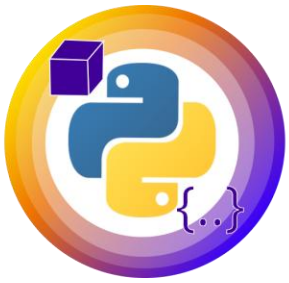

Python OOP: Special Methods

Key Takeaways

- **Special Methods**
  - They can be easily recognized by the double leading and trailing underscores around their names.
  - They are called indirectly when we call specific functions or use a specific syntax passing an instance of the class as the argument.

Special Method	Called by
__str__()	str() and print()
__len__()	len()
__getitem__()	<obj>[<index>]
__add__()	+
__bool__()	bool()

Key Takeaways

- **Special Methods**
  - Some of them have default implementations, but they can be "customized" by implementing them in the class.

```
class Dog:

    def __init__(self, name, age):
        self.name = name
        self.age = age

    def __str__(self):
        return f"I'm a Dog. My name is {self.name}"
```

`__lt__()`	<	Less than
`__le__()`	<=	Less Than or Equal to
`__eq__()`	==	Equal to
`__ne__()`	!=	Not equal to
`__gt__()`	>	Greater than
`__ge__()`	>=	Greater than or Equal to