Data Analytics 101

PRESENTATION BY:
BRIAN A.N ft. FRANK ANDREW

Introduction to Data Analytics - Learning Outcomes

By the end of this session, you will have covered the following learning outcomes:

Print Function: Ability to use the print() function to display the output of a Python program.

Variables: Capacity to create variables to store data in a Python program.

Comments: Proficiency in writing comments to provide explanations of code functionality.

Arithmetic Operators: Competency in writing arithmetic operators to perform mathematical operations.

Data Types: Understanding and determination of data types for different types of data.

Data Structures: Utilization of lists, dictionaries, and sets to store data effectively.

Lesson One

Overview

- The Python programming language stands out as one of the foremost choices in data science. According to a Kaggle survey [Link], Python's popularity continues to soar owing to its user-friendly syntax, extensive developer community, and robust data science libraries.

Skill one: Python Programming Terminology

Comments: Explanatory notes within our code for better understanding.

Variables: Names assigned to memory locations storing values in a program.

Arithmetic Operators: Functions performing mathematical operations on two values.

Data Types: Classifications specifying the type of value a variable holds.

Lists: Compound data types for storing items of various data types.

Conditional (if) Statements: Decision-making tools executing code based on conditions.

Dictionaries: Data structures storing key-value pairs, allowing retrieval, addition, and modification by keys.

Sets: Unordered collections of unique elements, suitable for data types excluding mutable elements.

For Loops: Iteration tools for sequences like lists, dictionaries, or strings.

While Loops: Execute statements as long as a condition remains true.

Functions: Organized and reusable code blocks, enhancing program modularity.

Terminologies: Variables

Definition: Storage containers for data, allowing values to change based on conditions or input.

Naming Conventions: Variable names must start with a letter, underscore, or non-numeric character. Each programming language has its naming conventions.

Reserved Words: Programming languages have reserved words that cannot be used as variable names (e.g., "Date"). Alternative names should be chosen, following naming conventions.

Operations: Variables allow performing operations considering the stored values.

Example:

```
age = 25
```

```
name = "John"
```

Terminologies : Comments

Purpose: Explains how a program works without affecting its execution.

Goals of Comments:

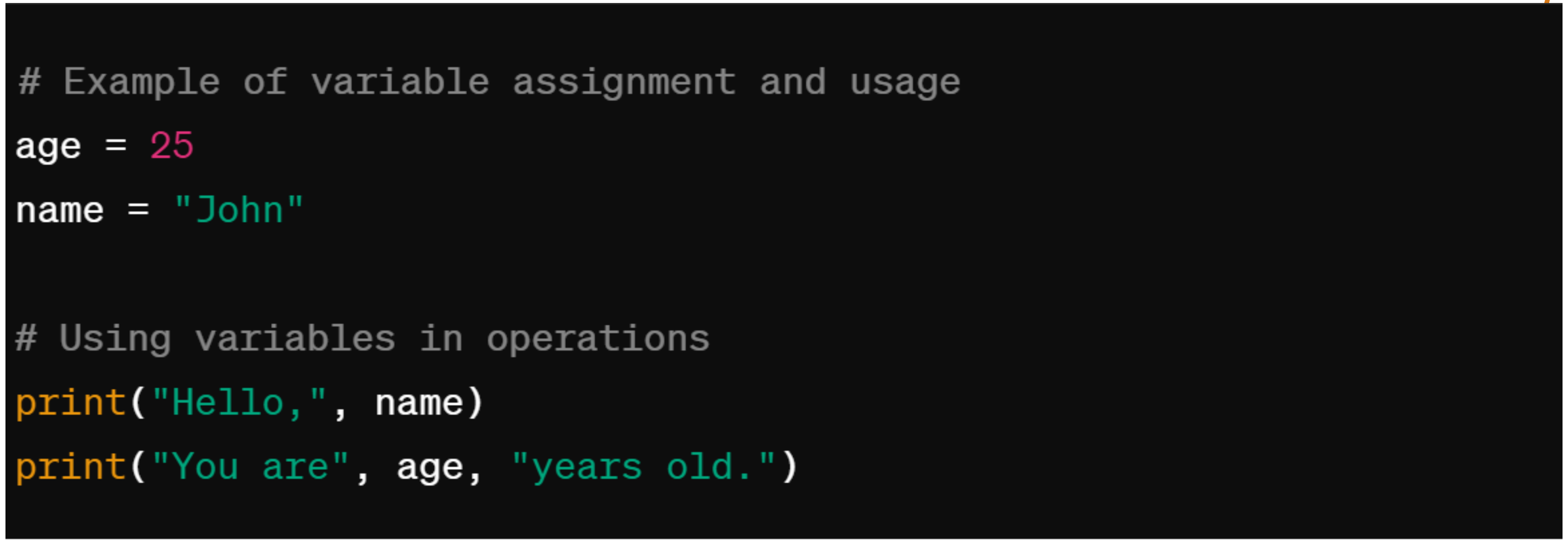
- Explain the functionality of specific code segments.
- Clarify aspects not immediately evident to the reader.
- Provide insight into the programmer's intentions.
- Serve as gentle reminders for future modifications.

```
# Example of variable assignment and usage

age = 25

name = "John"
```

Terminologies : Printing

print(): One of the most commonly used Python commands.

Functionality: The print() function is used to display output on the screen.



```python
# Example of variable assignment and usage
age = 25
name = "John"

# Using variables in operations
print("Hello,", name)
print("You are", age, "years old.")
```

Terminologies : Data Types

Purpose: Determine permissible mathematical operations and functionalities for data manipulation.

Python Data Types:

Integer: Positive or negative whole numbers.

Float: Real numbers with floating-point representation.

String: Sequence of characters.

Lists: Collection of data of different data types.

Dictionaries: Ordered set of key-value pair items.

Other Data Types: Tuples and sets.

```
# Examples of different data types in Python
integer_var = 10
float_var = 3.14
string_var = "Hello, World!"
list_var = [1, 2, 3, "a", "b", "c"]
dictionary_var = {"name": "John", "age": 25, "city": "New York"}
```

Terminologies: Variables

Definition: Storage containers for data, allowing values to change based on conditions or input.

Naming Conventions: Variable names must start with a letter, underscore, or non-numeric character. Each programming language has its naming conventions.

Reserved Words: Programming languages have reserved words that cannot be used as variable names (e.g., "Date"). Alternative names should be chosen, following naming conventions.

Operations: Variables allow performing operations considering the stored values.

Example:

```
age = 25
```

```
name = "John"
```

Complex types: Sets vs. Dictionaries vs. Lists

Unique Elements: Contains only unique elements with no duplicates.

Unordered: Elements are not stored in any particular order.

Set Example

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

Key-Value Pairs: Stores elements as key-value pairs, facilitating retrieval based on keys.

Mutable: Allows modification, addition, and removal of elements.

# Example of a dictionary

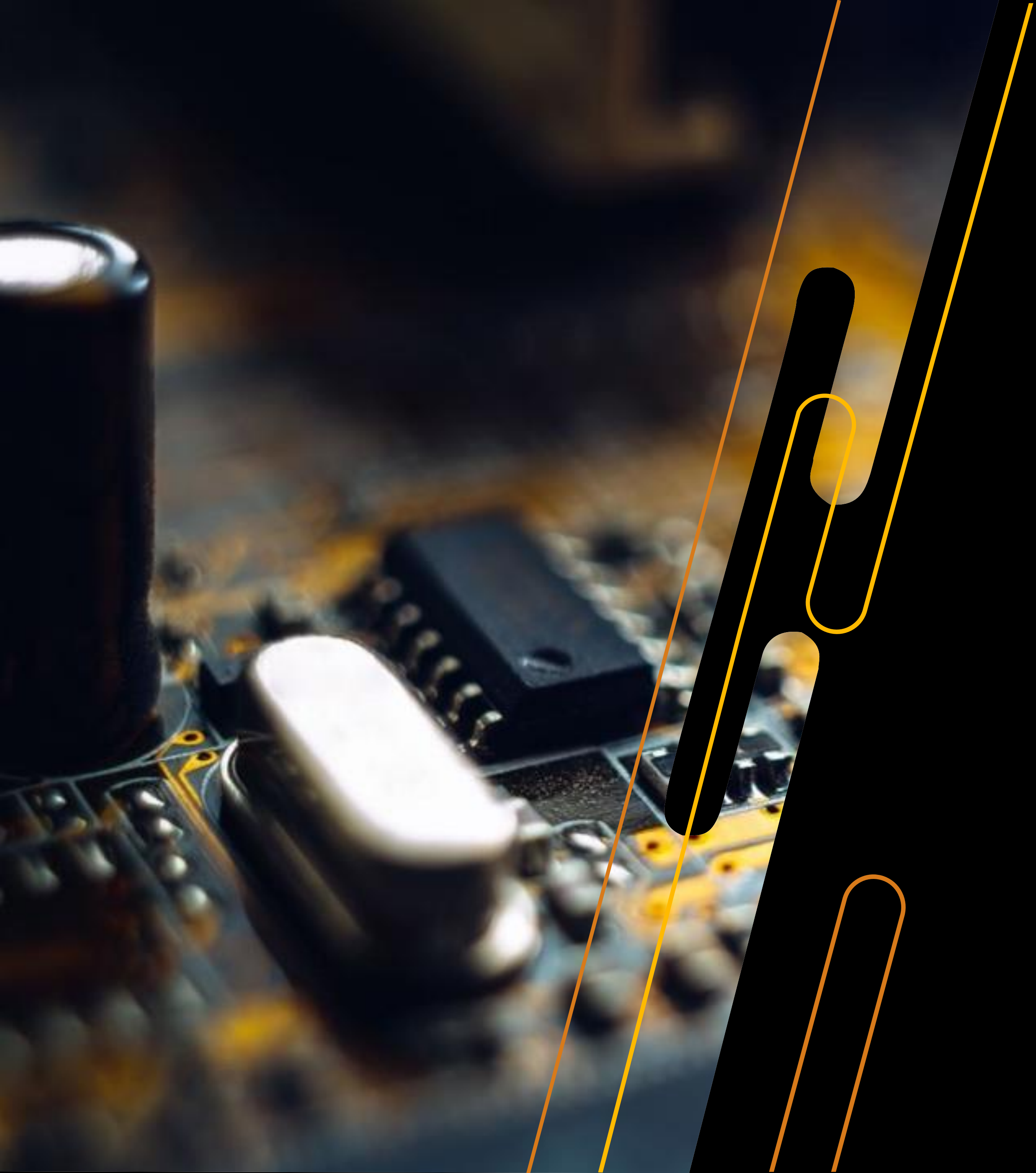
```
my_dict = {"name": "John", "age": 25, "city": "New York"}
```

Ordered: Elements are stored in a specific order, allowing indexing and slicing.

Mutable: Allows modification, addition, and removal of elements.

# Example of a list

```
my_list = [1, 2, 3, 4, 5]
```



Resources

What we’ve learned so far

Python Documentation

Link: https://docs.python.org/3/

Hands On Lab

Link: https://t.ly/_6uM-

Happy Coding