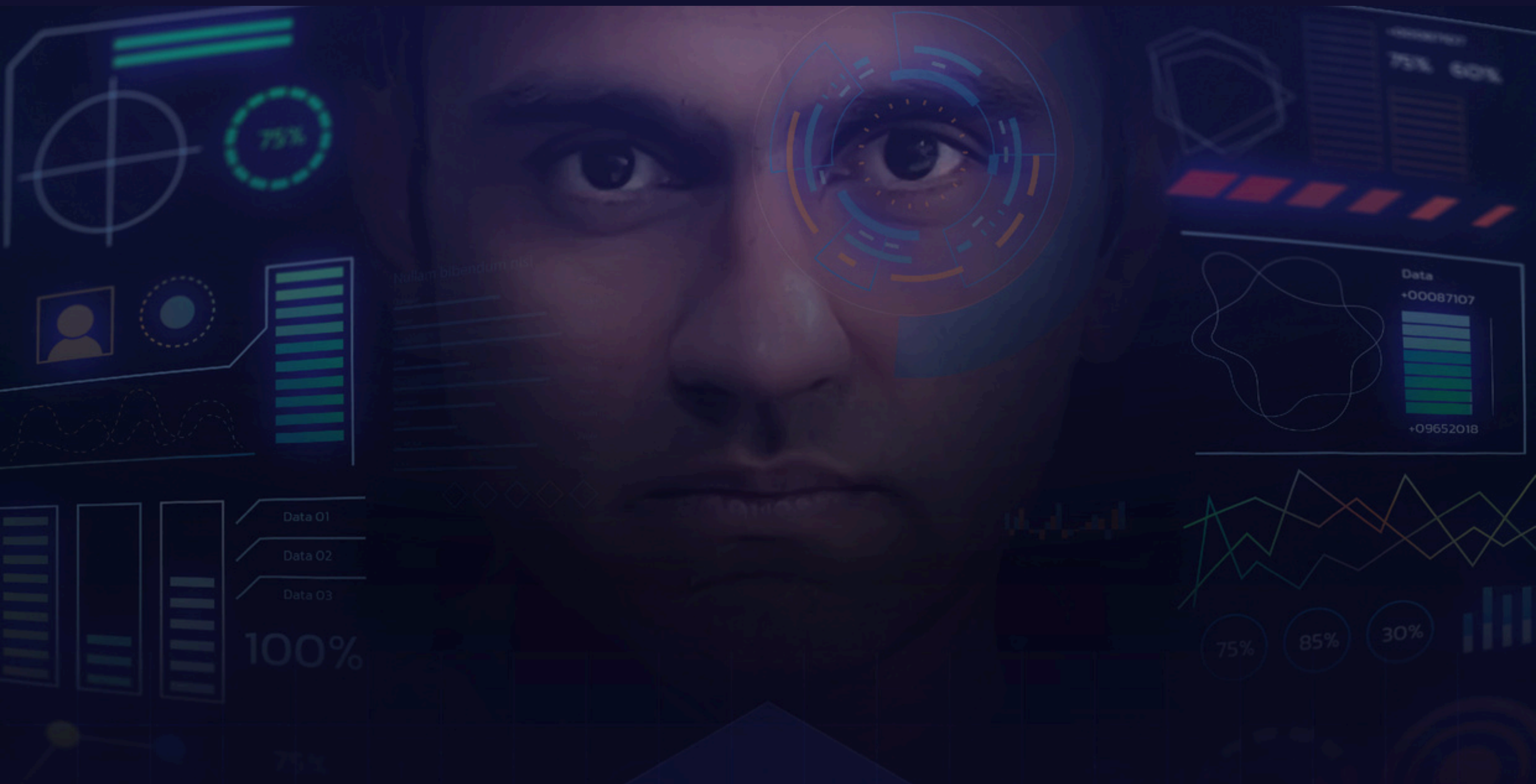

CODE BASICS

Python Advanced: Comprehensions, Sets

TAKEAWAYS

codebasics.io

Set and Frozenset

1 Sets are mutable collections that only store unique elements and support efficient membership checks, additions, and deletions.

2 Frozensets are immutable versions of sets, making them hashable and usable as dictionary keys or set elements.

3 Both sets and frozensets support mathematical set operations like union, intersection, difference, and symmetric difference.

4 Sets can be modified using methods like **add(), remove(), and update()**, while frozensets do not offer any methods that alter their content.

5 Understanding sets and frozensets is crucial for optimizing performance in scenarios requiring the storage of non-repeating elements and fast lookups.

List, Dict, Set Comprehensions

1 List comprehensions provide a concise way to create lists by running a for loop in a single-line.

2 Comprehensions are generally more readable and faster than using loops for building collections, making them a preferred choice for such tasks in Python.

3 Dictionary comprehensions allow for the dynamic construction of dictionaries by running a for loop in a single line.

4 Set comprehensions work similarly to list comprehensions but produce a set, automatically removing duplicate elements during creation.

5 All comprehensions can include conditionals to filter elements, providing a powerful tool for creating customized collections efficiently.

Pep8 Naming Convention

1 PEP 8 specifies naming conventions that enhance code readability: use **CamelCase** for class names, **lowercase_with_underscores** for function and variable names, and **ALL_CAPS** for constants.

2 Consistency with these conventions helps in maintaining a standard format across Python codebases, making it easier for other developers to read and maintain the code.

3 Adhering to PEP 8 naming conventions is considered a best practice in Python development and is often enforced in professional coding environments through code reviews and linters.

Code Debugging Using PyCharm

1 Debugging is an important skill that any Python program should acquire.

2 Debugging allows to monitor code execution. It is used frequently to find bugs or troubleshoot issues.

3 The majority of the IDEs (Integrated Development Environments) such as PyCharm provide built-in support for debugging.

4 Conditional break point allows you to break in the code when a certain condition is satisfied.