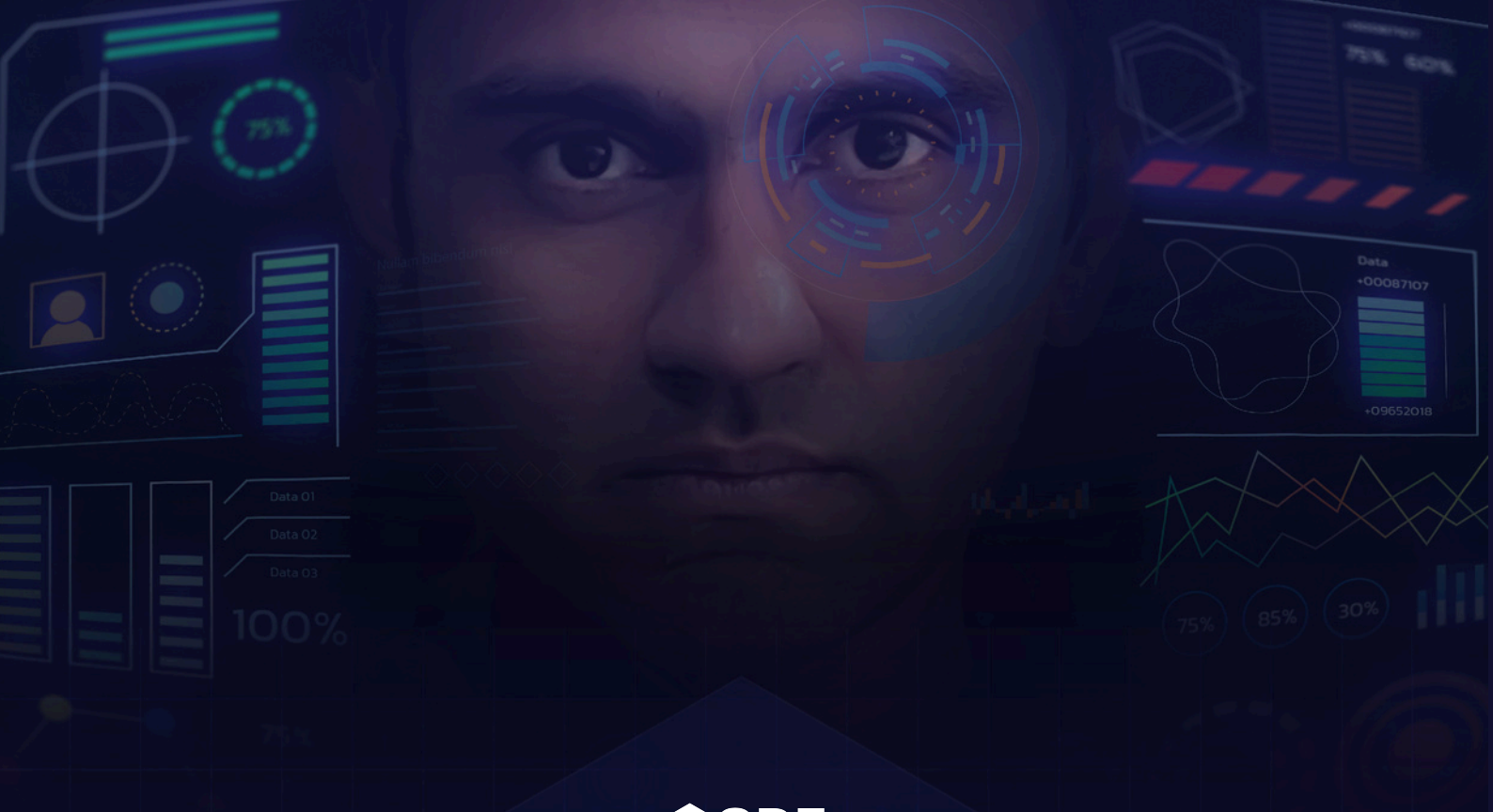

CODE BASICS

Python Advanced: Generators, Decorators, JSON

TAKEAWAYS

codebasics.io

Working with JSON

1. JSON (JavaScript Object Notation) is a lightweight data format widely used for data exchange between servers and web applications.
2. Python provides the **json** module to encode Python objects into JSON format using **json.dumps()** and to decode JSON data into Python objects using **json.loads().**
3. When working with files, use **json.dump()** to serialize an object to a file and **json.load()** to deserialize a file containing JSON data into a Python object.
4. JSON in Python supports basic data types like strings, numbers, lists, and dictionaries, making it versatile for data storage and communication needs.
5. Handling exceptions like **JSONDecodeError** is essential for robust JSON data parsing, especially when dealing with data from untrusted sources.

Generators and Iterators

1. Generators in Python are a simple way to create iterators using functions that yield values instead of returning a single value.
2. Iterators are objects that implement the **__iter__()** and **__next__()** methods, which allow Python to iterate over collections of items, such as in a for loop.
3. Generators help in managing memory efficiently by yielding items one at a time, only holding one item in memory, unlike lists which store all elements.
4. The **yield** statement is used in a function to turn it into a generator, suspending the function's state until the next value is needed.
5. Generators and iterators are powerful for handling large data sets, infinite sequences, and pipelines that transform data through a series of steps.

Decorators

1. Decorators in Python are functions that modify the behavior of another function, method, or class without permanently modifying the original.
2. They provide a flexible way to extend and modify the behavior of callable objects (functions, methods) through a clean syntax using the **@decorator_name** notation just above the function definition.
3. The majority of the IDEs (Integrated Development Environments) such as PyCharm provide built-in support for debugging.