I
E & C
T
IIT Guwahati

Advanced Certification Programme in Data Science Business Analytics

Python Fundamentals

Topics Covered

- What is Python?
- Introduction to Python Syntax
- Hands on Exercise- Python Data Types
- Hands on Exercise- expressions
- Hands on Exercise- String operations
- Async Recap
- Q & A

Async Recap

1. Installing Anaconda for Python Development
We covered the steps to download and install Anaconda, a distribution that simplifies Python setup and package management.

2. Python's Versatility and Key Features
Python is a general-purpose, high-level language known for its simplicity, readability, and extensive libraries. Its features include being easy to learn, open source, object-oriented, and portable.

3. Basic Python Syntax Rules
We explored fundamental syntax rules such as case sensitivity, the importance of indentation for defining code blocks, the use of comments for explaining code, and writing statements.

4. Launching and Using Jupyter Notebooks
Jupyter Notebook provides an interactive environment for writing and running Python code, visualizing data, and collaborating. We learned how to launch notebooks and create our first Python program.

5. Essential String Operations in Python
We introduced string operations, which are used to manipulate, query, and transform text. Common string methods include lower(), upper(), strip(), replace(), split(), find(), and count()..

What is Python?

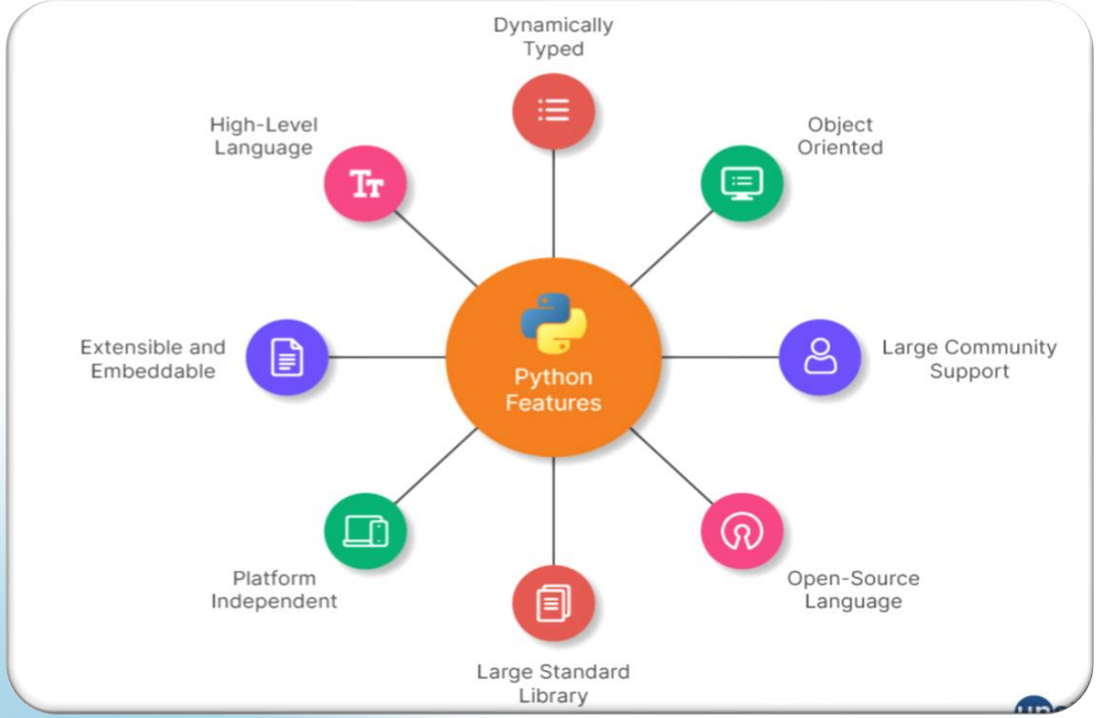
Introduction to Python

Simple to Learn, Powerful in Use

- Discover a rare mix of simplicity and power
- Focus on solving problems, not syntax
- Get surprised by its ease of use
- Ignore complex structures while coding
- Enjoy programming without distraction

Features of Python

Why It’s a Popular Choice?

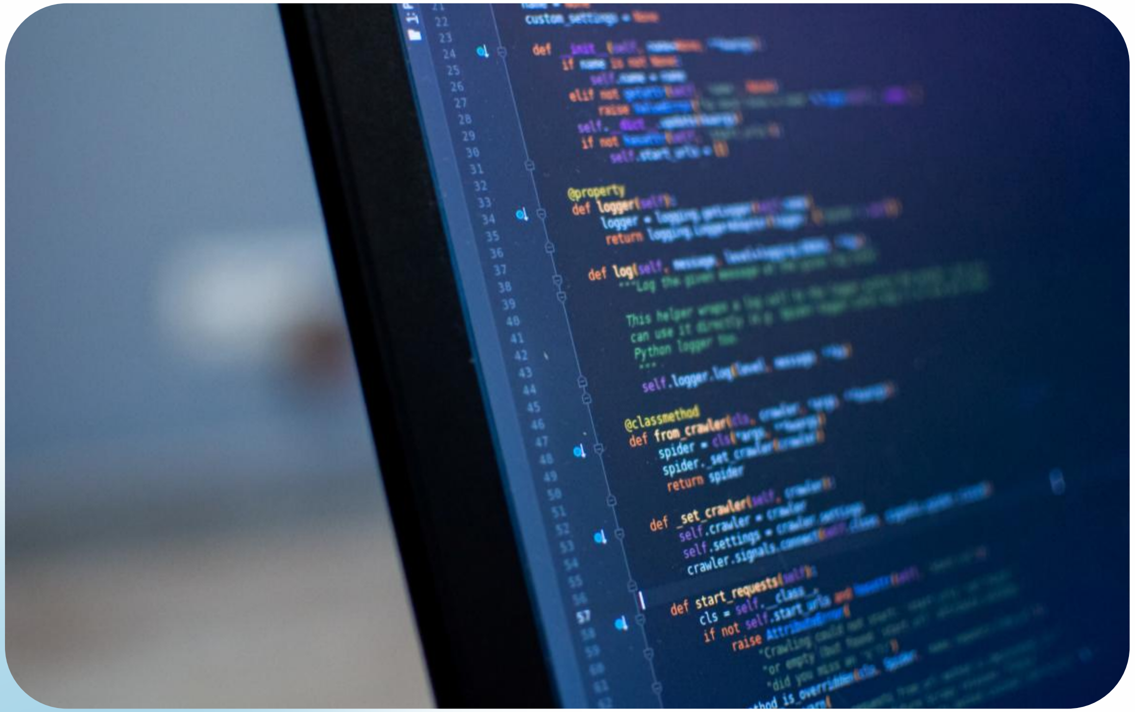


- Dynamically typed
- Object oriented
- Large community support
- Open-source language
- Large standard library
- Platform independent
- Extensible and embeddable
- High-level language

Introduction to Python Syntax

Syntax Rules in Python

The Foundation of Clean and Readable Code

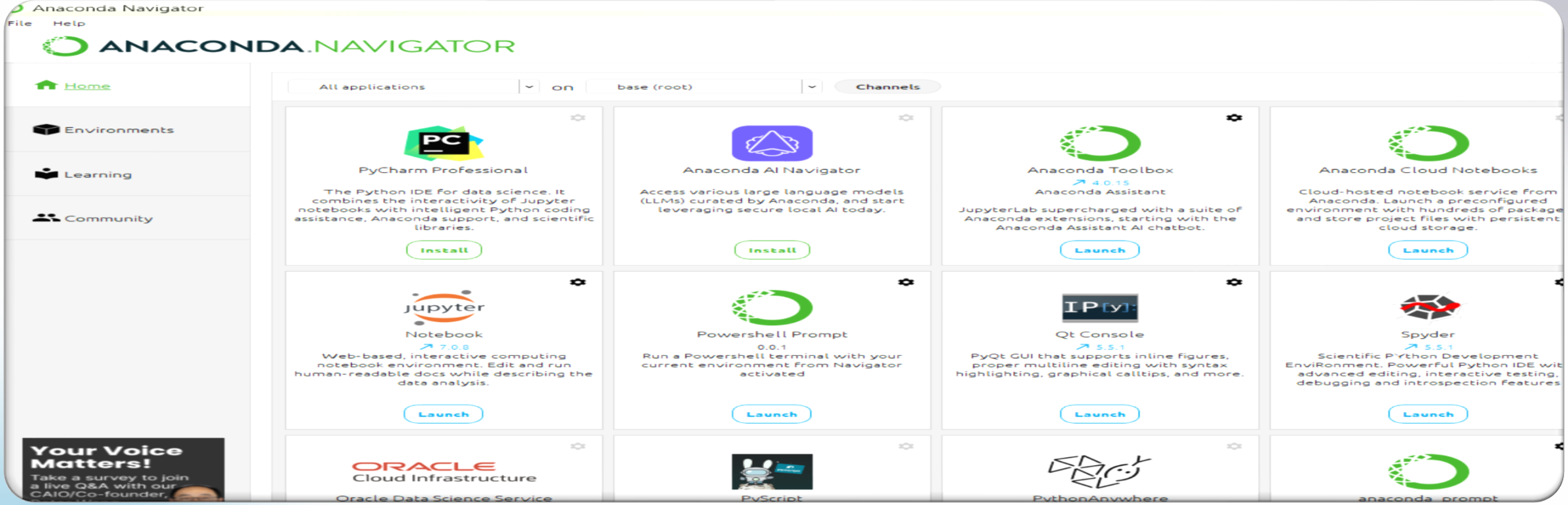


Reduces the need for extensive boilerplate code

- Case sensitivity
- Indentation
- Comments
- Statements
- No semicolons required

Launch a New Notebook

Start a New Python Workspace



The Python IDE for data science. It combines the interactivity of Jupyter notebooks with intelligent Python coding assistance, Anaconda support, and scientific libraries.

Access various large language models (LLMs) curated by Anaconda, and start leveraging secure local AI today.

Anaconda Assistant

JupyterLab supercharged with a suite of Anaconda extensions, starting with the Anaconda Assistant AI chatbot.

Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of package and store project files with persistent cloud storage.

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Run a Powershell terminal with your current environment from Navigator activated

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Scientific PYthon Development EnviRonment. Powerful Python IDE wit advanced editing, interactive testing, debugging and introspection features

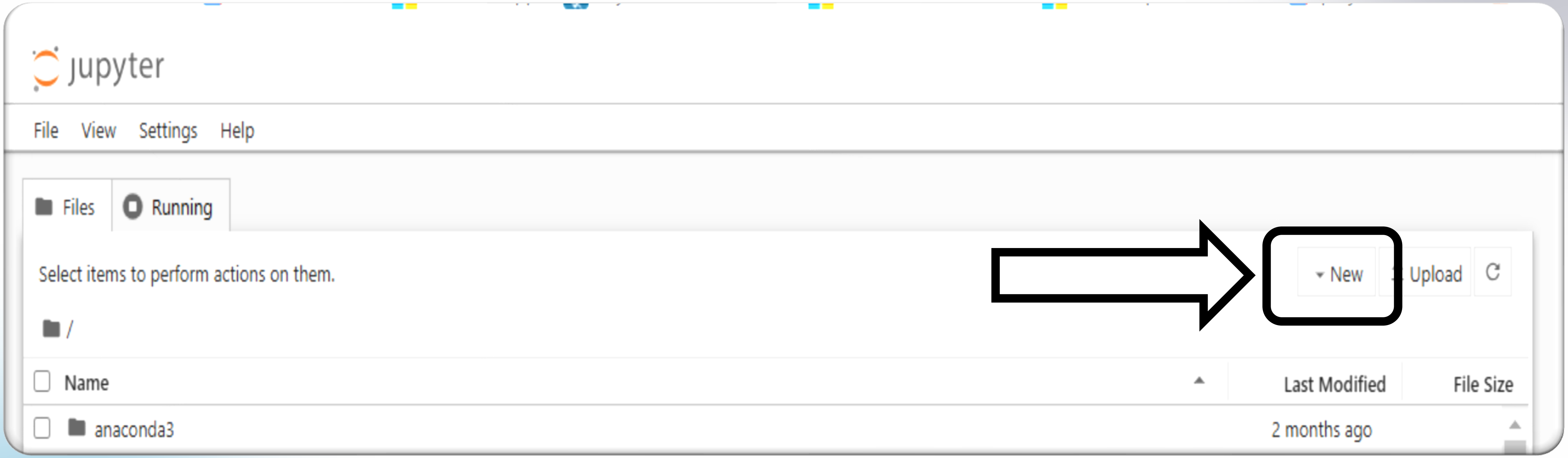
Your Voice Matters!

Take a survey to join a live Q&A with our CAIO/Co-founder,

- Launch the Jupyter notebook using the Anaconda navigator

Launch a New Notebook

Start Writing Your First Program



jupyter

Select items to perform actions on them.

/

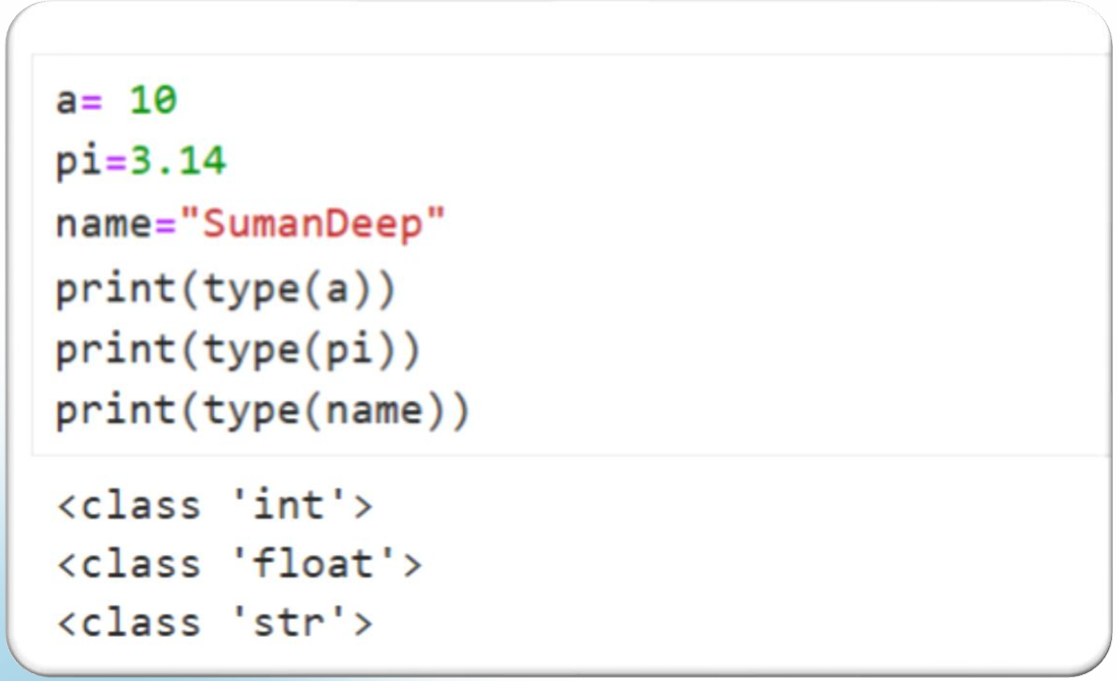
☐ Name	Last Modified	File Size
☐ anaconda3	2 months ago	

- Take a new notebook and start writing your first program

Hands on Exercise- Python Data Types

Practical Exercises on Python Data Types

Activity 1: Check the Data Type



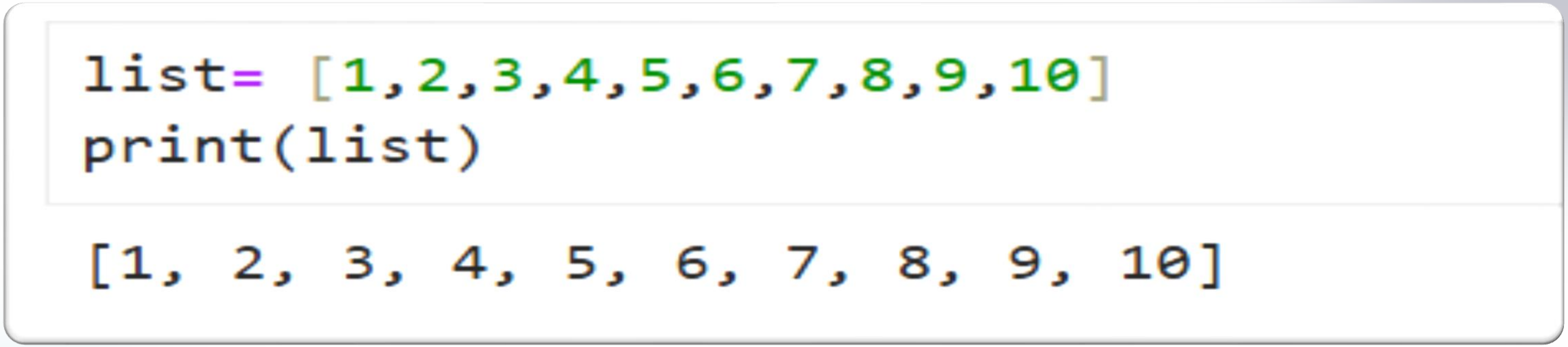
```
a= 10
pi=3.14
name="SumanDeep"
print(type(a))
print(type(pi))
print(type(name))
```

```
<class 'int'>
<class 'float'>
<class 'str'>
```

- Take three variables
- Assign values of different data type
- Check their data type

Practical Exercises on Python Data Types

Activity 2: Create a List



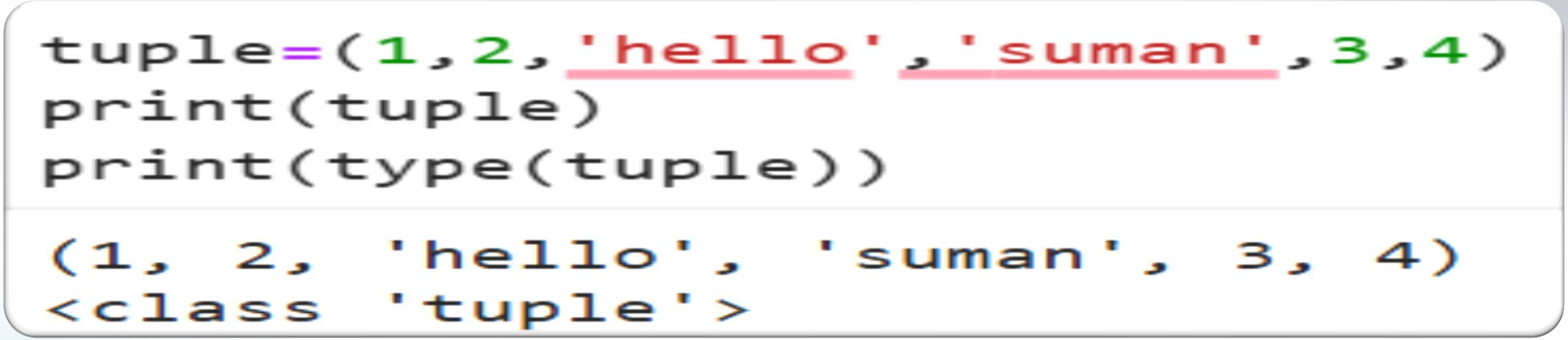
```
list= [1,2,3,4,5,6,7,8,9,10]
print(list)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

- Create a list showing numbers 1 to 10

Practical Exercises on Python Data Types

Activity 3: Create a Tuple



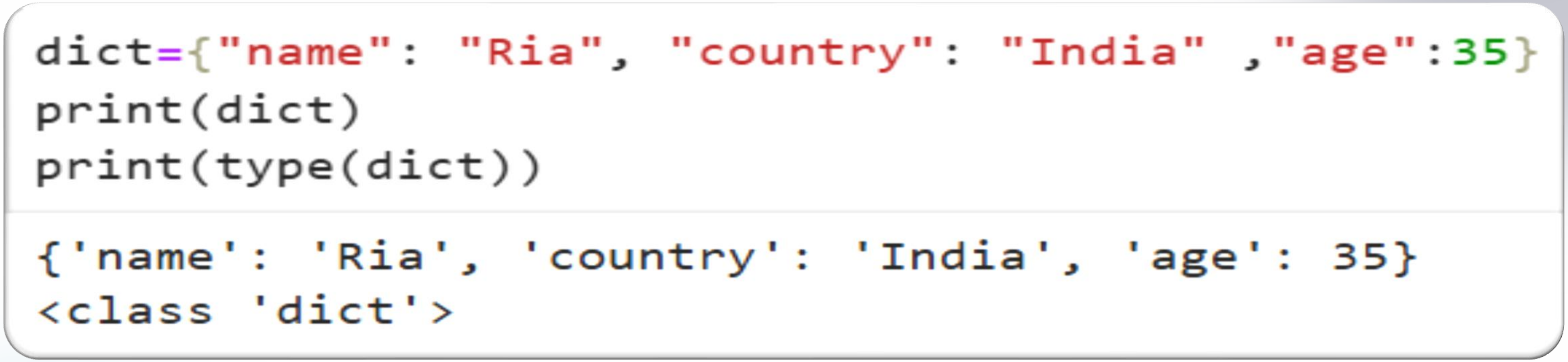
```
tuple=(1,2,'hello','suman',3,4)
print(tuple)
print(type(tuple))
```

```
(1, 2, 'hello', 'suman', 3, 4)
<class 'tuple'>
```

- Create a tuple showing numbers following data (1,2,'hello','suman',3,4)

Practical Exercises on Python Data Types

Activity 4: Create a Dictionary



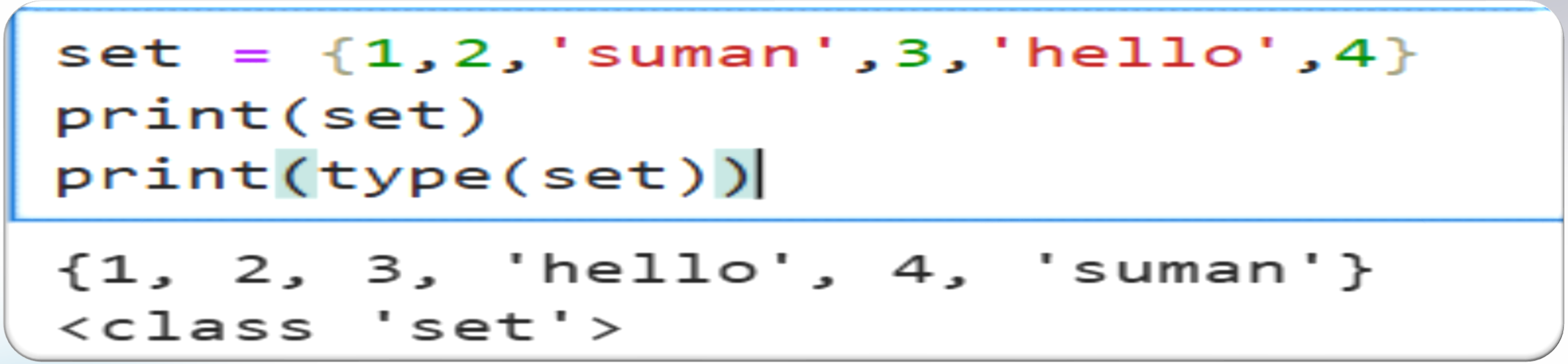
```
dict={"name": "Ria", "country": "India" ,"age":35}
print(dict)
print(type(dict))
```

```
{'name': 'Ria', 'country': 'India', 'age': 35}
<class 'dict'>
```

- Create a dictionary to show following data
- {"name": "Ria", "country": "India" ,"age":35}

Practical Exercises on Python Data Types

Activity 5: Create a Set



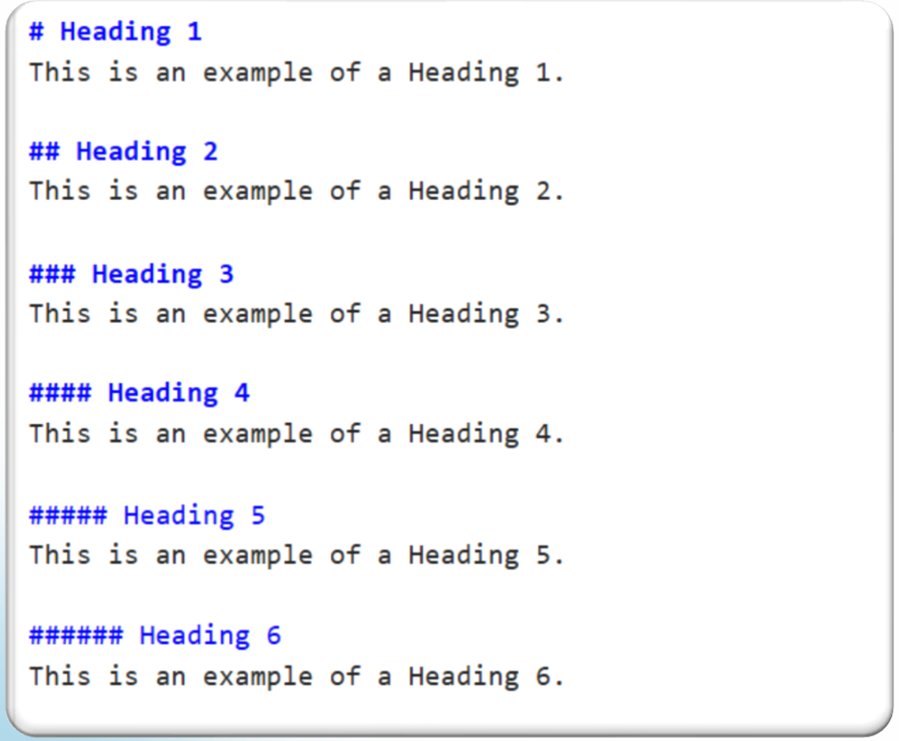
```
set = {1,2,'suman',3,'hello',4}
print(set)
print(type(set))
```

```
{1, 2, 3, 'hello', 4, 'suman'}
<class 'set'>
```

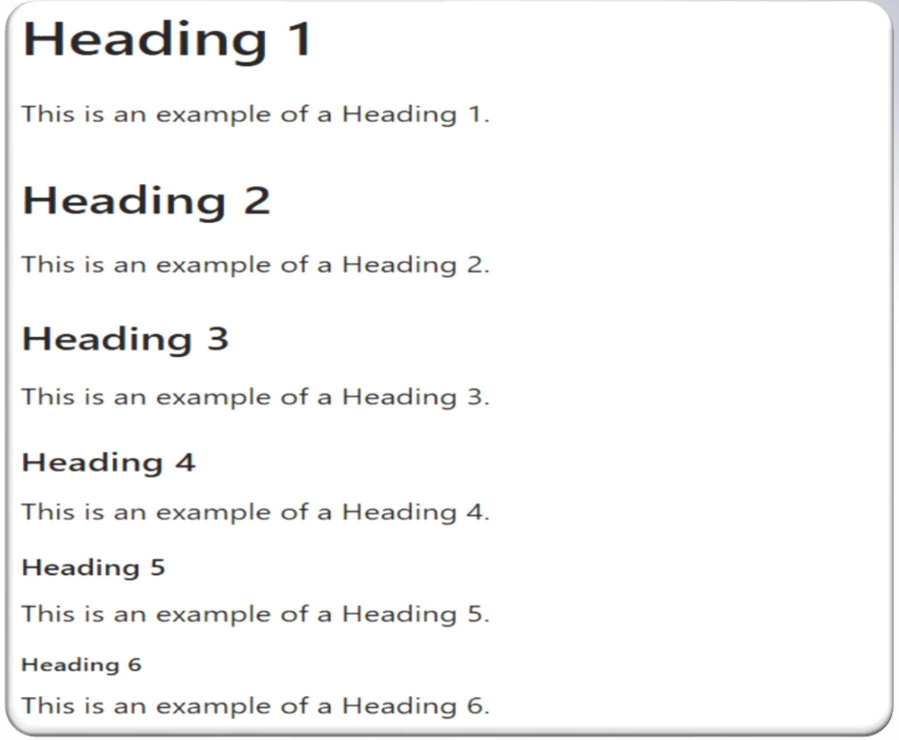
- Create a set showing following data
- {1,2,'suman', 3,'hello',4}

Practical Exercises on Python Data Types

Activity 6: Create a markdown cell



```
# Heading 1
This is an example of a Heading 1.

## Heading 2
This is an example of a Heading 2.

### Heading 3
This is an example of a Heading 3.

#### Heading 4
This is an example of a Heading 4.

##### Heading 5
This is an example of a Heading 5.

###### Heading 6
This is an example of a Heading 6.
```



Heading 1

This is an example of a Heading 1.

Heading 2

This is an example of a Heading 2.

Heading 3

This is an example of a Heading 3.

Heading 4

This is an example of a Heading 4.

Heading 5

This is an example of a Heading 5.

Heading 6

This is an example of a Heading 6.

- Create a markdown cell with different heading styles using the markdown syntax

List vs tuple

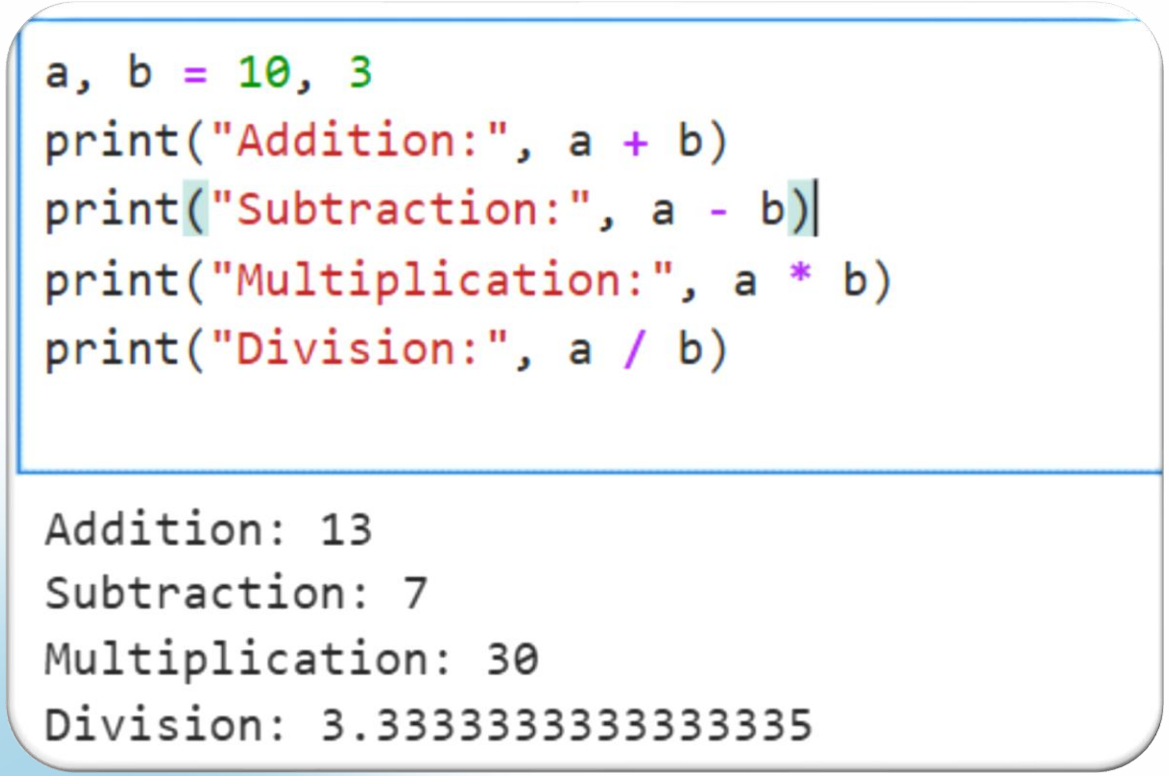
Knowing the Difference

Feature	List	Tuple
Definition	A mutable, ordered collection of items.	An immutable, ordered collection of items.
Syntax	Defined using square brackets [ ].	Defined using parenthesis ( ).
Mutability	Mutable: Elements can be added, removed or changed	Immutable: Once created, it cannot be modified.
Performance	Slower due to mutability overhead	Faster due to immutability and fixed size.
Use case	Used when elements may need to be modified	Used when elements should not change, such as constants.
Size	Dynamic: Can grow or shrink as needed	Fixed: Size is determined when created.

Hands on Exercise- Expressions

Practical Exercises on Expressions

Activity : Arithmetic Expressions



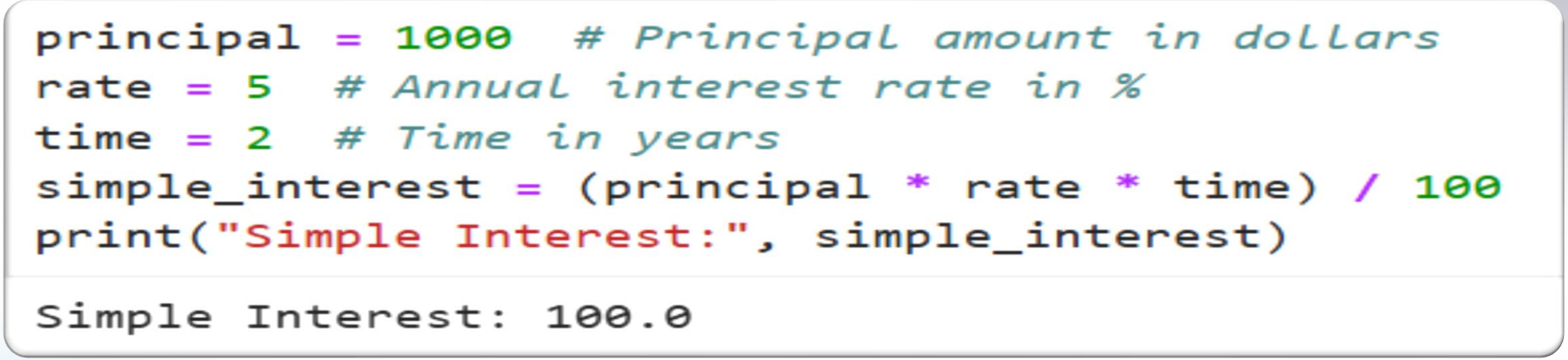
```
a, b = 10, 3
print("Addition:", a + b)
print("Subtraction:", a - b)
print("Multiplication:", a * b)
print("Division:", a / b)
```

```
Addition: 13
Subtraction: 7
Multiplication: 30
Division: 3.3333333333333335
```

- Take two variable and assign values to them
- Perform addition, subtraction, multiplication, division, and modulus operations

Practical Exercises on Expressions

Activity : Simple Interest Calculator



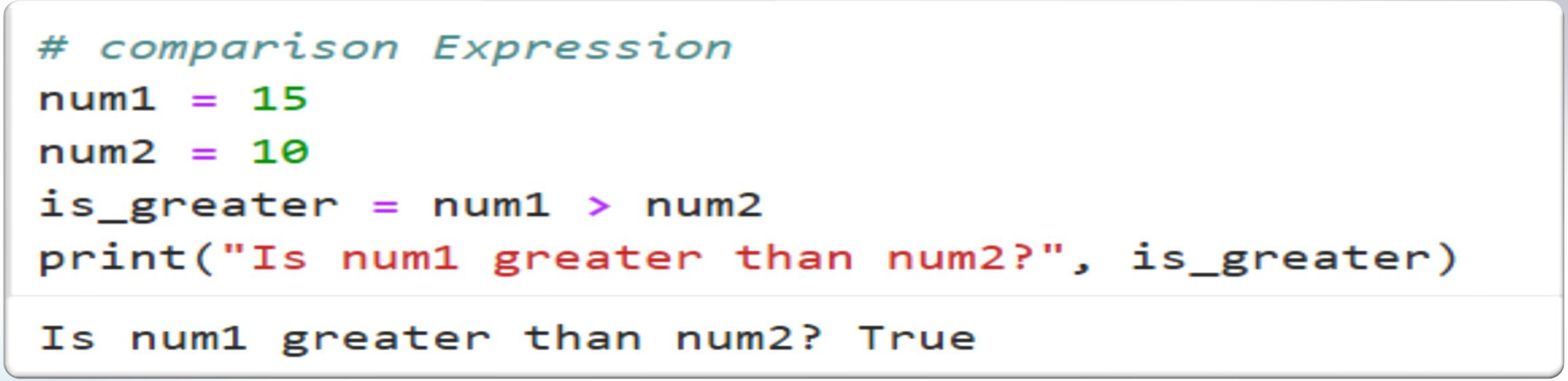
```
principal = 1000  # Principal amount in dollars
rate = 5  # Annual interest rate in %
time = 2  # Time in years
simple_interest = (principal * rate * time) / 100
print("Simple Interest:", simple_interest)
```

```
Simple Interest: 100.0
```

- Calculate simple interest SI as, SI = $\frac{P.T.R}{100}$

Practical Exercises on Expressions

Activity : Comparison Expressions



```python
# comparison Expression
num1 = 15
num2 = 10
is_greater = num1 > num2
print("Is num1 greater than num2?", is_greater)
```

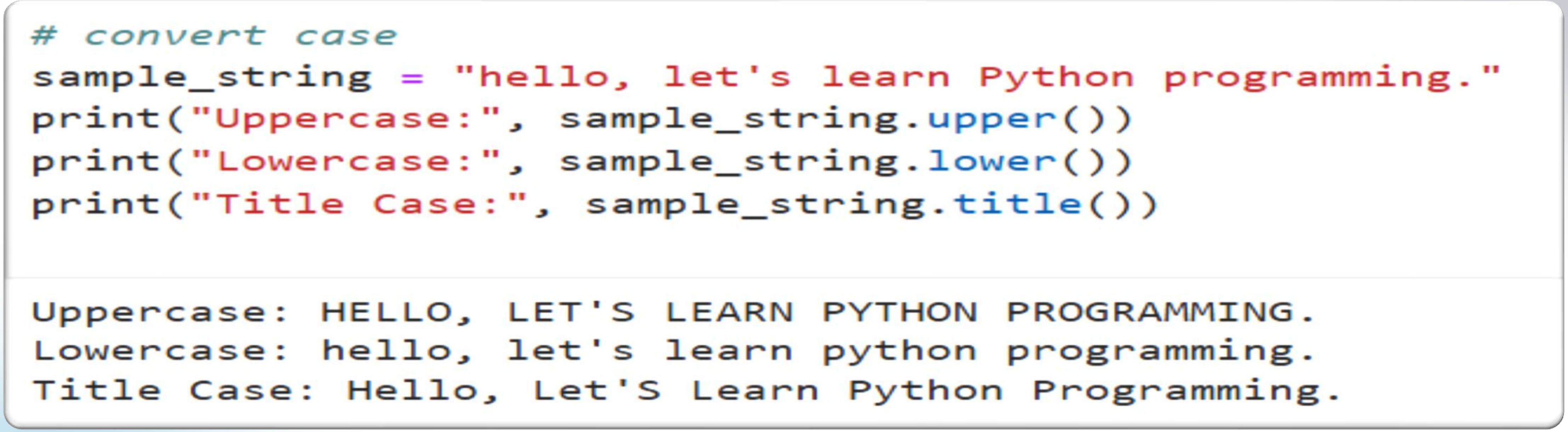
```
Is num1 greater than num2? True
```

- Take two variable compare them using any relational operator

Hands on Exercise- String operations

Practical Exercises on String Operations

Activity : Convert Case



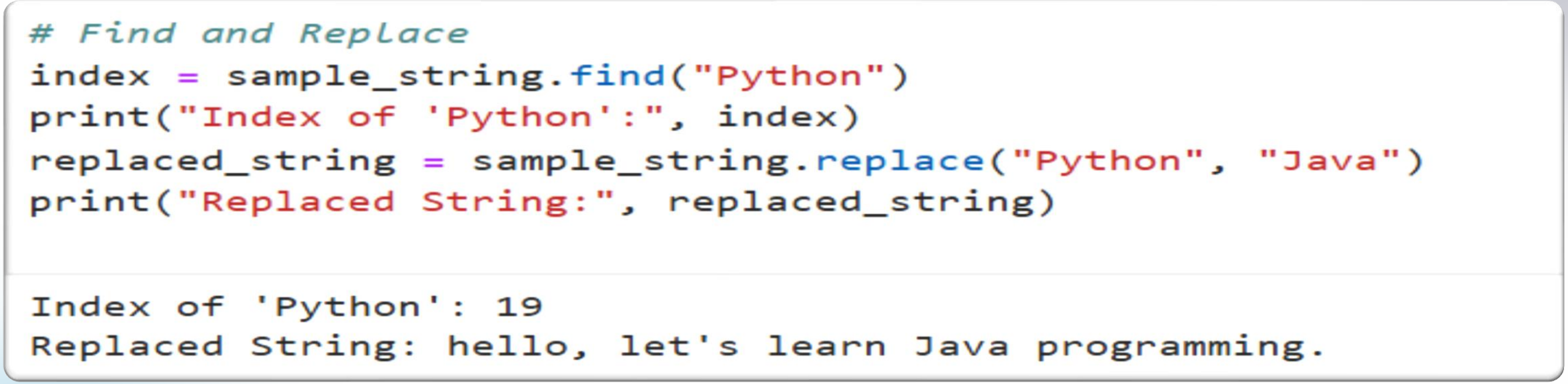
```
# convert case
sample_string = "hello, let's learn Python programming."
print("Uppercase:", sample_string.upper())
print("Lowercase:", sample_string.lower())
print("Title Case:", sample_string.title())
```

```
Uppercase: HELLO, LET'S LEARN PYTHON PROGRAMMING.
Lowercase: hello, let's learn python programming.
Title Case: Hello, Let'S Learn Python Programming.
```

- Convert a string to uppercase, lowercase and title case

Practical Exercises on String Operations

Activity : Find and Replace



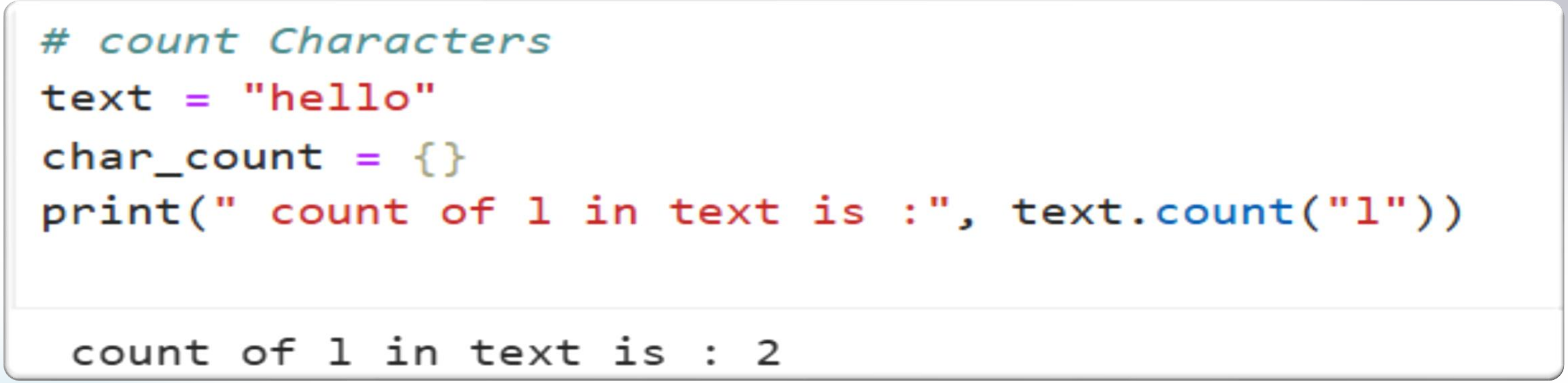
```
# Find and Replace
index = sample_string.find("Python")
print("Index of 'Python':", index)
replaced_string = sample_string.replace("Python", "Java")
print("Replaced String:", replaced_string)
```

```
Index of 'Python': 19
Replaced String: hello, let's learn Java programming.
```

- Find the index of "Python" and replace it with "Java"

Practical Exercises on String Operations

Activity : Count Characters



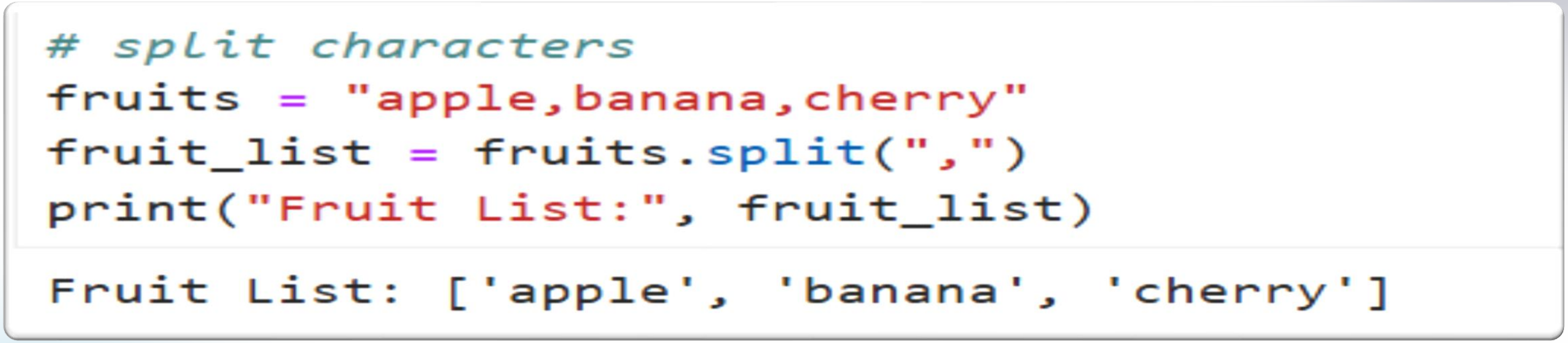
```
# count Characters
text = "hello"
char_count = {}
print(" count of l in text is :", text.count("l"))
```

```
 count of l in text is : 2
```

- Count the occurrences of each character in a string

Practical Exercises on String Operations

Activity : Split String



```
# split characters
fruits = "apple,banana,cherry"
fruit_list = fruits.split(",")
print("Fruit List:", fruit_list)
```

```
Fruit List: ['apple', 'banana', 'cherry']
```

- Split "apple,banana,cherry" into a list of fruits

Q & A

Thank you