

Dunder or Magic Methods

Dunder methods are special methods that start and end with the double underscores.

Magic methods are not meant to be invoked directly by you, but the invocation happens internally from the class on a certain action.

For example:

- when you add two numbers using the + operator, internally, the `__add__()` method will be called.
- Print `(len(a))` → `print(a.__len__())`
- Print `(a[1])` → `print(a.__getitem__(1))`

Built-in classes in Python define many magic methods.

Use the `dir()` function to see the methods and magic methods inherited by a class.

```
In [2]: print(dir("a"))
```

```
['_add_', '_class_', '_contains_', '_delattr_', '_dir_', '_doc_', '_eq_', '_format_', '_ge_', '_getattribute_', '_getitem_', '_getnewargs_', '_gt_', '_hash_', '_init_', '_init_subclass_', '_iter_', '_le_', '_len_', '_lt_', '_mod_', '_mul_', '_ne_', '_new_', '_reduce_', '_reduce_ex_', '_repr_', '_rmod_', '_rmul_', '_setattr_', '_sizeof_', '_str_', '_subclasshook_', 'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuffix', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

*Args & **Kwargs:

*Args: used to pass a variable number of arguments to the function

It's all about the * We can use any name after * such as : *num ,*names,...

* ➡ unpacking operator: deal with the inputs of the function as tuple

**Kwargs : used to pass a variable number of key word arguments to the function

** ➡ deal with the inputs of the function as dictionary.

Encapsulation

Enables the wrapping of all data (attributes and methods) into a single component(class) thereby preventing the accidental modification of data by imposing some restrictions on accessing variables or methods directly.

So Encapsulation used to restrict the access to the data stored in (attributes and methods)

Based on the visibility of the object's attribute in a class, the attributes are classified in two. They are **private** and **public**.

Public → Data (attributes and methods) is accessible from any part of the program

Private → The Data can be access from within the class only

Protected → can access the data with in the class that defined in or in the derived classes

Abstraction

is the process of hiding internal implementation of a process from user. The users only interact with the basic implementation of the function, but inner working is hidden.

Abstract Class: The classes that cannot be instantiated. This means that we cannot create objects of an abstract class and these are only meant to be inherited. Then an object of the derived class is used to access the features of the base class. These are specifically defined to lay a foundation of other classes that exhibit common behavior or characteristics.