Introduction
and your first python app

Python basic for everyone

NORDIC
< CODER >

Why Python

1. Dynamic programming language
2. Interactive language
3. Zen of python

Why Python - Zen of Python

- ***Readable***: Intuitive and strict syntax
- ***Productive***: saves a lot of code
- ***Portable:*** For every operating system
- ***Reloaded***: It comes with many libraries

What is Python used for?

- **Web Development**
    - Frameworks such as *Django*, *Flask*
- **Data Analysis**
    - Libraries such as NumPy and Pandas
    - Data visualisation libraries like Matplotlib and Seaborn
- **Internet Of Things**
    - Raspberry Pi + python
- **Web Scraping**
    - *Example*: https://scrapy.org/
- **Computer Vision**
    - *OpenCV* library
- **Machine Learning**
    - Libraries such as *Scikit-Learn*, *NLTK* and *TensorFlow*.

Python working Environments

1. Web interactive environment
2. Local development environment
3. Large scale, production environment

Web interactive environment

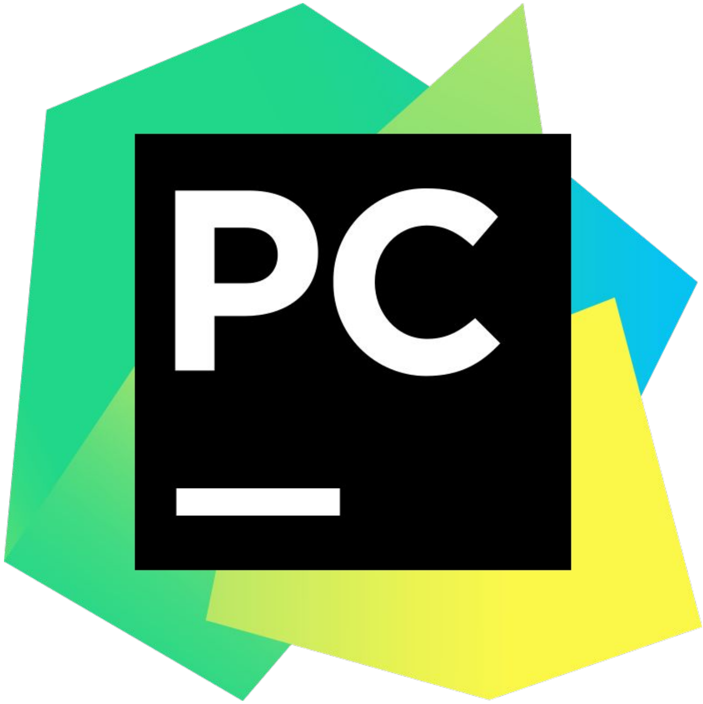
When:

1. Quick test and sharing.
2. Use where you don't have your laptop or working env setup for you.

https://repl.it/languages/python3

https://www.pythonanywhere.com

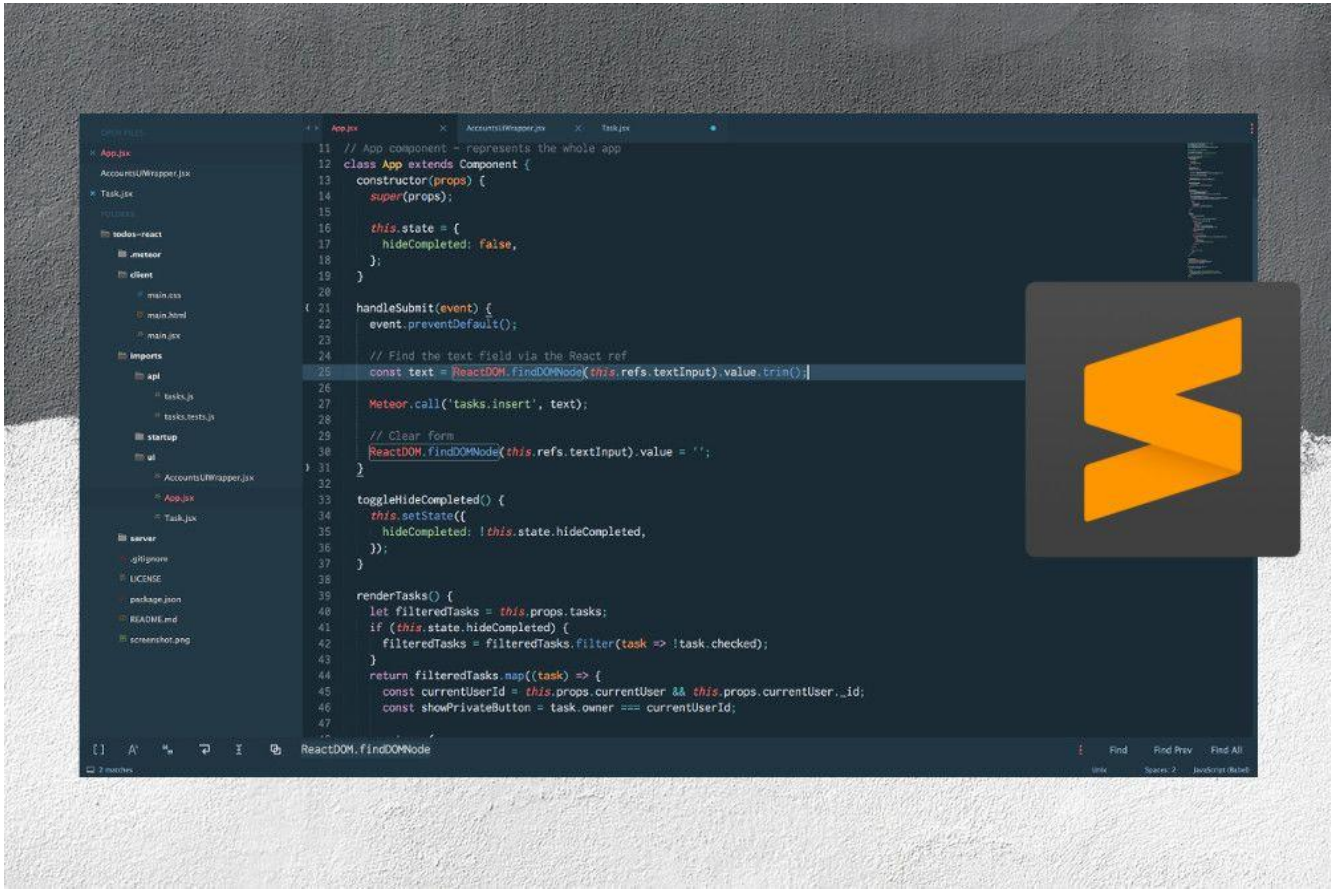
Local development environment



When:

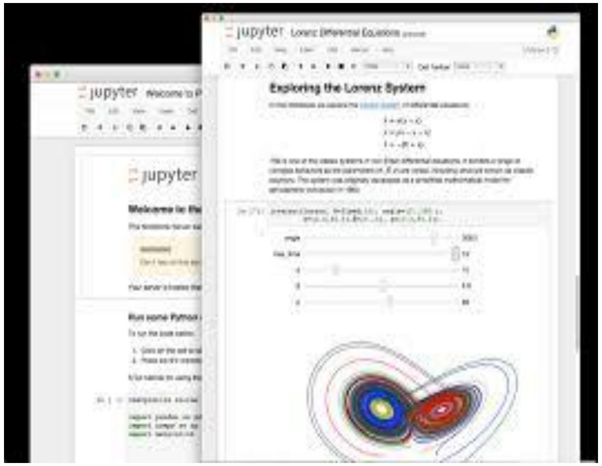
Develop most of your applications, scripts in Python.

Local development environment



```
// App component - represents the whole app
class App extends Component {
  constructor(props) {
    super(props);

    this.state = {
      hideCompleted: false,
    };
  }

  handleSubmit(event) {
    event.preventDefault();

    // Find the text field via the React ref
    const text = ReactDOM.findDOMNode(this.refs.textInput).value.trim();

    Meteor.call('tasks.insert', text);

    // Clear form
    ReactDOM.findDOMNode(this.refs.textInput).value = '';
  }

  toggleHideCompleted() {
    this.setState({
      hideCompleted: !this.state.hideCompleted,
    });
  }

  renderTasks() {
    let filteredTasks = this.props.tasks;
    if (this.state.hideCompleted) {
      filteredTasks = filteredTasks.filter(task => !task.checked);
    }
    return filteredTasks.map((task) => {
      const currentUserId = this.props.currentUser && this.props.currentUser._id;
      const showPrivateButton = task.owner === currentUserId;
```

Local development environment



Exploring the Lorenz System

When:

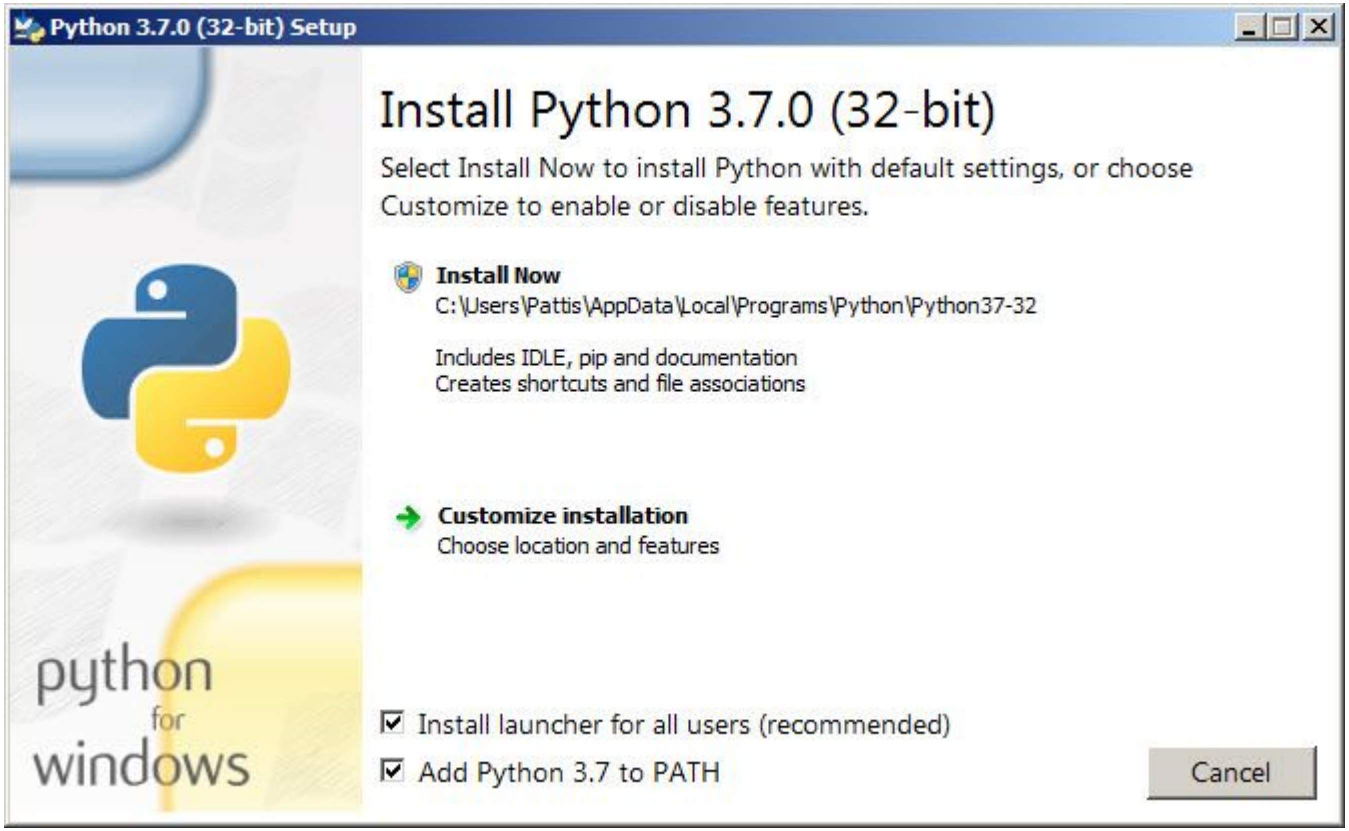
Create and share documents that contain live code, equations, visualizations and narrative text.

Large scale, production environment

When:

1. You publish your application to end users
2. You need big machine resource to run machine learning, data jobs ...

Installing Python



Python 3.7.0 (32-bit) Setup

Install Python 3.7.0 (32-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\Pattis\AppData\Local\Programs\Python\Python37-32

Includes IDLE, pip and documentation
Creates shortcuts and file associations

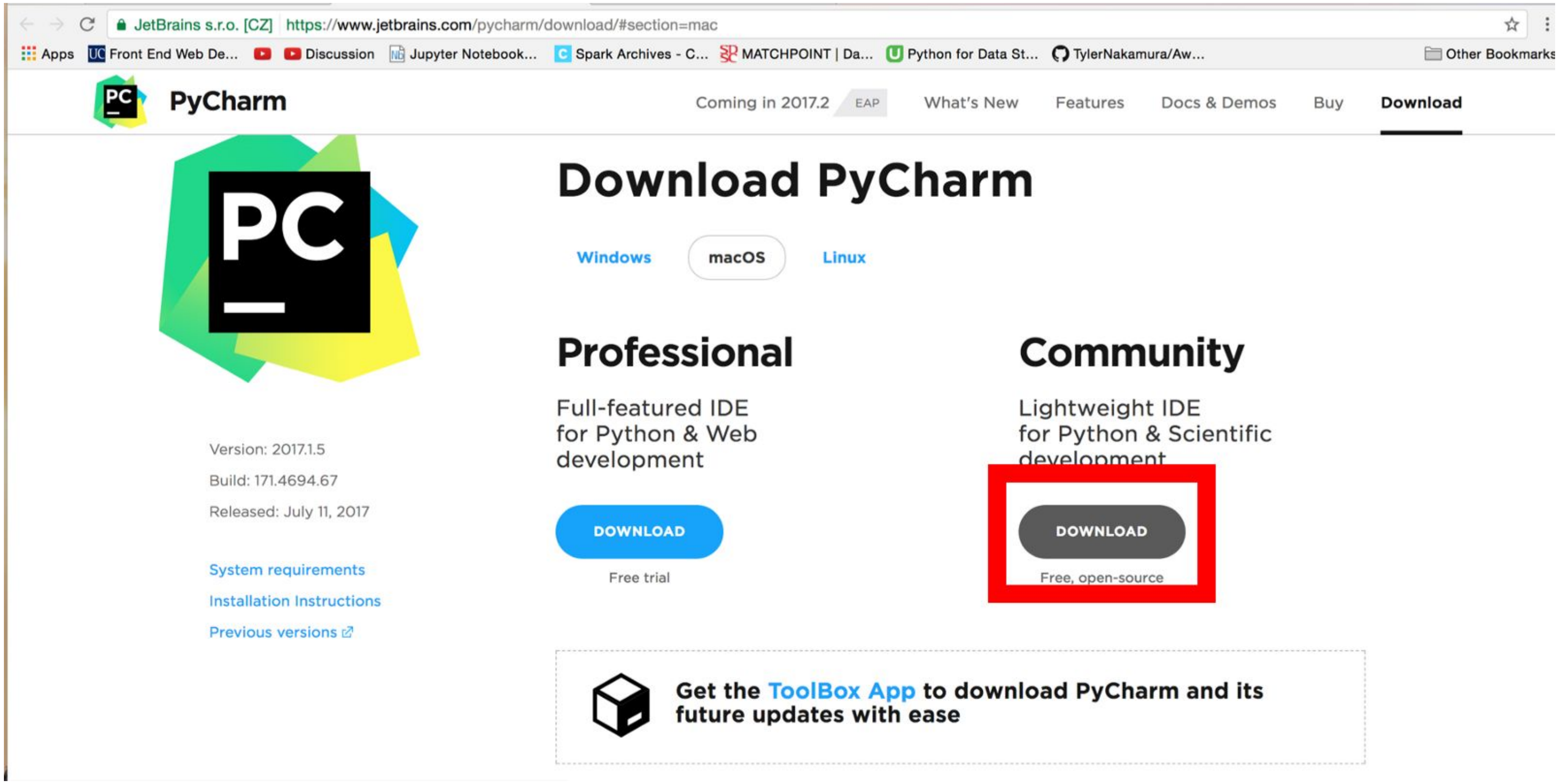
Customize installation
Choose location and features

☑ Install launcher for all users (recommended)

☑ Add Python 3.7 to PATH

https://www.python.org/downloads/

Installing text editor



Download PyCharm

Version: 2017.1.5

Build: 171.4694.67

Released: July 11, 2017

System requirements

Installation Instructions

Previous versions

Professional

Full-featured IDE for Python & Web development

Free trial

Community

Lightweight IDE for Python & Scientific development

Free, open-source

Get the ToolBox App to download PyCharm and its future updates with ease

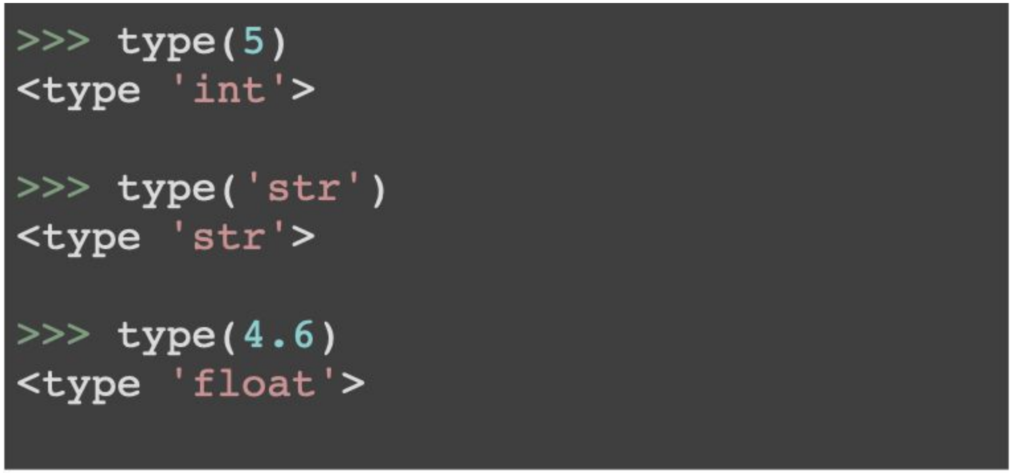
https://www.jetbrains.com/pycharm/download/#section=mac

Get help !!!

1. Python help()
2. Peers
3. Stackoverflow, Google

Data Types

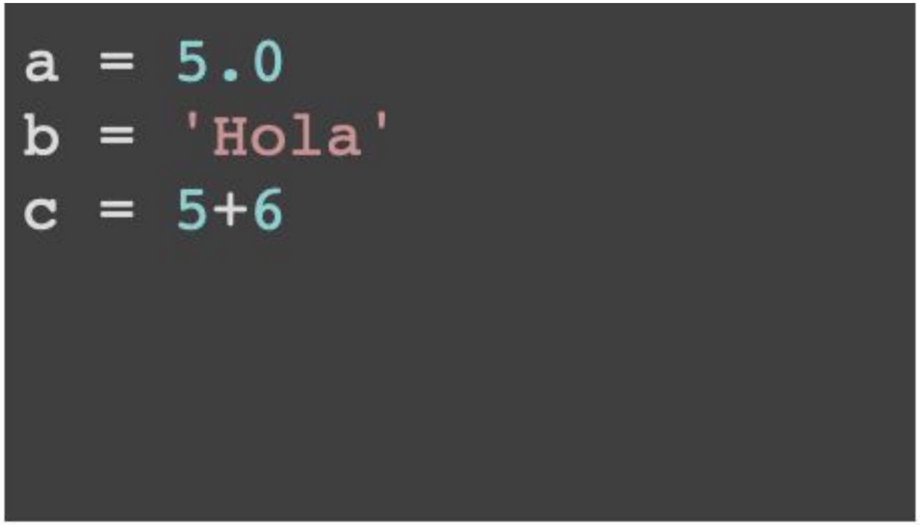
- Integer <int>
- Float <float>
- String <str>
- Boolean <bool>
- etc



```
>>> type(5)
<type 'int'>

>>> type('str')
<type 'str'>

>>> type(4.6)
<type 'float'>
```

Variables

Variables are used to store information to be referenced and manipulated in a computer program



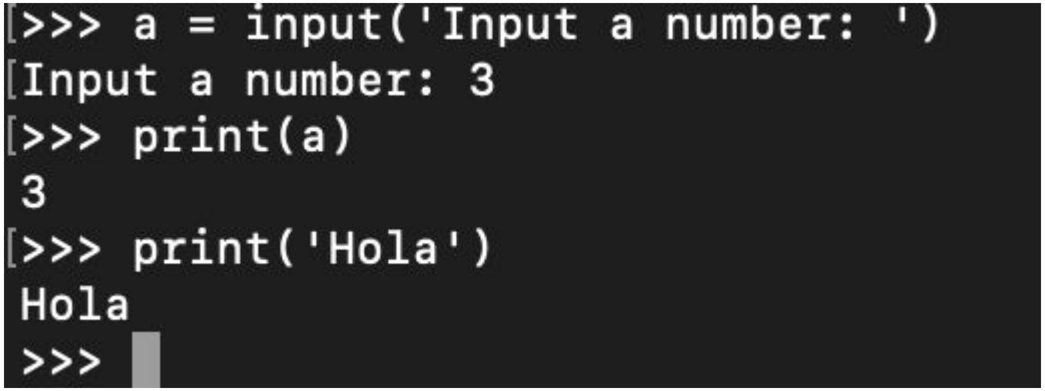
```
a = 5.0
b = 'Hola'
c = 5+6
```

Math operators

- + plus
- - minus
- /
- * multiply
- < less-than
- < greater-than
- <= less-than-equal
- >= greater-than-equal

Input/Output operators

Input and output (I/O) operators are used to take input and display output.

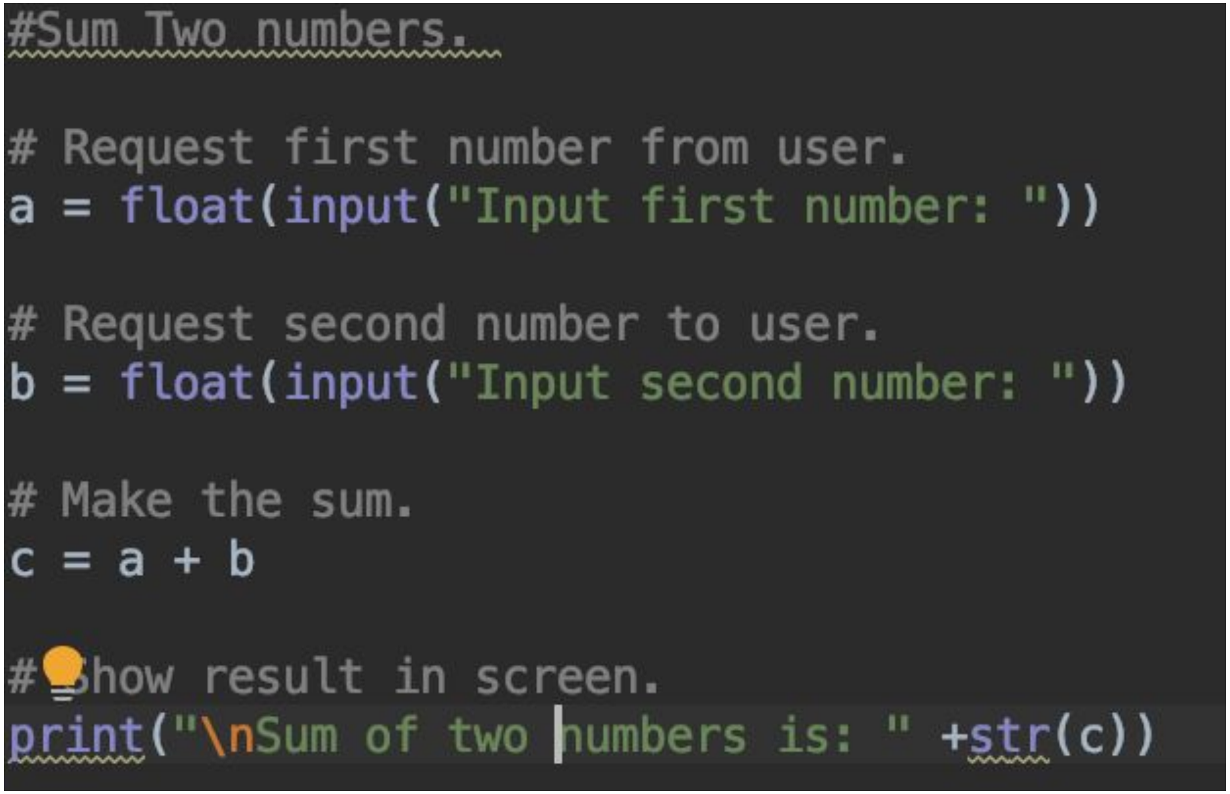


```
[>>> a = input('Input a number: ')
[Input a number: 3
[>>> print(a)
3
[>>> print('Hola')
Hola
>>> 
```

Exercise 1

1. Write a script that request two number from user
2. Make a sum with the two numbers and save result in a variable.
3. Finally print result variable

Answer 1



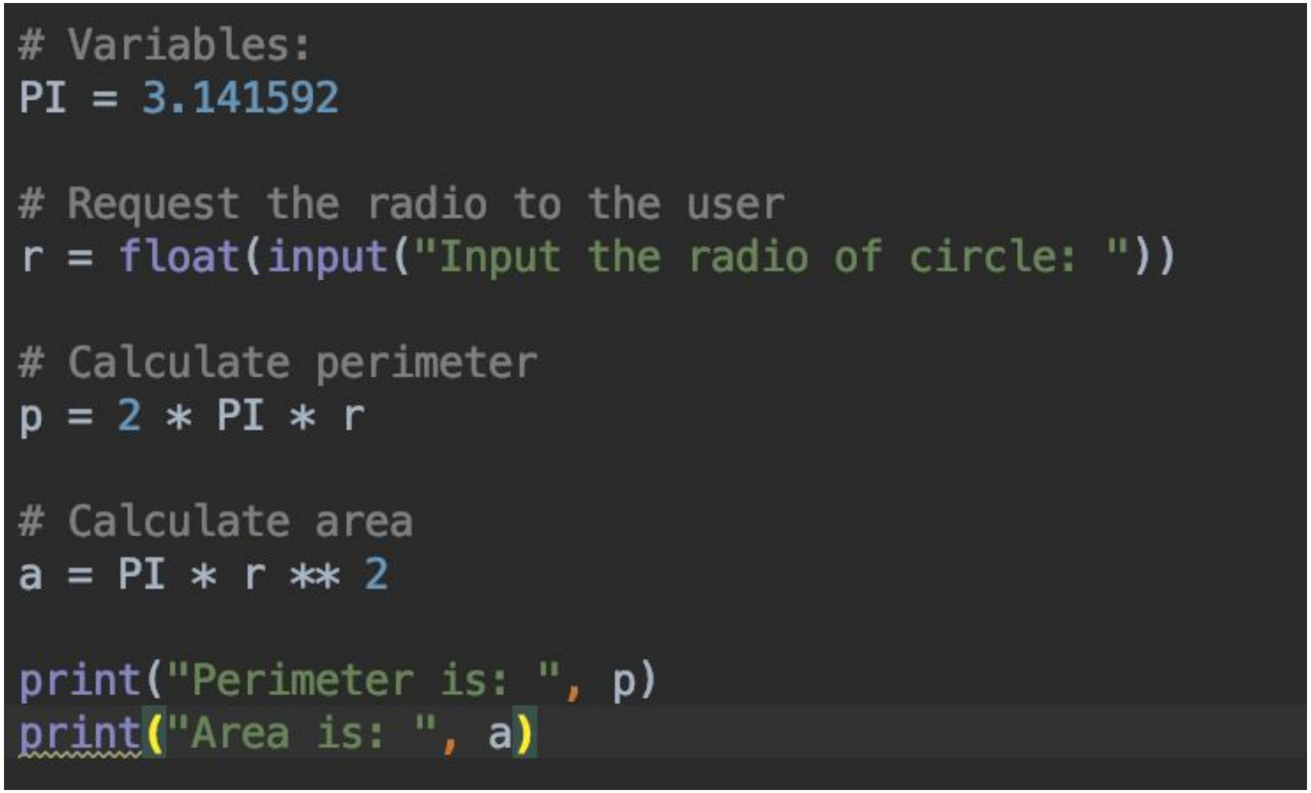
```python
#Sum Two numbers.

# Request first number from user.
a = float(input("Input first number: "))

# Request second number to user.
b = float(input("Input second number: "))

# Make the sum.
c = a + b

# Show result in screen.
print("\nSum of two numbers is: " +str(c))
```

Exercise 2

Create an algorithm to calculate perimeter and area of a circle

1. Create a variable with pi value
2. Request radio to the user
3. Calculate perimeter and save in variable
4. Calculate area and save in variable
5. Print in screen perimeter and area variables

Answer 2



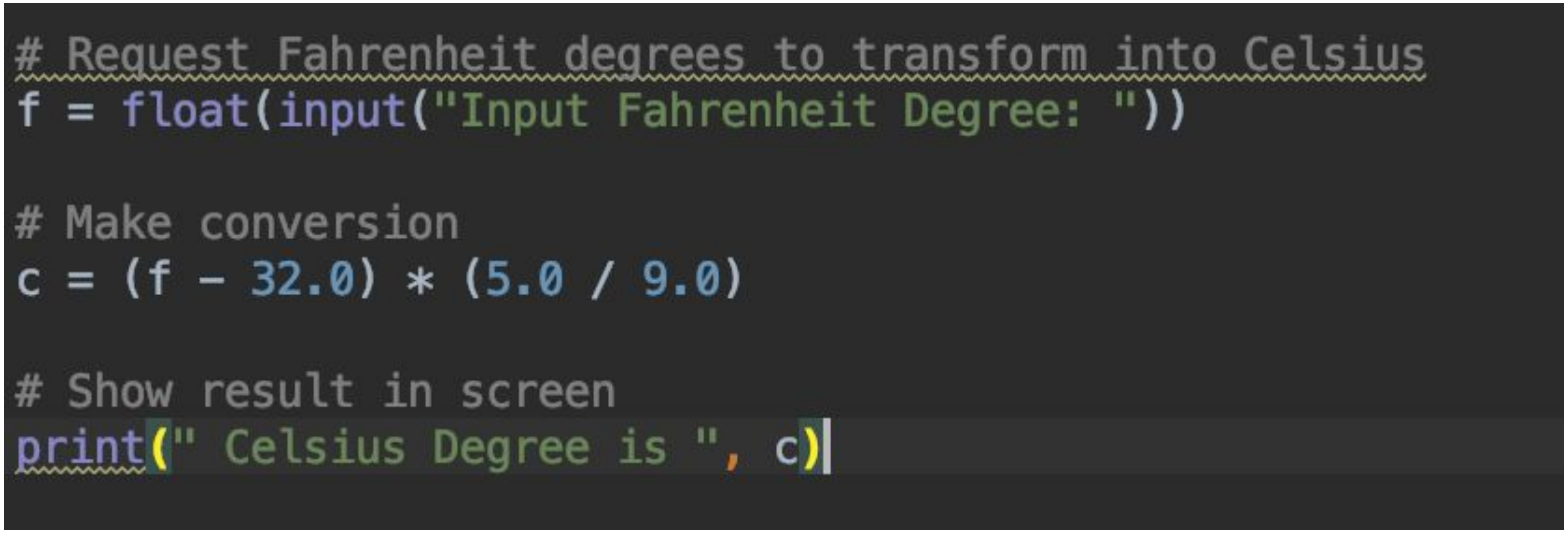
```python
# Variables:
PI = 3.141592

# Request the radio to the user
r = float(input("Input the radio of circle: "))

# Calculate perimeter
p = 2 * PI * r

# Calculate area
a = PI * r ** 2

print("Perimeter is: ", p)
print("Area is: ", a)
```

Exercise 3

Algorithm to transform Fahrenheit to Celsius

1. Request Fahrenheit to user
2. Make transformation
3. Show result in screen

Answer 3



```
# Request Fahrenheit degrees to transform into Celsius
f = float(input("Input Fahrenheit Degree: "))

# Make conversion
c = (f - 32.0) * (5.0 / 9.0)

# Show result in screen
print(" Celsius Degree is ", c)
```

NORDIC
< CODER >