

# Role of Underscore( )

08 March 2023 16:01

<https://www.datacamp.com/tutorial/role-underscore-python>

## Descriptive: Naming Styles

The following special forms using leading or trailing underscores are recognized (these can generally be combined with any case convention):

- **`_single_leading_underscore`**: weak "internal use" indicator. E.g. `from M import *` does not import objects whose name starts with an underscore.
- **`single_trailing_underscore_`**: used by convention to avoid conflicts with Python keyword, e.g.  
`Tkinter.Toplevel(master, class_='ClassName')`
- **`__double_leading_underscore`**: when naming a class attribute, invokes name mangling (inside class `FooBar`, `__boo` becomes `_FooBar__boo`; see below).
- **`__double_leading_and_trailing_underscore__`**: "magic" objects or attributes that live in user-controlled namespaces. E.g. `__init__`, `__import__` or `__file__`. Never invent such names; only use them as documented.