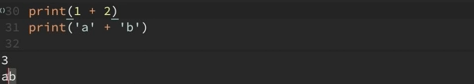
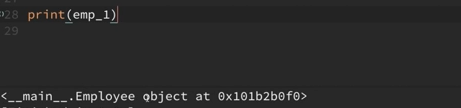
**Special Magic Methods**

These methods help emulate built-in methods with in python and also depict operator overloading. An example of operator overloading:



In the above example, the operator ‘+’ has different roles in both the statements.

In our case when we print the following:



We get the employee object printed. Using magic methods we can change the built-in behavior and choose what and how to display when print(emp\_1) is printed

An example of magic methods is:



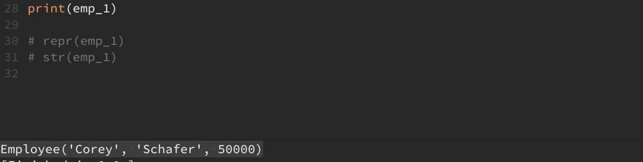
They are also called ‘dunder’ methods because they start and end with double underscores called ‘dunders’.

The other two commonly used magic methods are \_\_repr\_\_() and \_\_str\_\_().

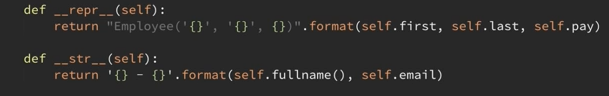
When the following snippet is defined in the class Employee:



Upon printing the following we get:



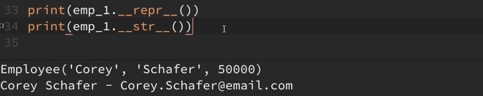
But now if we define both retr() and str():



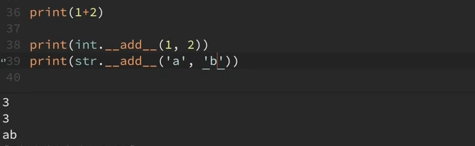
The output pertaining to str() is printed:



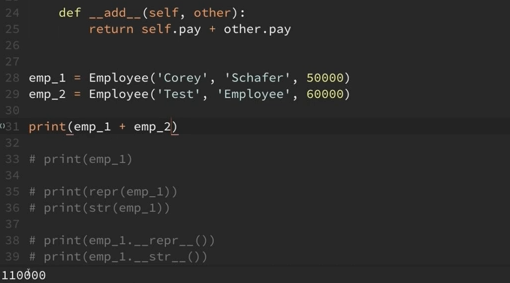
In order to access the result of retr() specifically use:



Operations like addition make use of \_\_add\_\_() in the background:



The following snippet creates a dunder method to add salaries of two employees:



Another example:



Using the same in our Employee class:



There are many other dunder methods present in the documentation.