**Python Decorators: How to Create and Use with Examples**

|  |  |  |
| --- | --- | --- |
| **Topic Name** | **Description** | **Approximate Number of Words** |
| Introduction | * Python Objects * First-Class Objects. * - Inner Functions. * - Higher-Order Functions. |  |
| Introduction to Decorators | * - It is also called meta programming * - Syntax of Python Decorators * - Reuse these customised behaviors on multiple functions/classes. * - Python decorator modify data |  |
| Writing your first decorator |  |  |
| How to Add Arguments to Decorators in Python |  |  |
| Returning Values from Decorated Functions |  |  |
| Reusing a Decorator on Multiple Functions |  |  |
| Multiple Decorators on a Single Function |  |  |
| A Decorator Within a Function |  |  |
| Preserving function Metadata |  |  |
| Decorating a Class |  |  |
| Classes as a Decorator |  |  |
| Built-in Fancy Decorators |  |  |
| Stateful Decorators |  |  |
| Debugging Decorators |  |  |
| Example Use Cases of Decorators |  |  |
| Where you don’t need decorators |  |  |
| Summary |  |  |
| Further reading |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| Total number of words |  | 1500-words max |

* Follow this style guide in your writing:
* Don’t repeat any data in the content. The same content/fact should not be written in intro features, pro and cons.
* Check your Hemming way Score and should pass Copyscape/ Grammarly
* Refer this How to article checklist <https://docs.google.com/spreadsheets/d/e/2PACX-1vSpJXNSPCbCShRNCS-Zq2KU5TtSO4FAVFSrziirhNQYKQkkGY8hrzNjtLDKerTq5pB12UEYJqfr6Abr/pubhtml?gid=128986070&single=true>