

### `__init__`

```
class my_class:
    def __init__(self,
field_one, field_two):
        self.field_one = field_one
        self.field_two = field_two
my_instance = my_class(42, "hello") # __init__ is called here
```

The constructor method called automatically when a new instance of a class is created. Receives the arguments passed to the class and uses them to set up the instance's fields. Without it you cannot pass any values when creating an instance.

### `__repr__`

```
class my_class:
    def __repr__(self):
        return f"my -
class (field_one={self.field_one}, field_two={self.field_two}) "
my_instance = my_class(42, "hello")
print(my_instance) # my_class(field_one=42, field_two=hello)
```

Defines how an object is represented as a string when printed or inspected. Should return a string that clearly describes the object and its contents. Used by `print()`, debuggers, and the interactive console.

### `!r`

```
my_value = "some text"
f"{my_value}" # some text
f"{my_value!r}" # 'some text'
```

Adds `repr()` formatting to a value inside an f-string, wrapping strings in quotes and showing escape characters explicitly. Useful for debugging and error messages where you want to see the exact value including its type hints. Without `!r` a string is inserted as plain text; with `!r` it is wrapped in quotes so it's unambiguous.

### `__eq__`

```
class my_class:
    def __eq__(self, other):
        return self.field_one == other.field_one and self.field_two == other.field_two
instance_a = my_class(42, "hello")
instance_b = my_class(42, "hello")
instance_a == instance_b # True
# same field values
instance_a is instance_b # False
# different objects in memory
```

Defines what equality means for two instances of a class. Without it, `==` checks if two variables point to the same object in memory rather than comparing their contents. Should return `True` if all relevant fields are equal, `False` otherwise.



By **blakecromar**

[cheatography.com/blakecromar/](https://cheatography.com/blakecromar/)

Not published yet.

Last updated 2nd July, 2026.

Page 1 of 1.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>