

به نام خدا

موضوع: کاربرد جبر خطی در پایتون

تیه کننده: ایمان جوادی سیسی

شماره دانشجویی: _____

استاد: جناب آقای دکتر حمید رسولی

درس: جبر خطی

دانشگاه آزاد اسلامی واحد علوم و تحقیقات

تعريف جبر خطى

جبر خطى چیست ؟

جبر خطى شاخه‌ای از ریاضیات است که به بررسی و مطالعه ماتریس‌ها، بردارها، فضاهای برداری (فضاهای خطى)، تبدیلات خطى، و دستگاه‌های معادلات خطى می‌پردازد.

کاربرد جبر خطى چیست ؟

جبر خطى کاربرد های مختلف در علوم مختلف دارد . برای مثال در خود ریاضیات در جبر مجرد، آنالیز تابعی، هندسه تحلیلی، آنالیز عددی ، فیزیک ، پردازش تصویر ، مخابرات ، بازی های ویدیویی ، گرافیک رایانه ای ، پردازش سیگنال های پزشکی و کاربرد دارد . اما یکی از مهم ترین کاربرد های جبر خطى در بحث هوش مصنوعی می باشد که خود دارای زیر مجموعه های مختلفی میباشد .

پایتون

جامع ترین و پر کاربردترین زبان برنامه نویسی حال حاضر دنیا که یادگیری اون ب شدت سادست و به همین دلیل به نو آوران توصیخ میشود با این زبان شروع کنند.



مزایای پایتون :

۱) شی گرا بودن

۲) کتابخانه های مرجع و جامع

۳) قابلیت main platform

۴) ساختار کاربردی ساده

امروزه غول های دنیا از این برنامه استفاده میکنند. شرکت هایی مانند (گوگل . یوتیوب . فیسبوک . ناسا و) با پیشرفت علم و تکنولوژی و ظهور ماشین لرنینگ بیشتر از هر وقت دیگه ای پایتون تونسته خودش رو محبوب تر کنه و روز ب روز کاربرد این زبان محبوب در حال افزایش .

برخی از کابرد های پایتون :

۱) ماشین لرنینگ ۲) دیتا ساینس ۳) توسعه نرم افزار ۴) مدیریت (سیستم و آی تی و وب)

کتابخانه های پایتون

کتابخانه های پایتون به دو صورت `built in` و خارجی می بانشد.

کتابخانه های `built in` کتابخانه هایی هستند که بصورت پیش فرض در پایتون قرار دارند.

کتابخانه های خارجی، کتابخانه هایی هستند که باید با استفاده از دستور `pip` نصب شوند.



Pip چیست؟

Pip یک سیستم مدیریت بسته (package manager) برای پایتون است. سیستم مدیریت بسته، ابزاری است که امکان نصب و مدیریت کتابخانه‌ها و متعلقات (depandancy) اعضای پایتون را به برنامه نویسان میدهد.

لازم به ذکر است که با کمک Pip میتوان کتابخانه‌ها و وابستگی‌هایی را نصب و مدیریت کرد که به عنوان بخشی از کتابخانه استاندارد توزیع نشده‌اند.

نحوه نصب Pip

```
imam@imam-IdeaPad-3-15IML05:~$ sudo apt install python3-pip
[sudo] password for iman:
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (20.0.2-5ubuntu1.6).
0 to upgrade, 0 to newly install, 0 to remove and 17 not to upgrade.
imam@imam-IdeaPad-3-15IML05:~$
```

نحوه نصب کتابخانه ها در پایتون

برای نصب یک کتابخانه در پایتون با کمک Pip، از دستور زیر استفاده میکنیم:

```
Windows #
pip install packagename
```

```
Mac/Linux #
sudo pip install packagename
```

❖ به جای packagename نام بسته مورد نظر خود برای نصب را بنویسید.

این دستور، کتابخانه نوشته شده را برای ما دانلود کرده و سپس نصب میکند.

بروز رسانی کتابخانه ها در پایتون

به منظور بروز رسانی یک کتابخانه میتوانیم دستور زیر را وارد نماییم:

```
Windows #
pip install --upgrade packagename
```

```
Mac/Linux #
sudo pip install --upgrade
packagename
```

اگر کتابخانه مربوط به روز باشد، بسته ای دانلود نشده و نصب نمیگردد.

حذف و پاکسازی بسته ها در پایتون

برای حذف و پاکسازی یک بسته از سیستم از دستور زیر استفاده مینماییم:

Windows #

```
pip uninstall packagename
```

Mac/Linux #

```
sudo pip uninstall packagename
```

جستجوی کتابخانه ها در پایتون

اگر بخواهیم نام یک کتابخانه را جستجو نماییم، می توانیم با استفاده از دستور زیر عبارتی را وارد کرده تا pip در لیست بسته های موجود به جستجوی آن بپردازد.

```
"pip search "query"
```

NumPy

Numpy یک کتابخانه برای زبان برنامه نویسی پایتون است.

با استفاده از این کتابخانه، امکان استفاده از آرایه ها و ماتریس های بزرگ چند بعدی فراهم می شود. همچنین می توان از تابع های ریاضیاتی سطح بالا بر روی این آرایه ها استفاده کرد.

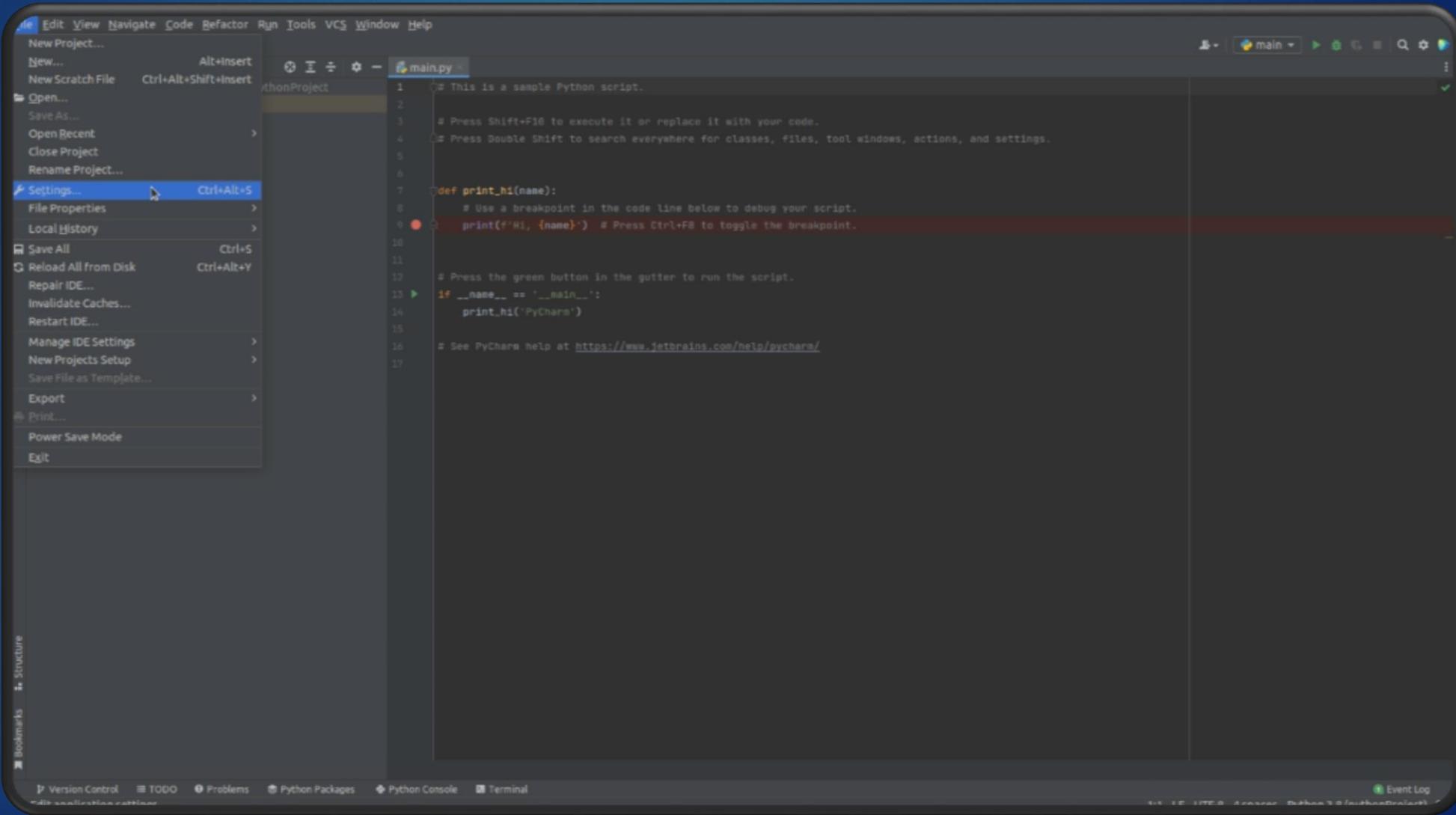


NumPy

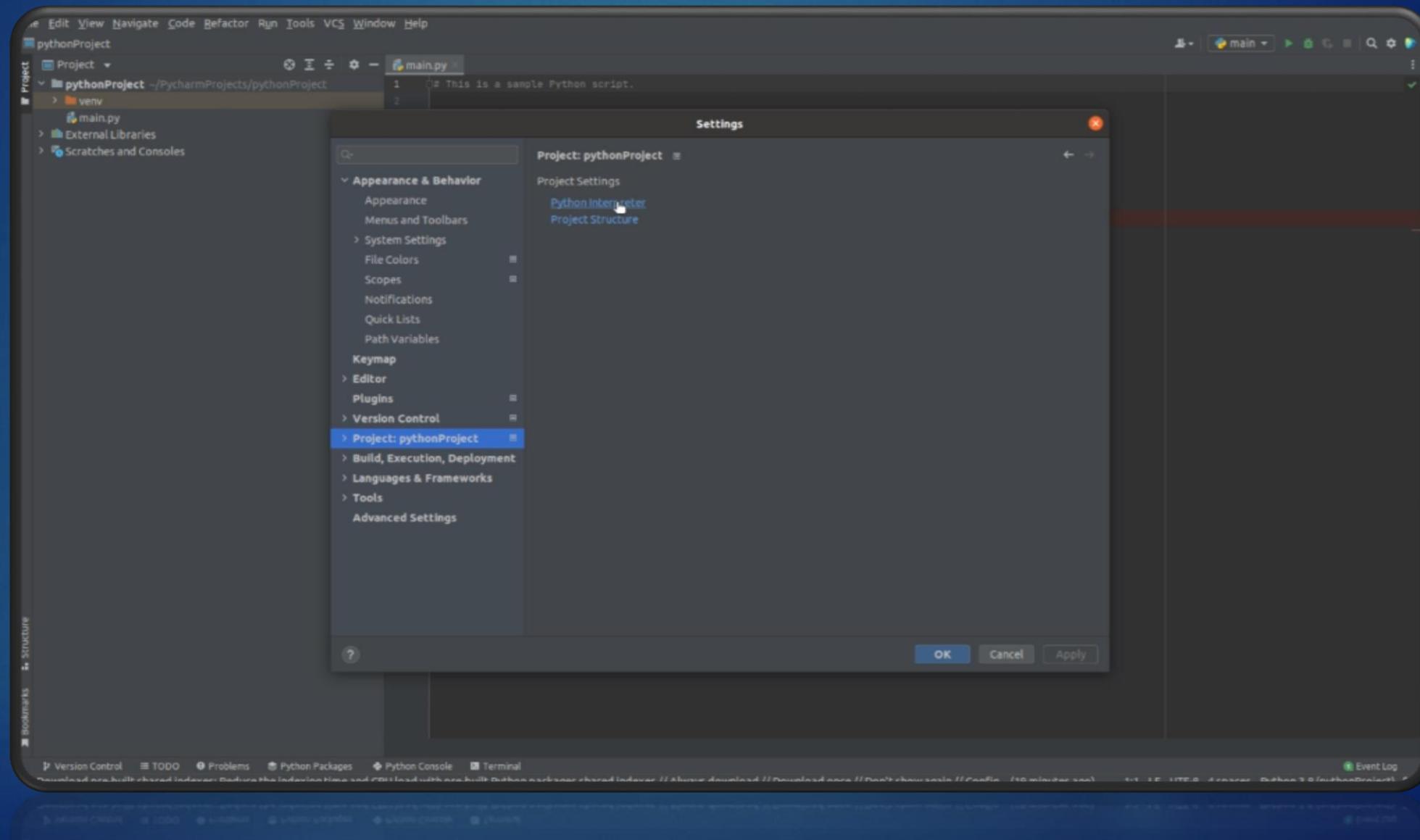
نحوه نصب NumPy

```
imam@imam-IdeaPad-3-15IML05:~$ sudo pip3 install numpy
Collecting numpy
  Downloading numpy-1.22.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(16.8 MB)
|██████████| 16.8 MB 2.1 MB/s
Installing collected packages: numpy
Successfully installed numpy-1.22.0
imam@imam-IdeaPad-3-15IML05:~$
```

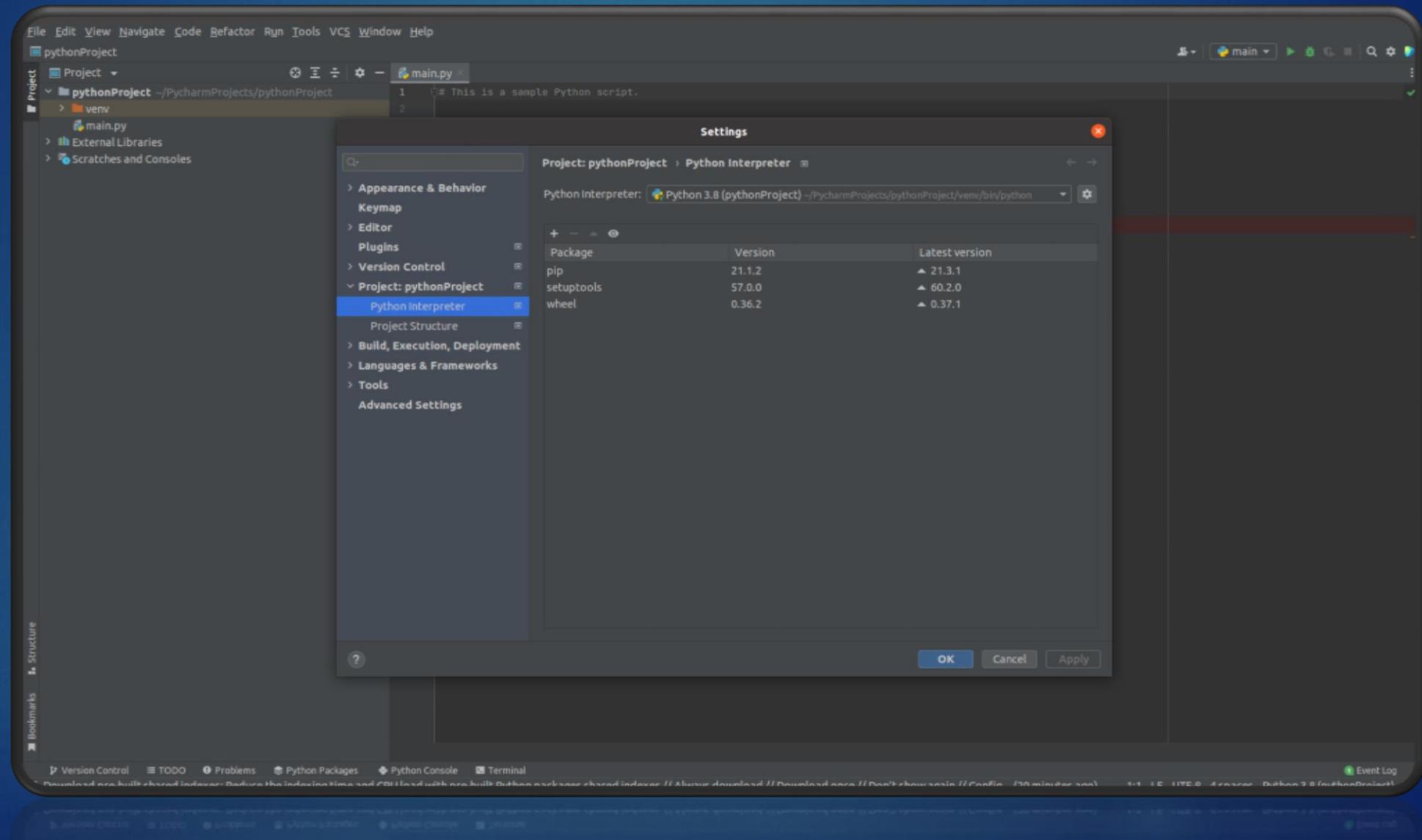
مرحله اول نصب PyCharm در NumPy



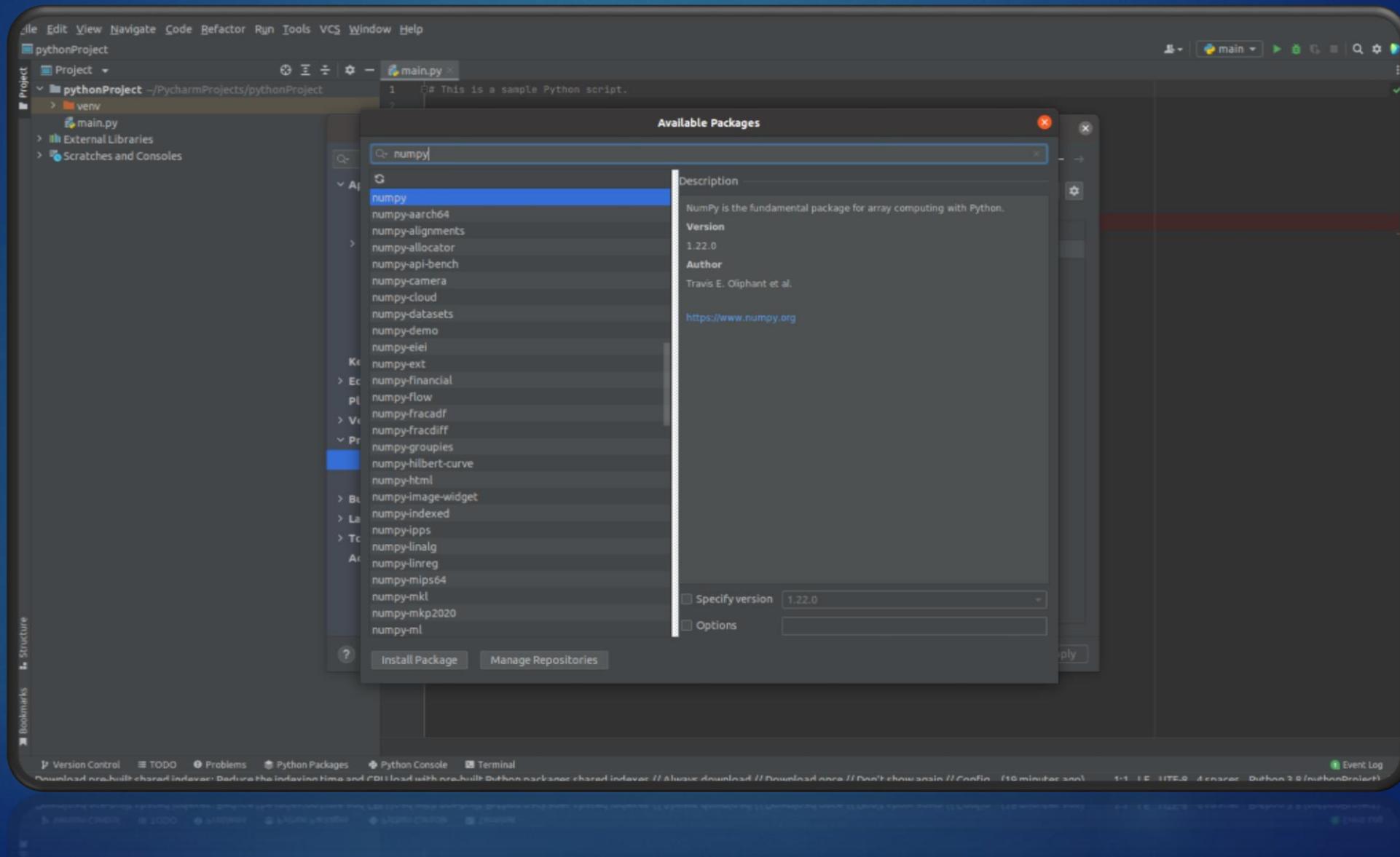
مرحله دوم نصب NumPy در PyCharm



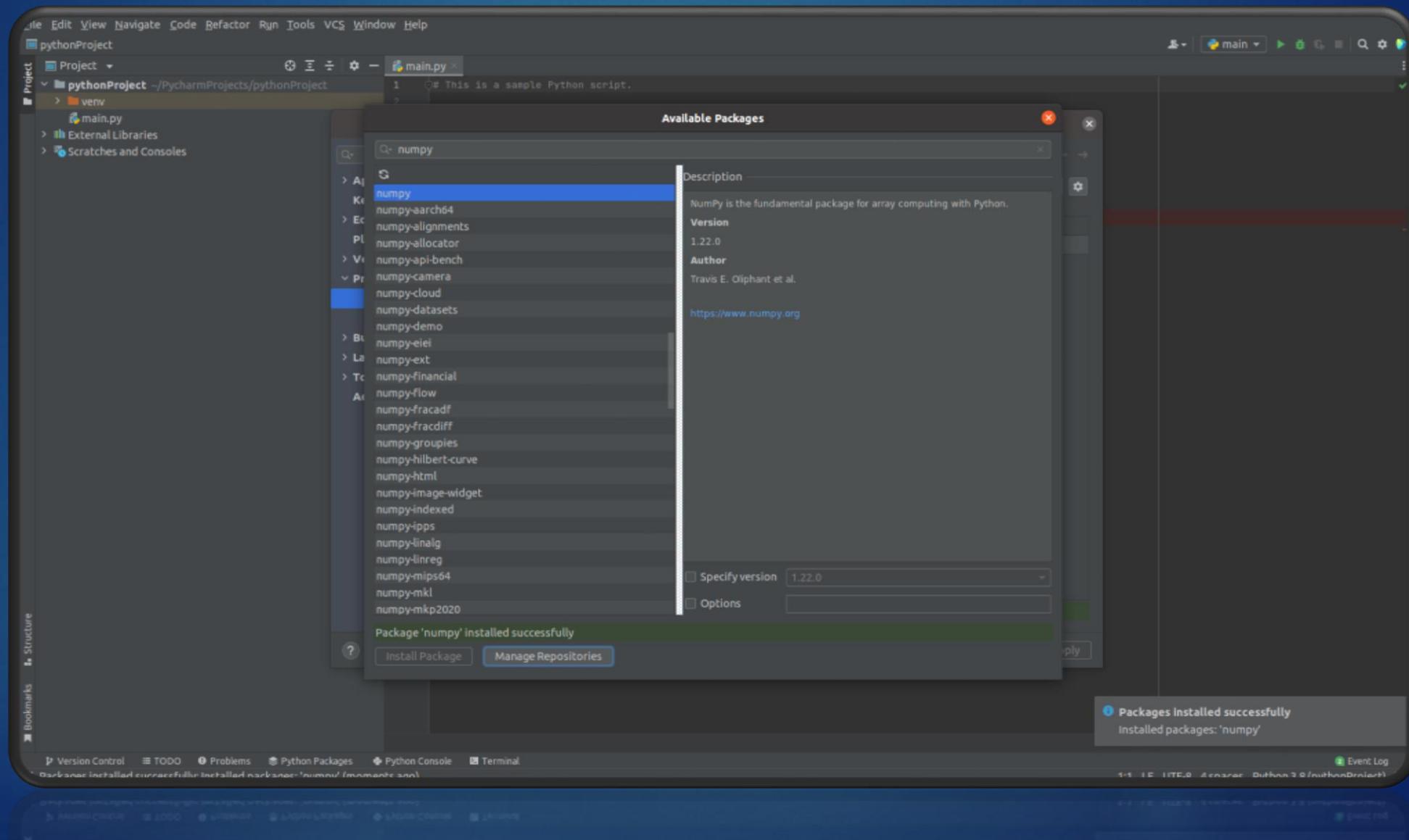
مرحله سوم نصب NumPy در PyCharm



مرحله چهارم نصب NumPy در PyCharm



مرحله آخر نصب NumPy در PyCharm



معکوس کردن و ضرب ماتریس ها

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

- Title Bar:** pythonProject - llinarg.py
- File Menu:** File Edit View Navigate Code Refactor Run Tools VCS Window Help
- Project Tree:** Shows the file llinarg.py.
- Code Editor:** Contains Python code for generating a 3x3 matrix, calculating its inverse, and multiplying it back by the original matrix to verify the result.

```
# numpy در جای خالی در اینجا بنویس
import numpy as np

x = np.random.randint(0, size=(3,3))

print(x)

# اعداد صحیح مثبت که ماتریس آن ۶ و ماتریس آن ۳×۳ است ( ۲ بعدی )
y = np.linalg.inv(x)
print(y)
# معکوس کردن ماتریس با
print(np.matmul(y, x))
# x , y ضرب ماتریس
```

- Run Tab:** Shows the output of running the script.

```
/home/imam/PycharmProjects/pythonProject/venv/bin/python /home/imam/PycharmProjects/pythonProject/llinarg.py
[[1 2 1]
 [1 4 4]
 [4 4 3]]
[[-0.4 -0.2 0.4]
 [ 1.3 -0.1 -0.3]
 [-1.2 0.4 0.2]]
[[ 1.0000000e+00  0.0000000e+00 -1.11022302e-16]
 [-2.22044605e-16  1.0000000e+00 -1.11022302e-16]
 [ 0.0000000e+00 -1.11022302e-16  1.0000000e+00]]

Process finished with exit code 0
```

- Bottom Navigation:** Version Control, Run, TODO, Problems, Python Packages, Python Console, Terminal, Event Log.

توابع جبر خطی

pythonProject - llinarg.py

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject llinarg.py
Project llinarg.py
1 # نumpy در
2
3 import numpy as np
4
5 x = np.random.randint(6, size=(3,3))
6
7 print(x)
8
9 آرایه x با اعداد صحیح مثبت که ما کسیم آن ۶ و سایر آن ۳×۳ است ( ۲ بعدی )
10
11 print(np.trace(x)) # جمع اعضای قطر اصلی را به ما برمیگرداند
12
13 print(np.linalg.det(x)) # دترمینان ماتریس را برمیگرداند
14
15 print(np.linalg.norm(x)) # نرم ماتریس را برمیگرداند
16
17 print(np.linalg.matrix_rank(x)) # رتبه ماتریس را به ما برمیگرداند
18
19 print(np.linalg.matrix_power(x, 5)) # ماتریس را به توان ۵ برمیگرداند
20
21
```

Run: llinarg ×

```
/home/imam/PycharmProjects/pythonProject/venv/bin/python /home/imam/PycharmProjects/pythonProject/llinarg.py
[[3 0 4]
 [0 2 0]
 [1 2 1]]
6
-1.099999999999993
5.916079783899616
3
[[ 987 1840 1220]
 [ 0 32 0]
 [ 305 350 377]]

Process finished with exit code 0
```

Version Control Run TODO Problems Python Packages Python Console Terminal Event Log

DEB R-W301 blank line at end of file 21-1 LF UTF-8 decoded Python 3.8 (PythonProject)

توابع جبر خطی

The screenshot shows a PyCharm IDE window titled "pythonProject - llinarg.py". The code in the editor performs the following steps:

- Imports the numpy library.
- Creates a 3x3 matrix x using np.random.randint with size $(3, 3)$.
- Prints the matrix x .
- Prints the condition number of x using np.linalg.cond(x).
- Performs QR decomposition on x using np.linalg.qr(x) and prints the matrices q and r .
- Prints the result of the dot product $np.dot(q, r)$.

The Run tab shows the output of the script:

```
[[2 3 4]
 [4 0 3]
 [5 1 5]]
54.393729794067086
[[-0.2981424  0.92869742 -0.22052714]
 [-0.59628479 -0.36161669 -0.7367132 ]
 [-0.74535599 -0.08218561  0.66158142]]
[[[-0.78820393 -1.63978318 -6.70820393]
 [ 0.          2.79398664  2.21901153]
 [ 0.          0.          0.27565892]]]
True

Process finished with exit code 0
```

مقدار ویژه و بردارهای ویژه راست

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

- Title Bar:** pythonProject - llinarg.py
- File Menu:** File Edit View Navigate Code Refactor Run Tools VCS Window Help
- Project Tree:** Shows a single file named llinarg.py.
- Code Editor:** Contains Python code to calculate eigenvalues and eigenvectors of a 3x3 matrix. The code includes comments in Persian explaining the steps.

```
# numpy در جبر خطی در
import numpy as np
x = np.random.randint(6, size=(3,3))
print(x)
# آرایه x با اعداد صحیح مثبت که ماتریس آن ۶ و سایز آن ۳x۳ است ( ۲ بعدی )
evalue, evectors = np.linalg.eig(x)
print(evalue) # مقدارهای ویژه
print(evectors) # بردارهای ویژه راست
```

- Run Tab:** Shows the command run: llinarg and the output of the executed code.

```
/home/inan/PycharmProjects/pythonProject/venv/bin/python /home/inan/PycharmProjects/pythonProject/llinarg.py
[[5 2]
 [1 3 2]
 [2 5 5]]
[9.35889894 3. 0.64110106]
[[ 0.67076698  0.88045091  0.52577545]
 [ 0.3164543 -0.17609018 -0.66867059]
 [ 0.67076698 -0.44022545  0.52577545]]

Process finished with exit code 0
```

- Bottom Navigation:** Version Control, Run, TODO, Problems, Python Packages, Python Console, Terminal, Event Log.

سیستم معادلات خطی

The screenshot shows the PyCharm IDE interface with a dark theme. The top bar displays "pythonProject - llinarg.py". The main window shows the code editor with the following Python script:

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject > llinarg.py
pythonProject > llinarg.py x
Project > 1 # numpy جبر خطی در
> 2
> 3 import numpy as np
> 4
> 5 # سیستم معادلات خطی
> 6
> 7
> 8 # x0 + x1 = 2 , x0 + 2*x1 = 0.5
> 9
> 10 a = [[1,1], [1,-2]]
> 11 b = [2,0.5]
> 12
> 13 x = np.linalg.solve(a, b)
> 14
> 15 print(x)
> 16
> 17
> 18
```

The "Run" tab at the bottom shows the output of running the script:

```
Run: llinarg x
/home/imam/PycharmProjects/pythonProject/venv/bin/python /home/imam/PycharmProjects/pythonProject/llinarg.py
[1.5 0.5]
Process finished with exit code 0
```

The bottom navigation bar includes tabs for Version Control, Run, TODO, Problems, Python Packages, Python Console, Terminal, and Event Log.

با تشکر از توجه شما