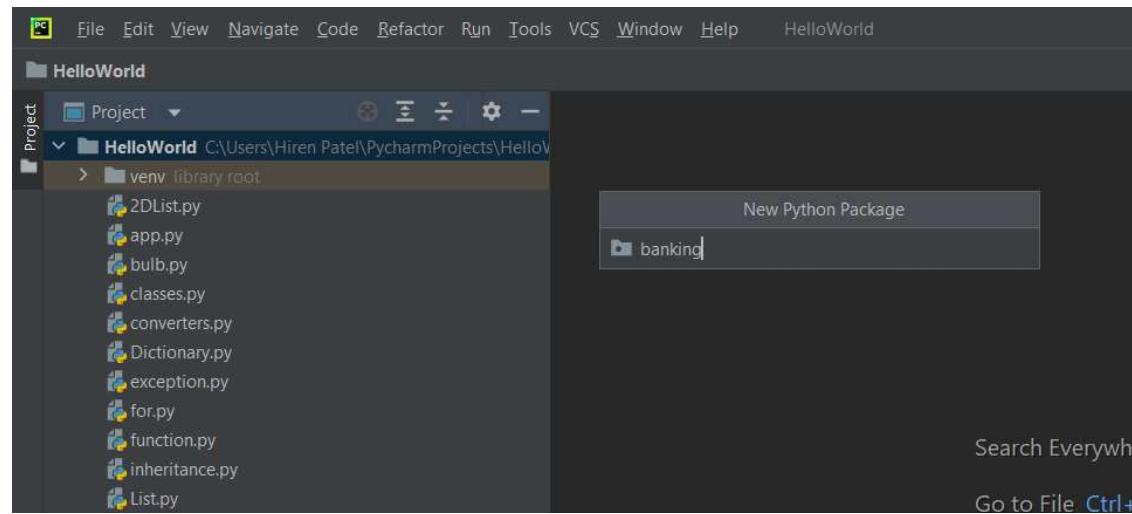
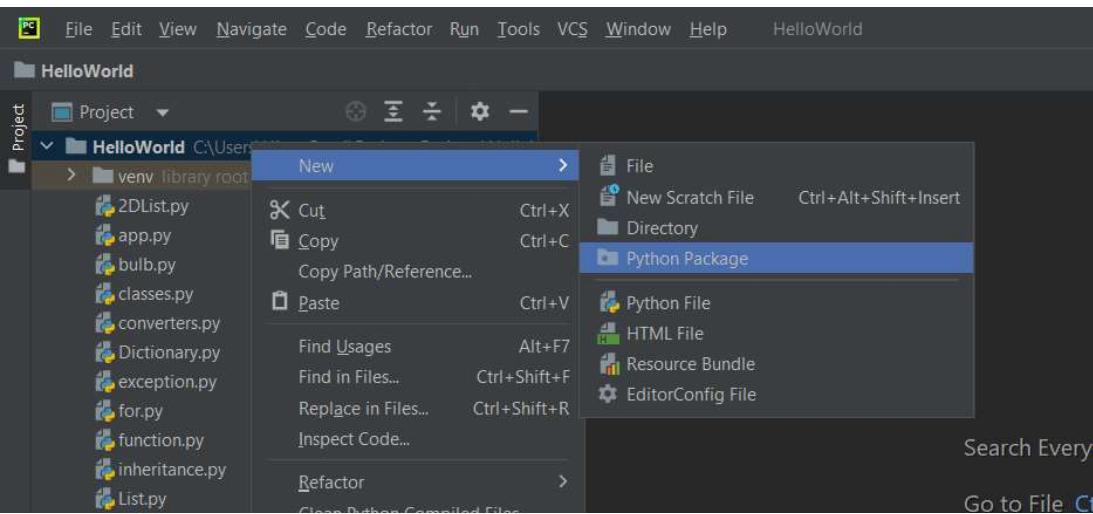
```
import converters

numbers = [11, -5, 39, 45, 12]
print(converters.find_max(numbers))
```

OUTPUT

```
45
```

Packages



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, and Tool. The title bar indicates the current file is "services.py" under the "banking" package of the "HelloWorld" project. The left sidebar displays the "Project" structure, which includes a "HelloWorld" folder containing a "banking" folder with files like __init__.py and services.py, and a "venv" folder with library.root and other Python files such as 2DList.py, app.py, bulb.py, classes.py, converters.py, Dictionary.py, exception.py, for.py, function.py, inheritance.py, List.py, ListOperation.py, main.py, module.py, NestedFor.py, packageBank.py, Tuples.py, and while.py. Below these are External Libraries and CMakeLists.txt.

services.py

```
def loan():
    print('This is the loan function under services module in banking package')

def deposit():
    print('This is the deposit function under services module in banking package')

def credit_card():
    print('This is the credit card function under services module in banking package')
```

packageBank.py

```
import banking.services
banking.services.loan()
```

This is the loan function under services module in banking package

```
from banking import services
services.loan()
services.deposit()
services.credit_card()
```

packageBank.py

```
from banking.services import deposit
deposit()
```

This is the deposit function under services module in banking package

packageBank.py

```
from banking.services import loan, deposit, credit_card
loan()
deposit()
credit_card()
```

This is the loan function under services module in banking package
This is the deposit function under services module in banking package
This is the credit card function under services module in banking package

```
def quicksort(arr):
    if len(arr) <= 1:
        return arr
    pivot = arr[len(arr) // 2]
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]
    return quicksort(left) + middle + quicksort(right)

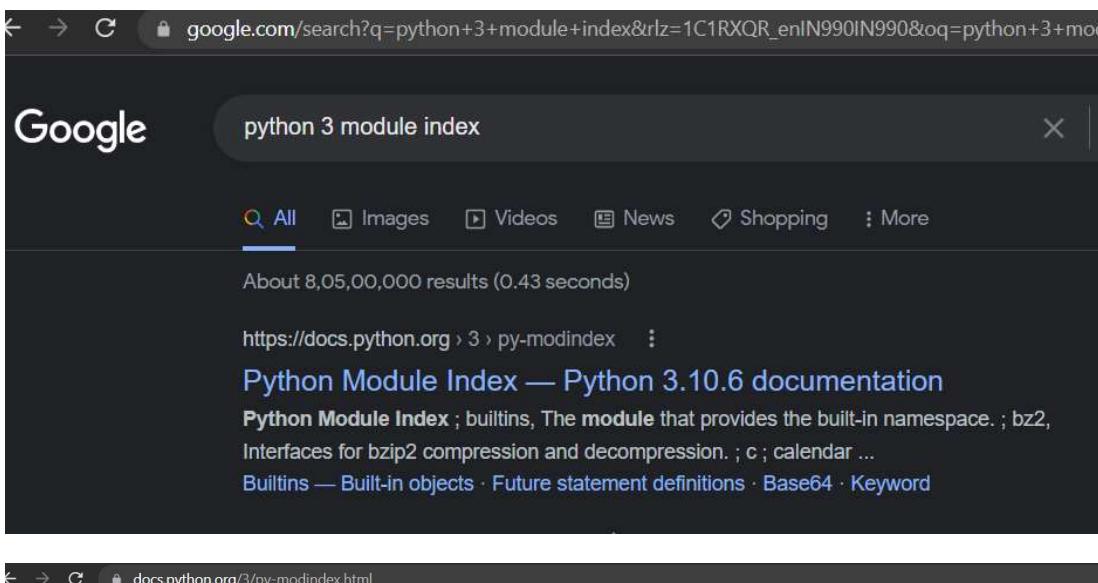
print(quicksort([3,6,8,10,1,2,1]))
```

[1, 1, 2, 3, 6, 8, 10]

```
s = "hello"
print(s.capitalize()) # Capitalize a string; prints "Hello"
print(s.upper()) # Convert a string to uppercase; prints "HELLO"
print(s.rjust(7)) # Right-justify a string, padding with spaces; prints " hello"
print(s.center(7)) # Center a string, padding with spaces; prints " hello "
print(s.replace('l', '(ell)')) # Replace all instances of one substring with another;
                                # prints "he(ell)(ell)o"
print(' world '.strip()) # Strip leading and trailing whitespace; prints "world"

Hello
HELLO
    hello
hello
he(ell)(ell)o
world
```

Random Numbers



google.com/search?q=python+3+module+index&rlz=1C1RXQR_enIN990IN990&oq=python+3+mod

Google

python 3 module index

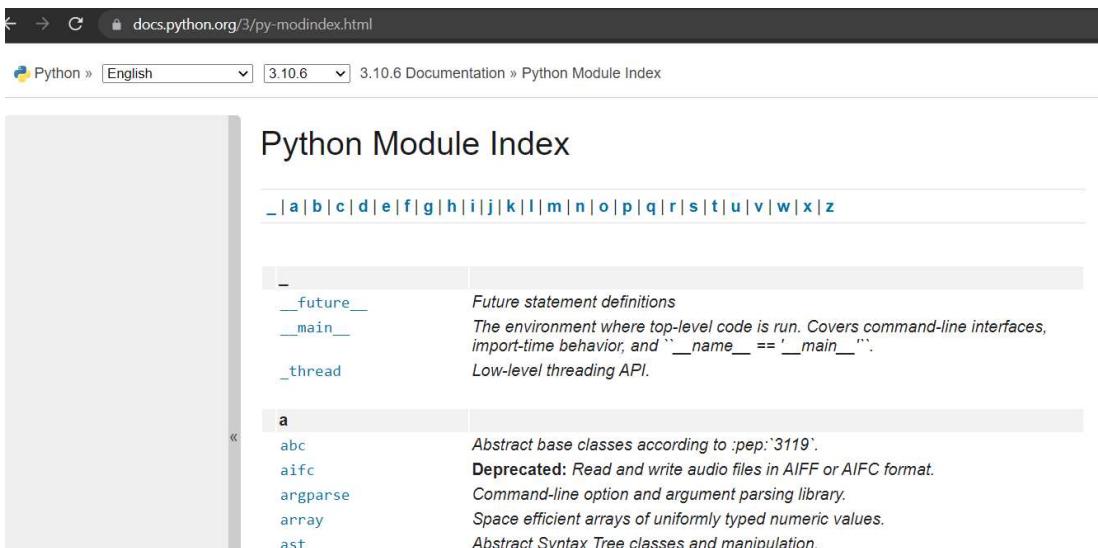
All Images Videos News Shopping More

About 8,05,00,000 results (0.43 seconds)

<https://docs.python.org/3/py-modindex.html> ::

Python Module Index — Python 3.10.6 documentation

Python Module Index ; builtins, The module that provides the built-in namespace. ; bz2, Interfaces for bzip2 compression and decompression. ; c ; calendar ...
Builtins — Built-in objects · Future statement definitions · Base64 · Keyword



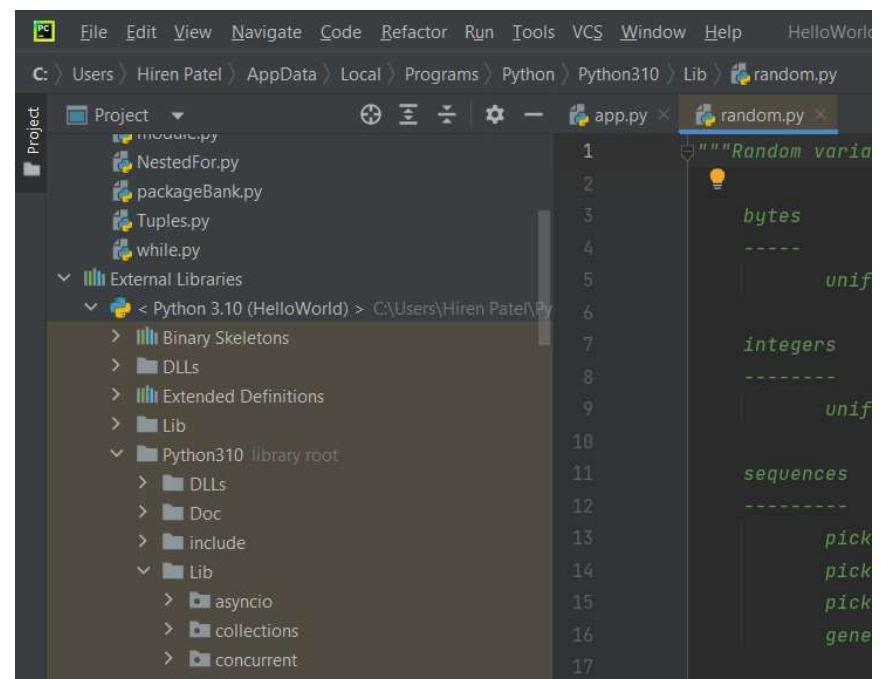
docs.python.org/3/py-modindex.html

Python » English 3.10.6 3.10.6 Documentation » Python Module Index

Python Module Index

_ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | z

_	_future_	Future statement definitions
	main	The environment where top-level code is run. Covers command-line interfaces, import-time behavior, and ``__name__ == '__main__'``.
	_thread	Low-level threading API.
a	abc	Abstract base classes according to :pep:`3119`.
	aifc	Deprecated: Read and write audio files in AIFF or AIFFC format.
	argparse	Command-line option and argument parsing library.
	array	Space efficient arrays of uniformly typed numeric values.
	ast	Abstract Syntax Tree classes and manipulation.



C:\Users\Hiren Patel\AppData\Local\Programs\Python\Python310\Lib\random.py

File Edit View Navigate Code Refactor Run Tools VCS Window Help HelloWorld

Project app.py random.py

```
"""Random variate generation functions.

bytes
-----
unif
integers
-----
sequences
-----
pick
pick
pick
gene
```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

random.py

Module contents

NestedFor.py

packageBank.py

Tuples.py

while.py

External Libraries

< Python 3.10 (HelloWorld) > C:\Users\Hiren Patel\Py

Binary Skeletons

DLLs

Extended Definitions

Lib

Python310\library root

DLLs

Doc

include

Lib

asyncio

collections

concurrent

Random Numbers

```
import random
for i in range(3):
    print(random.random())

for i in range(3):
    print(random.randint(10, 20))

team_members = ['Hiren', 'Sanjay', 'Pradip', 'Vijay', 'Parimal']
leader = random.choice(team_members)
print(leader)
```

```
0.9007723077084383
0.07188337323429828
0.7289552940868703
20
13
17
Sanjay
```

```
import random

class Dice:
    def roll(self):
        first = random.randint(1, 6)
        second = random.randint(1, 6)
        return first, second

dice = Dice()
print(dice.roll())
```

```
(5, 4)
```

Files and Directories

```
from pathlib import Path

# Absolute Path
# C:\Users\Hiren Patel\PycharmProjects\HelloWorld
# Relative Path

# Check whether the directory exists or not
path = Path("banking")
print(path.exists())

# Creating a new directory in existing path
path = Path("Emails")
print(path.mkdir())
```

```
from pathlib import Path

# Removing existing director
path = Path("Emails")
print(path.rmdir())
```

```
from pathlib import Path

path = Path()
print(path.glob('*.*py'))

for file in path.glob('*.*py'):
    print(file)
```

```
<generator object Path.glob at 0x0000021428169930>
2DList.py
app.py
bulb.py
classes.py
converters.py
Dictionary.py
exception.py
for.py
function.py
inheritance.py
List.py
ListOperation.py
main.py
module.py
NestedFor.py
packageBank.py
Tuples.py
while.py
```

Reading CSV Files

```
student.csv
1 11,"Hiren",9.8
2 22,"Pradip",7.7
3 33,"Sanjay",8.2
4 44,"Vijay",8.0
5 55,"Parimal",7.9

readCSV.py
import csv
with open("student.csv") as dataFile:
    data = csv.reader(dataFile)
    for row in data:
        print(row)
```

```
readCSV
"C:\Users\Hiren Patel\PycharmProjects\edureka\venv\Scripts\python.exe" "C:/Users/Hiren Patel/PycharmProjects/edureka/venv/readCSV.py"
['11', 'Hiren', '9.8']
['22', 'Pradip', '7.7']
['33', 'Sanjay', '8.2']
['44', 'Vijay', '8.0']
['55', 'Parimal', '7.9']

Process finished with exit code 0
```

Control Run Python Packages TODO Python Console Problems Terminal Services

Reading CSV Files

The screenshot shows a PyCharm interface with two files open: 'student.csv' and 'readCSV.py'. The 'readCSV.py' file contains Python code that reads a CSV file and prints the name and grade for each student based on their CGPA. The output window shows the results for five students: Hiren, Pradip, Sanjay, Vijay, and Parimal.

```
student.csv × readCSV.py ×
1 import csv
2 grade = []
3 counter = 0
4 with open("student.csv") as dataFile:
5     data = csv.reader(dataFile)
6     for row in data:
7         cpi = float(row[2])
8         if(cpi>9):
9             grade.append("Excellent")
10        elif(cpi>8):
11            grade.append("Very Good")
12        elif(cpi>7):
13            grade.append("Good")
14        elif(cpi>6):
15            grade.append("Average")
16        else:
17            grade.append("Poor")
18        print(f"{row[1]} has '{grade[counter]}' grade")
19        counter += 1
```

```
C:\Users\Hiren Patel\PycharmProjects\edur
Hiren has 'Excellent' grade
Pradip has 'Good' grade
Sanjay has 'Very Good' grade
Vijay has 'Good' grade
Parimal has 'Good' grade
```

Reading CSV Files using pandas library

```
import pandas  
data = pandas.read_csv("student.csv")  
print(data)
```

```
   ID    Name  Marks  
0  11    Hiren  9.8  
0  22    Pradip  7.7  
1  33    Sanjay  8.2  
2  44    Vijay  8.0  
3  55    Parimal  7.9
```

Processing CSV Files using pandas library

```
student.csv x  readCSV.py x
1 rollno, name, cpi
2 11, "Hiren", 9.8
3 22, "Pradip", 7.7
4 33, "Sanjay", 8.2
5 44, "Vijay", 8.0
6 55, "Parimal", 7.9
```

```
rollno      name    cpi
0       11     Hiren   9.8
1       22    Pradip   7.7
2       33    Sanjay   8.2
3       44     Vijay   8.0
4       55   Parimal   7.9
{'rollno': {0: 11, 1: 22, 2: 33, 3: 44, 4: 55}, 'name': {0: 'Hiren', 1: 'Pradip', 2: 'Sanjay', 3: 'Vijay', 4: 'Parimal'}, 'cpi': {0: 9.8, 1: 7.7, 2: 8.2, 3: 8.0, 4: 7.9}}
[9.8, 7.7, 8.2, 8.0, 7.9]
8.32
```

```
import pandas
data = pandas.read_csv("student.csv")
print(data)

#converting data file into dictionary
dataDictionary = data.to_dict()
print(dataDictionary)

#converting data file into list
dataList = data["cpi"].to_list()
print(dataList)

#Processing the data in the data file
print(data["cpi"].mean())
```

Creating CSV Files using pandas library

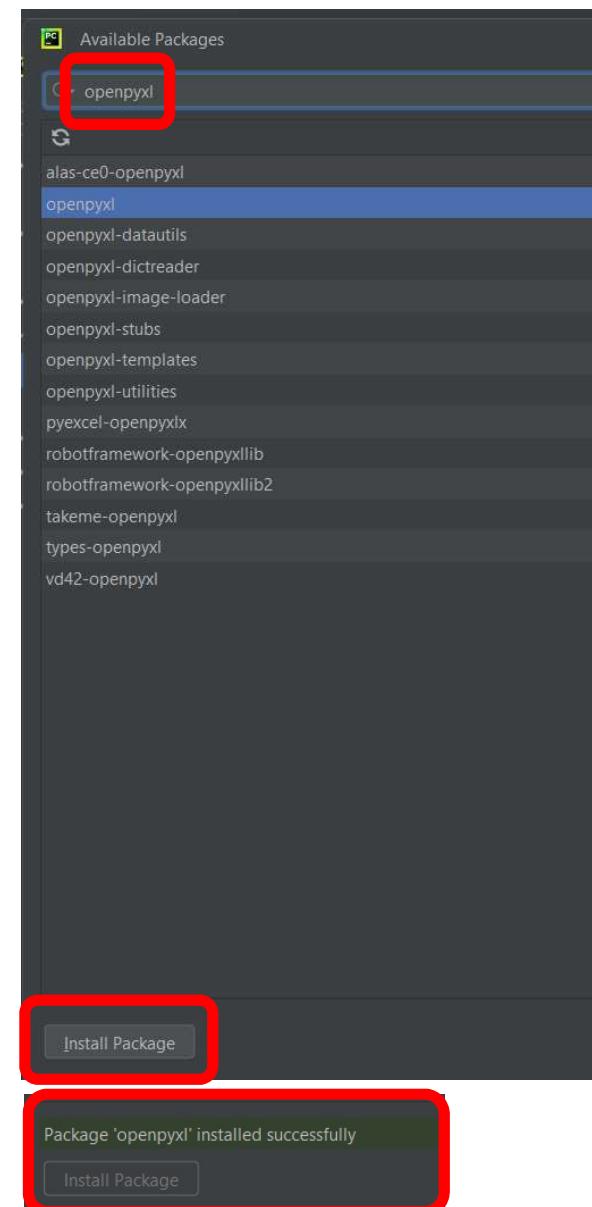
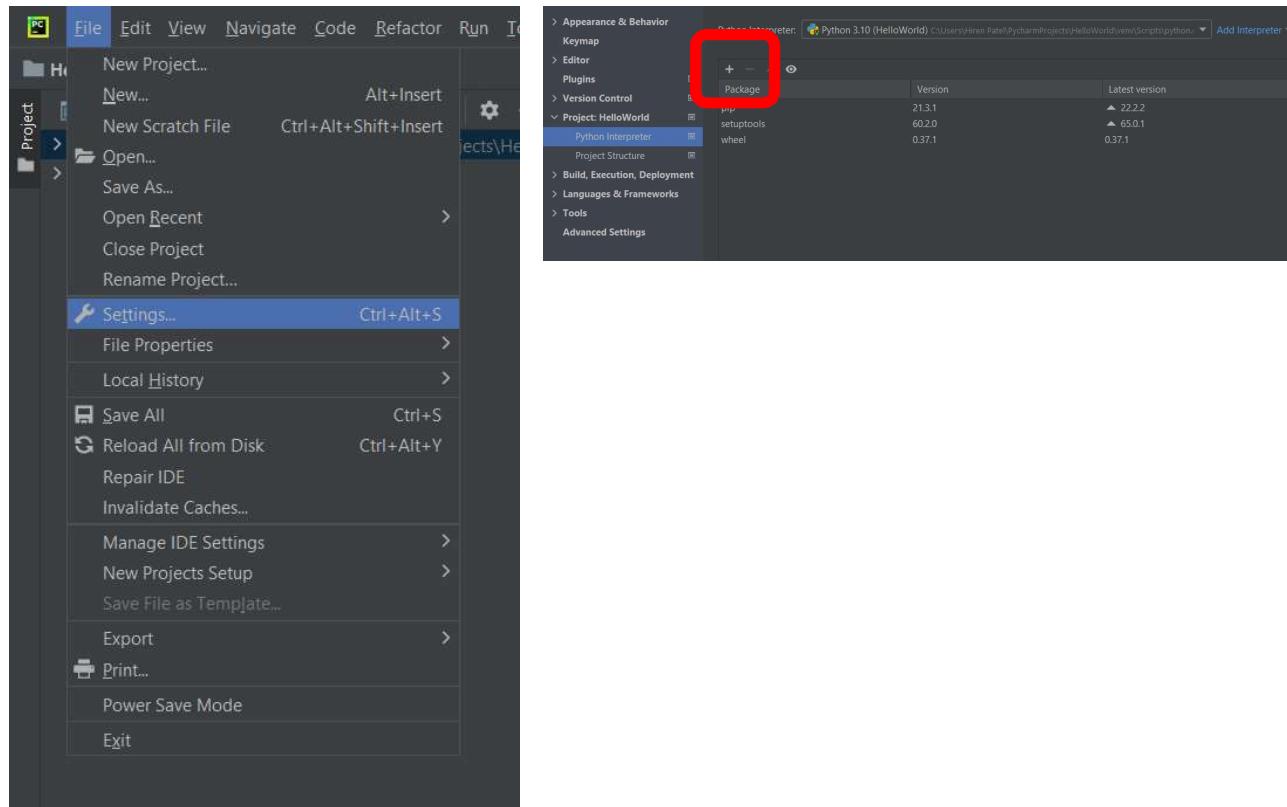
```
dict2dataFrame.py newFile.csv
1 import pandas
2 dataDictionary = {
3     "rollno": [11, 22, 33, 44, 55],
4     "name": ["Hiren", "Pradip", "Sanjay", "Vijay", "Parimal"],
5     "cpi": [9.8, 7.7, 8.2, 8.0, 7.0]
6 }
7 df = pandas.DataFrame(dataDictionary)
8 df.to_csv("newFile.csv")
9
```

```
dict2dataFrame.py newFile.csv
1 ,rollno,name,cpi
2 0,11,Hiren,9.8
3 1,22,Pradip,7.7
4 2,33,Sanjay,8.2
5 3,44,Vijay,8.0
6 4,55,Parimal,7.0
7
```

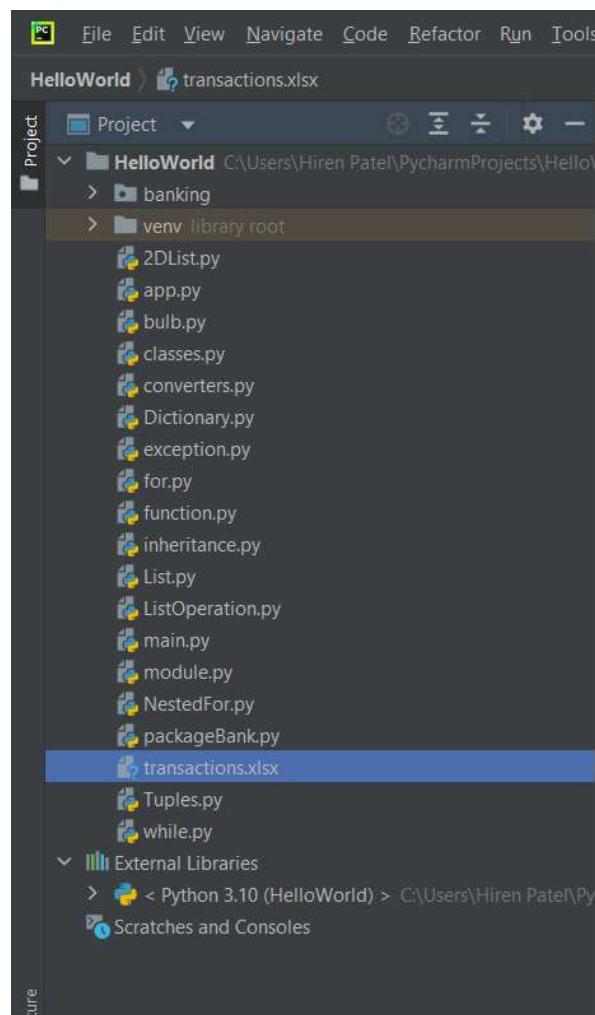
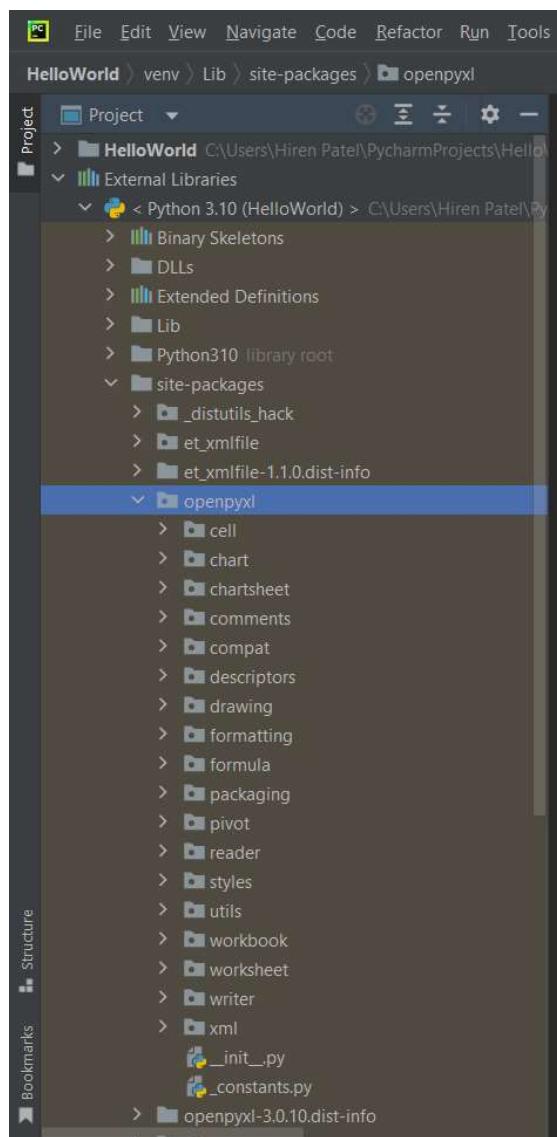
PyPI and openpyxl installation

```
Command Prompt - python
Microsoft Windows [Version 10.0.22000.708]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hiren Patel>python
Python 3.10.6 (tags/v3.10.6:9c7b4bd, Aug 1 2022, 21:53:49) [MSC v.1932 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> pip install openpyxl
```



Excel Spreadsheets



	A	B	C
1	transaction_id	product_id	price
2	1001	1	\$5.95
3	1002	2	\$6.95
4	1003	3	\$7.95
5			
6			

Download the file <https://github.com/RagingLeviathan/HelloWorld-mosh-python/raw/master/transactions.xlsx> and copy it in C:\Users\Hiren Patel\PycharmProjects\HelloWorld

```

import openpyxl as xl
wb =
xl.load_workbook('transactions.xlsx')
sheet = wb['Sheet1']
# First Way
cell = sheet['a1']
# Another Way
cell = sheet.cell(1, 1)
print(cell.value)

```

transaction_id

```

import openpyxl as xl
wb = xl.load_workbook('transactions.xlsx')
sheet = wb['Sheet1']
# First Way
cell = sheet['a1']
# Another Way
cell = sheet.cell(1, 1)
print(sheet.max_row)
for row in range(1, sheet.max_row+1):
    print(row)

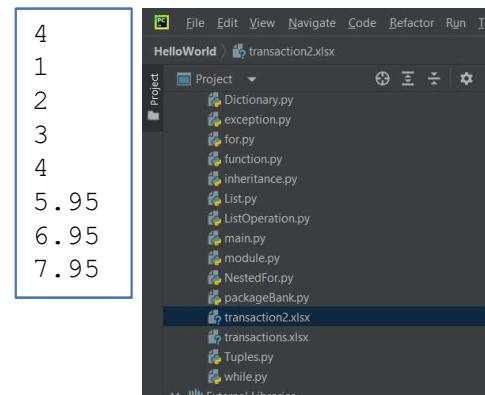
for row in range(2, sheet.max_row+1):
    cell = sheet.cell(row, 3)
    print(cell.value)

for row in range(2, sheet.max_row+1):
    cell = sheet.cell(row, 3)
    corrected_price = cell.value * 0.9
    corrected_price_cell = sheet.cell(row, 4)
    corrected_price_cell.value = corrected_price

wb.save('transaction2.xlsx')

```

	A	B	C
1	transaction_id	product_id	price
2		1001	1 \$5.95
3		1002	2 \$6.95
4		1003	3 \$7.95
5			
6			



	A	B	C	D
1	transaction_id	product_id	price	
2		1001	1 \$5.95	5.355
3		1002	2 \$6.95	6.255
4		1003	3 \$7.95	7.155

```

import openpyxl as xl
from openpyxl.chart import BarChart, Reference

wb = xl.load_workbook('transactions.xlsx')
sheet = wb['Sheet1']
cell = sheet['a1']

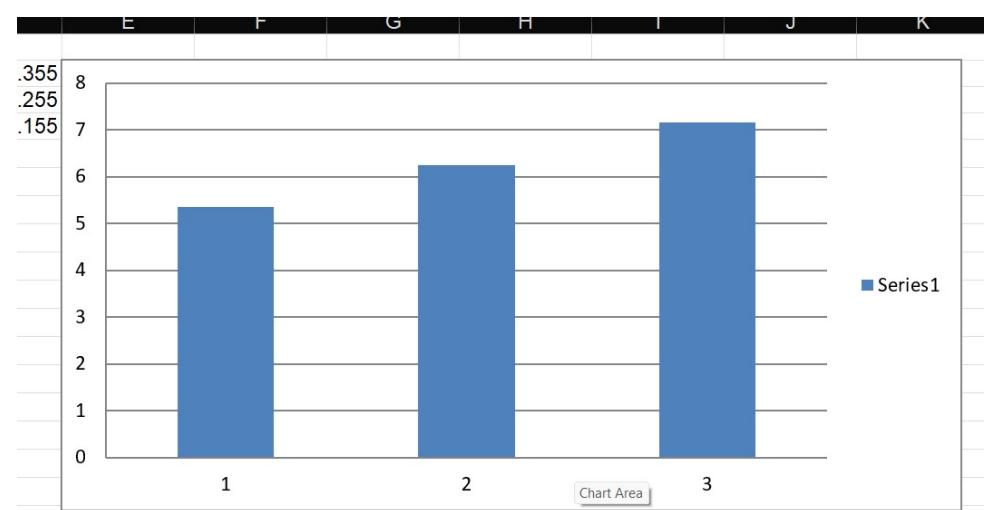
for row in range(2, sheet.max_row+1):
    cell = sheet.cell(row, 3)
    corrected_price = cell.value * 0.9
    corrected_price_cell = sheet.cell(row, 4)
    corrected_price_cell.value = corrected_price

values = Reference(sheet,
                   min_row=2,
                   max_row=sheet.max_row,
                   min_col=4,
                   max_col=4)
chart = BarChart()
chart.add_data(values)
sheet.add_chart(chart, 'e2')

wb.save('transaction2.xlsx')

```

	A	B	C	D	E
1	transaction_id	product_id	price		
2	1001	1	\$5.95	5.355	8
3	1002	2	\$6.95	6.255	7
4	1003	3	\$7.95	7.155	6
5					-
6					
7					



```

from tkinter import *

#Creating a new window and configurations
window = Tk()
window.title("Widget Examples")
window.minsize(width=500, height=500)

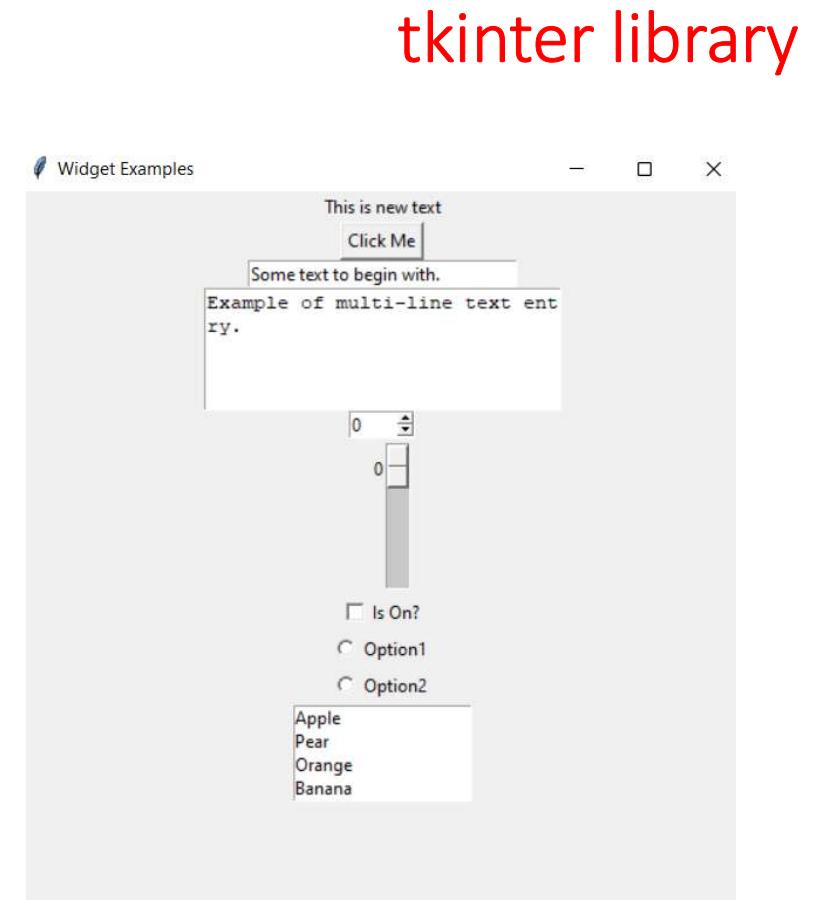
#Labels
label = Label(text="This is old text")
label.config(text="This is new text")
label.pack()

#Buttons
def action():
    print("Do something")

#calls action() when pressed
button = Button(text="Click Me", command=action)
button.pack()

#Entries
entry = Entry(width=30)
#Add some text to begin with
entry.insert(END, string="Some text to begin with.")
#Gets text in entry
print(entry.get())
entry.pack()

```



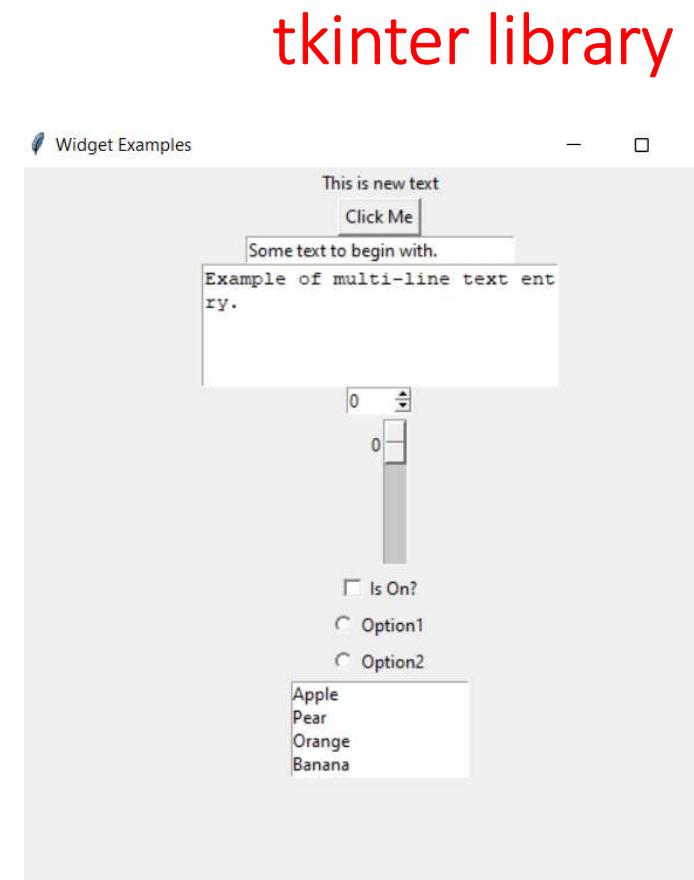
```

#Text
text = Text(height=5, width=30)
#Puts cursor in textbox.
text.focus()
#Adds some text to begin with.
text.insert(END, "Example of multi-line text entry.")
#Get's current value in textbox at line 1, character 0
print(text.get("1.0", END))
text.pack()

#Spinbox
def spinbox_used():
    #gets the current value in spinbox.
    print(spinbox.get())
spinbox = Spinbox(from_=0, to=10, width=5, command=spinbox_used)
spinbox.pack()

#Scale
#Called with current scale value.
def scale_used(value):
    print(value)
scale = Scale(from_=0, to=100, command=scale_used)
scale.pack()

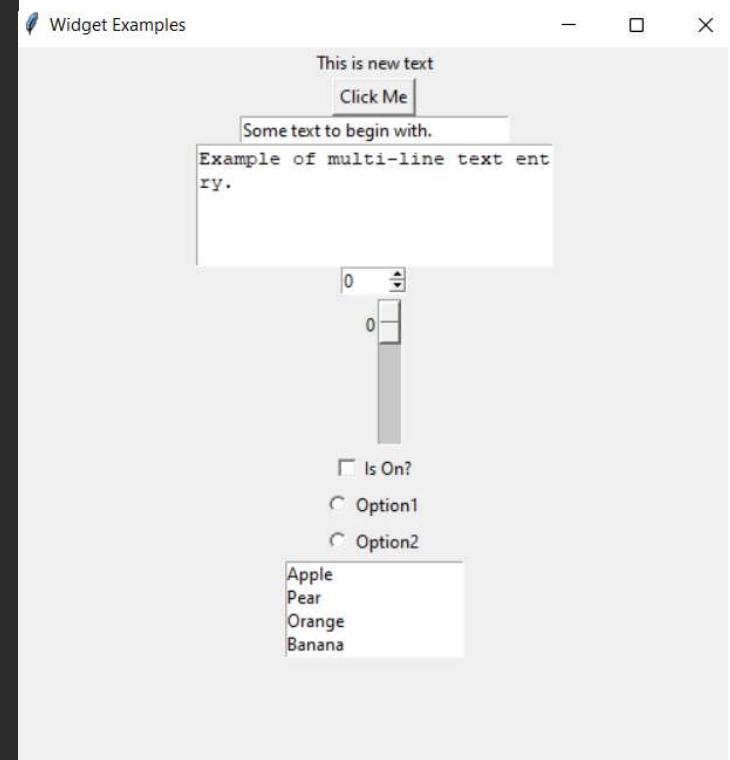
```



tkinter library

```
#Checkbutton
def checkbutton_used():
    #Prints 1 if On button checked, otherwise 0.
    print(checked_state.get())
#variable to hold on to checked state, 0 is off, 1 is on.
checked_state = IntVar()
checkbutton = Checkbutton(text="Is On?",  
variable=checked_state, command=checkbutton_used)
checked_state.get()
checkbutton.pack()

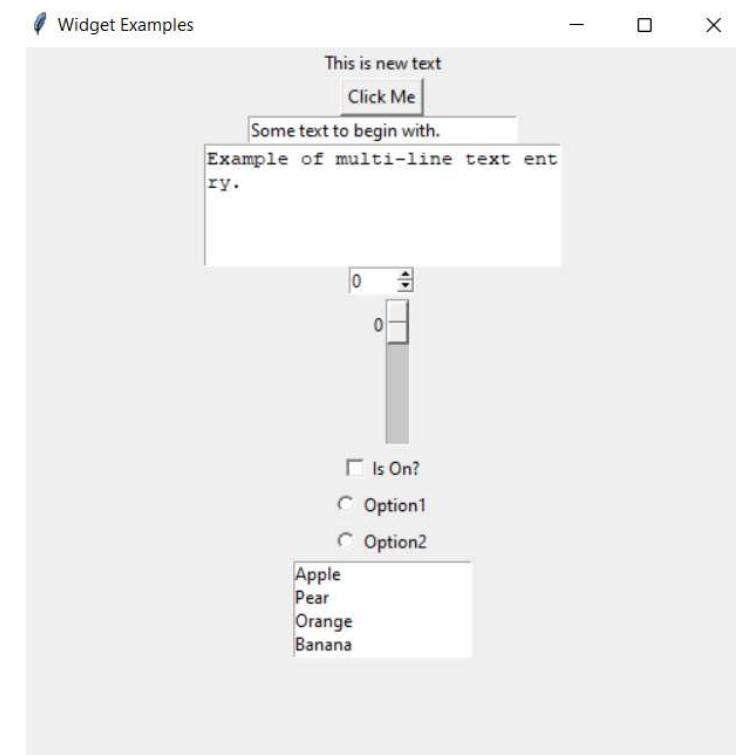
#Radiobutton
def radio_used():
    print(radio_state.get())
#Variable to hold on to which radio button value is checked.
radio_state = IntVar()
radiobutton1 = Radiobutton(text="Option1", value=1,  
variable=radio_state, command=radio_used)
radiobutton2 = Radiobutton(text="Option2", value=2,  
variable=radio_state, command=radio_used)
radiobutton1.pack()
radiobutton2.pack()
```



tkinter library

```
#Listbox
def listbox_used(event):
    # Gets current selection from listbox
    print(listbox.get(listbox.curselection()))

listbox = Listbox(height=4)
fruits = ["Apple", "Pear", "Orange", "Banana"]
for item in fruits:
    listbox.insert(fruits.index(item), item)
listbox.bind("<<ListboxSelect>>", listbox_used)
listbox.pack()
window.mainloop()
```



Python for Machine Learning

Applications of Machine Learning

Self-driving Cars

Robotics

Language Processing

Vision Processing

Forecasting Stock Market Trends

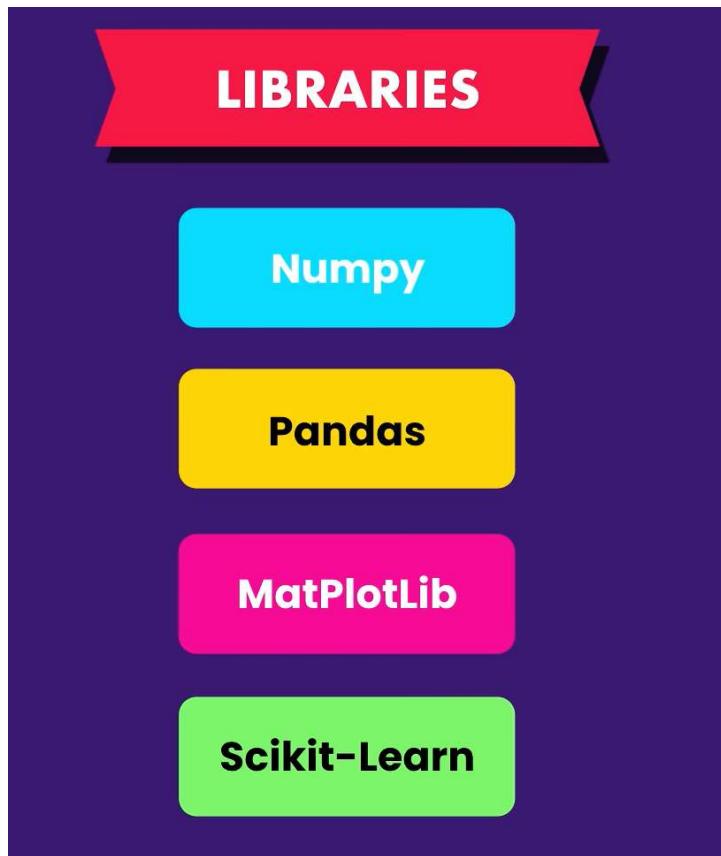


Steps of Machine Learning Project

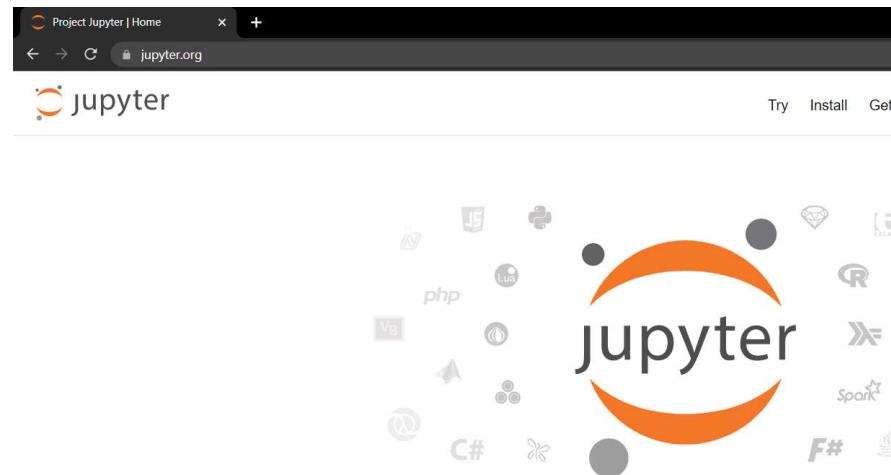
Steps

1. Import the Data
2. Clean the Data
3. Split the Data into Training/Test Sets
4. Create a Model
5. Train the Model
6. Make Predictions
7. Evaluate and Improve

Libraries and Tools for Machine Learning



Jupyter Environment for Writing ML Project



To install Jupyter, we use the platform of Anaconda



Data science technology for groundbreaking research.

Project Jupyter | Home x Anaconda | The World's Most Po| x +

← → C anaconda.com

ANACONDA. Products ▾ Pricing Solutions ▾ Resources ▾ Partners ▾ Blog Company ▾ Contact Sales

Data science technology for a better world.

Anaconda offers the easiest way to perform Python/R data science and machine learning on a single machine. Start working with thousands of open-source packages and libraries today.

Download 

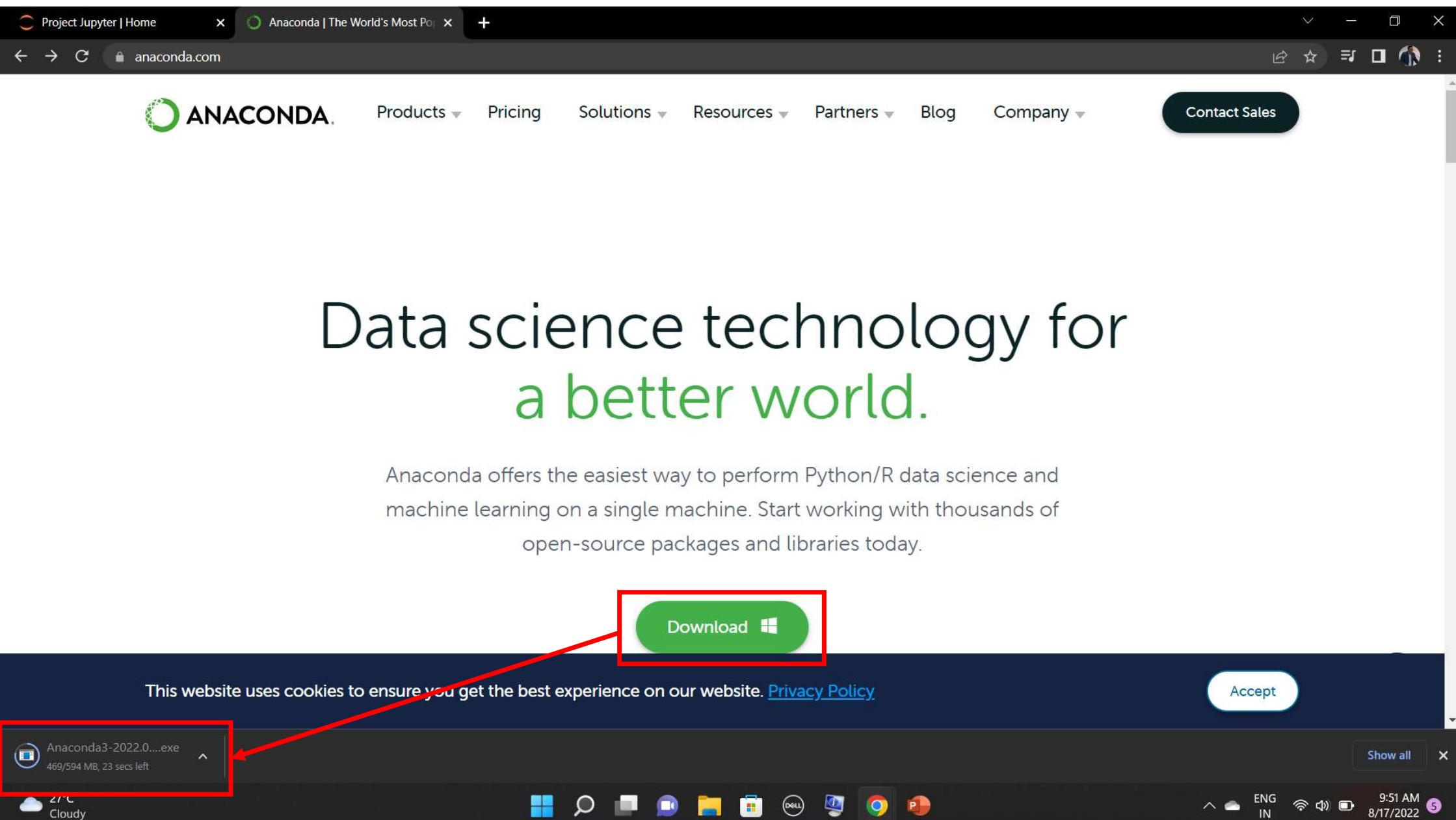
This website uses cookies to ensure you get the best experience on our website. [Privacy Policy](#)

Accept

Anaconda3-2022.0....exe
469/594 MB, 23 secs left

Cloudy 27°C Show all X

9:51 AM ENG IN 8/17/2022 5



Welcome to Anaconda3 2022.05 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2022.05 (64-bit).

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.

License Agreement

ANA CONDA

Please review the license terms before installing Anaconda3 2022.05 (64-bit).

Press Page Down to see the rest of the agreement.

```
=====
End User License Agreement - Anaconda Distribution
=====
```

Copyright 2015-2022, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

This End User License Agreement (the "Agreement") is a legal agreement between you and Anaconda, Inc. ("Anaconda") and governs your use of Anaconda Distribution (which was formerly known as Anaconda Individual Edition).

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 2022.05 (64-bit).

Anaconda, Inc. —

< Back I Agree Cancel

Select Installation Type

ANA CONDA

Please select the type of installation you would like to perform for Anaconda3 2022.05 (64-bit).

Install for:

Just Me (recommended)

All Users (requires admin privileges)

Anaconda, Inc. —

< Back Next > Cancel

Choose Install Location

ANA CONDA

Choose the folder in which to install Anaconda3 2022.05 (64-bit).

Setup will install Anaconda3 2022.05 (64-bit) in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder
C:\Users\Hiren Patel\anaconda3 Browse...

Space required: 3.5GB
Space available: 51.1GB

Anaconda, Inc. —

< Back Next > Cancel

Advanced Installation Options

ANA CONDA

Customize how Anaconda integrates with Windows

Advanced Options

Add Anaconda3 to my PATH environment variable

Not recommended. Instead, open Anaconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

Register Anaconda3 as my default Python 3.9

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.9 on the system.

Anaconda, Inc. —

< Back Install Cancel

Installing

ANA CONDA

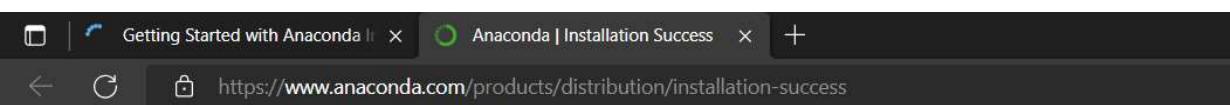
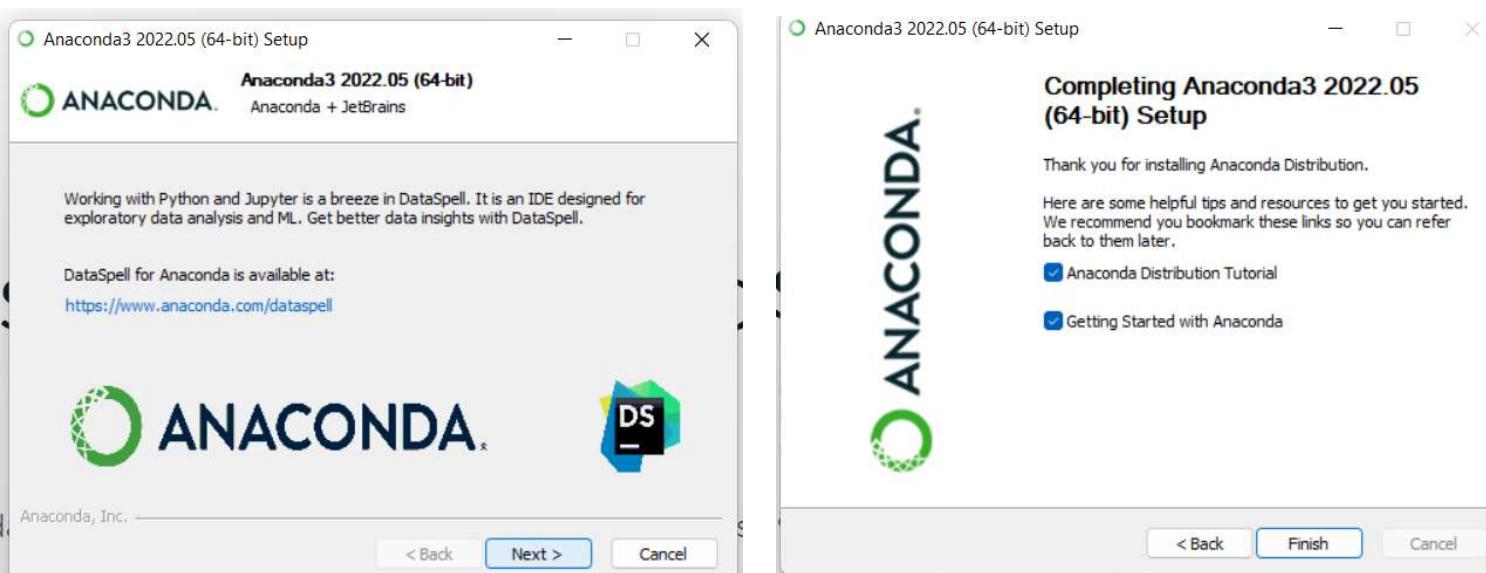
Please wait while Anaconda3 2022.05 (64-bit) is being installed.

Extract: pywin32-ctypes-0.2.0-py39haa95532_1000.conda

Show details

Anaconda, Inc. —

< Back Next > Cancel



Installation Success

Welcome to
Anaconda!

Visual Studio Code - Code Editing

https://code.visualstudio.com

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn Search Docs Download

Version 1.70 is now available! Read about the new features and fixes from July.

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

Download for Windows
Stable Build

The screenshot shows the Visual Studio Code interface. The main area displays a code editor with three files: App.js, index.js, and serviceWorker.js. The serviceWorker.js file contains JavaScript code related to service workers. To the left of the editor is a sidebar with icons for file operations like copy, paste, and search. Below the sidebar is a list of extensions available in the Marketplace, including Python, GitLens, C/C++, and ESLint. The status bar at the bottom shows the path 'serviceWorker.js - create-react-app - Visual Studio Code - In...'.

Documentation for Visual Studio

https://code.visualstudio.com/docs/?dv=win

Visual Studio Code Docs Updates Blog API Extensions FAQ Learn Search Download

Version 1.70 is now available! Read about the new features and fixes from July.

Thanks for downloading VS Code for Windows!

Download not starting? Try this [direct download link](#).
Please take a few seconds and help us improve ... [click to take survey](#).

Overview

SETUP

GET STARTED

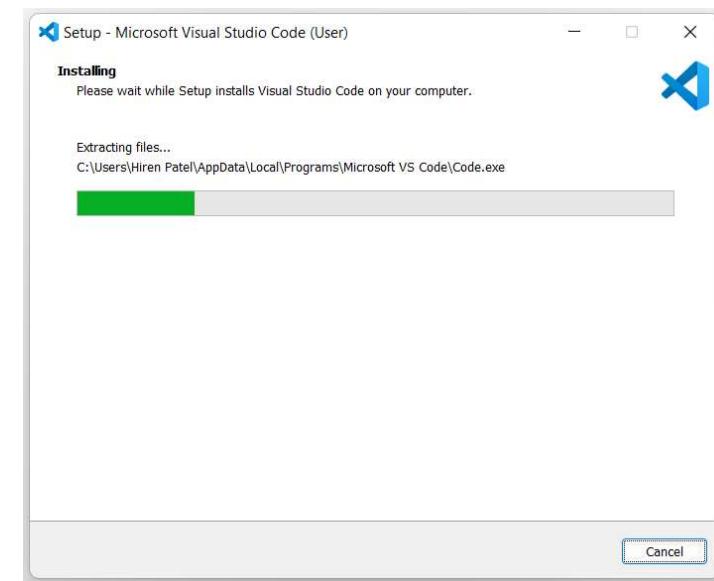
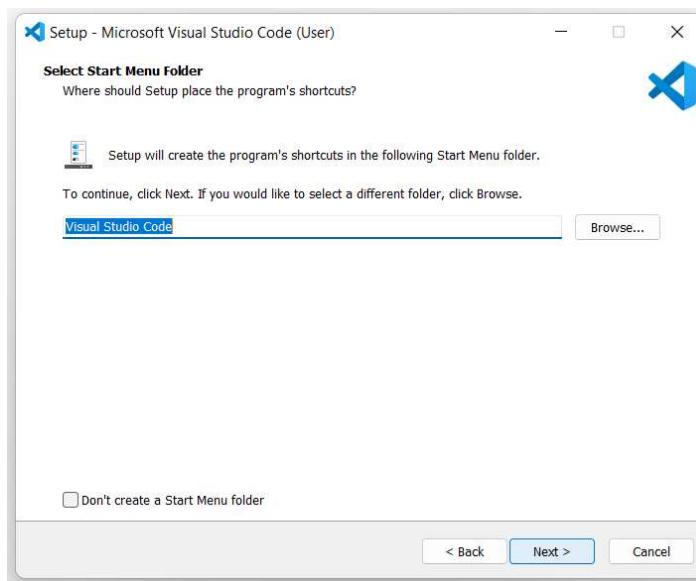
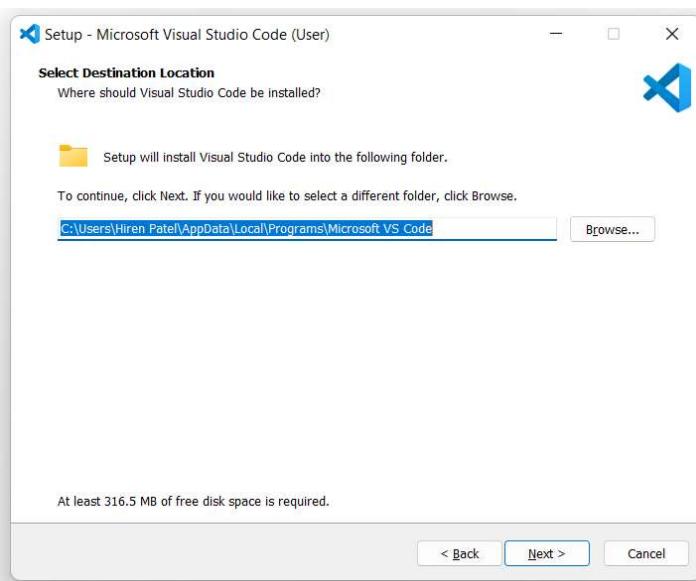
USER GUIDE

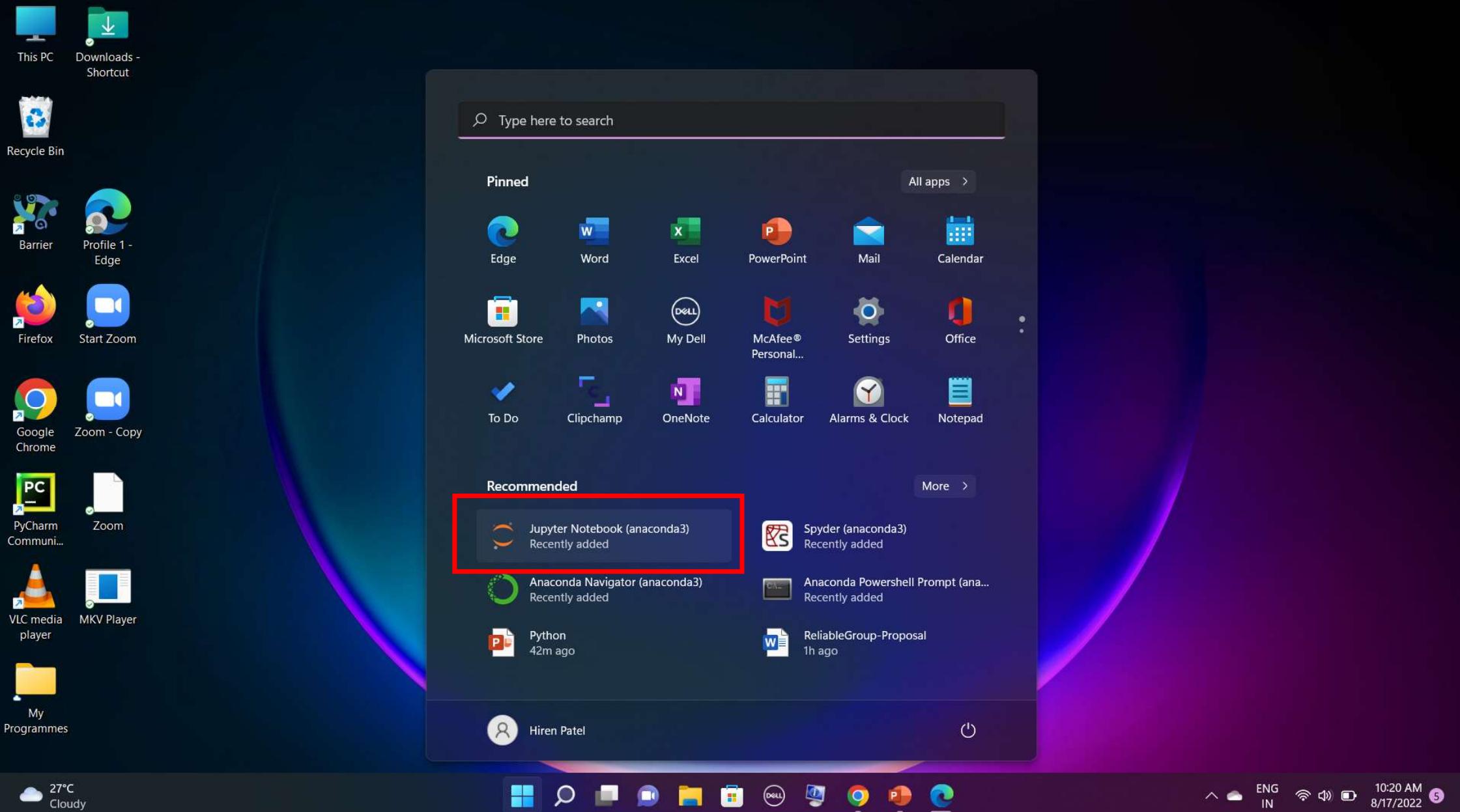
GETTING STARTED

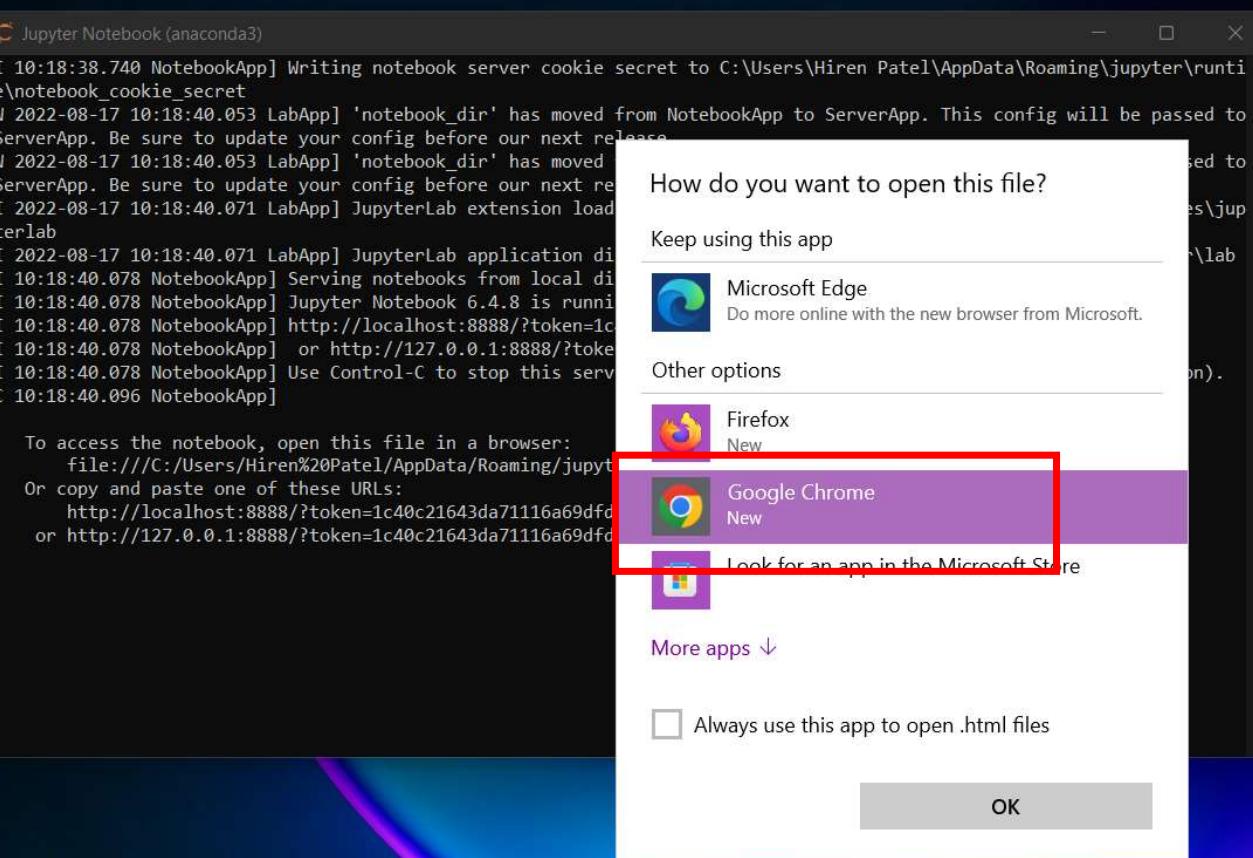
VS Code in Action
Top Extensions
First Steps
Keyboard Shortcuts

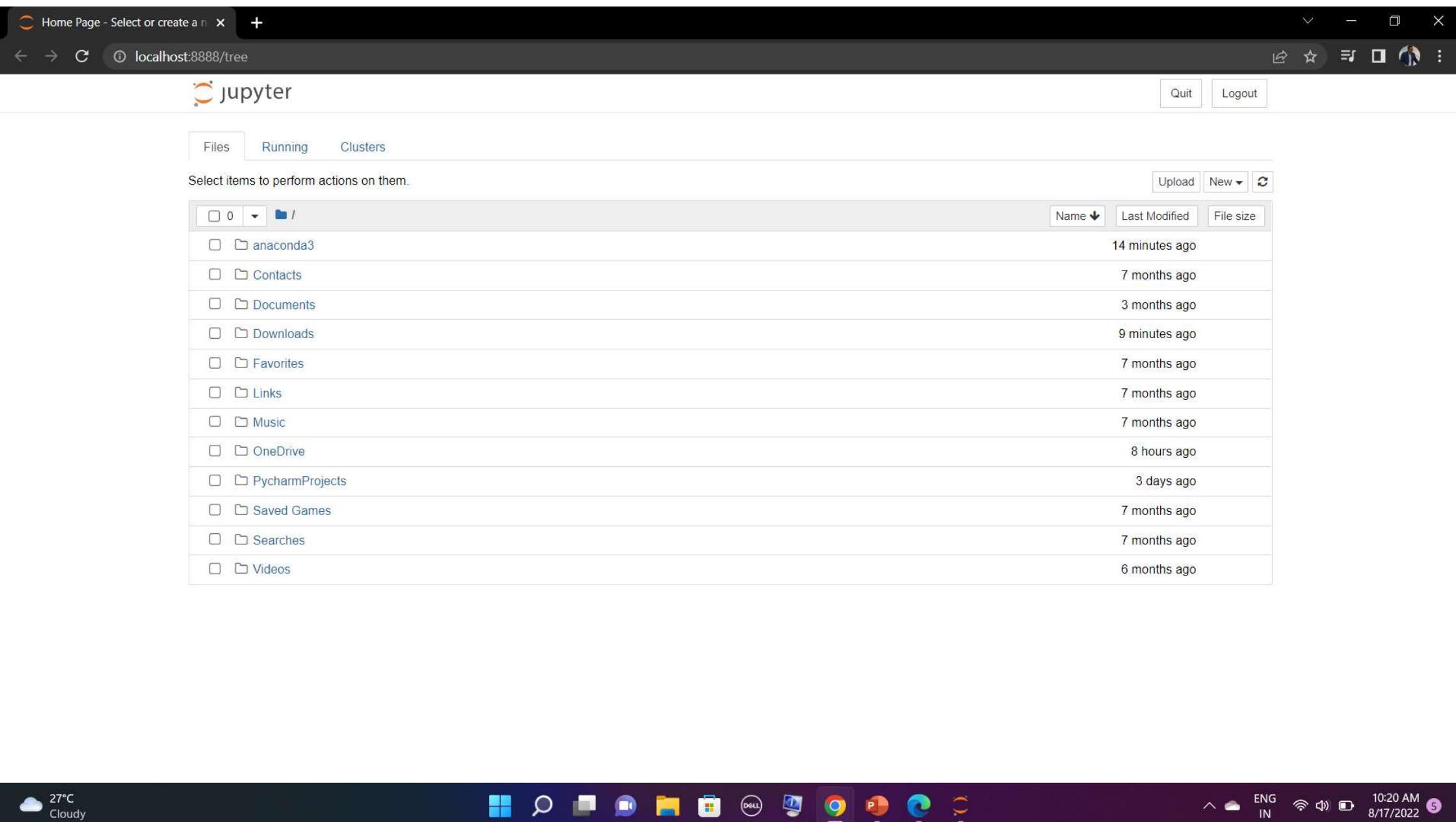
Downloads

VSCodeUserSetup-x64-1.70.1.exe
4.8 KB/s - 3.9 MB of 79.0 MB, 4 hours left









Home Page - Select or create a notebook

localhost:8888/tree#notebooks

jupyter

Files Running Clusters

Select items to perform actions on them.

	Name	Last Modified
<input type="checkbox"/>	0	7 months ago
<input type="checkbox"/>	/	
<input type="checkbox"/>	anaconda3	7 months ago
<input type="checkbox"/>	Contacts	7 months ago
<input type="checkbox"/>	Documents	7 months ago
<input type="checkbox"/>	Downloads	7 months ago
<input type="checkbox"/>	Favorites	7 months ago
<input type="checkbox"/>	Links	7 months ago
<input type="checkbox"/>	Music	7 months ago
<input type="checkbox"/>	OneDrive	8 hours ago
<input type="checkbox"/>	PycharmProjects	3 days ago
<input type="checkbox"/>	Saved Games	7 months ago
<input type="checkbox"/>	Searches	7 months ago
<input type="checkbox"/>	Videos	6 months ago

Upload New

Notebook: Python 3 (ipykernel)

Other: Text File, Folder, Terminal

localhost:8888/tree#notebooks

27°C Cloudy

ENG IN

10:22 AM 8/17/2022

Home x + v - □ x

localhost:8888/tree Logout

jupyter

Files Running Clusters

Rename Move Delete

Enter a new directory name:

Desktop

Cancel Rename

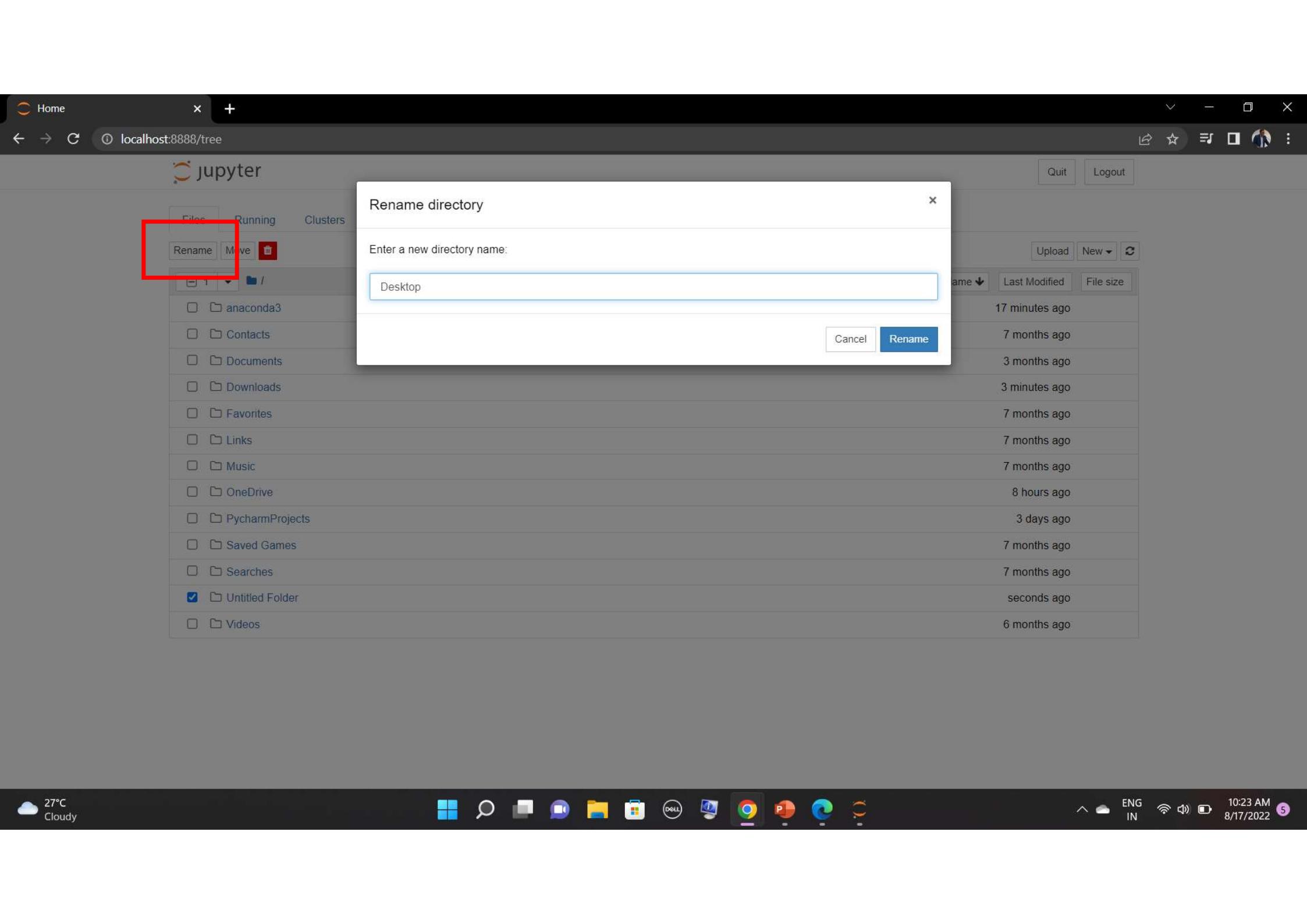
Rename directory

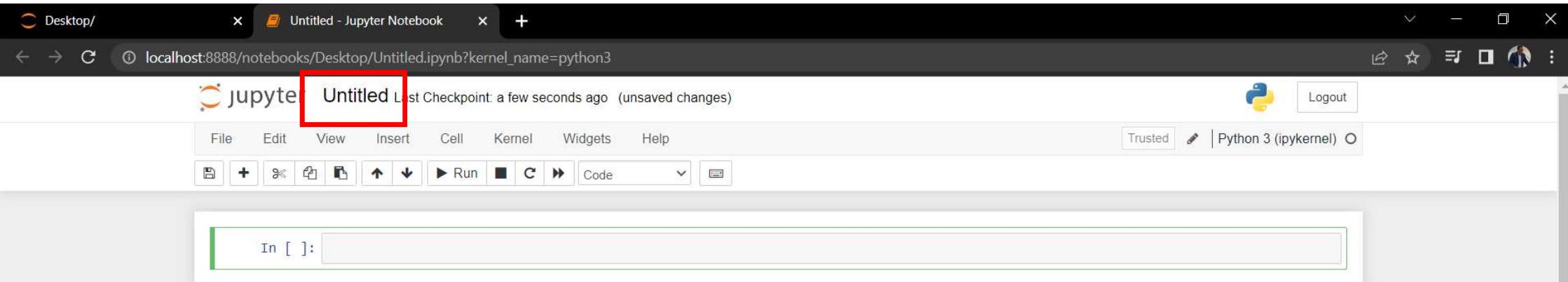
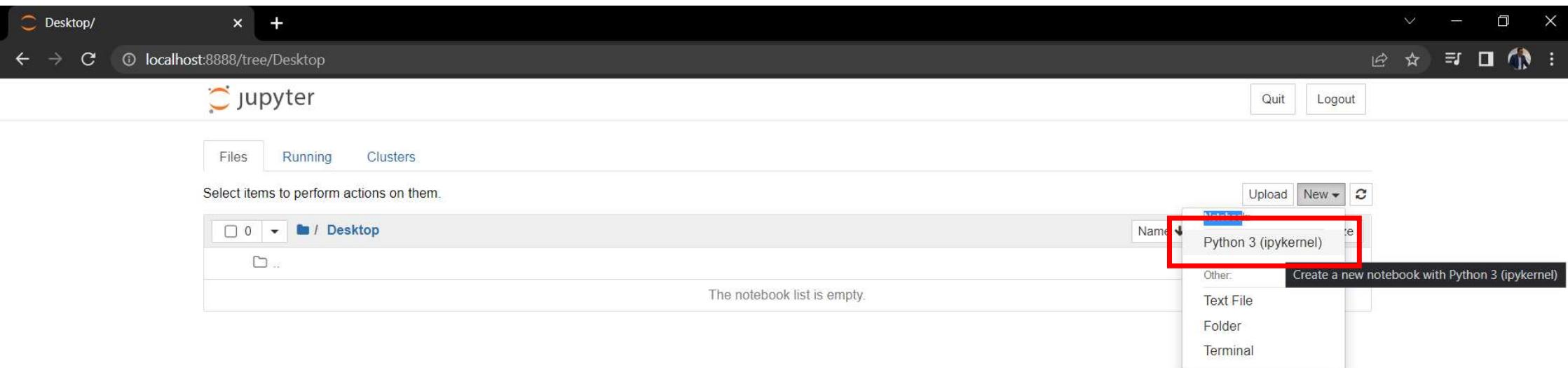
Upload New Copy

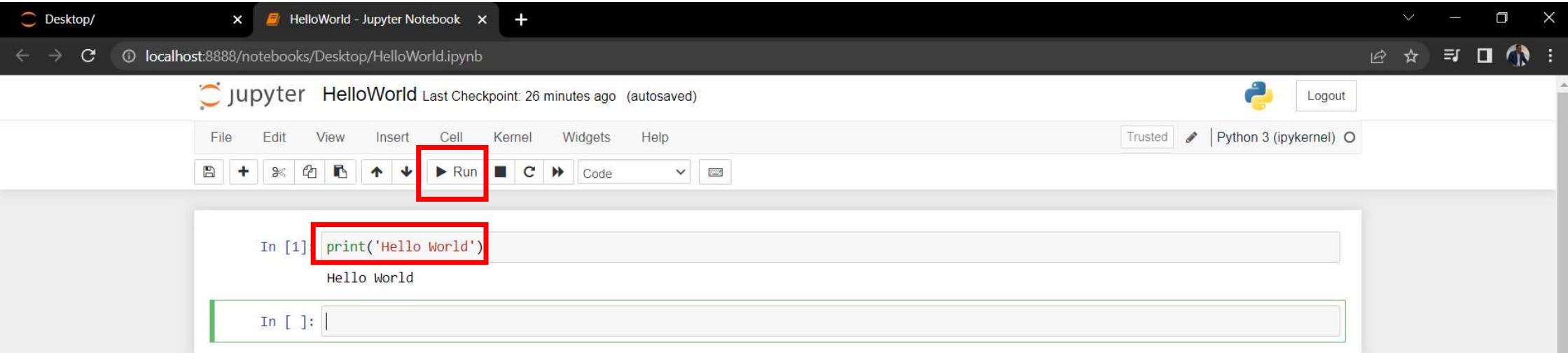
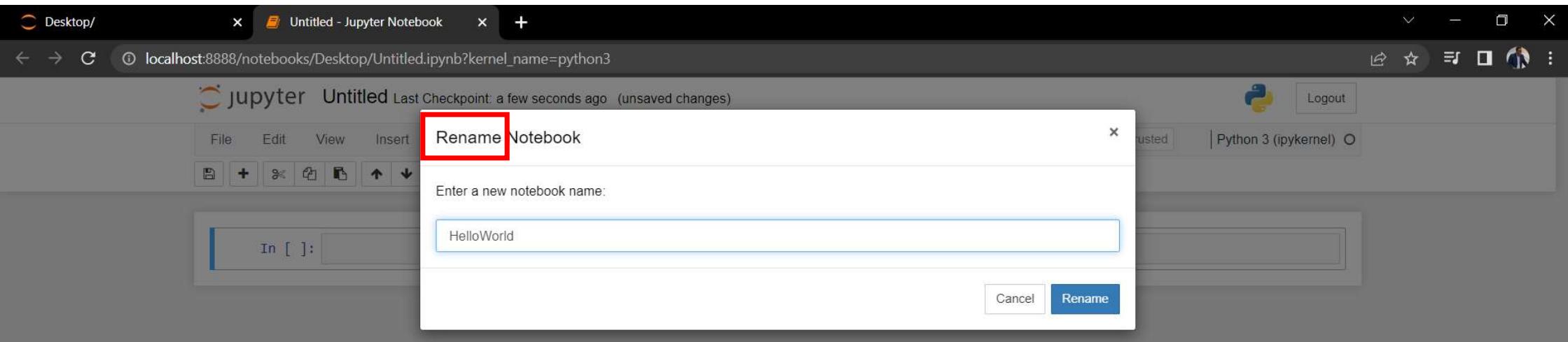
Name Last Modified File size

Name	Last Modified	File size
Desktop	17 minutes ago	
Untitled Folder	7 months ago	
anaconda3	3 months ago	
Contacts	3 minutes ago	
Documents	7 months ago	
Downloads	7 months ago	
Favorites	7 months ago	
Links	7 months ago	
Music	7 months ago	
OneDrive	8 hours ago	
PycharmProjects	3 days ago	
Saved Games	7 months ago	
Searches	7 months ago	
Videos	6 months ago	

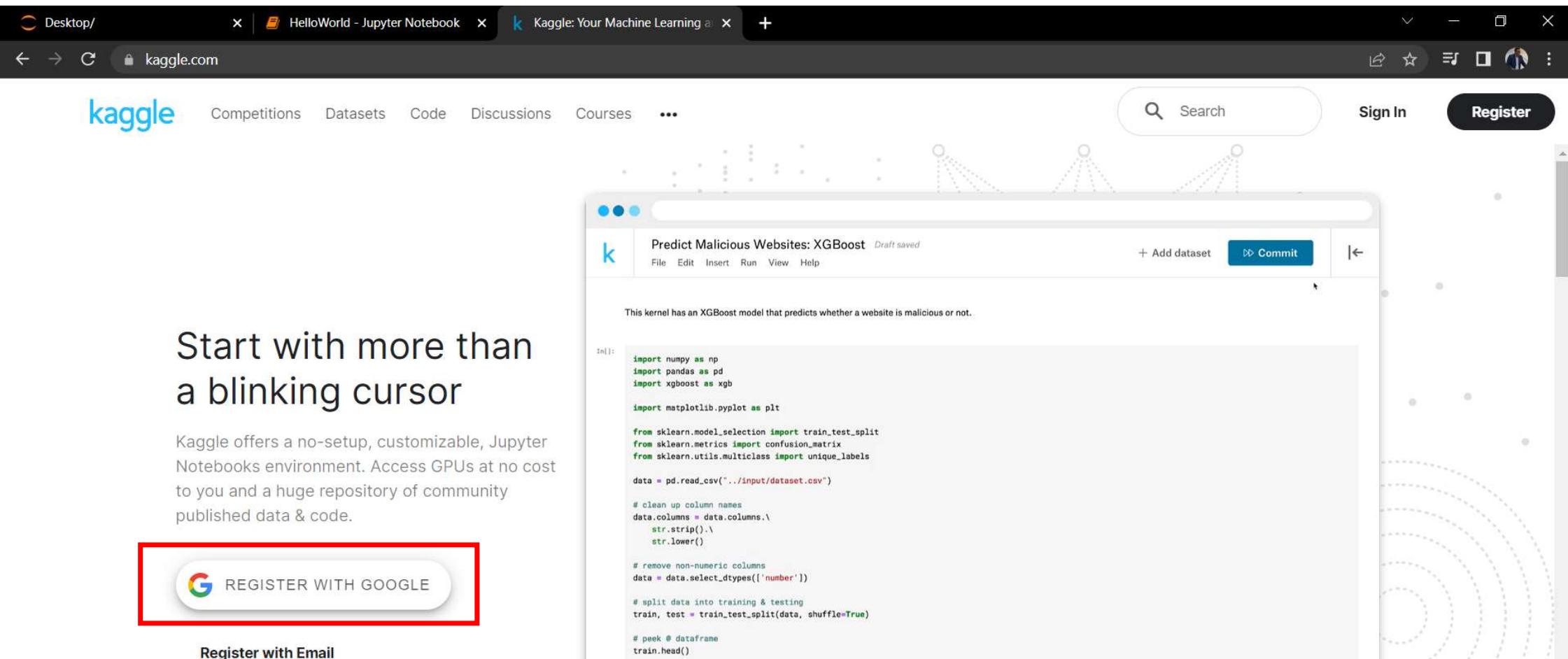
27°C Cloudy ENG IN 10:23 AM 8/17/2022 5







Importing/Loading Dataset (CSV) in Jupyter



The screenshot shows the Kaggle website interface. At the top, there are three tabs: "Desktop/" (active), "HelloWorld - Jupyter Notebook", and "Kaggle: Your Machine Learning". Below the tabs, the URL "kaggle.com" is visible. The main navigation bar includes links for "Competitions", "Datasets", "Code", "Discussions", "Courses", and a "Sign In" button. A search bar with the placeholder "Search" is also present.

The central area displays a Jupyter Notebook kernel titled "Predict Malicious Websites: XGBoost". The notebook has a "Draft saved" message. The code cell (In[1]) contains the following Python code:

```
import numpy as np
import pandas as pd
import xgboost as xgb

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.utils.multiclass import unique_labels

data = pd.read_csv("../input/dataset.csv")

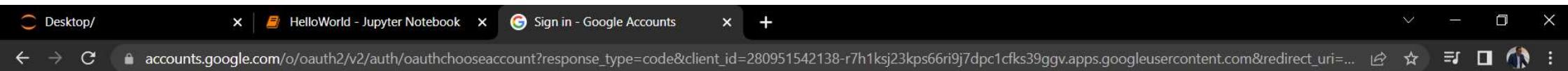
# clean up column names
data.columns = data.columns.\n    str.strip().\\
    str.lower()

# remove non-numeric columns
data = data.select_dtypes(['number'])

# split data into training & testing
train, test = train_test_split(data, shuffle=True)

# peek @ dataframe
train.head()
```

On the left side of the page, there is a large text block with the heading "Start with more than a blinking cursor" followed by a paragraph about Kaggle's Jupyter Notebook environment. Below this text is a red-bordered button labeled "REGISTER WITH GOOGLE" with a Google logo. At the bottom left, there is a link "Register with Email".



Complete Registration

FULL NAME (DISPLAYED)

Your profile URL
kaggle.com/hbpatel1976 [edit](#)

That username is already taken.

Desktop/ x | HelloWorld - Jupyter Notebook x k Kaggle: Your Home for Data Sci x +

← → C kaggle.com Search

kaggle Create

Home Competitions Datasets Code Discussions Courses More

Your Work View Active Events

27°C Cloudy ENG IN 11:09 AM 8/17/2022 5

Welcome, hbpatel1976!

Kaggle is the place to learn data science and build a portfolio

How to start: Choose a focus for today

Help us make relevant suggestions for you

Learn to compete on Kaggle
Improve and test your skills
[Get started](#) →

Take a short course
Our courses are the fastest way to learn data science
[Get started](#) →

Browse inspiring data and code
Improve your data science projects
[Get started](#) →

A red box highlights the search bar at the top of the page.

Desktop/ x | HelloWorld - Jupyter Notebook x k Search | Kaggle x +

kaggle.com/search?q=video

← video game sales | 🔍 X

View all results for "video game sales"

video game sales with ratings

video game sales 2020

Datasets View all 46

 Video Game Sales 6 years ago • 1 file (csv) • 390 kB • ^ 4676

 Video Game Sales with Ratings 6 years ago • 2 MB • ^ 913

Notebooks View all 845

 EDA - VIDEO GAME SALES 2 years ago • 3s to run • R • ^ 372

 Video Game Sales Exploratory Data Analysis 3 years ago • 24s to run • Python • ^ 150

Comments View all 495

 Reply to Video Game Sales 2 years ago • General • ^ 8

 Reply to Video Game Sales 2 years ago • General • ^ 3

Topics View all 134

27°C Cloudy ENG IN 11:10 AM 8/17/2022 5

Desktop/ x | HelloWorld - Jupyter Notebook x k Video Game Sales | Kaggle x +

kaggle.com/datasets/gregorut/videogamesales

kaggle Search

GREGORYSMITH - UPDATED 6 YEARS AGO 4686 New Notebook Download (390 kB) :

Video Game Sales

Analyze sales data from more than 16,500 games.



Data Code (909) Discussion (35) Metadata

About Dataset

This dataset contains a list of video games with sales greater than 100,000 copies. It was generated by a scrape of vgchartz.com.

Fields include

- Rank - Ranking of overall sales
- Name - The game's name
- Platform - Platform of the game's release (i.e. PC, PS4, etc.)
- Year - Year of the game's release
- Genre - Genre of the game
- Publisher - Publisher of the game
- NA_Sales - Sales in North America (in millions)

Usability 5.88

License Unknown

Expected update frequency Not specified

27°C Cloudy ENG IN 11:11 AM 8/17/2022 5

Desktop/ x | HelloWorld - Jupyter Notebook x | Video Game Sales | Kaggle x +

kaggle.com/datasets/gregorut/videogamesales

Search

Video Game Sales

Data Code (909) Discussion (35) Metadata ▲ 4686 New Notebook Download (390 kB) :

Games Video Games

vgsales.csv (1.36 MB)

Detail Compact Column 10 of 11 columns ▾

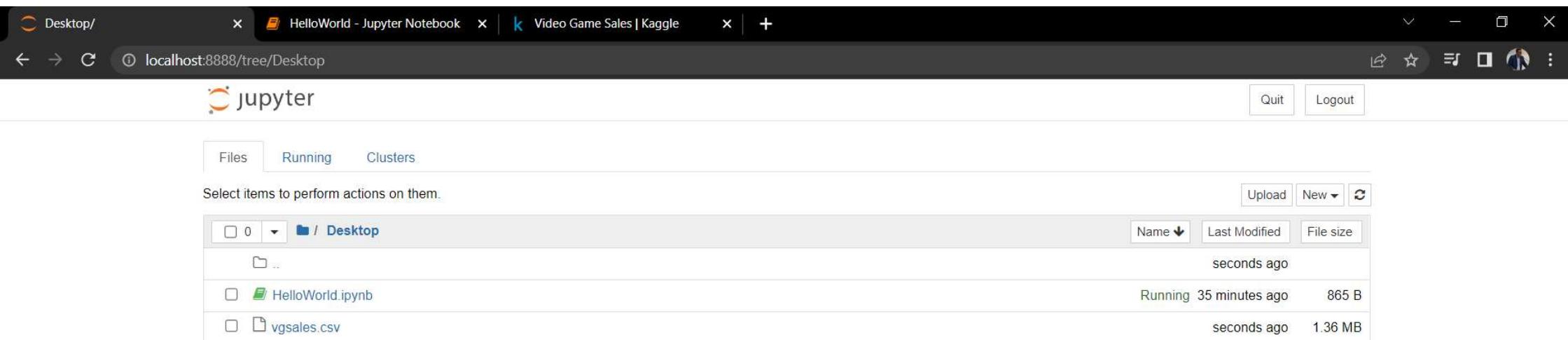
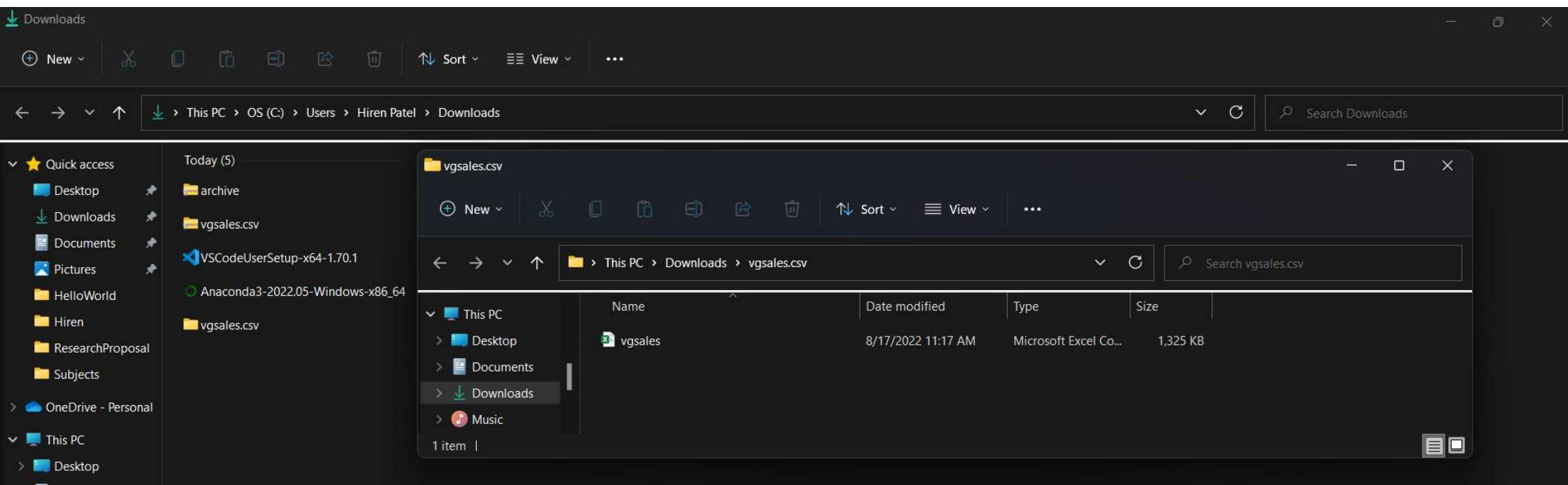
# Rank	Name	Platform	Year	Genre	P			
1 16.6k	11493 unique values	DS PS2 Other (12274)	13% 13% 74%	2009 2008 Other (13739)	9% 9% 83%	Action Sports Other (10936)	20% 14% 66%	Elect Act Oth
1	Wii Sports	Wii	2006	Sports	Nin			
2	Super Mario Bros.	NES	1985	Platform	Nin			
3	Mario Kart Wii	Wii	2008	Racing	Nin			
4	Wii Sports Resort	Wii	2009	Sports	Nin			
5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nin			
6	Tetris	GB	1989	Puzzle	Nin			

Data Explorer
Version 2 (1.36 MB)
vgsales.csv

vgsales.csv.zip

27°C Cloudy Show all X

ENG IN 11:16 AM 8/17/2022 5



Desktop/ HelloWorld - Jupyter Notebook Video Game Sales | Kaggle

localhost:8888/notebooks/Desktop>HelloWorld.ipynb

jupyter HelloWorld Last Checkpoint: an hour ago (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [2]: `import pandas as pd
df = pd.read_csv('vgsales.csv')
df`

Out[2]:

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	
0	Wii Sports	Wii	2006.0	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74	
1	Super Mario Bros.	NES	1985.0	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	
2	Mario Kart Wii	Wii	2008.0	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82	
3	Wii Sports Resort	Wii	2009.0	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00	
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	
...	
16593	16596	Woody Woodpecker in Crazy Castle 5	GBA	2002.0	Platform	Kemco	0.01	0.00	0.00	0.00	0.01
16594	16597	Men in Black II: Alien Escape	GC	2003.0	Shooter	Infogrames	0.01	0.00	0.00	0.00	0.01
16595	16598	SCORE International Baja 1000: The Official Game	PS2	2008.0	Racing	Activision	0.00	0.00	0.00	0.00	0.01
16596	16599	Know How 2	DS	2010.0	Puzzle	7G//AMES	0.00	0.01	0.00	0.00	0.01
16597	16600	Spirits & Spells	GBA	2003.0	Platform	Wanadoo	0.01	0.00	0.00	0.00	0.01

16598 rows × 11 columns

In []:

archive.zip vgsales.csv.zip Show all

27°C Cloudy ENG IN 11:24 AM 8/17/2022 5

```
In [3]: import pandas as pd  
df = pd.read_csv('vgsales.csv')  
df.shape
```

```
Out[3]: (16598, 11)
```

```
In [ ]:
```

```
In [4]: df.describe()
```

```
Out[4]:
```

	Rank	Year	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
count	16598.000000	16327.000000	16598.000000	16598.000000	16598.000000	16598.000000	16598.000000
mean	8300.605254	2006.406443	0.264667	0.146652	0.077782	0.048063	0.537441
std	4791.853933	5.828981	0.816683	0.505351	0.309291	0.188588	1.555028
min	1.000000	1980.000000	0.000000	0.000000	0.000000	0.000000	0.010000
25%	4151.250000	2003.000000	0.000000	0.000000	0.000000	0.000000	0.060000
50%	8300.500000	2007.000000	0.080000	0.020000	0.000000	0.010000	0.170000
75%	12449.750000	2010.000000	0.240000	0.110000	0.040000	0.040000	0.470000
max	16600.000000	2020.000000	41.490000	29.020000	10.220000	10.570000	82.740000

```
In [5]: df.values
```

```
Out[5]: array([[1, 'Wii Sports', 'Wii', ..., 3.77, 8.46, 82.74],  
               [2, 'Super Mario Bros.', 'NES', ..., 6.81, 0.77, 40.24],  
               [3, 'Mario Kart Wii', 'Wii', ..., 3.79, 3.31, 35.82],  
               ...,  
               [16598, 'SCORE International Baja 1000: The Official Game', 'PS2',  
                ..., 0.0, 0.0, 0.01],  
               [16599, 'Know How 2', 'DS', ..., 0.0, 0.0, 0.01],  
               [16600, 'Spirits & Spells', 'GBA', ..., 0.0, 0.0, 0.01]],  
              dtype=object)
```

music.csv

1. Import Data

The image shows two windows side-by-side. On the left is a Microsoft Excel spreadsheet titled 'music.csv'. The first few rows of data are:

	A	B	C	D
1	age	gender	genre	
2	20	1	HipHop	
3	23	1	HipHop	
4	25	1	HipHop	
5	26	1	Jazz	
6	29	1	Jazz	
7	30	1	Jazz	
8	31	1	Classical	
9	33	1	Classical	
10	37	1	Classical	
11	20	0	Dance	
12	21	0	Dance	
13	25	0	Dance	
14	26	0	Acoustic	
15	27	0	Acoustic	
16	30	0	Acoustic	
17	31	0	Classical	
18	34	0	Classical	
19	35	0	Classical	
20				

On the right is a Jupyter Notebook interface. The code cell In [11] contains:

```
import pandas as pd
music_data = pd.read_csv('music.csv')
music_data
```

The output cell Out[11] displays the same data as the Excel spreadsheet:

	age	gender	genre
0	20	1	HipHop
1	23	1	HipHop
2	25	1	HipHop
3	26	1	Jazz
4	29	1	Jazz
5	30	1	Jazz
6	31	1	Classical
7	33	1	Classical
8	37	1	Classical
9	20	0	Dance
10	21	0	Dance
11	25	0	Dance
12	26	0	Acoustic
13	27	0	Acoustic
14	30	0	Acoustic
15	31	0	Classical
16	34	0	Classical

2. Clean Data

Remove duplicates and clean NULL values but we don't have such thing in our current music.csv file

However, we may split the database into two table, first containing the input (first two column)s and second containing the output (last column)

```
In [12]: import pandas as pd  
music_data = pd.read_csv('music.csv')  
x = music_data.drop(columns=['genre'])  
x
```

Out[12]:

	age	gender
0	20	1
1	23	1
2	25	1
3	26	1
4	29	1
5	30	1
6	31	1
7	33	1
8	37	1
9	20	0
10	21	0
11	25	0
12	26	0
13	27	0
14	30	0
15	31	0

```
In [19]: import pandas as pd  
music_data = pd.read_csv('music.csv')  
x = music_data.drop(columns=['genre'])  
y = music_data['genre']  
y
```

Out[19]:

```
0      HipHop  
1      HipHop  
2      HipHop  
3      Jazz  
4      Jazz  
5      Jazz  
6    Classical  
7    Classical  
8    Classical  
9      Dance  
10     Dance  
11     Dance  
12   Acoustic  
13   Acoustic  
14   Acoustic  
15  Classical  
16  Classical  
17  Classical  
Name: genre, dtype: object
```

3. Learning and Predicting Data

```
In [43]: import pandas as pd
from sklearn.tree import DecisionTreeClassifier

#import dataset
music_data = pd.read_csv('music.csv')
#create input set
X = music_data.drop(columns=['genre'])
#create output set
y = music_data['genre']

#create a model
model = DecisionTreeClassifier()
#train the model
model.fit(X, y)

music_data
```

Out[43]:

	age	gender	genre
0	20	1	HipHop
1	23	1	HipHop
2	25	1	HipHop
3	26	1	Jazz
4	29	1	Jazz
5	30	1	Jazz
6	31	1	Classical
7	33	1	Classical

```
In [25]: import pandas as pd
from sklearn.tree import DecisionTreeClassifier

music_data = pd.read_csv('music.csv')
X = music_data.drop(columns=['genre'])
y = music_data['genre']

model = DecisionTreeClassifier()
model.fit(X, y)

predictions = model.predict([[21, 1], [22, 0]])
predictions
```

Out[25]: array(['HipHop', 'Dance'], dtype=object)

In []:

Q: What does a male (1) of age (21) like? [21, 1]
Prediction: HipHop

Q: What does a female (0) of age (22) like? [22, 0]
Prediction: Dance

4. Calculating Accuracy

Divide the dataset into two sets viz. (A) training dataset (70-80%), (B) testing dataset (20-30%)

Instead of passing one or two samples, we'll pass testing dataset and compare the resultant predictions with the actual values. Based on that, we can calculate the accuracy.

accuracy_score(y_test, predictions): y_test: Expected values, predictions: Actual values.

```
In [26]: import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

music_data = pd.read_csv('music.csv')
X = music_data.drop(columns=['genre'])
y = music_data['genre']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = DecisionTreeClassifier()
model.fit(X_train, y_train)
|
predictions = model.predict(X_test)

score = accuracy_score(y_test, predictions)
score

Out[26]: 1.0

Out[29]: 0.75
```

You may run this again and again with different size of training/testing data (E.g. test_size=0.5)

5. Persisting Models

```
In [39]: import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.externals import joblib

music_data = pd.read_csv('music.csv')
X = music_data.drop(columns=['genre'])
y = music_data['genre']

model = DecisionTreeClassifier()
model.fit(X, y)

joblib.dump(model, 'music-recommender.joblib')
```

```
Out[39]: ['music-recommender.joblib']
```



```
In [41]: import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn.externals import joblib

# music_data = pd.read_csv('music.csv')
# X = music_data.drop(columns=['genre'])
# y = music_data['genre']

# model = DecisionTreeClassifier()
# model.fit(X, y)

model = joblib.load('music-recommender.joblib')
predictions = model.predict([[21, 1]])
predictions
```

Out[41]: array(['HipHop'], dtype=object)

6. Visualizing a Decision Tree

```
In [45]: import pandas as pd
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree

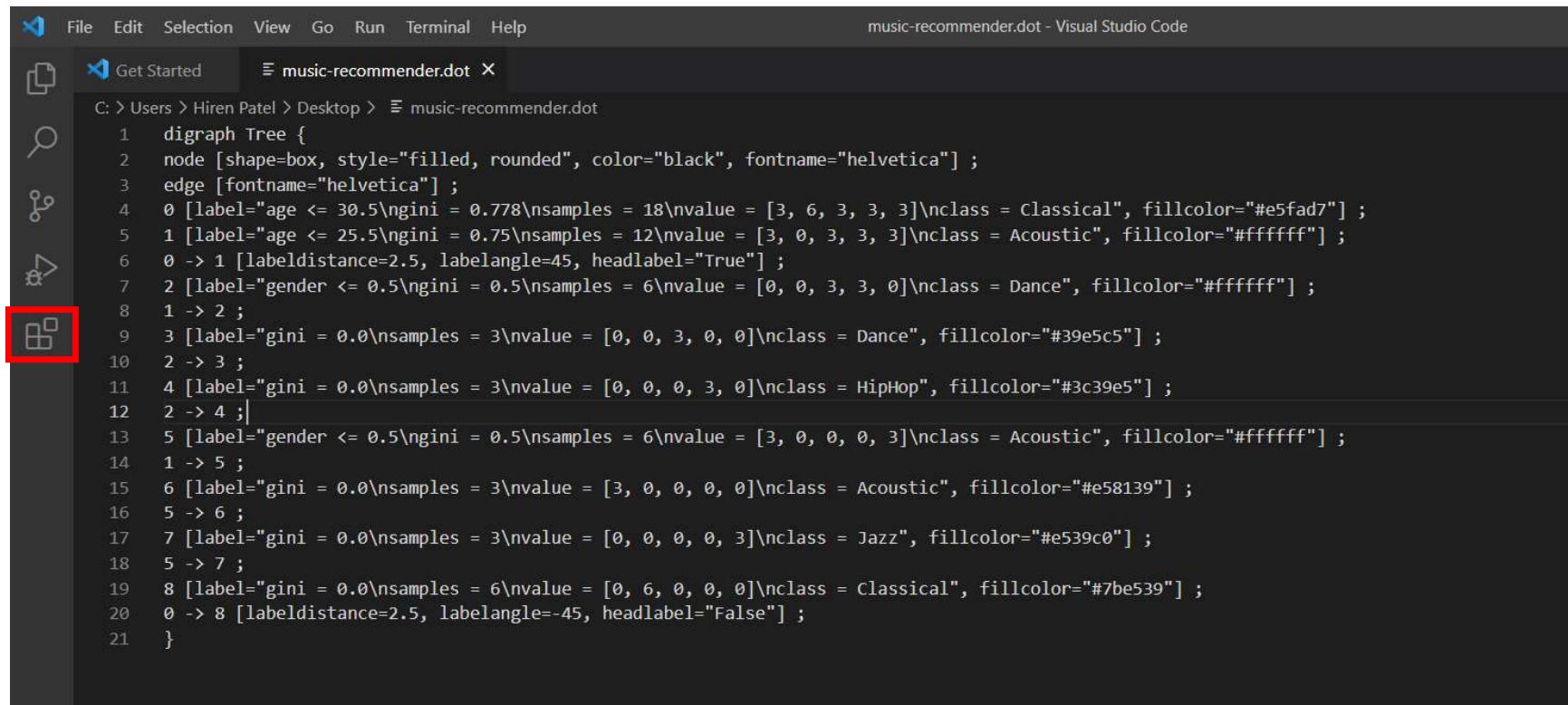
music_data = pd.read_csv('music.csv')
X = music_data.drop(columns=['genre'])
y = music_data['genre']

model = DecisionTreeClassifier()
model.fit(X, y)

tree.export_graphviz(model, out_file='music-recommender.dot',
                     feature_names=['age', 'gender'],
                     class_names=sorted(y.unique()),
                     label='all',
                     rounded=True,
                     filled=True)
```

		Name	Last Modified	File size
□	0	...	seconds ago	
□	HelloWorld.ipynb		Running a minute ago	4.04 kB
□	music-recommender.dot		a minute ago	1.3 kB
□	music-recommender.joblib		23 minutes ago	2.54 kB
□	music.csv		3 hours ago	270 B
□	vgsales.csv		8 hours ago	1.36 MB

Open the music-recommender.dot [from C:\Users\Hiren Patel\Desktop] file in Visual Studio Code.



```
File Edit Selection View Go Run Terminal Help
music-recommender.dot - Visual Studio Code
Get Started music-recommender.dot X
C: > Users > Hiren Patel > Desktop > music-recommender.dot
1 digraph Tree {
2 node [shape=box, style="filled, rounded", color="black", fontname="helvetica"] ;
3 edge [fontname="helvetica"] ;
4 0 [label="age <= 30.5\n gini = 0.778\n samples = 18\n value = [3, 6, 3, 3, 3]\n class = Classical", fillcolor="#e5fad7"] ;
5 1 [label="age <= 25.5\n gini = 0.75\n samples = 12\n value = [3, 0, 3, 3, 3]\n class = Acoustic", fillcolor="#ffffff"] ;
6 0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
7 2 [label="gender <= 0.5\n gini = 0.5\n samples = 6\n value = [0, 0, 3, 3, 0]\n class = Dance", fillcolor="#ffffff"] ;
8 1 -> 2 ;
9 3 [label="gini = 0.0\n samples = 3\n value = [0, 0, 3, 0, 0]\n class = Dance", fillcolor="#39e5c5"] ;
10 2 -> 3 ;
11 4 [label="gini = 0.0\n samples = 3\n value = [0, 0, 0, 3, 0]\n class = HipHop", fillcolor="#3c39e5"] ;
12 2 -> 4 ;
13 5 [label="gender <= 0.5\n gini = 0.5\n samples = 6\n value = [3, 0, 0, 0, 3]\n class = Acoustic", fillcolor="#ffffff"] ;
14 1 -> 5 ;
15 6 [label="gini = 0.0\n samples = 3\n value = [3, 0, 0, 0, 0]\n class = Acoustic", fillcolor="#e58139"] ;
16 5 -> 6 ;
17 7 [label="gini = 0.0\n samples = 3\n value = [0, 0, 0, 0, 3]\n class = Jazz", fillcolor="#e539c0"] ;
18 5 -> 7 ;
19 8 [label="gini = 0.0\n samples = 6\n value = [0, 6, 0, 0, 0]\n class = Classical", fillcolor="#7be539"] ;
20 0 -> 8 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;
21 }
```

File Edit Selection View Go Run Terminal Help

Extension: Graphviz (dot) language support for Visual Studio Code - Visual Studio Code

EXTENSIONS: MARKET... Get Started Extension: Graphviz (dot) language support for Visual Studio Code

dot

.ENV DotENV Support for dotenv file syntax 3M 5 João Pinto | 173,535 ★★★★★(12)

Graphviz (dot)... This extension provides Gra... 173K 5 João Pinto | Install

Graphviz (dot) ... 88K 5 Stephanvs | Install

Dot Template create new files according t... 4K Mora Qiu | Install

Dot Log Make it easy to console.log i... 3K 5 jaluk | Install

Dot Thing This is a Vscode plugin passi... 12 5 Victor Bo | Install

dot-icons Icons with a dot 474 5 Andreas Weber | Install

dot-creator a simple code creator of Bce... 33 SimbaNana | Install

Dot Developer ... VS Code Dot Developer The... 265 5 n-devs | Install

Dot Net Complete ... this extension creates norm... 705 David Kroods | Install

t-dot-svelte-vscode Translate your app with ease 83 jjnguy | Install

Graphviz (dot) language support for Visual Studio Code v0.0.6

João Pinto | 173,535 ★★★★★(12)

This extension provides GraphViz (dot) language support for Visual Studio Code

Install

Details Feature Contributions

Graphviz Support

A vscode extension that provides language support and live preview for the Graphviz format.

The preview uses the [Viz.js](#) library.

The extension can be activated in two ways

Features

Snippets

Try typing one of the following prefixes to see available snippets: `graph`, `>`, `var`, `dir`, `prop`, `path` or `rank` and efficiently create graphs, variables, properties, paths or ranks.

Graph preview

- Toggle Preview - `ctrl+shift+v` (Mac: `cmd+shift+v`)
- Open Preview to the Side - `ctrl+k v` (Mac: `cmd+k shift+v`)

Categories

Programming Languages Snippets

Extension Resources

Marketplace Repository License João Pinto

More Info

Released on 3/26/2018, 02:22:37
Last updated 2/26/2020, 15:46:02
Identifier joaompinto.vscode-graphviz

28°C Cloudy ENG IN 7:18 PM 8/17/2022

Reload Visual Studio Code.

The screenshot shows the Visual Studio Code interface with a dark theme. The top bar includes File, Edit, Selection, View, Go, Run, Terminal, and Help. A tab bar shows "music-recommender.dot - Visual Studio Code". The Explorer sidebar on the left has a "TIMELINE" section with one item. The main editor area contains the following graphviz dot code:

```
C: > Users > Hiren Patel > Desktop > music-recommender.dot
1 digraph Tree {
2 node [shape=box, style="filled, rounded", color="black", fontname="helvetica"] ;
3 edge [fontname="helvetica"] ;
4 0 [label="age <= 30.5\ngini = 0.778\nsamples = 18\nvalue = [3, 6, 3, 3, 3]\nclass = Classical", fillcolor="#e5f5ff"] ;
5 1 [label="age <= 25.5\ngini = 0.75\nsamples = 12\nvalue = [3, 0, 3, 3, 3]\nclass = Acoustic", fillcolor="#ffffff"] ;
6 0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
7 2 [label="gender <= 0.5\ngini = 0.5\nsamples = 6\nvalue = [0, 0, 3, 3, 0]\nclass = Dance", fillcolor="#ffffff"] ;
8 1 -> 2 ;
9 3 [label="gini = 0.0\nsamples = 3\nvalue = [0, 0, 3, 0, 0]\nclass = Dance", fillcolor="#39e5c5"] ;
10 2 -> 3 ;
11 4 [label="gini = 0.0\nsamples = 3\nvalue = [0, 0, 0, 3, 0]\nclass = HipHop", fillcolor="#3c39e5"] ;
12 2 -> 4 ;
13 5 [label="gender <= 0.5\ngini = 0.5\nsamples = 6\nvalue = [3, 0, 0, 0, 3]\nclass = Acoustic", fillcolor="#ffffff"] ;
14 1 -> 5 ;
15 6 [label="gini = 0.0\nsamples = 3\nvalue = [3, 0, 0, 0, 0]\nclass = Acoustic", fillcolor="#e58139"] ;
16 5 -> 6 ;
17 7 [label="gini = 0.0\nsamples = 3\nvalue = [0, 0, 0, 0, 3]\nclass = Jazz", fillcolor="#e539c0"] ;
18 5 -> 7 ;
19 8 [label="gini = 0.0\nsamples = 6\nvalue = [0, 6, 0, 0, 0]\nclass = Classical", fillcolor="#7be539"] ;
20 0 -> 8 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;
21 }
```

A context menu is open on the right side of the editor, listing "Show Opened Editors", "Close All" (Ctrl+K W), "Close Saved" (Ctrl+K U), "Enable Preview Editors" (checked), "Open Preview to the Side" (Ctrl+K V), and "Open Preview in New Tab" (Ctrl+K X).

File Edit Selection View Go Run Terminal Help

Preview: 'music-recommender.dot' - Visual Studio Code

EXPLORER: TIMELINE

The active editor cannot provide timeline information.

```
C: > Users > Hiren Patel > Desktop > music-recommender.dot
```

```
1 digraph Tree {  
2 node [shape=box, style="stroke:#000,stroke-width:1px"];  
3 edge [fontname="helvetica, sans-serif"];  
4 0 [label="age <= 30"];  
5 1 [label="age <= 25"];  
6 0 -> 1 [label="distance <= 1000"];  
7 2 [label="gender <= 0.5"];  
8 1 -> 2 ;  
9 3 [label="gini = 0.0"];  
10 2 -> 3 ;  
11 4 [label="gini = 0.0"];  
12 2 -> 4 ;  
13 5 [label="gender <= 0.5"];  
14 1 -> 5 ;  
15 6 [label="gini = 0.0"];  
16 5 -> 6 ;  
17 7 [label="gini = 0.0"];  
18 5 -> 7 ;  
19 8 [label="gini = 0.0"];  
0 -> 8 [label="distance <= 1000"];  
21 }
```

```
graph TD; Root[age <= 30] --> Node1[age <= 25.5]; Node1 -- True --> Node2[gender <= 0.5]; Node2 --> Leaf1[gini = 0.0, samples = 3, value = [0, 0, 3, 0, 0], class = Dance]; Node2 -- False --> Node3[gender <= 0.5]; Node3 --> Leaf2[gini = 0.0, samples = 3, value = [0, 0, 0, 3, 0], class = HipHop]; Node3 --> Leaf3[gini = 0.0, samples = 3, value = [3, 0, 0, 0, 0], class = Acoustic]; Root -- False --> Leaf4[gini = 0.0, samples = 6, value = [0, 6, 0, 0, 0], class = Classical];
```

100 % ▲ ▼ 1:1 ↶ ↷ ↸

28°C Cloudy ENG IN 7:22 PM 8/17/2022 5

Django Project

django - Google Search

google.com/search?q=django&rlz=1C1RXQR_enIN90IN990&oq=django&aqs=chrome..69i57j0i131i433i512i3j0i433i512j46i131i433i512j0i433i512j0i131i433i512l2j0i512.1319j0j78sourceid=...

Google

django

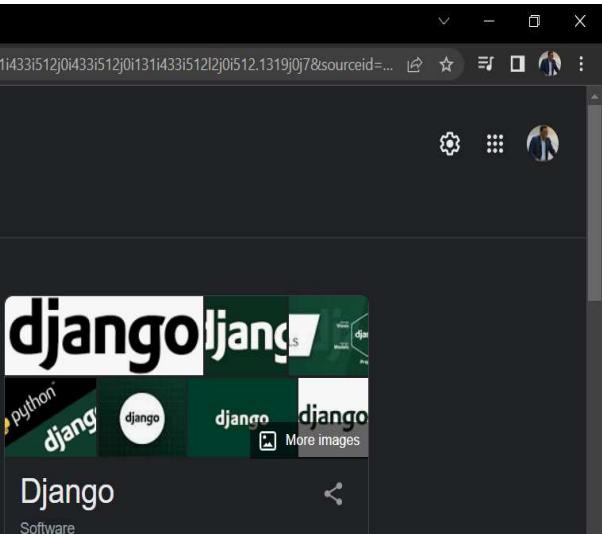
All Books Images Videos News More Tools

About 19,70,00,000 results (0.39 seconds)

<https://www.djangoproject.com/> Django: The web framework for perfectionists with deadlines
Django is a high-level Python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of ...

[Documentation](#)
Django at a glance - Quick install guide - Getting started - Models

[Writing your first Django app](#)



dj The web framework for perfectionists with deadlines

djangoproject.com

django The web framework for perfectionists with deadlines.

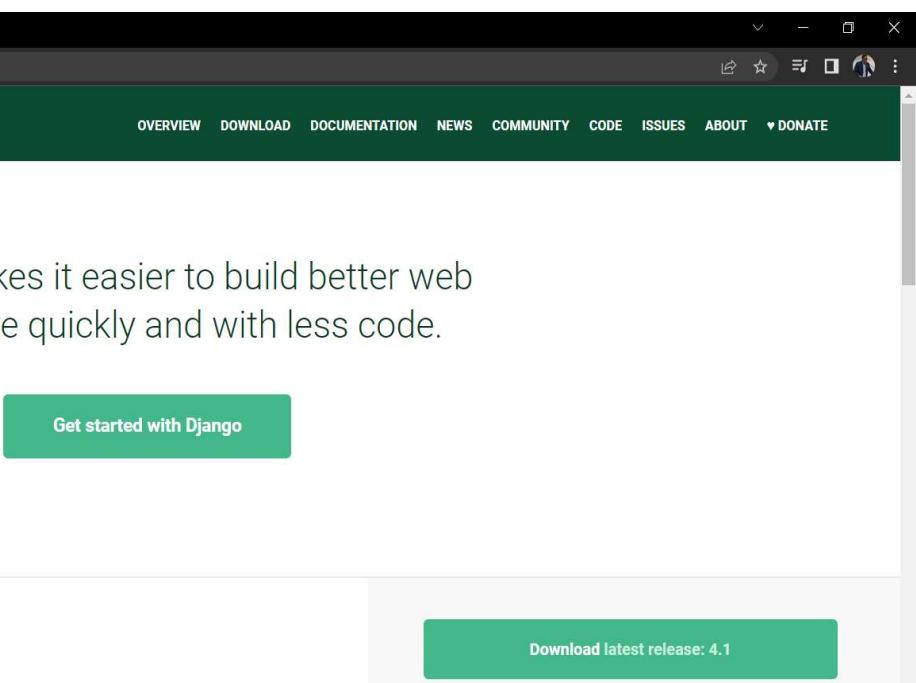
OVERVIEW DOWNLOAD DOCUMENTATION NEWS COMMUNITY CODE ISSUES ABOUT ▾ DONATE

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

Get started with Django

Meet Django

Download latest release: 4.1

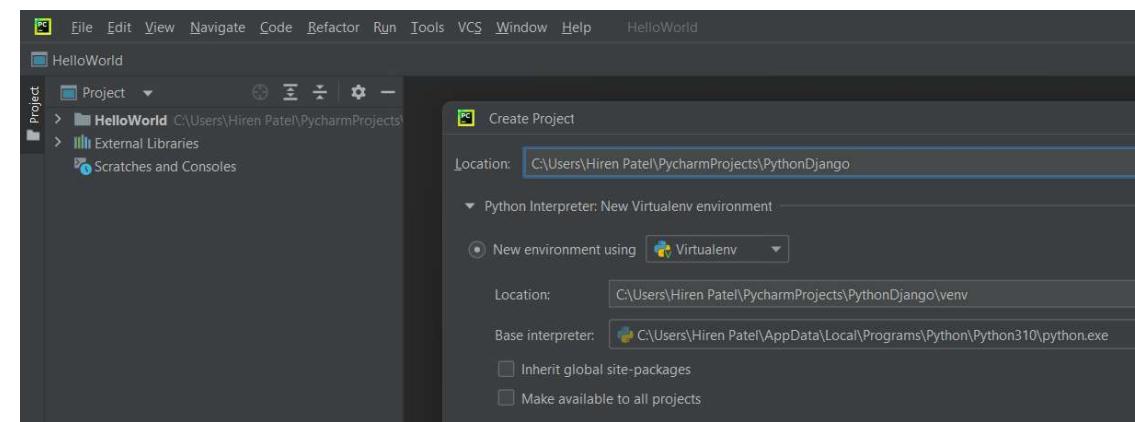
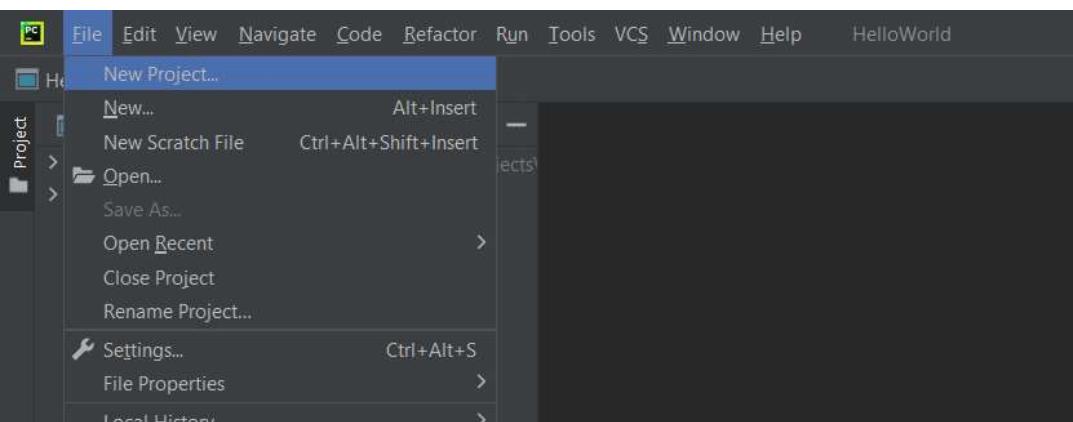


Top websites build using Django Framework

[Disqus](#)
[Pinterest](#)
[Facebook](#)
[Instagram](#)
[Spotify](#)
[Netflix](#)
[Youtube](#)
[The Washington Post](#)
[The Onion](#)
[National Geographic](#)
[Nasa](#)
[Dropbox](#)
[Opesta](#)
[Prezi](#)
[Eventbrite](#)
[Uber](#)
[Lyft](#)
[BitBucket](#)
[Mozilla](#)
[Summary](#)

What is a Framework:

- Library of reusable modules
- Defines a structure for an application



The screenshot shows the PyCharm interface with the code editor open. The file 'main.py' is displayed. The code is as follows:

```
# This is a sample Python script.

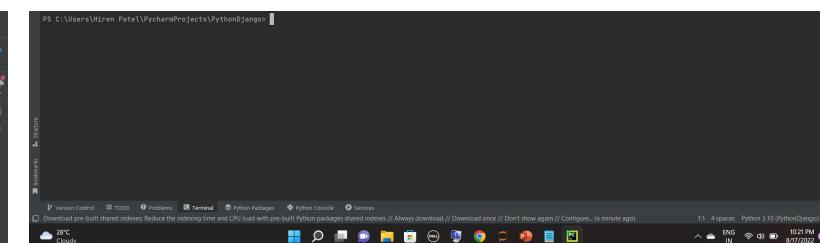
# Press Shift+F10 to execute it or replace it with your code.
# Press Double Shift to search everywhere for classes, files, actions, and settings.

def print_hi(name):
    # Use a breakpoint in the code line below to debug your script.
    print(f'Hi, {name}') # Press Ctrl+F8 to toggle the breakpoint.

# Press the green button in the gutter to run the script.
if __name__ == '__main__':
    print_hi('PyCharm')

# See PyCharm help at https://www.jetbrains.com/help/pycharm/
```

The code editor has a red box highlighting the 'Terminal' tab in the bottom navigation bar.



File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - main.py

PythonDjango > main.py

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

main.py

External Libraries

Scratches and Consoles

Terminal: C:\Windows\system32\rshell.exe

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> pip install django==4.1
Collecting django==4.1
  Downloading Django-4.1-py3-none-any.whl (8.1 MB)
    8.1/8.1 MB 234.4 kB/s eta 0:00:00
Collecting sqlparse>=0.2.2
  Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
    42.3/42.3 kB 341.2 kB/s eta 0:00:00
Collecting asgiref<4,>=3.5.2
  Downloading asgiref-3.5.2-py3-none-any.whl (22 kB)
Collecting tzdata
  Downloading tzdata-2022.2-py2.py3-none-any.whl (336 kB)
    336.4/336.4 kB 260.9 kB/s eta 0:00:00
Installing collected packages: tzdata, sqlparse, asgiref, django
Successfully installed asgiref-3.5.2 django-4.1 sqlparse-0.4.2 tzdata-2022.2

[notice] A new release of pip available: 22.2.1 -> 22.2.2
[notice] To update, run: python.exe -m pip install --upgrade pip
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```

Version Control TODO Problems Terminal Python Packages Python Console Services

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (6 minutes ago)

28°C Cloudy

1:1 4 spaces Python 3.10 (PythonDjango) 10:25 PM 8/17/2022 6

The screenshot shows the PyCharm IDE interface. The top bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonDjango - main.py. The Project tool window on the left shows a folder structure with a red box around the 'PythonDjango' folder. The main editor window displays 'main.py' with sample code. The terminal at the bottom shows the command: PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> django-admin startproject PythonDjango .

The screenshot shows the PyCharm IDE interface. The top bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, and PythonDjango > PythonDjango > settings.py. The Project tool window on the left shows a folder structure with a red box around the 'PythonDjango' folder. The main editor window displays 'settings.py'. The terminal at the bottom shows the command: PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py runserver. The output indicates the server is running at http://127.0.0.1:8000/.

The install worked successfully!  +

127.0.0.1:8000

 django View [release notes](#) for Django 4.1



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

 [Django Documentation](#)
Topics, references, & how-to's

 [Tutorial: A Polling App](#)
Get started with Django

 [Django Community](#)
Connect, get help, or contribute

28°C Cloudy

Windows Start button, Search, File, Settings, Task View, Chrome, File Explorer, Dell logo, Taskbar icons, Network, ENG IN, 10:33 PM, 8/17/2022, Battery icon

Django Project

Order Management (Django APP #1)	Customer Management (Django APP #2)
Product Management (Django APP #3)	Other (Django APP #4)

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - PythonDjango\urls.py

PythonDjango > PythonDjango > urls.py

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

- products
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py
- PythonDjango
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
- venv\library root
 - db.sqlite3
 - main.py
 - manage.py

External Libraries

Scratches and Consoles

Structure

Bookmarks

Version Control TODO Problems Terminal Python Packages Python Console Services

You are using the Django framework PyCharm Professional Edition has special support for it. (24 minutes ago)

29°C Cloudy 21:48 CRLF UTF-8 4 spaces Python 3.10 (PythonDjango) (2) ENG INTL 9:35 PM 8/18/2022 5

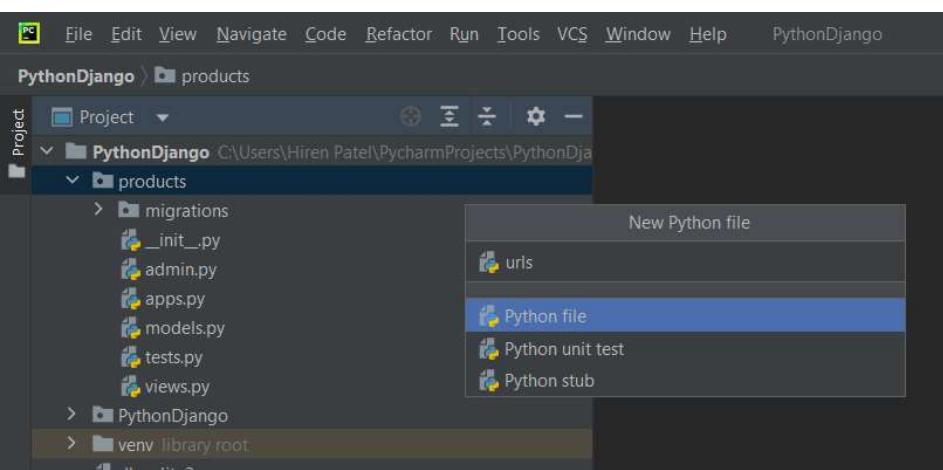
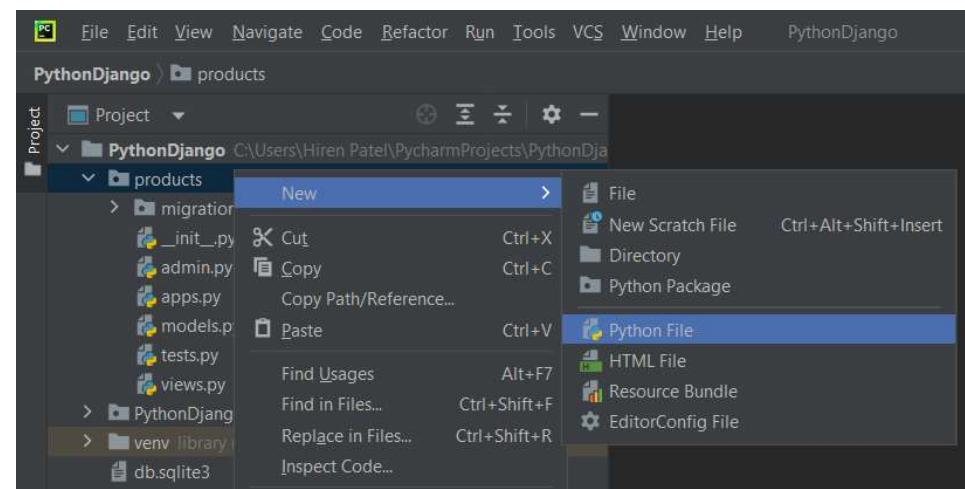
products\urls.py PythonDjango\urls.py views.py

```
1 """PythonDjango URL Configuration
2
3     The `urlpatterns` list routes URLs to views. For more information please see:
4         https://docs.djangoproject.com/en/4.1/topics/http/urls/
5     Examples:
6         Function views
7             1. Add an import: from my_app import views
8             2. Add a URL to urlpatterns: path('', views.home, name='home')
9
10    Class-based views
11        1. Add an import: from other_app.views import Home
12        2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13
14    Including another URLconf
15        1. Import the include() function: from django.urls import include, path
16        2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
17
18
19    from django.contrib import admin
20    from django.urls import path, include
21
22    urlpatterns = [
23        path('admin/', admin.site.urls),
24        path('products/', include('products.urls'))]
```

PyCharm interface showing the code editor for `views.py` in the `products` application of the `PythonDjango` project. The code defines a single view function `index` that returns a `HttpResponse` with the message "Hello Hiren". The `views.py` file is highlighted with a red box.

```
from django.http import HttpResponse
from django.shortcuts import render

def index(request):
    return HttpResponse('Hello Hiren')
```

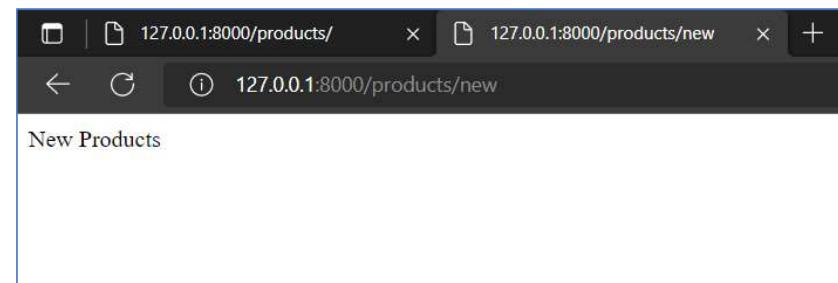
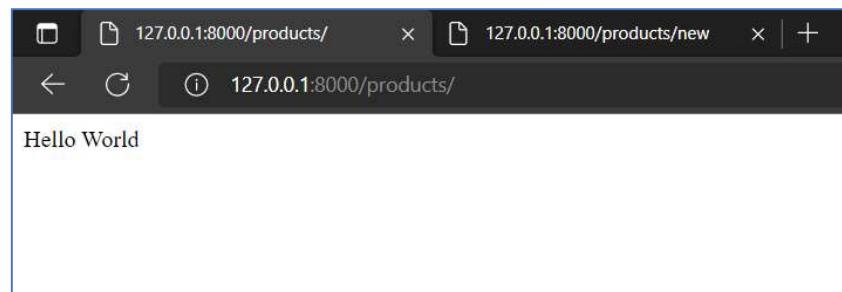
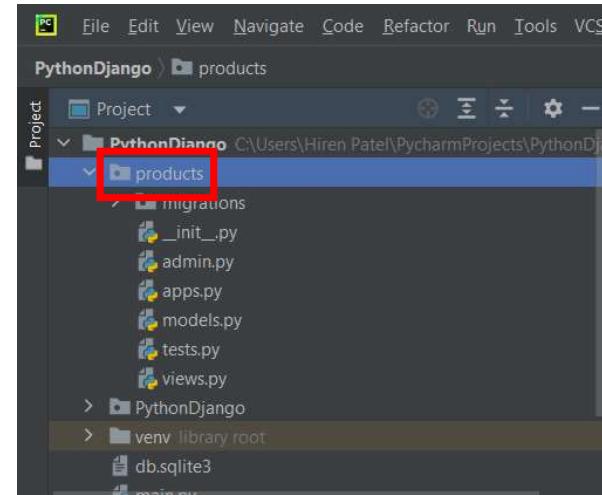


PyCharm interface showing the code editor for `urls.py` in the `products` application of the `PythonDjango` project. The file contains URL patterns for two views: `index` and `new`. The `urls.py` file is highlighted with a red box.

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index),
    path('new', views.new)
```

```
Terminal: C:\Windows\rshell.exe x C:\Windows\rshell.exe (2) x + v
Go to File Ctrl+Shift+F
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py startapp products
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```



Models

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonDjango - models.py. The left sidebar displays the project structure under 'PythonDjango' with a 'products' folder highlighted by a red box. Inside 'products', 'migrations', '_init_.py', 'admin.py', and 'models.py' are listed, with 'models.py' also highlighted by a red box. The main editor window shows the contents of 'models.py':

```
from django.db import models

# Create your models here.

class Product(models.Model):
    name = models.CharField(max_length=255)
    price = models.FloatField()
    stock = models.IntegerField()
    image_url = models.CharField(max_length=2083)
```

```
from django.db import models

class Product(models.Model):
    name = models.CharField(max_length=255)
    price = models.FloatField()
    stock = models.IntegerField()
    image_url = models.CharField(max_length=2083)
```

The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonDjango - db.sqlite3. The left sidebar displays the project structure under 'PythonDjango' with 'db.sqlite3' highlighted by a red box. Other files like 'main.py' and 'manage.py' are also visible. The main editor window shows the contents of 'db.sqlite3':

Google search results for "db server for sqlite" are shown, with the first result being "DB Browser for SQLite".

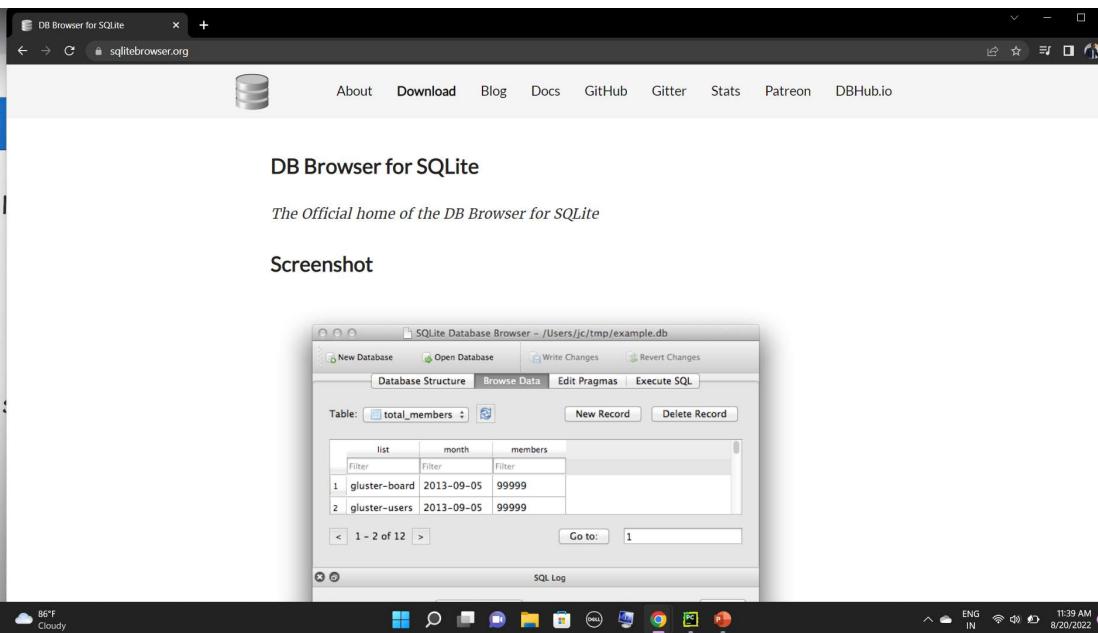
Google search results for "db server for sqlite":

About 1,84,00,000 results (0.44 seconds)

<https://sqlitebrowser.org> :: DB Browser for SQLite

DB Browser for SQLite (DB4S) is a high quality, visual, open source tool to create, design, and edit database files compatible with SQLite.

Downloads



Downloads

(Please consider sponsoring us on Patreon 😊)

Windows

Our latest release (3.12.2) for Windows:

- DB Browser for SQLite – Standard installer for 32-bit Windows
- DB Browser for SQLite – .zip (no installer) for 32-bit Windows
- **DB Browser for SQLite – Standard installer for 64-bit Windows**
- DB Browser for SQLite – .zip (no installer) for 64-bit Windows

Windows PortableApp

- DB Browser for SQLite – PortableApp

Note – If for any reason the standard Windows release does not work (e.g. gives an error), try a nightly build (below).

Downloads

(Please consider sponsoring us on Patreon 😊)

Windows

Our latest release (3.12.2) for Windows:

- [DB Browser for SQLite – Standard installer](#)
- [DB Browser for SQLite – .zip \(no installer\)](#)
- [DB Browser for SQLite – Standard installer for 64-bit Windows](#)
- [DB Browser for SQLite – .zip \(no installer\) for 64-bit Windows](#)

Windows PortableApp

- [DB Browser for SQLite – PortableApp](#)

Note – If for any reason the standard Windows release does not work (e.g. gives an error), try a nightly build (below).

Downloads

(Please consider sponsoring us on Patreon 😊)

Windows

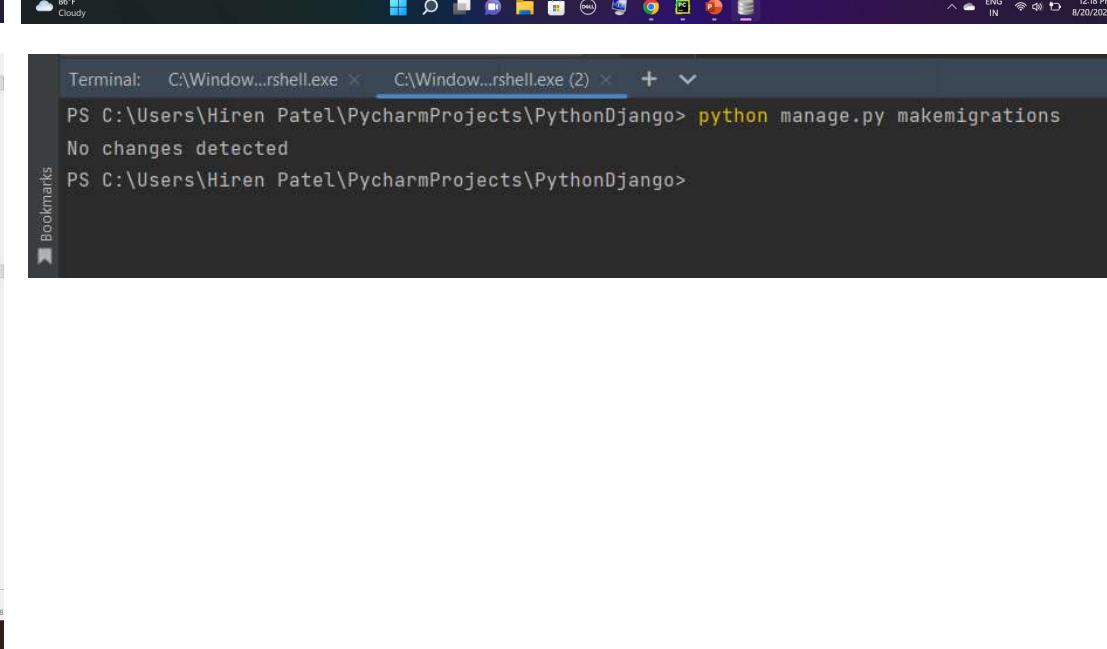
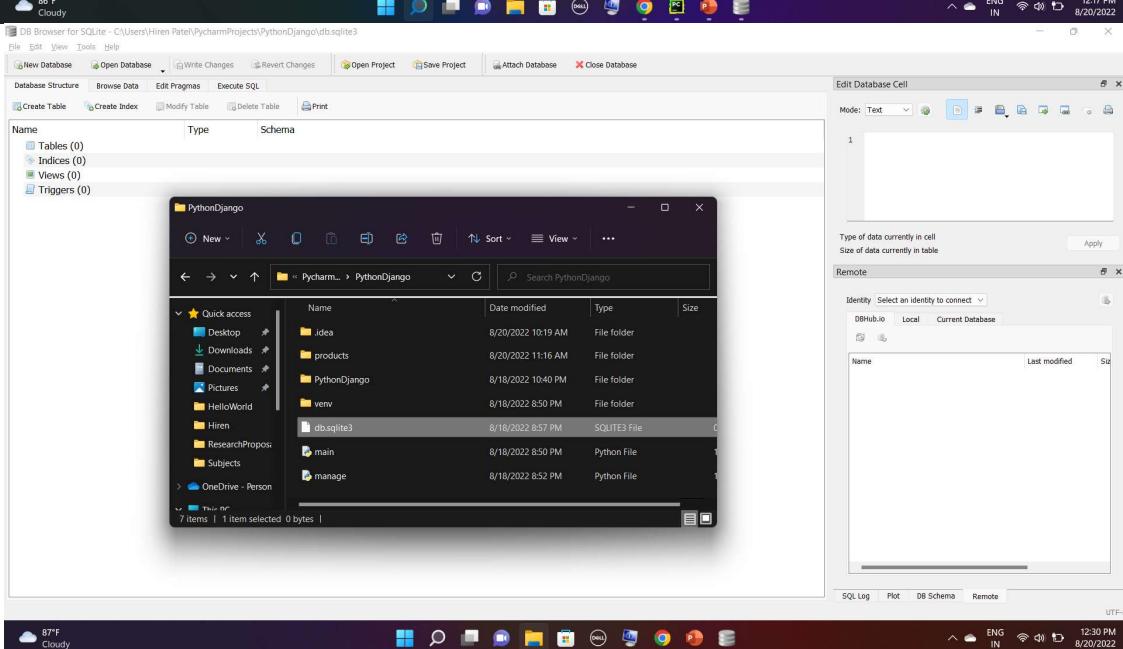
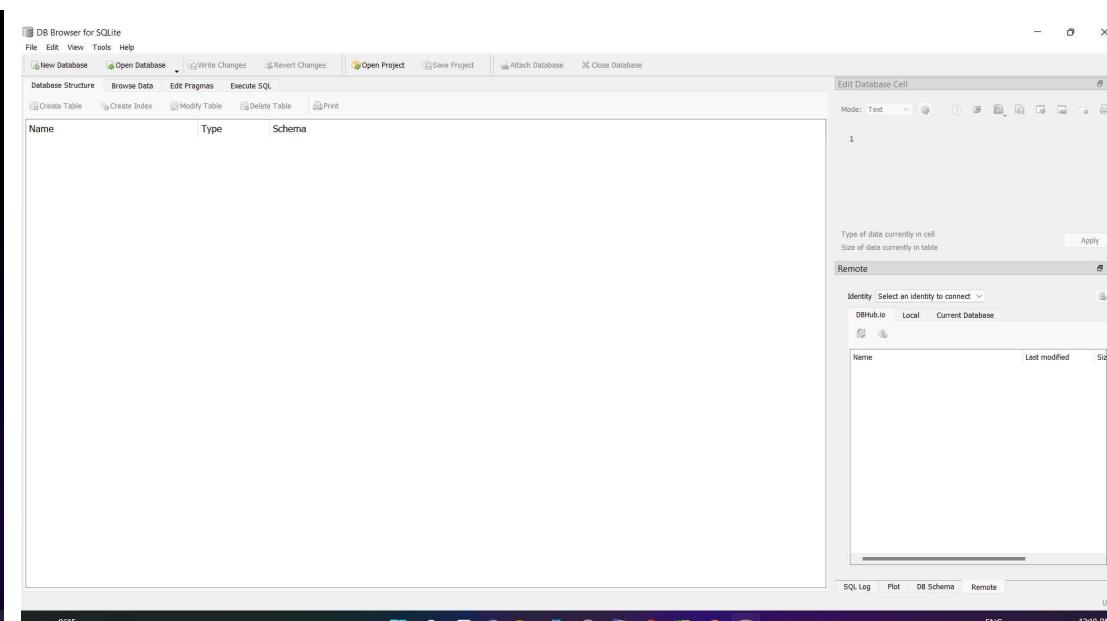
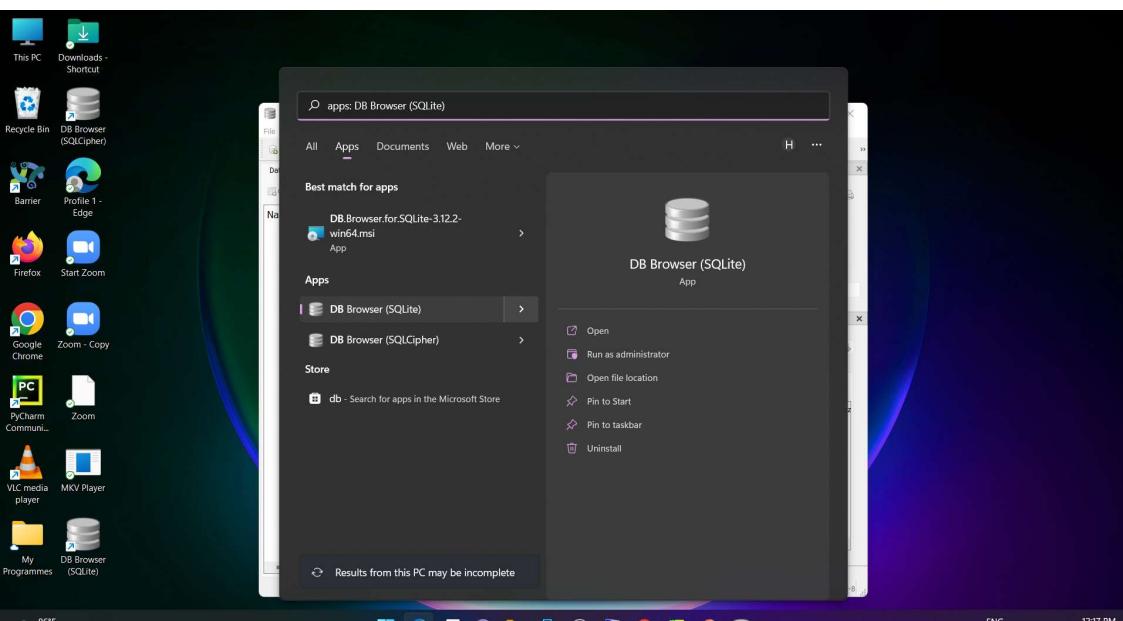
Our latest release (3.12.2) for Windows:

- [DB Browser for SQLite – Standard installer](#)
- [DB Browser for SQLite – .zip \(no installer\)](#)
- [DB Browser for SQLite – Standard installer for 64-bit Windows](#)
- [DB Browser for SQLite – .zip \(no installer\) for 64-bit Windows](#)

Windows PortableApp

- [DB Browser for SQLite – PortableApp](#)

Note – If for any reason the standard Windows release does not work (e.g. gives an error), try a nightly build (below).



File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - settings.py

PythonDjango > PythonDjango > settings.py

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

- products
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py
- PythonDjango
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py
 - wsgi.py
- venv library root
- db.sqlite3

models.py db.sqlite3 settings.py

```
24
25     # SECURITY WARNING: don't run with debug turned on in production!
26     DEBUG = True
27
28     ALLOWED_HOSTS = []
29
30     # Application definition
31
32     INSTALLED_APPS = [
33         'django.contrib.admin',
34         'django.contrib.auth',
35         'django.contrib.contenttypes',
36         'django.contrib.sessions',
37         'django.contrib.messages',
38         'django.contrib.staticfiles',
39     ]
40 ]
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - apps.py

PythonDjango > products > apps.py

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

- products
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py

models.py db.sqlite3 settings.py apps.py

```
1 from django.apps import AppConfig
2
3
4 class ProductsConfig(AppConfig):
5     default_auto_field = 'django.db.models.BigAutoField'
6     name = 'products'
7 ]
```

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'products.apps.ProductsConfig'
]
```

Terminal: C:\Windows\system32\cmd.exe x C:\Windows\system32\cmd.exe (2) x + ▾

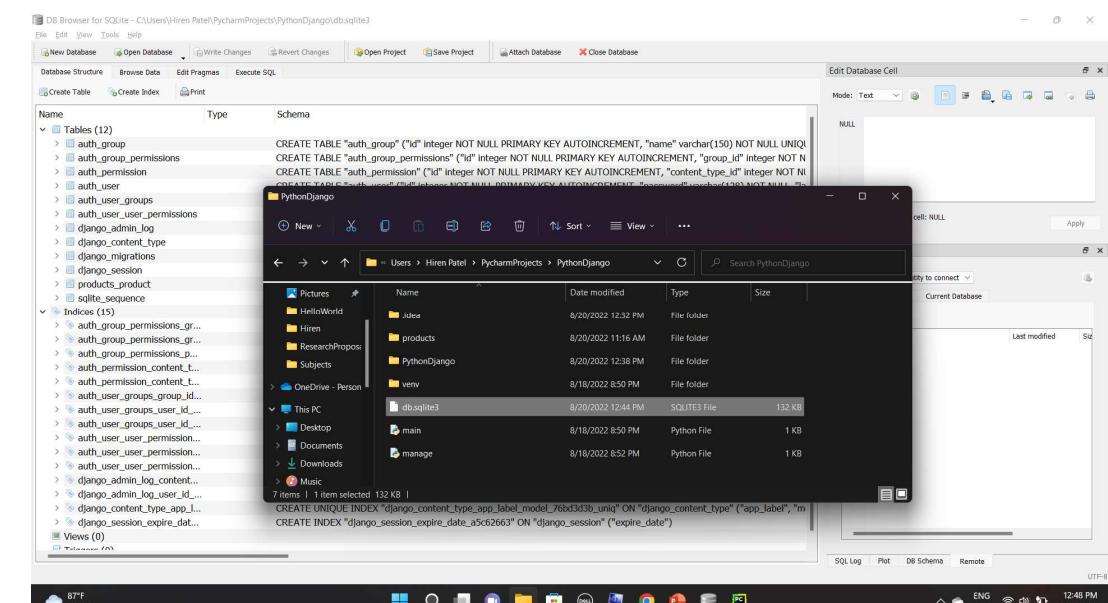
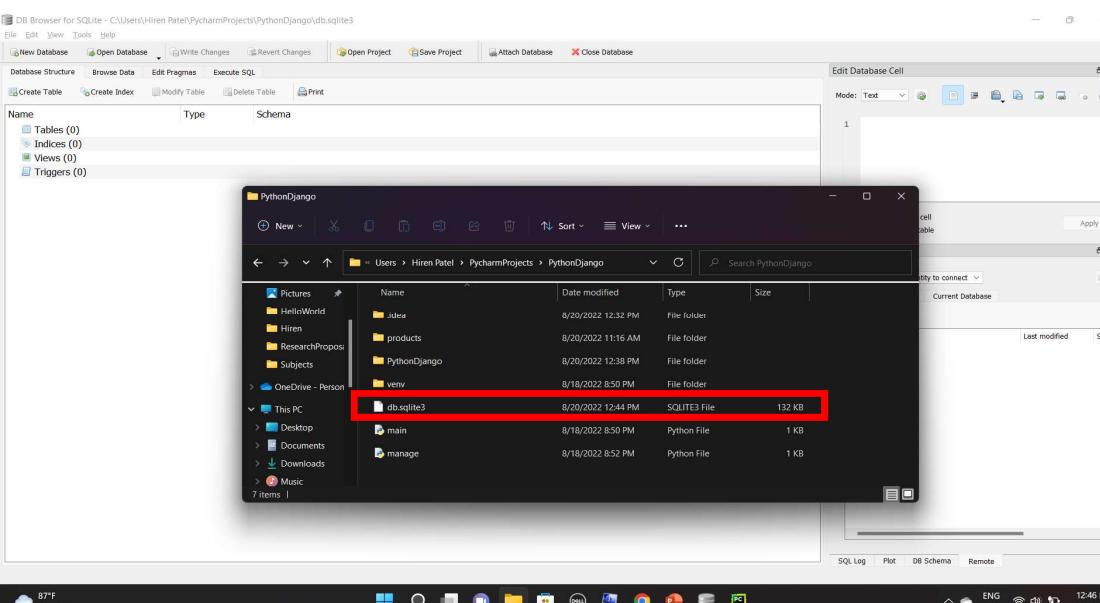
```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py makemigrations
No changes detected
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py makemigrations
Migrations for 'products':
  products\migrations\0001_initial.py
    - Create model Product
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```

The screenshot shows the PyCharm IDE interface with the following details:

- Project View:** On the left, the project structure is displayed. A red box highlights the `migrations` folder within the `products` app directory.
- Code Editor:** The main window shows the `0001_initial.py` migration file. The code defines a single migration class `Migration` with an `initial` value of `True`, no dependencies, and one operation: creating a `Product` model with fields `id`, `name` (max length 255), `price`, `stock`, and `image_url` (max length 2083).
- Terminal:** At the top, there are two terminal tabs labeled `C:\Windows\system32\cmd.exe` and `C:\Windows\system32\cmd.exe (2)`. The second tab is active, showing the command `python manage.py makemigrations` being run twice, resulting in "No changes detected" both times.

Terminal: C:\Windows\rshell.exe x C:\Windows\rshell.exe (2) x + v

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py migrate
Operations to perform:
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying products.0001_initial... OK
  Applying sessions.0001_initial... OK
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```



DB Browser for SQLite - C:\Users\Hiren Patel\PycharmProjects

File Edit View Tools Help

New Database Open Database Write Changes

Database Structure Browse Data Edit Pragmas Execute SQL

Create Table Create Index Print

Name	Type
Tables (12)	
> auth_group	
> auth_group_permissions	
> auth_permission	
> auth_user	
> auth_user_groups	
> auth_user_user_permissions	
> django_admin_log	
> django_content_type	
> django_migrations	
> django_session	
> products_product	
> sqlite_sequence	
Indices (15)	

DB Browser for SQLite - C:\Users\Hiren Patel\PycharmProjects\PythonDjango\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes

Open Project Save Project

Database Structure Browse Data Edit Pragmas Execute SQL

Create Table Create Index Print

Name	Type	Schema
Tables (12)		
> auth_group		CREATE TABLE "auth_group" ("id" integer NOT NULL)
> auth_group_permissions		CREATE TABLE "auth_group_permissions" ("id" integer NOT NULL)
> auth_permission		CREATE TABLE "auth_permission" ("id" integer NOT NULL)
> auth_user		CREATE TABLE "auth_user" ("id" integer NOT NULL)
> auth_user_groups		CREATE TABLE "auth_user_groups" ("id" integer NOT NULL)
> auth_user_user_permissions		CREATE TABLE "auth_user_user_permissions" ("id" integer NOT NULL)
> django_admin_log		CREATE TABLE "django_admin_log" ("id" integer NOT NULL)
> django_content_type		CREATE TABLE "django_content_type" ("id" integer NOT NULL)
> django_migrations		CREATE TABLE "django_migrations" ("id" integer NOT NULL)
> django_session		CREATE TABLE "django_session" ("session_key" varchar(40) NOT NULL, "session_data" blob NOT NULL, "expire_date" datetime NOT NULL)
> products_product		CREATE TABLE "products_product" ("id" integer NOT NULL, "name" varchar(255) NOT NULL, "price" real NOT NULL, "stock" integer NOT NULL, "image_url" varchar(2083) NOT NULL)
> sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py migrate
Operations to perform:
```

```
    Apply all migrations: admin, auth, contenttypes, products, sessions
```

```
Running migrations:
```

```
    No migrations to apply.
```

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```

The screenshot shows the PyCharm IDE interface with a Django project named "PythonDjango".

Project View: The left sidebar shows the project structure. The "products" app contains files: migrations, __init__.py, admin.py, apps.py, and models.py. A red box highlights the "models.py" file.

Code Editor: The main window displays the contents of "models.py". It defines two models: "Product" and "Offer". The "Offer" model has fields: code (CharField, max_length=10), description (CharField, max_length=255), and discount (FloatField). A red box highlights the "Offer" class definition.

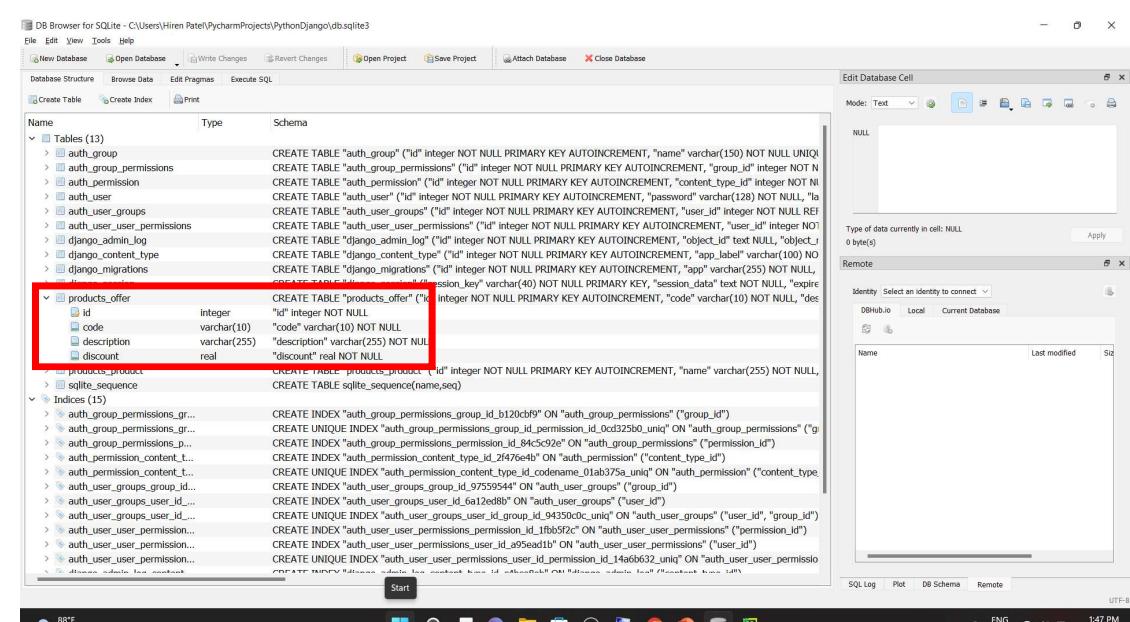
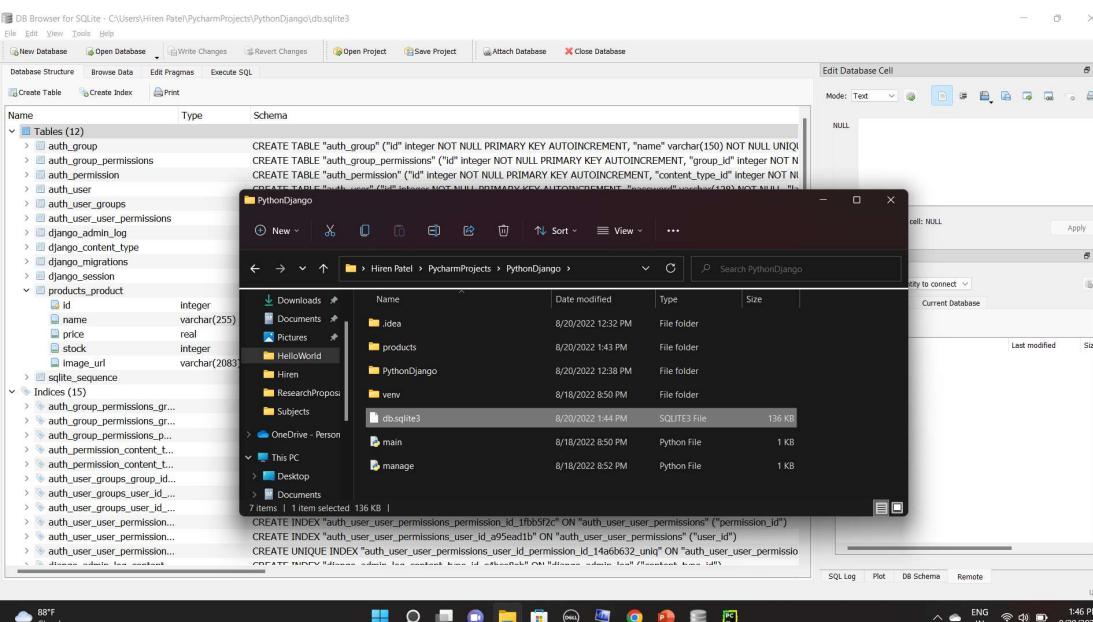
```
class Product(models.Model):
    name = models.CharField(max_length=255)
    price = models.FloatField()
    stock = models.IntegerField()
    image_url = models.CharField(max_length=2083)

class Offer(models.Model):
    code = models.CharField(max_length=10)
    description = models.CharField(max_length=255)
    discount = models.FloatField()
```

Terminal: The bottom panel shows a terminal window with the following command history:

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py makemigrations
Migrations for 'products':
  products\migrations\0002_offer.py
    - Create model Offer
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, products, sessions
Running migrations:
  Applying products.0002_offer... OK
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```

Bottom Status Bar: The status bar at the bottom provides system information: 88°F Cloudy, a taskbar with various icons, and the date/time: 8/20/2022, 1:44 PM.

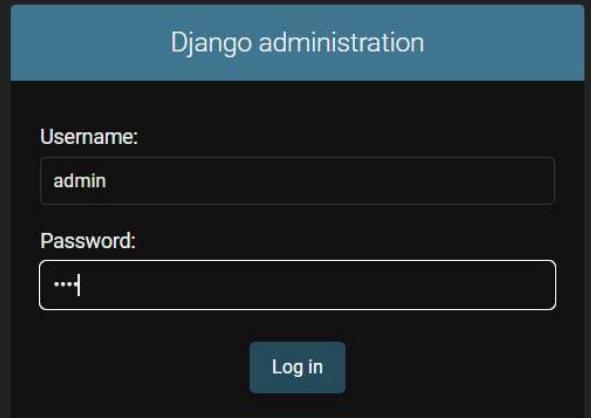


Admin Panel

The screenshot shows a dual-pane interface. The top pane is a web browser window titled "127.0.0.1:8000/admin" displaying the "Django administration" login screen. It features two input fields for "Username" and "Password" and a blue "Log in" button. The bottom pane is a terminal window titled "Terminal" showing the command-line process of creating a superuser using Python's manage.py script.

```
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango> python manage.py createsuperuser
Username (leave blank to use 'hirenpatel'): admin
Email address: hbpatel1976@gmail.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
PS C:\Users\Hiren Patel\PycharmProjects\PythonDjango>
```

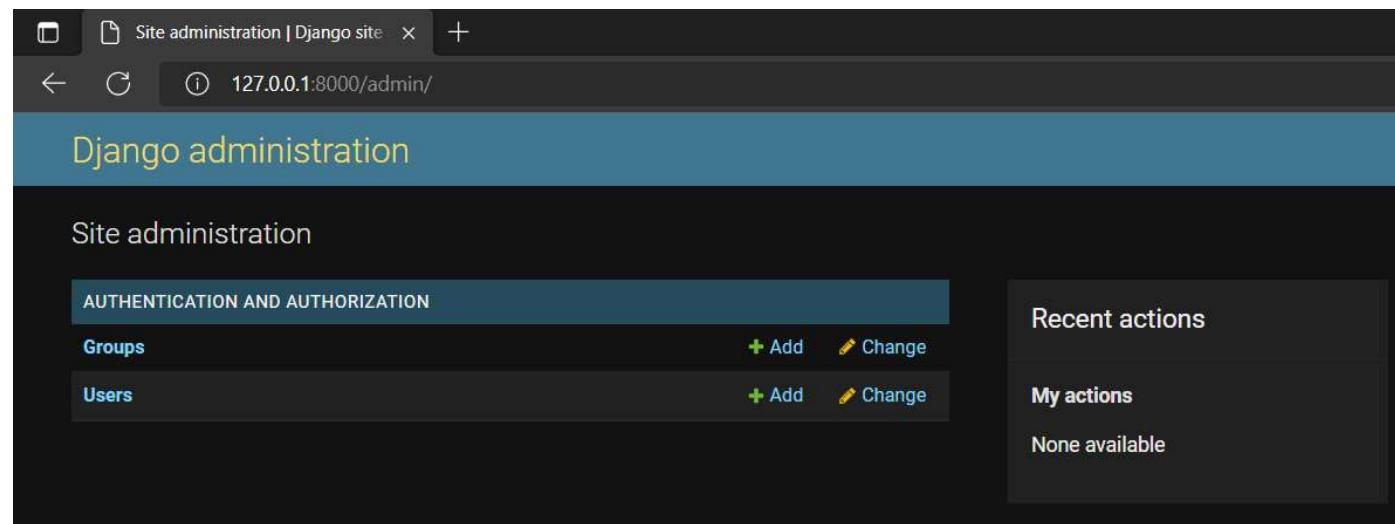
Username: admin
Password: 1234



Django administration

Username:

Password:



Site administration | Django site

127.0.0.1:8000/admin/

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups	<input type="button" value="Add"/> <input type="button" value="Change"/>
Users	<input type="button" value="Add"/> <input type="button" value="Change"/>

Recent actions

My actions

None available

Select user to change | Django si x +

127.0.0.1:8000/admin/auth/user/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Authentication and Authorization › Users

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

ADD USER +

Select user to change

Action: — Go 0 of 1 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	admin	hbpatel1976@gmail.com			<input checked="" type="checkbox"/>

1 user

FILTER

By staff status

All
Yes
No

By superuser status

All
Yes
No

By active

All
Yes
No

88°F Cloudy

ENG IN

2:02 PM 8/20/2022

PC File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - admin.py

PythonDjango > products > admin.py

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango 1
 products
 migrations
 init.py
 admin.py (highlighted with red box)
 apps.py
 models.py
 tests.py
 urls.py
 views.py
 PythonDjango
 init.py

```
1 from django.contrib import admin
2 from .models import Product
3
4 admin.site.register(Product)
5
```

Site administration | Django site +
127.0.0.1:8000/admin/

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups	+ Add	Change
Users	+ Add	Change

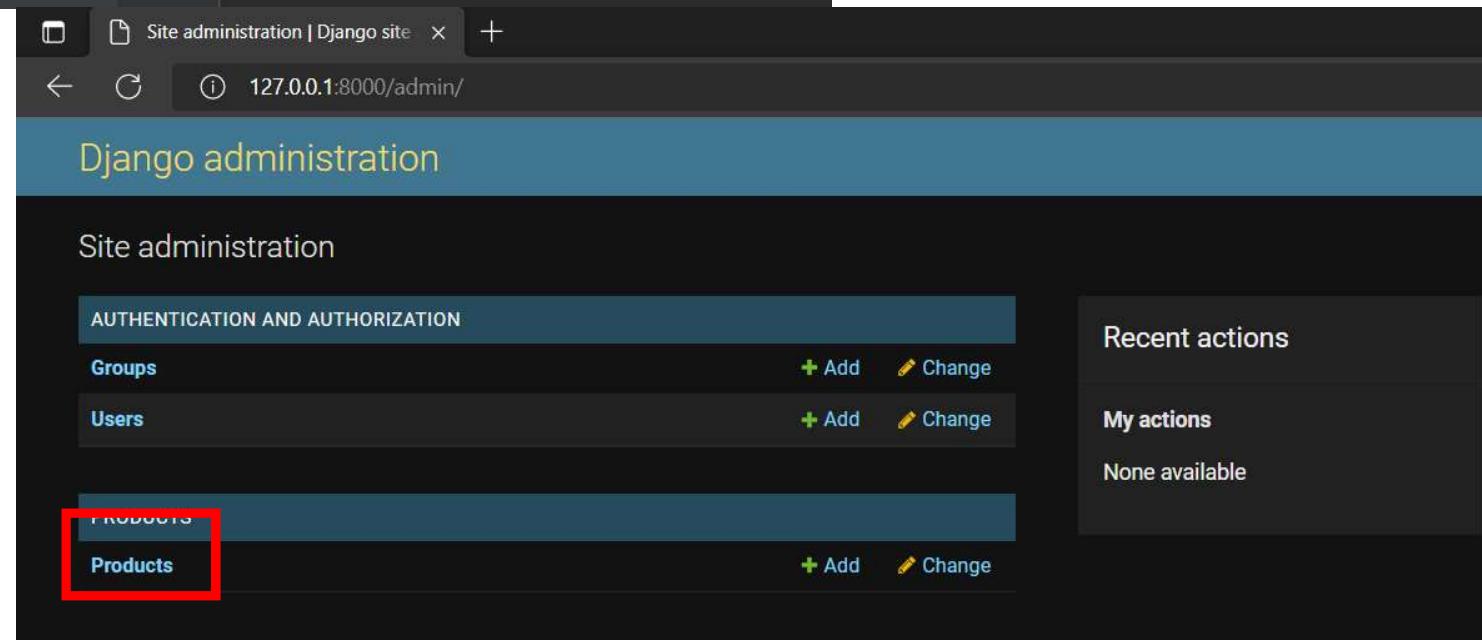
PRODUCTS

Products	+ Add	Change
----------	-------	--------

Recent actions

My actions

None available



Select product to change | Django × +

127.0.0.1:8000/admin/products/product/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Products › Products

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

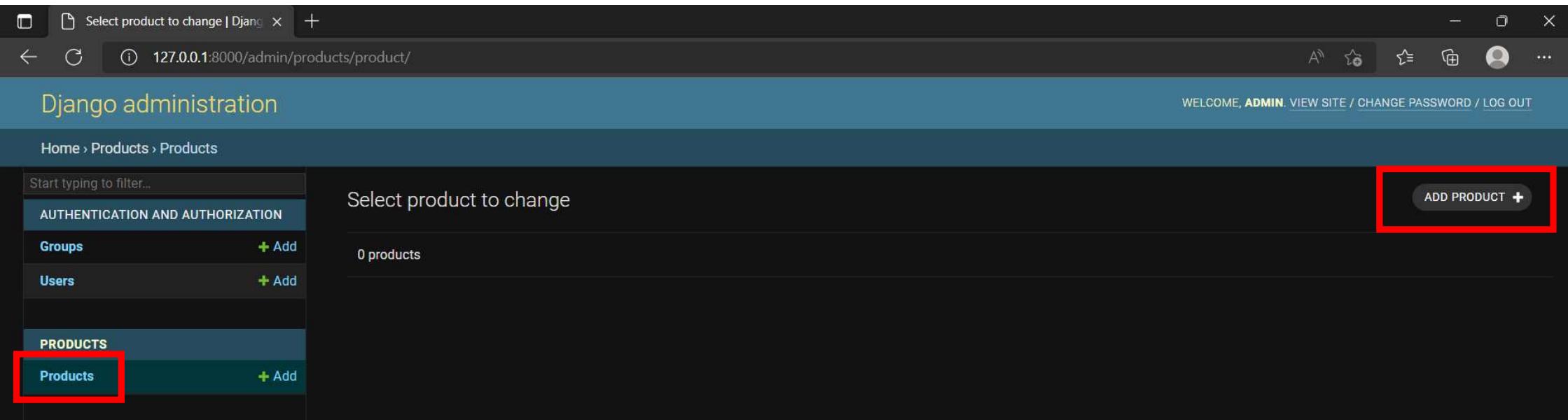
PRODUCTS

- Products + Add

Select product to change

0 products

ADD PRODUCT +



Add product | Django site admin × +

127.0.0.1:8000/admin/products/product/add/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Products › Products › Add product

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

PRODUCTS

- Products + Add

Add product

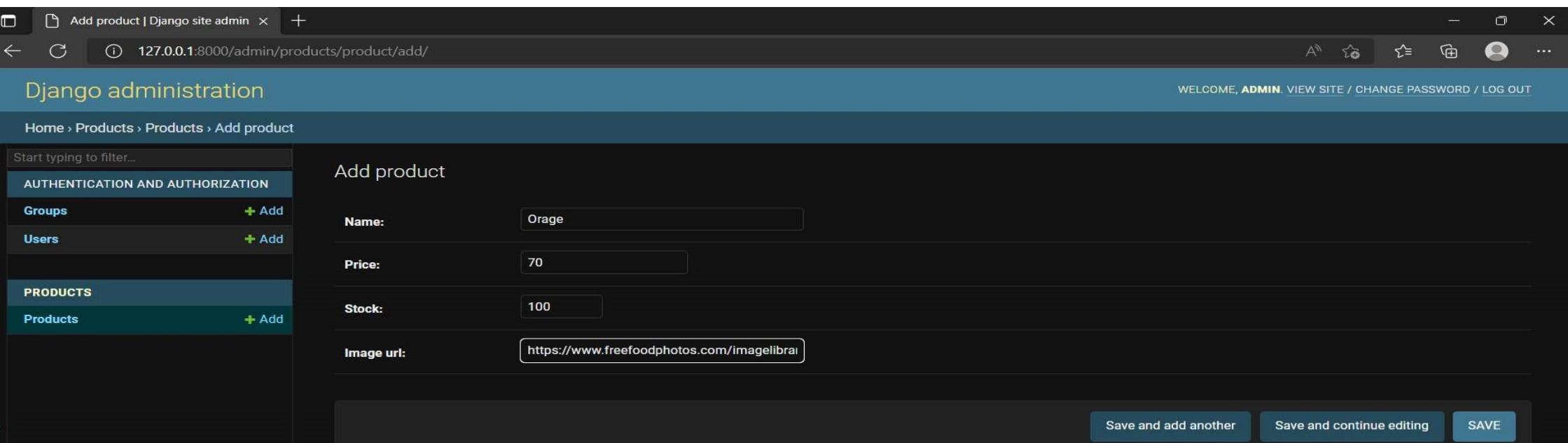
Name:

Price:

Stock:

Image url:

Save and add another **Save and continue editing** **SAVE**



Select product to change | Django +

127.0.0.1:8000/admin/products/product/

Django administration

Welcome, ADMIN. View site / Change password / Log out

Home > Products > Products

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users [+ Add](#)

PRODUCTS

- Products [+ Add](#)

Select product to change

Action: — Go 0 of 1 selected

- PRODUCT
- Product object (1)

1 product

[ADD PRODUCT +](#)

Product object (1) | Change prod +

127.0.0.1:8000/admin/products/product/1/change/

Django administration

Welcome, ADMIN. View site / Change password / Log out

Home > Products > Products > Product object (1)

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users [+ Add](#)

PRODUCTS

- Products [+ Add](#)

Change product

Product object (1)

Name: Orage

Price: 70.0

Stock: 100

Image url: <https://www.freefoodphotos.com/imagelibrary>

Delete

Save and add another

Save and continue editing

SAVE

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - admin.py

PythonDjango > products > admin.py

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

- products
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py
- PythonDjango

```
1 from django.contrib import admin
2 from .models import Product
3
4 class ProductAdmin(admin.ModelAdmin):
5     list_display = ('name', 'price', 'stock')
6
7
8 admin.site.register(Product, ProductAdmin)
```

Select product to change | Django x +

127.0.0.1:8000/admin/products/product/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Products

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

PRODUCTS

Products + Add

Select product to change

Action: — Go 0 of 1 selected

	NAME	PRICE	STOCK
<input type="checkbox"/>	Orage	70.0	100

1 product

ADD PRODUCT +

Select product to change | Django × +

127.0.0.1:8000/admin/products/product/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Products › Products

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

PRODUCTS

Products + Add

The product "Product object (5)" was added successfully.

Select product to change

Action: — Go 0 of 5 selected

	NAME	PRICE	STOCK
<input type="checkbox"/>	Grapes	130.0	30
<input type="checkbox"/>	Apple	120.0	80
<input type="checkbox"/>	Banana	20.0	100
<input type="checkbox"/>	Strawberry	120.0	50
<input type="checkbox"/>	Orage	70.0	100

5 products

ADD PRODUCT +

This screenshot shows the Django Admin interface for managing products. A success message at the top indicates a new product was added. The main area displays a list of five products: Grapes, Apple, Banana, Strawberry, and Orage, each with its name, price, and stock level. An 'ADD PRODUCT +' button is visible in the top right. The left sidebar shows navigation links for authentication, products, and other Django models.

Select product to change | Django × +

127.0.0.1:8000/products/

127.0.0.1:8000/products/

Hello World

This screenshot shows a simple web application running at the URL 127.0.0.1:8000/products/. The page displays the text "Hello World". The browser's address bar also shows the URL 127.0.0.1:8000/products/.

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - admin.py

PythonDjango > products > admin.py

```
Project
PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
  + products
    + migrations
      + __init__.py
      + admin.py
      + apps.py
      + models.py
      + tests.py
      + urls.py
      + views.py
  + PythonDjango
    + __init__.py
    + asgi.py
    + settings.py
    + urls.py
    + wsgi.py

PythonDjango > products > admin.py
```

settings.py admin.py 0001_initial.py models.py app

```
1 from django.contrib import admin
2 from .models import Product, Offer
3
4
5 class OfferAdmin(admin.ModelAdmin):
6     list_display = ('code', 'discount')
7
8
9 class ProductAdmin(admin.ModelAdmin):
10    list_display = ('name', 'price', 'stock')
11
12
13 admin.site.register(Offer, OfferAdmin)
14 admin.site.register(Product, ProductAdmin)
```

Site administration | Django site 127.0.0.1:8000/products/ 127.0.0.1:8000/admin/

Django administration

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups + Add Change

Users + Add Change

PRODUCTS

Offers + Add Change

Products + Add Change

Select offer to change | Django s 127.0.0.1:8000/products/ 127.0.0.1:8000/admin/products/offer/

127.0.0.1:8000/admin/products/offer/

Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Products > Offers

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

PRODUCTS

Offers + Add

Products + Add

The offer "Offer object (1)" was changed successfully.

Select offer to change

Action: — Go 0 of 2 selected

	CODE	DISCOUNT
<input type="checkbox"/>	DIWALI15	0.15
<input type="checkbox"/>	INDEPEN10	0.1

2 offers

ADD OFFER +

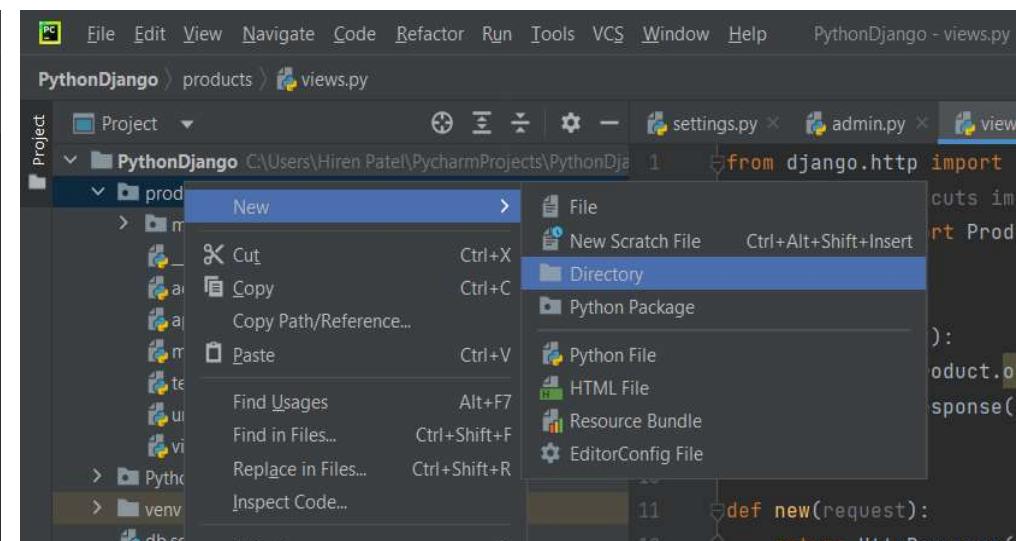
File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - views.py

PythonDjango > products > views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render
from .models import Product

def index(request):
    products = Product.objects.all()
    return HttpResponseRedirect('Hello World')

def new(request):
    return HttpResponseRedirect('New Products')
```



File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - views.py

PythonDjango > products > views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render
from .models import Product

def index(request):
    products = Product.objects.all()
    return HttpResponseRedirect('Hello World')

def new(request):
    return HttpResponseRedirect('New Products')
```

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

products

migrations

__init__.py

admin.py

apps.py

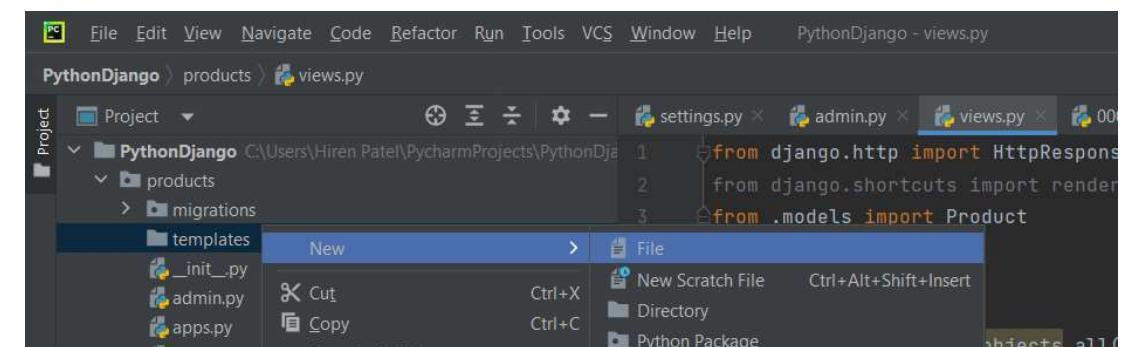
templates

New Directory

index.html

```
def index(request):
    products = Product.objects.all()
    return HttpResponseRedirect('Hello World')

def new(request):
    return HttpResponseRedirect('New Products')
```



File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - views.py

PythonDjango > products > views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render
from .models import Product

def index(request):
    products = Product.objects.all()
    return HttpResponseRedirect('Hello World')

def new(request):
    return HttpResponseRedirect('New Products')
```

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

products

migrations

templates

__init__.py

admin.py

apps.py

New File

index.html

```
def index(request):
    products = Product.objects.all()
    return HttpResponseRedirect('Hello World')

def new(request):
    return HttpResponseRedirect('New Products')
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - index.html

PythonDjango > products > templates > index.html

Project

```
PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
  products
    migrations
    templates
      index.html
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
```

index.html

```
<h1>Products</h1>
<ul>
  <li> Item 1 </li>
  <li> Item 2 </li>
  <li> Item 2 </li>
  <li> Item 4 </li>
</ul>
```

views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render
from .models import Product

def index(request):
    products = Product.objects.all()
    return render(request, 'index.html')

def new(request):
    return HttpResponseRedirect('New Products')
```

Select offer to change | Django s X 127.0.0.1:8000/products/ +

← ⌂ ① 127.0.0.1:8000/products/

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - views.py

PythonDjango > products > views.py

Project

```
PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
  products
    migrations
    templates
      index.html
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      urls.py
      views.py
  PythonDjango
  manage.py
```

views.py

```
from django.http import HttpResponseRedirect
from django.shortcuts import render
from .models import Product

def index(request):
    products = Product.objects.all()
    return render(request, 'index.html')

def new(request):
    return HttpResponseRedirect('New Products')
```

Products

- Item 1
- Item 2
- Item 3
- Item 4

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - views.py

PythonDjango > products > views.py

```
Project
PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
  + products
    + migrations
    + templates
      + index.html
      + __init__.py
      + admin.py
      + apps.py
      + models.py
      + tests.py
      + urls.py
      + views.py
    + PythonDjango
    + venv\library root
      + db.sqlite3
      + main.py

index.html x views.py x Press Esc to exit full screen
from django.http import HttpResponseRedirect
from django.shortcuts import render
from .models import Product

def index(request):
    products = Product.objects.all()
    return render(request,
                  'index.html',
                  {'products': products})

def new(request):
    return HttpResponseRedirect('New Products')
```

Select offer to change | Django s x 127.0.0.1:8000/products/

← ⌂ ⌂ 127.0.0.1:8000/products/

Products

- Orage
- Strawberry
- Banana
- Apple
- Grapes

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - index.html

PythonDjango > products > templates > index.html

```
Project
PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango
  + products
    + migrations
    + templates
      + index.html
      + __init__.py
      + admin.py
      + apps.py
      + models.py

index.html x views.py x
<h1>Products</h1>
<ul>
    {% for product in products %}
        <li> {{ product.name }} </li>
    {% endfor %}
</ul>
```

The screenshot shows the PyCharm IDE interface. At the top, the menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help, and PythonDjango - index.html. The Project tool window on the left shows the PythonDjango project structure with files like migrations, templates (index.html), __init__.py, admin.py, and apps.py. The main editor window displays the contents of index.html:

```
<h1>Products</h1>
<ul>
    {% for product in products %}
        <li>{{ product.name }} (Rs. {{ product.price }})</li>
    {% endfor %}
</ul>
```

The screenshot shows a web browser window with the URL 127.0.0.1:8000/products/. The page title is "Products". The content is a list of fruits with their prices:

- Orage (Rs. 70.0)
- Strawberry (Rs. 120.0)
- Banana (Rs. 20.0)
- Apple (Rs. 120.0)
- Grapes (Rs. 130.0)

The screenshot shows a Google search results page for the query "bootstrap framework". The search bar contains "bootstrap framework". Below it, there are filters for All, Images, Videos, News, Shopping, and More. The text "About 8,86,00,000 results (0.50 seconds)" is displayed. A snippet from the Bootstrap website is shown, featuring the Bootstrap logo and navigation links: Docs (highlighted with a red box), Examples, Icons, Themes, and Blog.

The screenshot shows the Bootstrap website homepage at getbootstrap.com. The header features the Bootstrap logo and navigation links: Docs (highlighted with a red box), Examples, Icons, Themes, and Blog. A yellow callout box in the top right corner says "New in v5.2 CSS variables, resp...".

B Get started with Bootstrap · Boot x +

getbootstrap.com/docs/5.2/getting-started/introduction/

Docs Examples Icons Themes Blog Search CTRL K v5.2

Getting started

Introduction

- Download
- Contents
- Browsers & devices
- JavaScript
- Webpack
- Parcel
- Vite
- Accessibility
- RFS
- RTL
- Contribute

Customize

- Overview
- Sass
- Options
- Color
- Components
- CSS variables
- Optimize

Get started with Bootstrap

[View on GitHub](#)

Bootstrap is a powerful, feature-packed frontend toolkit. Build anything—from prototype to production—in minutes.



Join us for demos, and best practices to modernize, unify, and innovate with data - Register now

ads via Carbon

On this page

- Quick start
- CDN links
- Next steps
- JS components
- Important globals
 - HTML5 doctype
 - Responsive meta tag
 - Box-sizing
 - Reboot
- Community

Quick start

Get started by including Bootstrap's production-ready CSS and JavaScript via CDN without the need for any build steps. See it in practice with this [Bootstrap CodePen demo](#).

1. Create a new `index.html` file in your project root. Include the `<meta name="viewport">` tag as well for proper responsive behavior in mobile devices.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
```



B Get started with Bootstrap · Boot x +

getbootstrap.com/docs/5.2/getting-started/introduction/ 🔍 ⭐ ⏺ 🌐 v5.2

Docs Examples Icons Themes Blog Search CTRL K

Getting started

Introduction

Download

Contents

Browsers & devices

JavaScript

Webpack

Parcel

Vite

Accessibility

RFS

RTL

Contribute

Customize

Overview

Sass

Options

Color

Components

CSS variables

Optimize

Quick start

Get started by including Bootstrap's production-ready CSS and JavaScript via CDN without the need for any build steps. See it in practice with this [Bootstrap CodePen demo](#).

1. Create a new `index.html` file in your project root. Include the `<meta name="viewport">` tag as well for proper responsive behavior in mobile devices.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
  </body>
</html>
```

Copy to clipboard

On this page

Quick start

CDN links

Next steps

JS components

Important globals

HTML5 doctype

Responsive meta tag

Box-sizing

Reboot

Community

90°F Cloudy

ENG IN

3:35 PM 8/20/2022

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - index.htm

PythonDjango > products > templates

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

- products
- migrations
- templates
 - New
 - File

index.html

index.html

views.py

1 <h1>Products</h1>

2

3 {% for product in products %}

Cut Ctrl+X

Copy Ctrl+C

Copy Path/Reference...

Paste Ctrl+V

Find Usages Alt+F7

Find in Files... Ctrl+Shift+F

New Scratch File Ctrl+Alt+Shift+Insert

Directory

Python Package

Python File

HTML File

Resource Bundle

EditorConfig File

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - index.htm

PythonDjango > products > templates

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

- products
- migrations
- templates
 - index.html
 - base.html
 - views.py
 - admin.py
 - apps.py
 - models.py

index.html

index.html

views.py

1 <h1>Product</h1>

2

3 {% for product in products %}

4

New File

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - base.html

PythonDjango > products > templates > base.html

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

- products
- migrations
- templates
 - base.html
 - index.html

base.html

base.html

views.py

1 <!doctype html>

2 <html lang="en">

3 <head>

4 <meta charset="utf-8">

5 <meta name="viewport" content="width=device-width, initial-scale=1">

6 <title>Bootstrap demo</title>

7 </head>

8 <body>

9 <h1>Hello, world!</h1>

10 </body>

11 </html>

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - base.html

PythonDjango > products > templates > base.html

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

- products
 - migrations
 - templates
 - base.html
 - index.html
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py

> PythonDjango

index.html x base.html x views.py x

```
1  <!doctype html>
2  <html lang="en">
3      <head>
4          <meta charset="utf-8">
5          <meta name="viewport" content="width=device-width, initial-scale=1">
6          <title>Bootstrap demo</title>
7      </head>
8      <body>
9          {% block content %}
10         {% endblock %}
11     </body>
12 </html>
```

File Edit View Navigate Code Refactor Run Tools VCS Window Help PythonDjango - index.html

PythonDjango > products > templates > index.html

Project

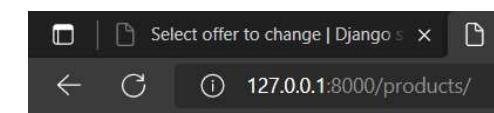
PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

- products
 - migrations
 - templates
 - base.html
 - index.html
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py

index.html x base.html x views.py x

```
1  {% extends 'base.html' %}

2  {% block content %}
3      <h1>Products</h1>
4      <ul>
5          {% for product in products %}
6              <li> {{ product.name }} (Rs. {{ product.price }}) </li>
7          {% endfor %}
8      </ul>
9  {% endblock %}
```



Products

- Orage (Rs. 70.0)
- Strawberry (Rs. 120.0)
- Banana (Rs. 20.0)
- Apple (Rs. 120.0)
- Grapes (Rs. 130.0)

B Cards · Bootstrap v5.2

getbootstrap.com/docs/5.2/components/card/

Docs Examples Icons Themes Blog

Search CTRL K

v5.2

Floating labels

Layout

Validation

Components

Accordion

Alerts

Badge

Breadcrumb

Buttons

Button group

Card

Carousel

Close button

Collapse

Dropdowns

List group

Modal

Navbar

Navs & tabs

Offcanvas

Pagination

Placeholders

Cards

[View on GitHub](#)

Bootstrap's cards provide a flexible and extensible content container with multiple variants and options.



Join us for AWS Innovate and learn to scale your applications and innovate with AWS Global Infrastructure

ads via Carbon

About

A **card** is a flexible and extensible content container. It includes options for headers and footers, a wide variety of content, contextual background colors, and powerful display options. If you're familiar with Bootstrap 3, cards replace our old panels, wells, and thumbnails. Similar functionality to those components is available as modifier classes for cards.

Example

Cards are built with as little markup and styles as possible, but still manage to deliver a ton of control and customization. Built with flexbox, they offer easy alignment and mix well with other Bootstrap components. They have no `margin` by default, so use `spacing utilities` as needed.

On this page

About

Example

Content types

Body

Titles, text, and links

Images

List groups

Kitchen sink

Header and footer

Sizing

Using grid markup

Using utilities

Using custom CSS

Text alignment

Navigation

Images

Image caps

Image overlays

Horizontal

Card styles

Background and color

Border

Mixins utilities

Card layout

HTML

Copy to clipboard

```
<div class="card" style="width: 18rem;">
  
  <div class="card-body">
    <h5 class="card-title">Card title</h5>
    <p class="card-text">Some quick example text to build on the card title and make up the bulk of
      <a href="#" class="btn btn-primary">Go somewhere</a>
    </div>
</div>
```

Project

PythonDjango C:\Users\Hiren Patel\PycharmProjects\PythonDjango

- products
 - migrations
 - templates
 - base.html
 - index.html
- __init__.py
- admin.py
- apps.py
- models.py
- tests.py
- urls.py
- views.py

index.html

```
1  {% extends 'base.html' %} 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

Notifications



Products



Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

[Go somewhere](#)



Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

[Go somewhere](#)



Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

[Go somewhere](#)

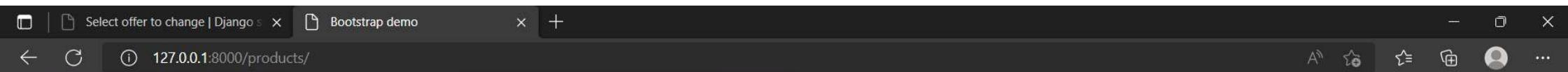


Card title

Some quick example text to build on the card title and make up the bulk of the card's content.

```
{% extends 'base.html' %}

{% block content %}
    <h1>Products</h1>
    <div class="row">
        {% for product in products %}
            <div class="col">
                <div class="card" style="width: 18rem;">
                    
                    <div class="card-body">
                        <h5 class="card-title"> {{ product.name }} </h5>
                        <p class="card-text"> Rs. {{ product.price }} </p>
                        <a href="#" class="btn btn-primary"> Add to Card </a>
                    </div>
                </div>
            </div>
        {% endfor %}
    </div>
    {% endblock %}
```



Orage

Rs. 70.0

[Add to Cart](#)



90°F
Cloudy



4:21 PM
8/20/2022

B Navbar · Bootstrap v5.2

getbootstrap.com/docs/5.2/components/navbar/

Docs Examples Icons Themes Blog

Search CTRL K

v5.2

Badge
Breadcrumb
Buttons
Button group
Card
Carousel
Close button
Collapse
Dropdowns
List group
Modal
Navbar
Navs & tabs
Offcanvas
Pagination
Placeholders
Popovers
Progress
Scrollspy
Spinners
Toasts
Toolips

Brand

The `.navbar-brand` can be applied to most elements, but an anchor works best, as some elements might require utility classes or custom styles.

Text

Add your text within an element with the `.navbar-brand` class.

Navbar

Navbar

HTML

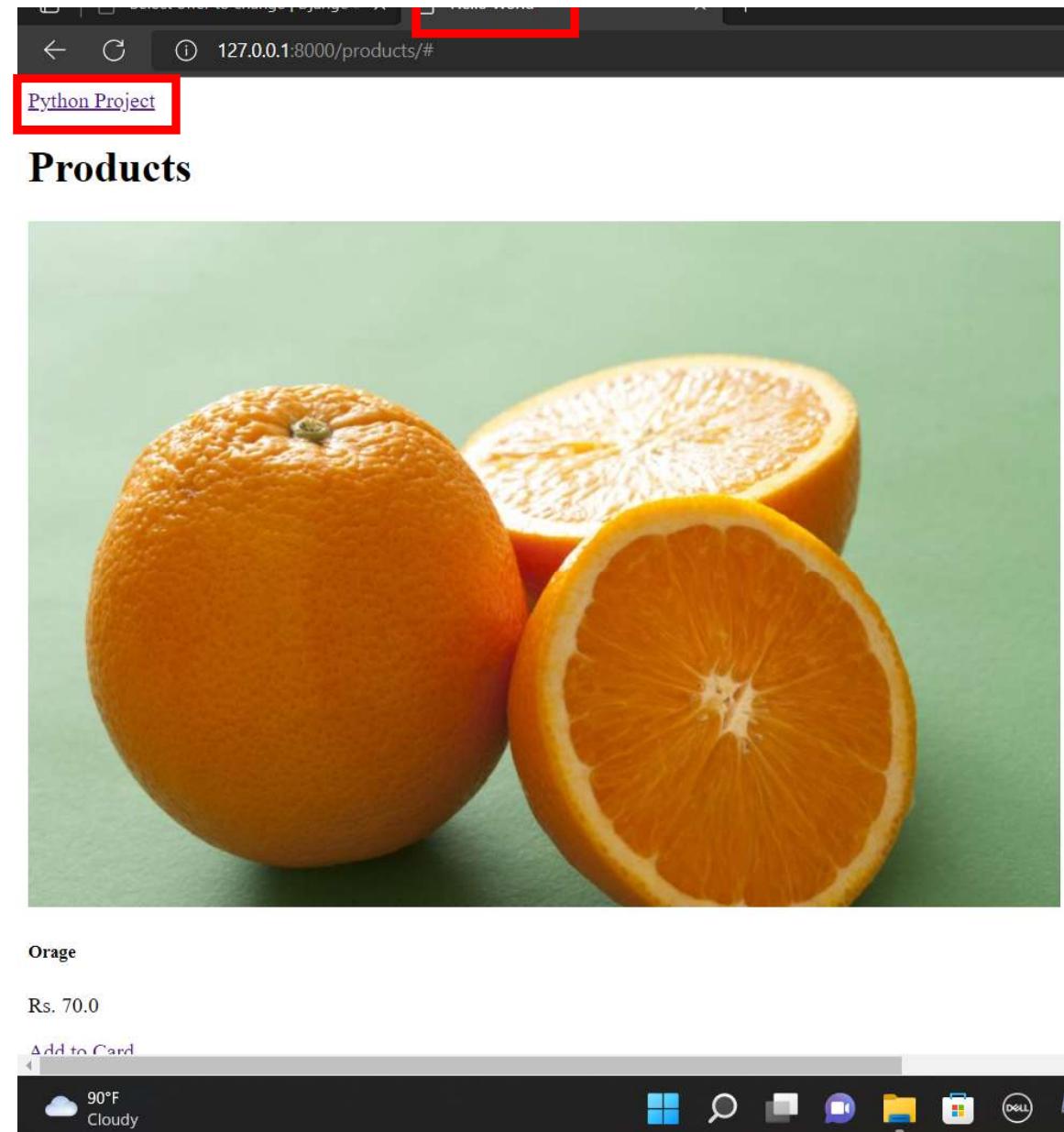
```
<!-- As a link -->
<nav class="navbar bg-light">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
  </div>
</nav>

<!-- As a heading -->
<nav class="navbar bg-light">
  <div class="container-fluid">
    <span class="navbar-brand mb-0 h1">Navbar</span>
  </div>
```

On this page

How it works
Supported content
Brand
Text
Image
Image and text
Nav
Forms
Text
Color schemes
Containers
Placement
Scrolling
Responsive behaviors
Toggler
External content
Offcanvas
CSS
Variables
Sass variables
Sass loop

```
ex.html x base.html x
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Hello World</title>
  </head>
  <body>
    <!-- As a link -->
    <nav class="navbar bg-light">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">Python Project</a>
      </div>
    </nav>
    {% block content %}
    {% endblock %}
  </body>
</html>
```



Python Libraries

Panda, SciPy, Matplotlib, Seaborn, NumPy

Using colab
(Source: Edureka: <https://youtu.be/-6RqxhNO2yY>)

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Table of contents

- Getting started
- Data science
- Machine learning
- More Resources
- Featured examples

+ Code + Text Copy to Drive

Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

New notebook

Open notebook Ctrl+O

Upload notebook

Rename

Save a copy in Drive

Save a copy as a GitHub Gist

+ Code + Text Copy

Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.

Welcome To Colaboratory

Untitled1.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

Play

{x}

Welcome To Colaboratory

panda.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

Play

{x}

panda.ipynb - Colaboratory

colab.research.google.com/drive/1nPBF6WYPv8KQhzAeYe2i3wDnYXSDJEg

panda.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

RAM Disk Editing

```
import pandas as pd

my_web = {'Day':[1,2,3,4,5,6], 'Site_Visitors':[1000,500,2000,1500,800,2100], 'Bounce_Rate': [10,20,30,35,25,5]}

df = pd.DataFrame(my_web)

print(df)
```

	Day	Site_Visitors	Bounce_Rate
0	1	1000	10
1	2	500	20
2	3	2000	30
3	4	1500	35
4	5	800	25
5	6	2100	5

```
import pandas as pd

my_web = {'Day':[1,2,3,4,5,6], 'Site_Visitors':[1000,500,2000,1500,800,2100], 'Bounce_Rate': [10,20,30,35,25,5]}

df = pd.DataFrame(my_web)

print(df)
```

Slicing

panda.ipynb - Colaboratory x +

colab.research.google.com/drive/1nPBF6WYPv8KQhzAeYe2i3wDnYXSDJEg

CO panda.ipynb ☆

File Edit View Insert Runtime Tools Help

+ Code + Text

Q {x} import pandas as pd

{x} my_web = {'Day':[1,2,3,4,5,6], 'Site_Visitors':[1000,500,2000,1500,800,2100], 'Bounce_Rate': [10,20,30,35,25,5]}

df = pd.DataFrame(my_web)

print(df.head(2))

	Day	Site_Visitors	Bounce_Rate
0	1	1000	10
1	2	500	20

panda.ipynb - Colaboratory x +

colab.research.google.com/drive/1nPBF6WYPv8KQhzAeYe2i3wDnYXSDJEg

CO panda.ipynb ☆

All changes saved

File Edit View Insert Runtime Tools Help

+ Code + Text

Q {x} import pandas as pd

{x} my_web = {'Day':[1,2,3,4,5,6], 'Site_Visitors':[1000,500,2000,1500,800,2100], 'Bounce_Rate': [10,20,30,35,25,5]}

df = pd.DataFrame(my_web)

print(df.tail(2))

	Day	Site_Visitors	Bounce_Rate
4	5	800	25
5	6	2100	5

Merging

panda.ipynb - Colaboratory +

colab.research.google.com/drive/1nPBF6WYPv8KQhzAeYe2i3wDnYXSDJEg#scrollTo=IphVaNTWdBeI

panda.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

import pandas as pd

{x} df1 = pd.DataFrame({'House_Price_Index':[55,60,70,65], 'Interest_Rate':[2.5,3.5,4.5,4.0], 'India_GDP':[37,42,35,49]}, index = [2001, 2002, 2003, 2004])

df2 = pd.DataFrame({'House_Price_Index':[55,60,70,65], 'Interest_Rate':[2.5,3.5,4.5,4.0], 'India_GDP':[37,42,35,49]}, index = [2005, 2006, 2007, 2008])

merge = pd.merge(df1, df2)

print(merge)

	House_Price_Index	Interest_Rate	India_GDP
0	55	2.5	37
1	60	3.5	42
2	70	4.5	35
3	65	4.0	49

```
import pandas as pd

df1 = pd.DataFrame({'House_Price_Index':[55,60,70,65], 'Interest_Rate':[2.5,3.5,4.5,4.0], 'India_GDP':[37,42,35,49]}, index = [2001, 2002, 2003, 2004])

df2 = pd.DataFrame({'House_Price_Index':[55,60,70,65], 'Interest_Rate':[2.5,3.5,4.5,4.0], 'India_GDP':[37,42,35,49]}, index = [2005, 2006, 2007, 2008])

merge = pd.merge(df1, df2)

print(merge)
```

panda.ipynb - Colaboratory +

colab.research.google.com/drive/1nPBF6WYPv8KQhzAeYe2i3wDnYXSDJEg#scrollTo=IphVaNTWdBel

CO **panda.ipynb** ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Q ✓

{x}

□

```
import pandas as pd

df1 = pd.DataFrame({'House_Price_Index':[55,60,70,65], 'Interest_Rate':[2.5,3.5,4.5,4.0], 'India_GDP':[37,42,35,49]}, index = [2001, 2002, 2003, 2004])

df2 = pd.DataFrame({'House_Price_Index':[55,60,70,65], 'Interest_Rate':[2.5,3.5,4.5,4.0], 'India_GDP':[37,42,35,49]}, index = [2005, 2006, 2007, 2008])

merge = pd.merge(df1, df2, on = "House_Price_Index")

print(merge)
```

	House_Price_Index	Interest_Rate_x	India_GDP_x	Interest_Rate_y	India_GDP_y
0	55	2.5	37	2.5	37
1	60	3.5	42	3.5	42
2	70	4.5	35	4.5	35
3	65	4.0	49	4.0	49

Joining

panda.ipynb - Colaboratory +

colab.research.google.com/drive/1nPBF6WYPv8KQhzAeYe2i3wDnYXSDJEg#scrollTo=IphVaNTWdBeI

panda.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

import pandas as pd

df1 = pd.DataFrame({'House_Price_Index':[55,60,70,65], 'Housing_Interest_Rate':[2.5,3.5,4.5,4.0], 'USA_GDP':[37,42,35,49]}, index = [2001, 2002, 2003, 2004])

df2 = pd.DataFrame({'Sensex':[5500,6000,7000,6500], 'Personal_Interest_Rate':[2.5,3.5,4.5,4.0], 'India_GDP':[37,42,35,49]}, index = [2001, 2003, 2004, 2005])

joined = df1.join(df2)

print(joined)

	House_Price_Index	Housing_Interest_Rate	USA_GDP	Sensex	India_GDP
2001	55	2.5	37	5500.0	37.0
2002	60	3.5	42	NaN	NaN
2003	70	4.5	35	6000.0	42.0
2004	65	4.0	49	7000.0	35.0

```
import pandas as pd

df1 = pd.DataFrame({'House_Price_Index':[55,60,70,65], 'Housing_Interest_Rate':[2.5,3.5,4.5,4.0], 'USA_GDP':[37,42,35,49]},  
                   index = [2001, 2002, 2003, 2004])  
df2 = pd.DataFrame({'Sensex':[5500,6000,7000,6500], 'Personal_Interest_Rate':[2.5,3.5,4.5,4.0], 'India_GDP':[37,42,35,49]},  
                   index = [2001, 2003, 2004, 2005])  
joined = df1.join(df2)

print(joined)
```

panda.ipynb - Colaboratory +

colab.research.google.com/drive/1nPBF6WYPv8KQhzAeYe2i3wDnYXSDJEg#scrollTo=IphVaNTWdBeI

panda.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

import pandas as pd

{x} df1 = pd.DataFrame({'House_Price_Index':[55,60,70,65], 'Housing_Interest_Rate':[2.5,3.5,4.5,4.0], 'USA_GDP':[37,42,35,49]}, index = [2001, 2002, 2003, 2004])

df2 = pd.DataFrame({'Sensex':[5500,6000,7000,6500], 'Personal_Interest_Rate':[2.5,3.5,4.5,4.0], 'India_GDP':[37,42,35,49]}, index = [2001, 2003, 2004, 2005])

joined = df2.join(df1)

print(joined)

	Sensex	Personal_Interest_Rate	India_GDP	House_Price_Index	\
2001	5500	2.5	37	55.0	
2003	6000	3.5	42	70.0	
2004	7000	4.5	35	65.0	
2005	6500	4.0	49	NaN	

	Housing_Interest_Rate	USA_GDP
2001	2.5	37.0
2003	4.5	35.0
2004	4.0	49.0
2005	NaN	NaN

Concatenation

A screenshot of a Google Colab notebook titled "panda.ipynb - Colaboratory". The code cell contains the following Python code:

```
import pandas as pd
df1 = pd.DataFrame({'Name':["Hiren", "Pradip", "Sanjay", "Vijay"],
                    'CPI':[8.8, 7.7, 9.4, 6.3]},
                   index = [11, 22, 33, 44])
df2 = pd.DataFrame({'Name':["Pragnesh", "Bipin", "Ashish", "Parimal"],
                    'CPI':[8.7, 7.8, 9.3, 6.4]},
                   index = [55, 66, 77, 88])
concat = pd.concat([df1, df2])
print(concat)
```

The output of the code is displayed below the code cell:

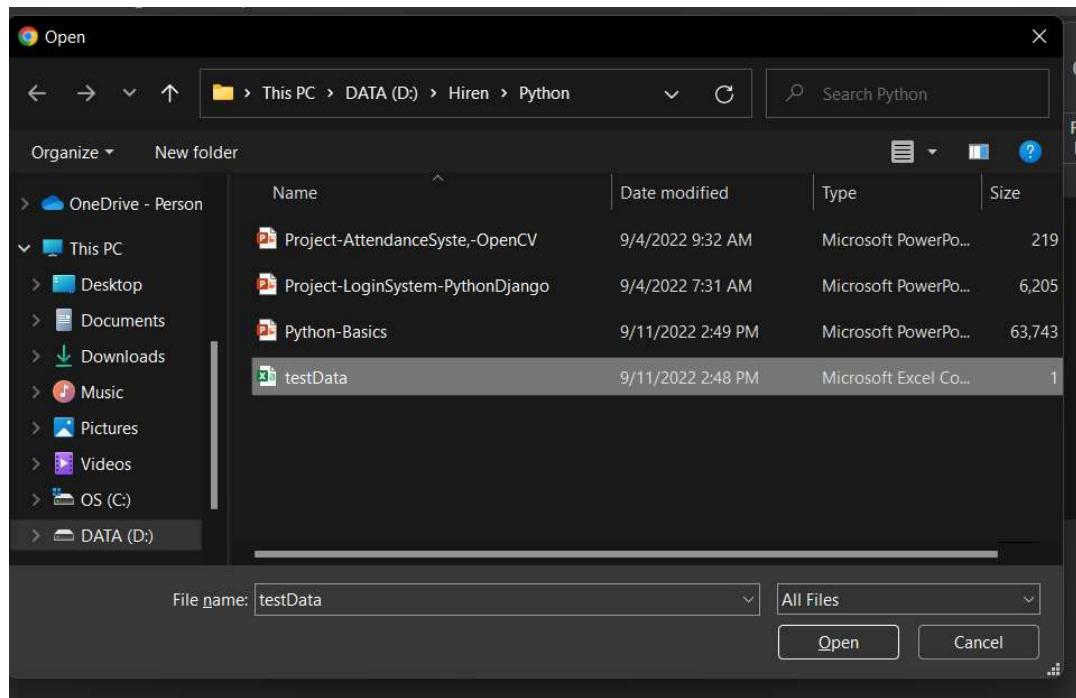
	Name	CPI
11	Hiren	8.8
22	Pradip	7.7
33	Sanjay	9.4
44	Vijay	6.3
55	Pragnesh	8.7
66	Bipin	7.8
77	Ashish	9.3
88	Parimal	6.4

```
import pandas as pd
df1 = pd.DataFrame({'Name':["Hiren", "Pradip", "Sanjay", "Vijay"],
                    'CPI':[8.8, 7.7, 9.4, 6.3]},
                   index = [11, 22, 33, 44])
df2 = pd.DataFrame({'Name':["Pragnesh", "Bipin", "Ashish", "Parimal"],
                    'CPI':[8.7, 7.8, 9.3, 6.4]},
                   index = [55, 66, 77, 88])
concat = pd.concat([df1, df2])
print(concat)
```

A screenshot of a Google Colab notebook interface. At the bottom, there is a file upload dialog with the following text:
Choose Files ...
No file chosen
Cancel upload

A screenshot of Microsoft Excel. The ribbon shows "Home" is selected. A table is displayed with the following data:

	A	B	C	D
1	Year	Sensex	Gold Price	
2	2011	15300	26400	
3	2012	19150	31050	
4	2013	20450	29600	
5	2014	28200	28006	
6	2015	25300	26343	
7	2016	26750	28623	
8	2017	34200	29667	
9	2018	36000	31438	
10	2019	41500	35220	
11	2020	46800	48651	
12	2021	59100	48720	
13				
14				



Convert CSV to HTML

Convert CSV to HTML

panda.ipynb - Colaboratory x python - Exporting Data from go x rates.html - Google

colab.research.google.com/drive/1nPBF6WYPv8KQhzAeYe2i3wDnYXSDJEg#

panda.ipynb

File Edit View Insert Runtime Tools Help Saving...

+ Code + Text

```
import pandas as pd
import io

from google.colab import files
uploaded = files.upload()

readData = pd.read_csv(io.BytesIO(uploaded['testData.csv']))
print(readData)

readData.to_html('rates.html')
files.download('rates.html')
```

Choose Files testData.csv

- testData.csv(text/csv) - 225 bytes, last modified: 9/11/2022 - 100% done

Saving testData.csv to testData (5).csv

	Year	Sensex	Gold Price
0	2011	15300	26400
1	2012	19150	31050
2	2013	20450	29600
3	2014	28200	28006
4	2015	25300	26343
5	2016	26750	28623
6	2017	34200	29667
7	2018	36000	31438
8	2019	41500	35220
9	2020	46800	48651
10	2021	59100	48720

Downloading "rates.html":

rates.html

File | C:/Users/Hiren%20Patel/Downloads/rates.html

	Year	Sensex	Gold Price
0	2011	15300	26400
1	2012	19150	31050
2	2013	20450	29600
3	2014	28200	28006
4	2015	25300	26343
5	2016	26750	28623
6	2017	34200	29667
7	2018	36000	31438
8	2019	41500	35220
9	2020	46800	48651
10	2021	59100	48720

```
import pandas as pd
import io

from google.colab import files
uploaded = files.upload()

readData = pd.read_csv(io.BytesIO(uploaded['testData.csv']))
print(readData)

readData.to_html('rates.html')
files.download('rates.html')
```

```

import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import style
import io
from google.colab import files

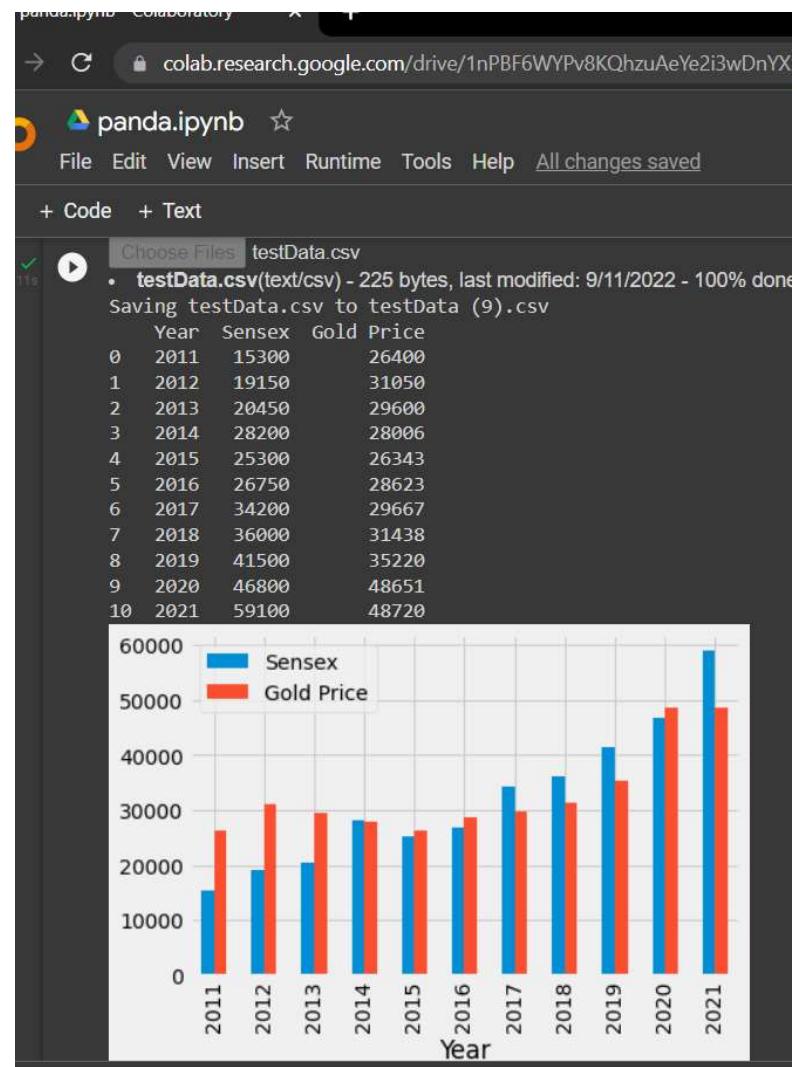
uploaded = files.upload()
readData = pd.read_csv(io.BytesIO(uploaded['testData.csv']))
print(readData)

readData = readData.set_index(["Year"])

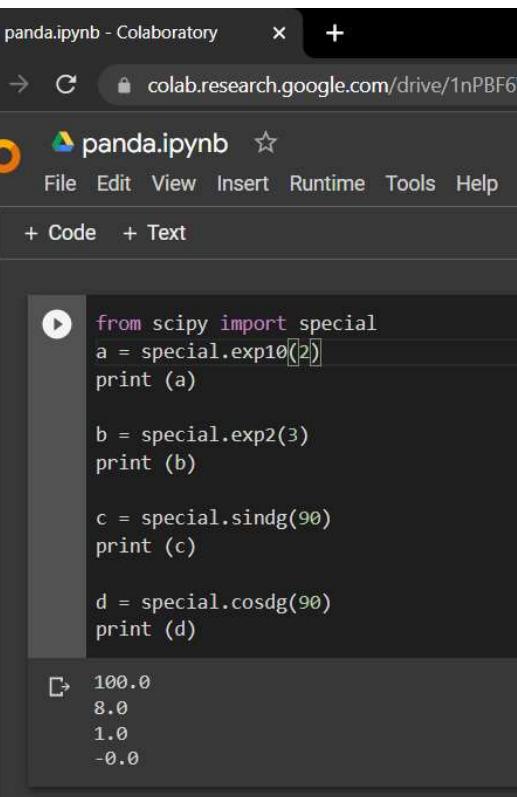
readData.plot(kind='bar')
plt.show()

```

Plotting the Graph



SciPy in Python (Special Function)



The screenshot shows a Google Colab interface with a Jupyter notebook titled "panda.ipynb". The code cell contains the following Python script:

```
from scipy import special
a = special.exp10(2)
print (a)

b = special.exp2(3)
print (b)

c = special.sindg(90)
print (c)

d = special.cosdg(90)
print (d)
```

The output of the cell is:

```
100.0
8.0
1.0
-0.0
```

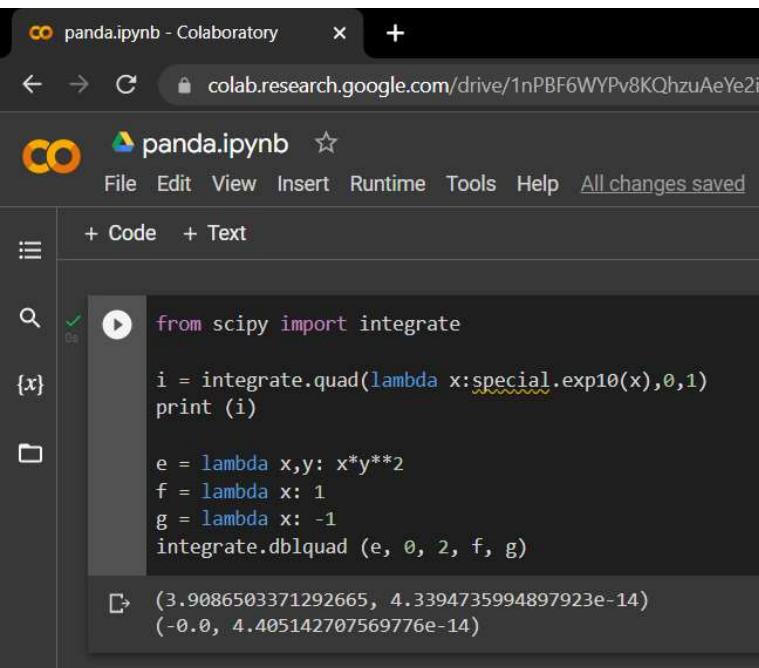
```
from scipy import special
a = special.exp10(2)
print (a)

b = special.exp2(3)
print (b)

c = special.sindg(90)
print (c)

d = special.cosdg(90)
print (d)
```

SciPy in Python (Integration)



A screenshot of a Jupyter Notebook cell. The code imports scipy.integrate and uses quad to integrate a function from 0 to 1. It then defines three functions e, f, and g, and uses dblquad to integrate them over different ranges. The output shows the results of both integrations.

```
from scipy import integrate

i = integrate.quad(lambda x: special.exp10(x),0,1)
print (i)

e = lambda x,y: x*y**2
f = lambda x: 1
g = lambda x: -1
integrate.dblquad (e, 0, 2, f, g)

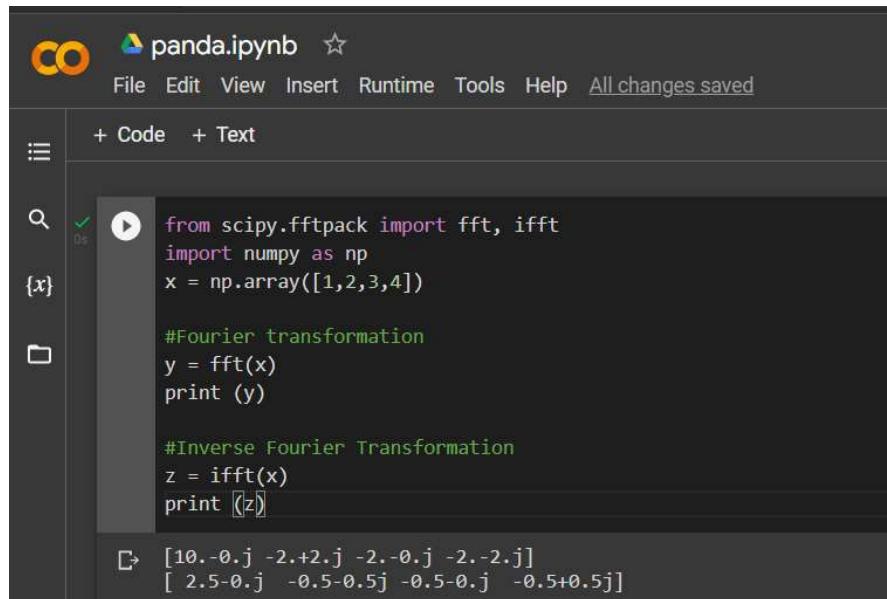
(3.9086503371292665, 4.3394735994897923e-14)
(-0.0, 4.405142707569776e-14)
```

```
from scipy import integrate

i = integrate.quad(lambda
x:special.exp10(x),0,1)
print (i)

e = lambda x,y: x*y**2
f = lambda x: 1
g = lambda x: -1
integrate.dblquad (e, 0, 2, f, g)
```

SciPy in Python (Fourier Transformation)



A screenshot of a Jupyter Notebook interface. The title bar says "panda.ipynb". The menu bar includes File, Edit, View, Insert, Runtime, Tools, Help, and "All changes saved". Below the menu is a toolbar with icons for Code and Text. The code cell contains Python code for Fourier transformation:

```
from scipy.fftpack import fft, ifft
import numpy as np
x = np.array([1,2,3,4])

#Fourier transformation
y = fft(x)
print (y)

#Inverse Fourier Transformation
z = ifft(x)
print (z)
```

The output cell shows the results:

```
[10.-0.j -2.+2.j -2.-0.j -2.-2.j]
[ 2.5-0.j -0.5-0.5j -0.5-0.j -0.5+0.5j]
```

```
from scipy.fftpack import fft, ifft
import numpy as np
x = np.array([1,2,3,4])

#Fourier transformation
y = fft(x)
print (y)

#Inverse Fourier Transformation
z = ifft(x)
print (z)
```

panda.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help

+ Code + Text

{x}

from scipy import linalg
#inverse of a matrix

a = np.array([[1,2],[3,4]])
b = linalg.inv (a)

print (b)

[[[-2. 1.]
 [1.5 -0.5]]]

```
from scipy import linalg
#inverse of a matrix

a = np.array([[1,2],[3,4]])

print (b)
```

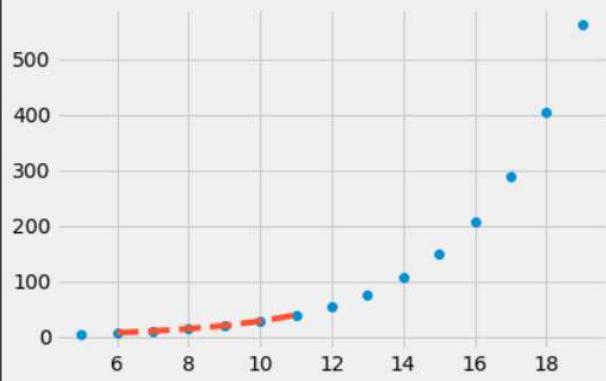
panda.ipynb - Colaboratory

File Edit View Insert Runtime Tools Help

+ Code + Text

{x}

import matplotlib.pyplot as plt
import numpy as np
from scipy import interpolate
x = np.arange(5, 20)
y = np.exp(x/3.0)
f = interpolate.interp1d(x, y)
x1 = np.arange(6, 12)
y1 = f(x1)
plt.plot(x, y, 'o', x1, y1, '--')
plt.show()



x	y	x1	y1
5	0	6	0
6	0	7	0
7	0	8	0
8	0	9	0
9	0	10	0
10	0	11	20
11	20	12	50
12	50	13	80
13	80	14	110
14	110	15	140
15	140	16	170
16	170	17	200
17	200	18	230
18	230	19	260
19	260	20	290

SciPy in Python (Linear Algebra)

SciPy in Python (Interpolation)

```
import matplotlib.pyplot as plt
import numpy as np
from scipy import interpolate
x = np.arange(5, 20)
y = np.exp(x/3.0)
f = interpolate.interp1d(x, y)
x1 = np.arange(6, 12)
y1 = f(x1)
plt.plot(x, y, 'o', x1, y1, '--')
plt.show()
```

panda.ipynb - Colaboratory +

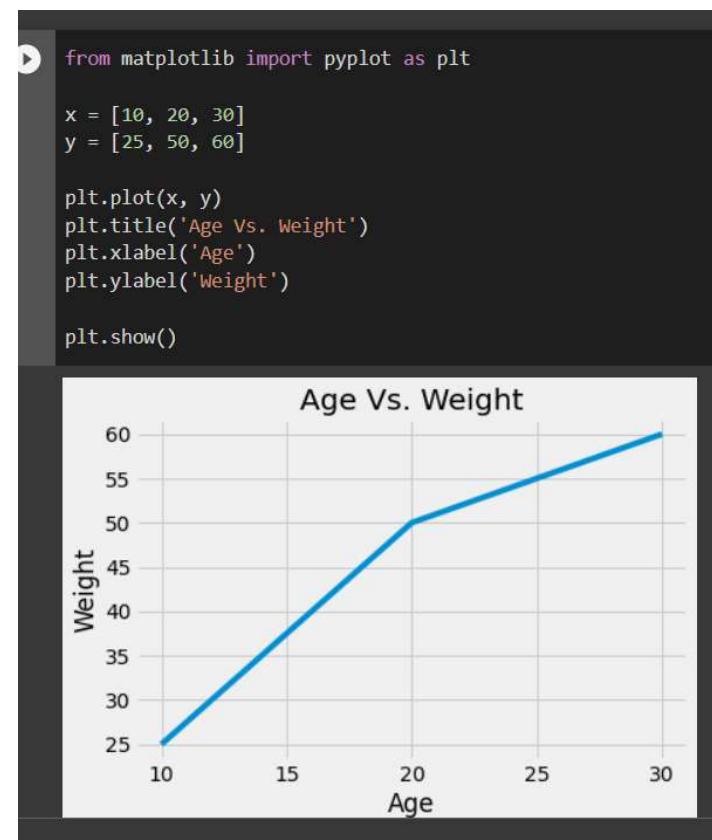
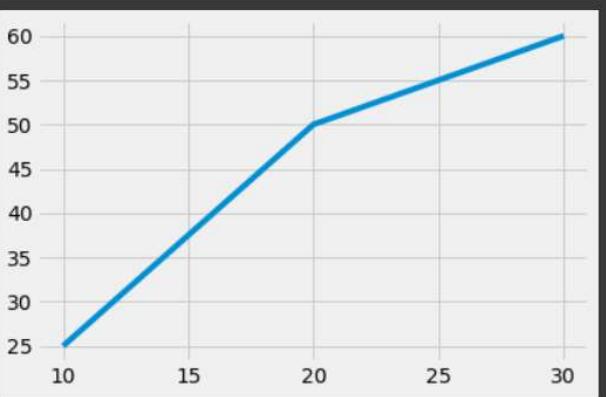
colab.research.google.com/drive/1nPBF6WYPv8KQhzAeYe2i3v

panda.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

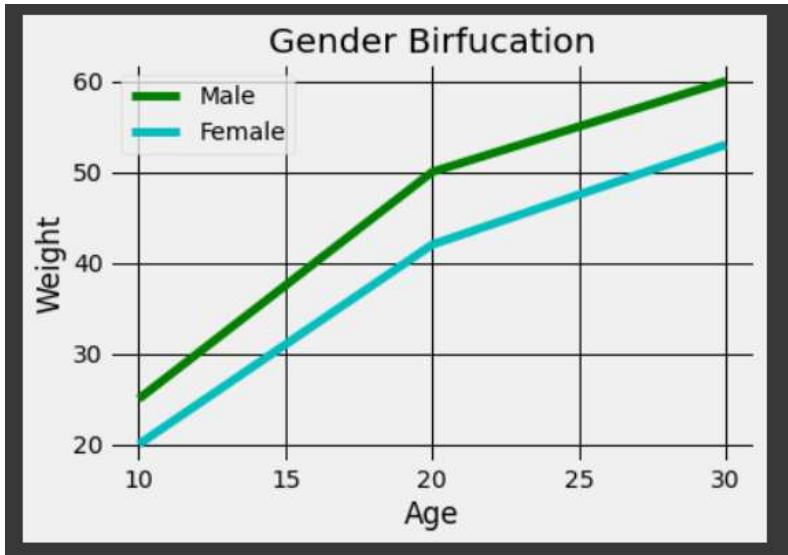
from matplotlib import pyplot as plt
plt.plot([10, 20, 30],[25, 50, 60])
plt.show()



```
from matplotlib import pyplot as plt  
  
x = [10, 20, 30]  
y = [25, 50, 60]  
  
plt.plot(x, y)  
plt.title('Age Vs. Weight')  
plt.xlabel('Age')  
plt.ylabel('Weight')  
  
plt.show()
```

```
from matplotlib import pyplot as plt  
  
plt.plot([10, 20, 30],[25, 50, 60])  
  
plt.show()
```

Python for Visualization (matplotlib)



Python for Visualization (matplotlib)

```
from matplotlib import pyplot as plt
male_age = [10, 20, 30]
male_weight = [25, 50, 60]
female_age = [10, 20, 30]
female_weight = [20, 42, 53]
plt.plot(male_age, male_weight, 'g', label='Male', linewidth=5)
plt.plot(female_age, female_weight, 'c', label='Female', linewidth=5)
plt.title('Gender Birfucation')
plt.xlabel('Age')
plt.ylabel('Weight')
plt.legend()
plt.grid(True, color='k')
plt.show()
```

```

import matplotlib.pyplot as plt

male_age = [10, 20, 30]
male_weight = [25, 50, 60]

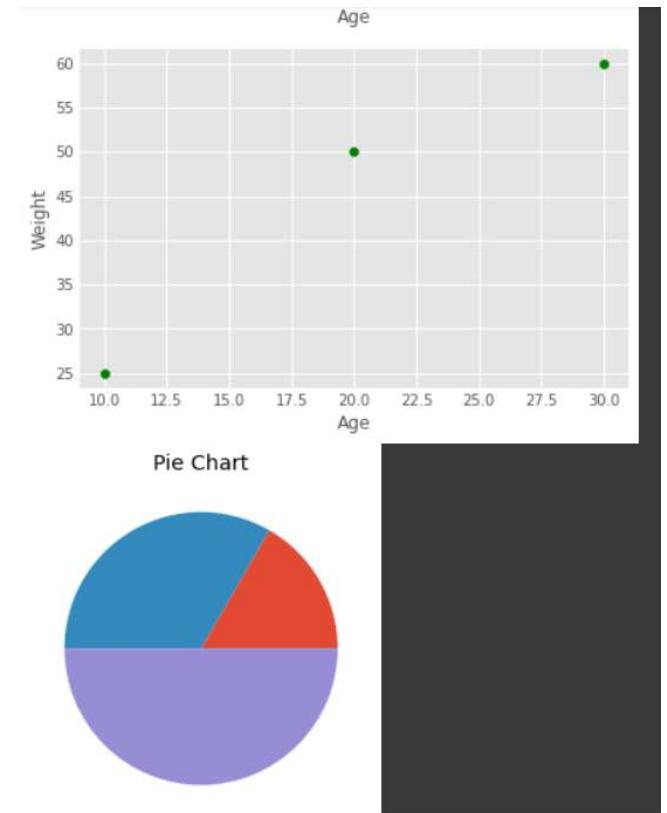
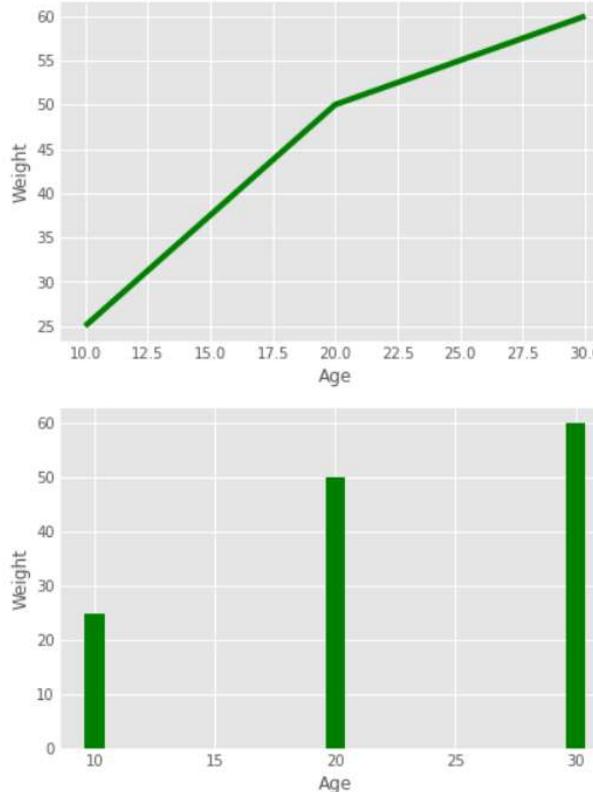
plt.xlabel('Age')
plt.ylabel('Weight')
plt.plot(male_age, male_weight,
color='g', label='Line Chart')
plt.show()

plt.xlabel('Age')
plt.ylabel('Weight')
plt.bar(male_age, male_weight,
color='g', label='Bar Chart')
plt.show()

plt.xlabel('Age')
plt.ylabel('Weight')
plt.scatter(male_age, male_weight,
color='g', label='Scatter Chart')
plt.show()

plt.pie(male_age)
plt.title('Pie Chart')
plt.show()

```



Python for Visualization (matplotlib)

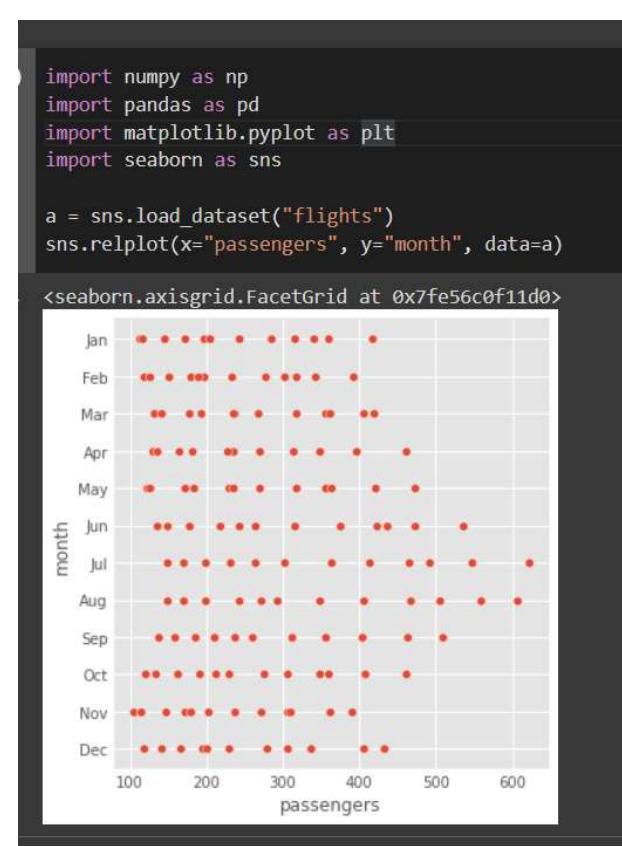
Seaborn

github.com/mwaskom/seaborn-data

dots.csv	Add dots dataset
dowjones.csv	Add dowjones dataset
exercise.csv	Add exercise dataset
flights.csv	Add flights dataset
fmri.csv	flights.csv Change sorting of events

github.com/mwaskom/seaborn-data/blob/master/flights.csv

year	month	passengers
1949	January	112
1949	February	118
1949	March	132
1949	April	129
1949	May	121
1949	June	135
1949	July	148
1949	August	148



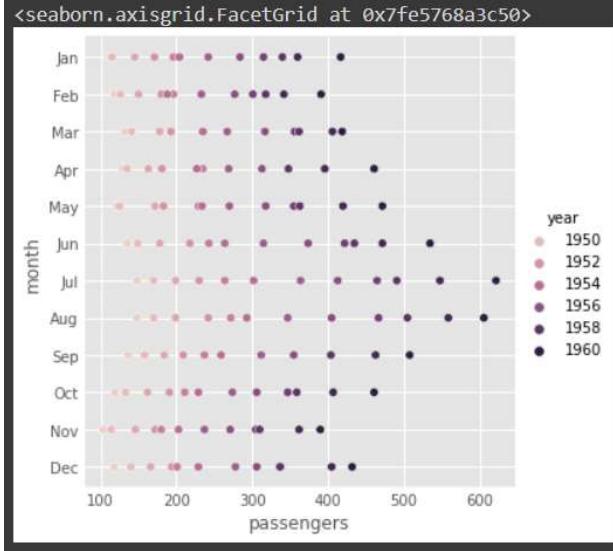
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

a = sns.load_dataset("flights")
sns.relplot(x="passengers", y="month", data=a)
```

Seaborn

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

a = sns.load_dataset("flights")
sns.relplot(x="passengers", y="month", hue='year', dat
```



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

a = sns.load_dataset("flights")
sns.relplot(x="passengers", y="month", hue='year', data=a)
```

Seaborn

github.com/mwaskom/seaborn-data/blob/master/tips.csv

245 lines (245 sloc) | 9.5 KB

Raw Blame

Search this file...

1	total_bill	tip	sex	smoker	day	time	size
2	16.99	1.01	Female	No	Sun	Dinner	2
3	10.34	1.66	Male	No	Sun	Dinner	3

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

b = sns.load_dataset("tips")
sns.relplot(x="time", y="tip", data=b, kind="line")
```



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

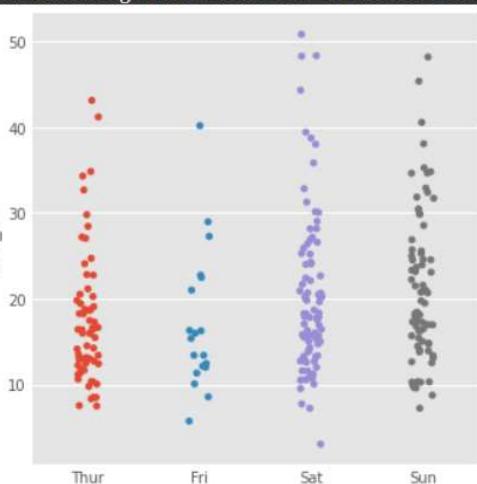
b = sns.load_dataset("tips")
sns.relplot(x="time", y="tip", data=b, kind="line")
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

b = sns.load_dataset("tips")

sns.catplot(x="day", y="total_bill", data=b)

<seaborn.axisgrid.FacetGrid at 0x7fe56c15e450>
```

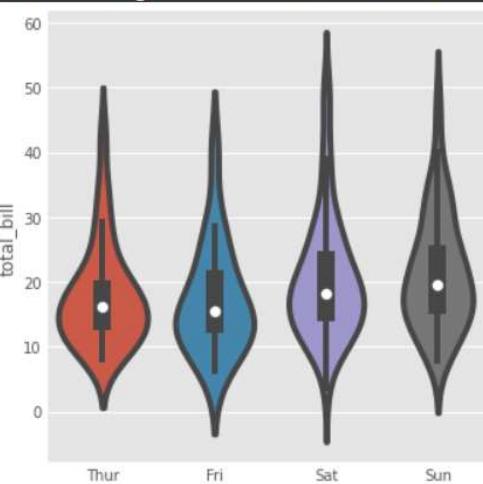
A scatter plot with four facets corresponding to the days of the week: Thursday, Friday, Saturday, and Sunday. The x-axis is labeled 'day' and the y-axis is labeled 'total_bill'. The data points are colored by day: Thursday is red, Friday is blue, Saturday is purple, and Sunday is grey. The plot shows a general increase in total bill from Thursday to Sunday.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

b = sns.load_dataset("tips")

sns.catplot(x="day", y="total_bill", kind="violin", data=b)

<seaborn.axisgrid.FacetGrid at 0x7fe56a6e6250>
```

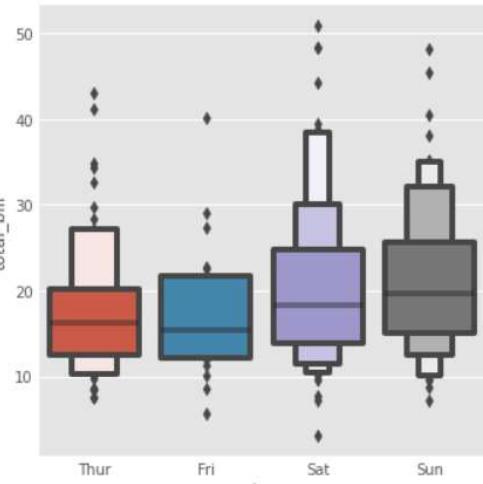
A facet grid with four violin plots showing the distribution of total_bill for each day of the week. The x-axis is labeled 'day' and the y-axis is labeled 'total_bill'. The violins are colored red for Thursday, blue for Friday, purple for Saturday, and grey for Sunday. Each violin has a central white dot representing the median.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

b = sns.load_dataset("tips")

sns.catplot(x="day", y="total_bill", kind="boxen", data=b)

<seaborn.axisgrid.FacetGrid at 0x7fe56bfa5350>
```

A facet grid with four boxen plots showing the distribution of total_bill for each day of the week. The x-axis is labeled 'day' and the y-axis is labeled 'total_bill'. The boxen are colored red for Thursday, blue for Friday, purple for Saturday, and grey for Sunday. Each boxen has a central white dot representing the median.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

b = sns.load_dataset("tips")

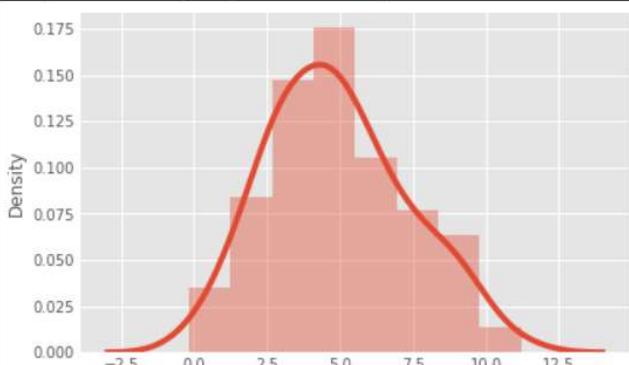
sns.catplot(x="day", y="total_bill", data=b)
```

Seaborn

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

c=sns.load_dataset("tips")
c = np.random.normal(loc=5, size=100, scale=2)
sns.distplot (c)

/usr/local/lib/python3.7/dist-packages/seaborn/distribution
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7fe564ffbdd0>


```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

c=sns.load_dataset("tips")
c = np.random.normal(loc=5, size=100, scale=2)
sns.distplot (c)
```

Seaborn

github.com/mwaskom/seaborn-data/blob/master/iris.csv

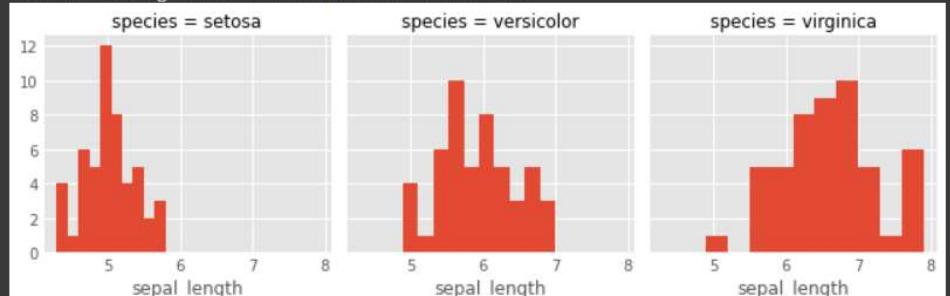
Search this file...

	sepal_length	sepal_width	petal_length	petal_width	species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

a = sns.load_dataset("iris")
b = sns.FacetGrid(a, col="species")
b.map(plt.hist, "sepal_length")
```

<seaborn.axisgrid.FacetGrid at 0x7fe565043110>



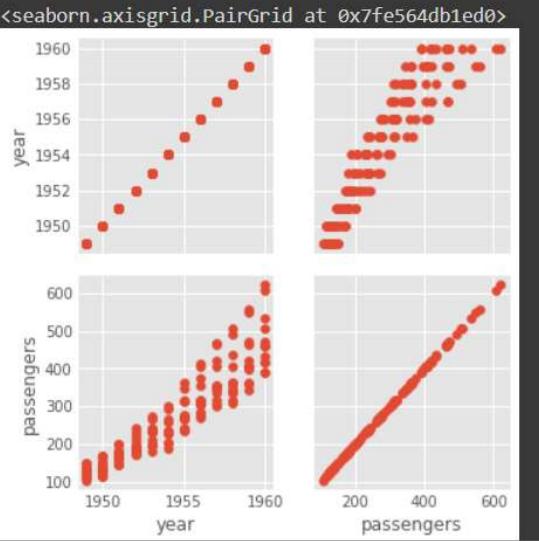
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

a = sns.load_dataset("iris")
b = sns.FacetGrid(a, col="species")
b.map(plt.hist, "sepal_length")
```

Multi-Plot Grids

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

a = sns.load_dataset("flights")
b = sns.PairGrid(a)
b.map(plt.scatter)
```



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

a = sns.load_dataset("flights")
b = sns.PairGrid(a)
b.map(plt.scatter)
```

Multi-Plot Grids

Tensorflow

```
import tensorflow as tf

node1 = tf.constant(3.0, tf.float32)
node2 = tf.constant(4.0)

print(node1, node2)

tf.Tensor(3.0, shape=(), dtype=float32) tf.Tensor(4.0, shape=(), dtype=float32)
```

```
import tensorflow as tf

node1 = tf.constant(3.0, tf.float32)
node2 = tf.constant(4.0)

print(node1, node2)
```

```
▶ import tensorflow as tf
tf.compat.v1.disable_eager_execution()

a = tf.constant([5.0])
b = tf.constant(6.0)

c = a * b

sess = tf.compat.v1.Session()

result = sess.run(c)
print(result)

30.0
```

```
import tensorflow as tf
tf.compat.v1.disable_eager_execution()

a = tf.constant(5.0)
b = tf.constant(6.0)

c = a * b

sess = tf.compat.v1.Session()

result = sess.run(c)
print(result)
```

NumPy

```
import numpy as np
import sys

a = np.array([1, 2, 3], dtype='int8')
b = np.array([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]])

print(a)

print("Dimension - a : ", a.ndim)
print("Dimension - b : ", b.ndim)

print("a - Type: ", a.dtype, "Size: ", a.itemsize, "bytes")
print("b - Type: ", b.dtype, "Size: ", b.itemsize, "bytes")

[1 2 3]
Dimension - a : 1
Dimension - b : 2
a - Type: int8 Size: 1 bytes
b - Type: float64 Size: 8 bytes
```

```
import numpy as np
import sys

a = np.array([1, 2, 3], dtype='int8')
b = np.array([[1.0, 2.0, 3.0], [4.0, 5.0, 6.0]])

print(a)

print("Dimension - a : ", a.ndim)
print("Dimension - b : ", b.ndim)

print("a - Type: ", a.dtype, "Size: ", a.itemsize, "bytes")
print("b - Type: ", b.dtype, "Size: ", b.itemsize, "bytes")
```

```
In [1]: import numpy as np
import sys
```

```
In [3]: a = np.array([[11,22,33,44,55],[10,20,30,40,50]])
print(a)
```

```
[[11 22 33 44 55]
 [10 20 30 40 50]]
```

```
In [5]: a.shape
```

```
Out[5]: (2, 5)
```

```
In [6]: a[0, 2]
```

```
Out[6]: 33
```

```
In [7]: a[1,-2]
```

```
Out[7]: 40
```

```
In [8]: a[0,:]
```

```
Out[8]: array([11, 22, 33, 44, 55])
```

```
In [9]: a[:, 0]
```

```
Out[9]: array([11, 10])
```

```
In [10]: a[1,4]=99  
print(a)
```

```
[[11 22 33 44 55]  
 [10 20 30 40 99]]
```

```
In [11]: a[:, 2]=101  
print(a)
```

```
[[ 11 22 101 44 55]  
 [ 10 20 101 40 99]]
```

```
In [12]: np.zeros(5)
```

```
Out[12]: array([0., 0., 0., 0., 0.])
```

```
In [14]: np.zeros((2, 3))
```

```
Out[14]: array([[0., 0., 0.],  
 [0., 0., 0.]])
```

```
In [15]: np.ones(5)
```

```
Out[15]: array([1., 1., 1., 1., 1.])
```

```
In [16]: np.ones((2, 3))
```

```
Out[16]: array([[1., 1., 1.],  
 [1., 1., 1.]])
```

```
In [18]: np.full((2,4),99)
```

```
Out[18]: array([[99, 99, 99, 99],  
 [99, 99, 99, 99]])
```

```
In [19]: np.full((2,4),99, dtype='float32')
```

```
Out[19]: array([[99., 99., 99., 99.],  
 [99., 99., 99., 99.]], dtype=float32)
```

```
In [20]: np.random.rand(4,2)
```

```
Out[20]: array([[0.59161756, 0.23972281],  
 [0.33114222, 0.47488643],  
 [0.12111486, 0.71373998],  
 [0.75847543, 0.78653875]])
```

```
In [21]: np.random.random_sample(a.shape)
```

```
Out[21]: array([[0.58526244, 0.51710963, 0.4514024 , 0.30242931, 0.2918369 ],  
 [0.86825353, 0.43792201, 0.47336005, 0.41503633, 0.55981663]])
```

```
In [23]: np.random.randint(10,size=(2,4))
```

```
Out[23]: array([[8, 3, 4, 2],  
 [9, 1, 5, 8]])
```

```
In [24]: np.random.randint(10,20,size=(2,4))
```

```
Out[24]: array([[18, 10, 13, 12],  
 [19, 15, 15, 17]])
```

```
In [25]: np.identity(3)
```

```
Out[25]: array([[1., 0., 0.],  
 [0., 1., 0.],  
 [0., 0., 1.]])
```

```
In [25]: np.identity(3)
```

```
Out[25]: array([[1., 0., 0.],  
 [0., 1., 0.],  
 [0., 0., 1.]])
```

```
In [27]: arr = np.array([1,2,3])  
rp = np.repeat(arr, 4)  
print (rp)
```

```
[1 1 1 1 2 2 2 2 3 3 3 3]
```

```
In [28]: a = np.array([1,2,3,4])  
print(a)
```

```
[1 2 3 4]
```

```
In [29]: a+2
```

```
Out[29]: array([3, 4, 5, 6])
```

```
In [30]: a-2
```

```
Out[30]: array([-1, 0, 1, 2])
```

```
In [31]: a*2
```

```
Out[31]: array([2, 4, 6, 8])
```

```
In [32]: a/2
```

```
Out[32]: array([0.5, 1. , 1.5, 2. ])
```

```
Out[33]: array([2, 2, 4, 4])
```

```
In [34]: a ** 2
```

```
Out[34]: array([ 1, 4, 9, 16], dtype=int32)
```

```
In [35]: np.sin(a)
```

```
Out[35]: array([ 0.84147098, 0.90929743, 0.14112001, -0.7568025 ])
```

```
In [36]: np.cos(a)
```

```
Out[36]: array([ 0.54030231, -0.41614684, -0.9899925 , -0.65364362])
```

NumPy

NumPy

```
In [37]: np.sin(a)
Out[37]: array([ 0.84147098,  0.90929743,  0.14112001, -0.7568025 ])

In [44]: a = np.ones((2, 3))
print (a)

b= np.full((3, 2), 4)
print (b)

c = np.matmul(a, b)
print (c)

[[1. 1. 1.]
 [1. 1. 1.]]
 [[4 4]
 [4 4]
 [4 4]]
 [[12. 12.]
 [12. 12.]]

In [47]: # Find determinant
c = np.identity(3)
np.linalg.det(c)

Out[47]: 1.0

In [50]: stats = np.array ([[1,2,3],[4,5,6]])
stats

Out[50]: array([[1, 2, 3],
 [4, 5, 6]])
```

```
In [52]: np.min(stats)
Out[52]: 1

In [53]: np.max(stats)
Out[53]: 6

In [56]: np.max(stats, axis=1)
Out[56]: array([3, 6])

In [57]: np.max(stats, axis=0)
Out[57]: array([4, 5, 6])

In [58]: np.sum(stats)
Out[58]: 21

In [59]: np.sum(stats, axis=0)
Out[59]: array([5, 7, 9])

In [60]: np.sum(stats, axis=1)
Out[60]: array([ 6, 15])
```

```
In [67]: v1 = np.array([1,2,3,4])
v2 = np.array([5,6,7,8])

np.vstack([v1, v2])
np.vstack([v1, v2, v2, v2])

Out[67]: array([[1, 2, 3, 4],
 [5, 6, 7, 8],
 [5, 6, 7, 8],
 [5, 6, 7, 8]])
```

```
In [70]: h1 = np.ones((2,4))
h2 = np.zeros((2,2))
np.hstack([h1, h2])

Out[70]: array([[1., 1., 1., 1., 0., 0.],
 [1., 1., 1., 1., 0., 0.]])
```

```
In [64]: pre = np.array([[1,2,3,4],[5,6,7,8]])
print(pre)

post = pre.reshape((1,8))
print(post)

[[1 2 3 4]
 [5 6 7 8]]
 [[1 2 3 4 5 6 7 8]]

In [62]: pre = np.array([[1,2,3,4],[5,6,7,8]])
print(pre)

post = pre.reshape((2,4))
print(post)

[[1 2 3 4]
 [5 6 7 8]]
 [[1 2 3 4]
 [5 6 7 8]]

In [63]: pre = np.array([[1,2,3,4],[5,6,7,8]])
print(pre)

post = pre.reshape((2,2,2))
print(post)

[[1 2 3 4]
 [5 6 7 8]]
 [[[1 2]
 [3 4]]
 [[5 6]
 [7 8]]]
```

NumPy

A screenshot of a web browser window showing a Jupyter Notebook interface. The title bar says "localhost:8888/tree". The main area shows a file browser with a sidebar for "anaconda3", "Contacts", "Desktop", "Documents", and "Downloads". A context menu is open over a blank area, with the "Text File" option highlighted by a red box. The menu also includes "Upload", "New", "Folder", and "Terminal". The status bar at the bottom right says "2 hours ago".

A screenshot of a web browser window showing a Jupyter Text Editor. The title bar says "localhost:8888/edit/data.txt". The main area shows a text editor with the file "data.txt" open. The content of the file is:

```
1 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
2 11, 22, 33, 44, 55, 66, 77, 88, 99, 111
3 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
```

NumPy

```
In [74]: np.genfromtxt('data.txt',delimiter=',')  
  
Out[74]: array([[ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.],  
                 [11., 22., 33., 44., 55., 66., 77., 88., 99., 111.],  
                 [10., 20., 30., 40., 50., 60., 70., 80., 90., 100.]])  
  
In [76]: filedata = np.genfromtxt('data.txt',delimiter=',')  
filedata  
  
Out[76]: array([[ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.],  
                 [11., 22., 33., 44., 55., 66., 77., 88., 99., 111.],  
                 [10., 20., 30., 40., 50., 60., 70., 80., 90., 100.]])  
  
In [78]: filedata = np.genfromtxt('data.txt',delimiter=',')  
filedata = filedata.astype('int32')  
filedata  
  
Out[78]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10],  
                 [11, 22, 33, 44, 55, 66, 77, 88, 99, 111],  
                 [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]])  
  
In [79]: filedata > 50  
  
Out[79]: array([[False, False, False, False, False, False, False, False,  
                  False],  
                 [False, False, False, False, True, True, True, True,  
                  True],  
                 [False, False, False, False, False, True, True, True,  
                  True]])
```

```
In [82]: filedata [filedata > 50]  
  
Out[82]: array([ 55,  66,  77,  88,  99, 111,  60,  70,  80,  90, 100])  
  
In [83]: np.any(filedata > 50, axis=0)  
  
Out[83]: array([False, False, False, False, True, True, True, True,  
                  True])  
  
In [84]: np.all(filedata > 50, axis=0)  
  
Out[84]: array([False, False, False, False, False, False, False, False,  
                  False])  
  
In [86]: ((filedata >50) & (filedata <100))  
  
Out[86]: array([[False, False, False, False, False, False, False, False,  
                  False],  
                 [False, False, False, False, True, True, True, True,  
                  True],  
                 [False, False, False, False, False, True, True, True,  
                  True]])  
  
In [87]: ~((filedata >50) & (filedata <100))  
  
Out[87]: array([[ True,  True,  True,  True,  True,  True,  True,  True,  
                  True],  
                 [ True,  True,  True,  True, False, False, False, False,  
                  True],  
                 [ True,  True,  True,  True,  True, False, False, False,  
                  True]])
```