
Аналіз даних в Python та R

Data science with Python and R

Осінь 2023/2024

Викладач:

Розова Людмила Вікторівна

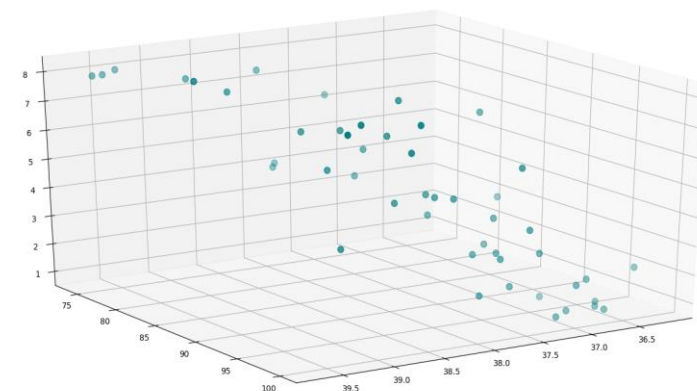
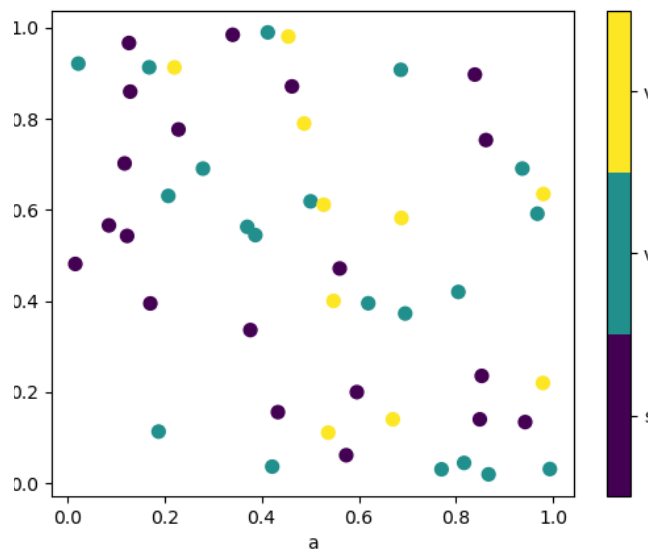
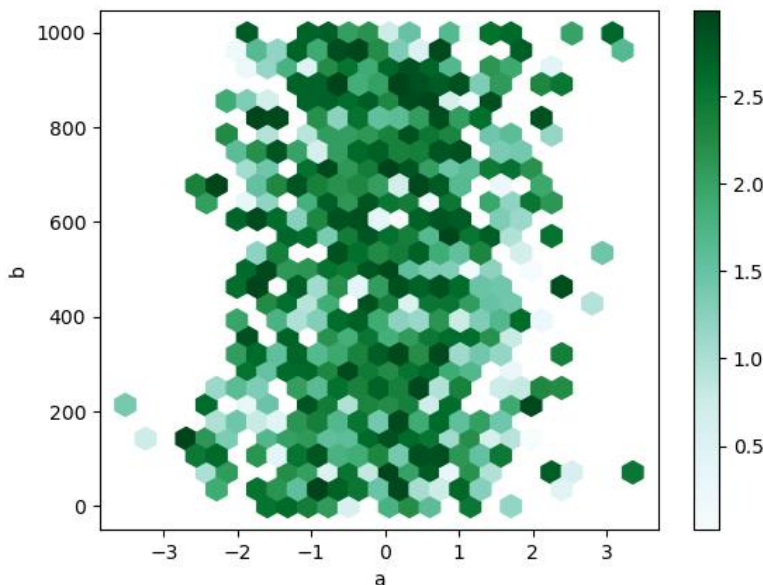
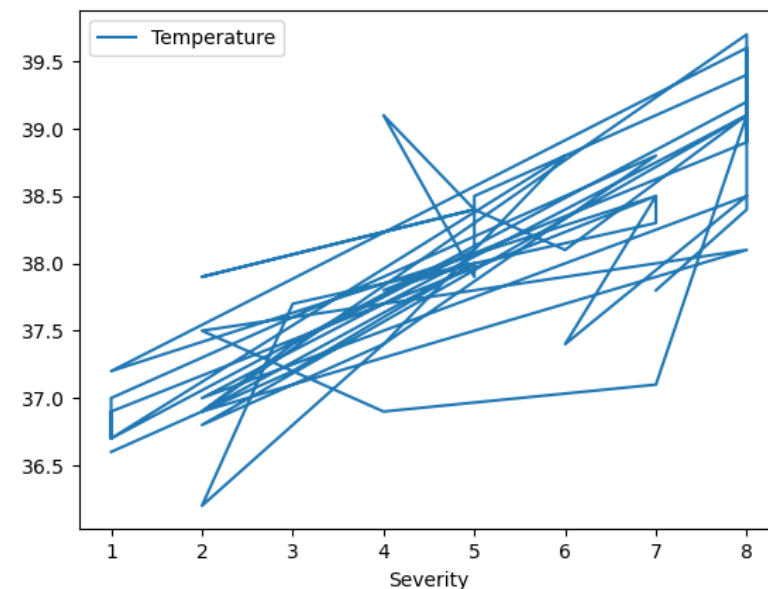
Чому необхідно обробляти дані для аналізу?

2

У нас більше даних, ніж будь-коли раніше.

Але самі собою дані не можуть багато розповісти нам про світ навколо нас. Необхідно інтерпретувати інформацію та виявляти приховані закономірності.

Тут на допомогу приходить наука про дані.



[illegible]

Що робить спеціалісти по даним Data Scientists?

4

Розв'язують вище перелічені проблеми:

- Визначають, які саме дані потрібно для тієї, чи іншої задачі
 - Знаходять шляхи отримання даних
 - Підготовлюють та обробляють великі обсяги даних.
 - Прибирають зайві дані
 - Створюють моделі
 - Тестують моделі, допрацьовують
 - Запускають їх у виробництво
-

Якими навичками повинен мати Data Scientist?

Знання:

- ✓ Статистичного аналізу та математики, які є важливими стовпами науки про дані.
- ✓ Машинного навчання — цінний інструмент пошуку закономірностей у великих наборах даних.
- ✓ Роботи з базами даних. Мова структурованих запитів (SQL) є обов'язковою навичкою для фахівців за даними. Проте популярність нереляційних баз даних (NoSQL) також зростає
- ✓ Мови програмування Python, R...

Які мови програмування популярні у науці про дані?

Python

- найпопулярніша мова програмування для науки про дані.

Це універсальна мова, для якої є безліч бібліотек.

R

Це мова призначена саме для статистичного аналізу

SQL

це мова запитів, розроблена для реляційних баз даних.

Python проти R: у чому різниця?

Вибираючи між Python та R, вам потрібно подумати про свої конкретні потреби.

- З одного боку, Python відносно легко освоїти, застосовується в багатьох дисциплінах, має суворий синтаксис

Велика бібліотека Python — одна з його найбільших переваг, оскільки вона надає програмістам різні інструменти, що підходять для вирішення різних завдань

- З іншого боку, R має понад 10 000 пакетів для маніпулювання даними, здатний легко створювати графіку публікаційної якості, має чудові можливості статистичного моделювання і ширше використовується в наукових колах, охороні здоров'я та фінансах.

Python – мультипарадигмальна мова, підтримує безліч підходів до програмування.

Python підтримує об'єктно-орієнтований стиль програмування.

Це дозволяє створити в коді об'єкти Python, які представляють об'єкти реального світу. Ці об'єкти можуть містити як дані, і функціональні можливості реального об'єкта.

Щоб створити об'єкт у Python, потрібний клас.

Класи Python мають атрибути для представлення даних та методи, які додають функціональності

Які обмеження Python?

Оскільки Python — це інтерпретована, динамічно типізована мова, вона повільніша порівняно з компільованою, статично типізованою мовою, такою як C. Тому Python корисний, коли швидкість не така важлива.

R

R — це мова програмування, призначена для математиків, статистиків та бізнес-аналітиків, у яких може не бути досвіду програмування.

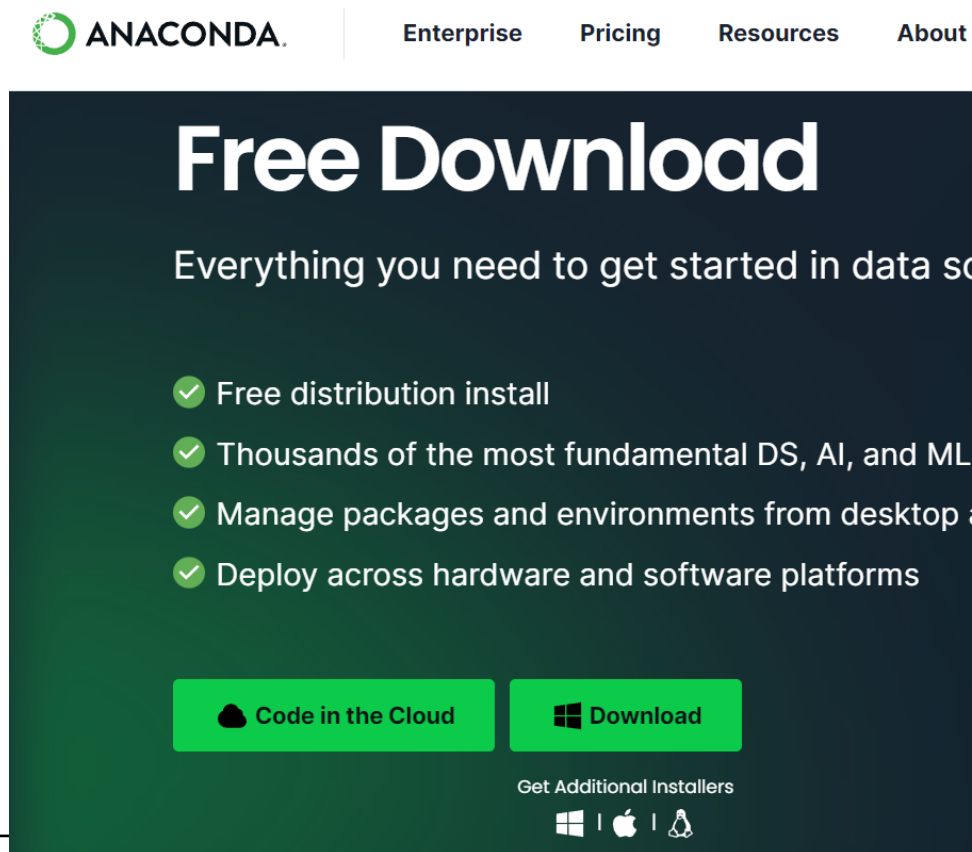
- Вона була створена спеціально для статистичного програмування. Корисна для обробки, очищення, аналізу та подання даних.
- R також є популярною мовою для проектів з науки даних. Більшість даних, що використовуються в науці про дані, можуть бути безладними і складними. У мові програмування R є функції та бібліотеки, призначені для очищення неорганізованих даних та спрощення обробки складних структур даних, яких немає в інших мовах.
- Вона також надає потужні інструменти візуалізації даних, які знаходити закономірності у великих наборах даних та подавати результати у виразних звітах.
- Машинне навчання - ще одна область, в якій корисна мова R. R надає розробникам широкий вибір бібліотек машинного навчання, які допоможуть їм знаходити тенденції даних і прогнозувати майбутні події.

Anaconda 3 for Windows Installer (Python 3.8 version)

<https://www.anaconda.com/download/>

Anaconda

—
дистрибутив для мов програмування Python та R з відкритим кодом для обробки даних великого об'єму, побудови аналітичних прогнозів і наукових обчислень.



The screenshot shows the Anaconda website's 'Free Download' section. At the top, the Anaconda logo is on the left, and navigation links for 'Enterprise', 'Pricing', 'Resources', and 'About' are on the right. The main heading 'Free Download' is in large white text. Below it, a subtitle reads 'Everything you need to get started in data science on your workstation.' A list of four features with green checkmarks follows: 'Free distribution install', 'Thousands of the most fundamental DS, AI, and ML packages', 'Manage packages and environments from desktop application', and 'Deploy across hardware and software platforms'. At the bottom, there are two green buttons: 'Code in the Cloud' with a cloud icon and 'Download' with a Windows icon. Below these buttons, the text 'Get Additional Installers' is followed by icons for Windows, macOS, and Linux.

ANACONDA. Enterprise Pricing Resources About

Free Download

Everything you need to get started in data science on your workstation.

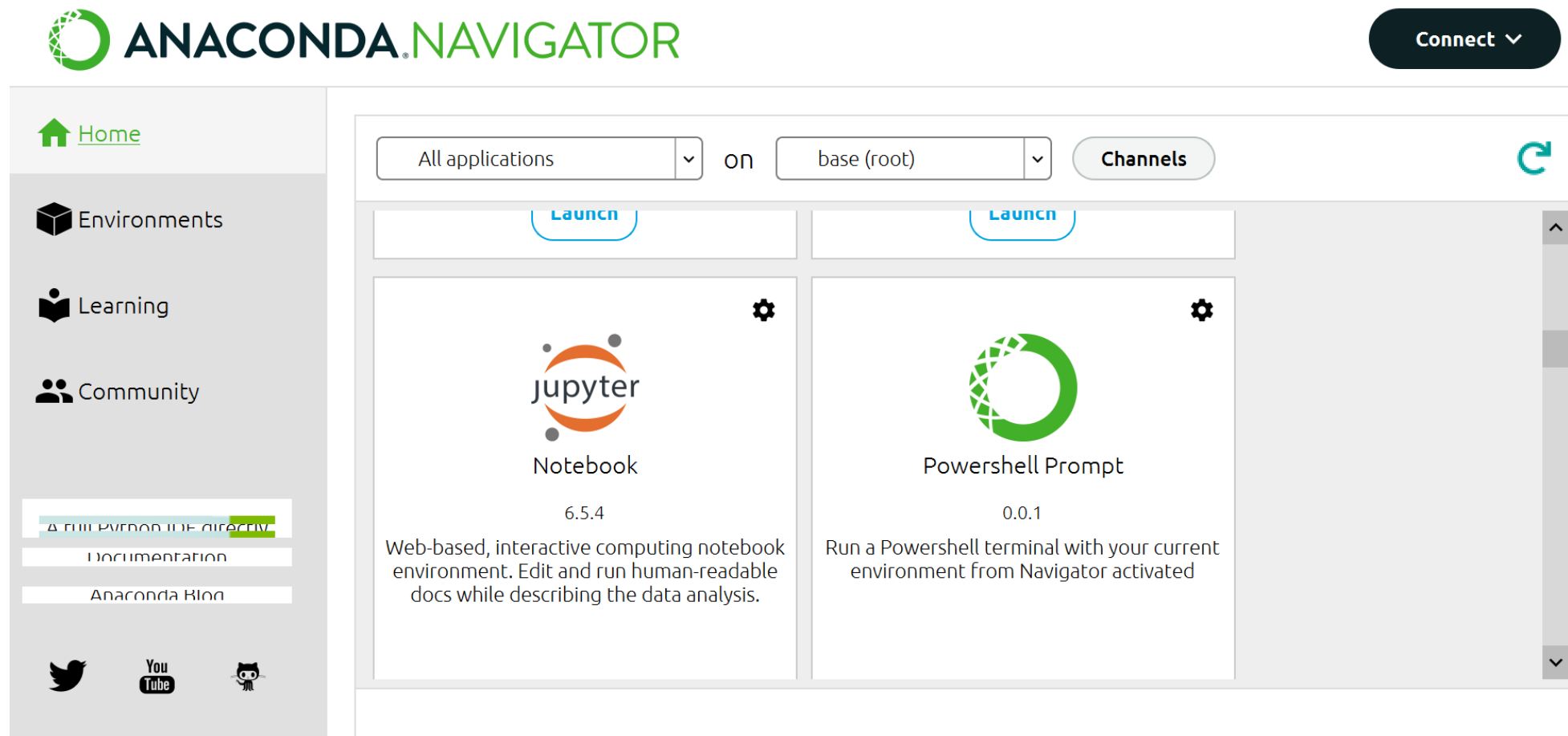
- ✓ Free distribution install
- ✓ Thousands of the most fundamental DS, AI, and ML packages
- ✓ Manage packages and environments from desktop application
- ✓ Deploy across hardware and software platforms

Code in the Cloud Download

Get Additional Installers

Windows macOS Linux

Програмне забезпечення



Програмне забезпечення



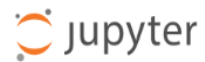
Quit Logout

Files Running Clusters

Select items to perform actions on them.

Upload New ↕

<input type="checkbox"/> 0	▼	📁 /	Name ▼	
<input type="checkbox"/>		📁 anaconda3		
<input type="checkbox"/>		📁 Contacts		
<input type="checkbox"/>		📁 Documents		
<input type="checkbox"/>		📁 Downloads		
<input type="checkbox"/>		📁 Favorites		7 місяців назад
<input type="checkbox"/>		📁 Links		7 місяців назад



Quit Logout

Files Running Clusters

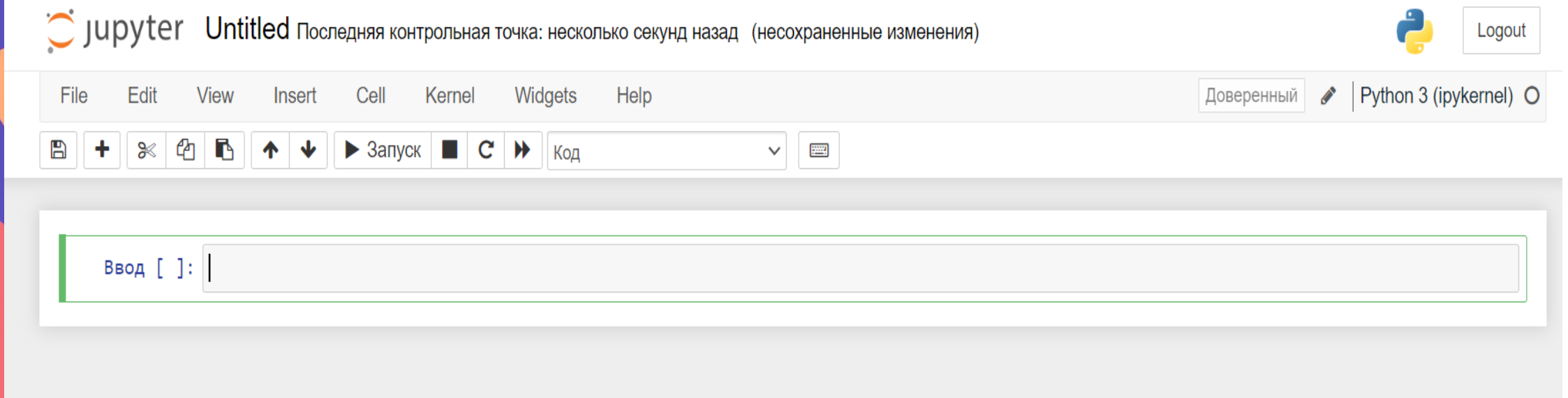
Select items to perform actions on them.

Upload New ↕

<input type="checkbox"/> 0	▼	📁 / Python_lessons	Name ▼	
		📁 ..		несколько
Список блокнотов пуст.				

Notebook:
Python 3 (ipykernel)
Other:
Text File
Folder
Terminal

Програмне забезпечення



Numbers

Integer

Positive or negative
No decimal point

Float

Positive or negative
With decimal point

Complex

$A+BJ$
A = Real Part
BJ = Imaginary Part

Python. Rewiev. Data type in Python

```
In [10]: a = 10
```

```
In [11]: type(a)
```

```
Out[11]: int
```

```
In [12]: b = 99.9
```

```
In [15]: type(b)
```

```
Out[15]: float
```

```
In [16]: c = 45.j
```

```
In [17]: type(c)
```

```
Out[17]: complex
```


Strings

Sequence of characters, literal constant

Text

'Python'
&
"Python"

Print and Input functions

```
In [19]: print("Welcome Python")
```

Welcome Python

```
In [20]: Name = input("What's your name?")
```

What's your name?John

```
In [21]: Name
```

```
Out[21]: 'John'
```

Python. Rewiev.

Conversions of types

```
In [22]: Variable_1 = 100
```

```
In [24]: type(Variable_1)
```

```
Out[24]: int
```

```
In [25]: float(Variable_1)
```

```
Out[25]: 100.0
```

```
In [27]: type(float(Variable_1))
```

```
Out[27]: float
```

```
In [28]: str(Variable_1)
```

```
Out[28]: '100'
```

Operators in Python

Arithmetic Operators

Comparison Operators

Assignment Operators

Logical Operators

```
In [10]: number = int(input("Please enter a number: "))  
if number < 0:  
    print("Your number is negative.")  
else:  
    print("Your number is positive.")
```

Python. Rewiev.

Conditions

```
In [10]: number = int(input("Please enter a number: "))
if number < 0:
    print("Your number is negative.")
else:
    print("Your number is positive.")
```

```
Model = int(input("Please choose 1-3: "))

if Model == 1:
    print("You choose Rolls-Royce.")
elif Model == 2:
    print("You choose Aston Martin.")
elif Model == 3:
    print("You choose Bentley.")
```

Loops

```
In [1]: i = 0
while i < 15:
    print(i)
    i += 1
```

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

```
for number in range(0,15):
    print(number)
```

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14

Sequences

Lists

Tuples

Dicts

Sets

```
MyList = [1,2,3,4,5,6,7,8]
```

```
type(MyList)
```

```
list
```

```
MyList_2 = [1,2,3,"apple","banana",13,25,'a','b',13.5]
```

```
type(MyList_2)
```

```
list
```

```
MyList_3 = [1,2,3,[1,2,3],"apple","banana",["apple","banana"]]
```

```
type(MyList_3)
```

```
list
```

```
MyTuple = (1,2,3,4,5,6,7,8)
```

```
type(MyTuple)
```

```
tuple
```

```
MyTuple_2 = (1,2,3,"apple","banana",13,25,"a","b",13.5)
```

```
type(MyTuple_2)
```

```
tuple
```

```
MyTuple_3 = (1,2,3,[1,2,3],"apple","banana",["apple","banana"])
```

```
type(MyTuple_3)
```

```
tuple
```

Python. Rewiev.

```
MyDict = {"Zero": 0, "One": 1, "Two": 2, "Three": 3, "Four": 4, "Five": 5}
```

```
type(MyDict)
```

```
dict
```

```
MyDict_E = {}
```

```
type(MyDict_E)
```

```
dict
```

```
MyDict_2 = {"January":31, "February":28,
```

```
type(MyDict_2)
```

```
dict
```

Modules

```
import
```

```
dir(ma
```

```
[ '_do  
, '_lo 18  
, '_name__',  
, '_package__',  
, '_spec__',  
, 'acos',  
, 'acosh',
```

```
def Sum(A,B):  
    """This function sum the variables"""  
    print(A+B)
```

```
Sum(4,5)
```

```
9
```

```
def Sum(A,B,C,D):  
    print(A+B+C+D)
```

```
def Multiply(A):  
    print(A*2)
```

```
Sum(3,4,5,6)
```

```
MySet = {1,2,3,4,5}
```

```
e(MySet)
```

```
et_2 = {"a","b","c","d","e"}
```

```
e(MySet_2)
```

```
or y).
```

```
s to radians.
```

```
remainder(x, y, 7)  
Difference between x and the closest integer multiple of y.
```

Дякую за увагу!

