

COMP  
110

# Classes in Memory + Magic Methods

# Question

When I call `print(x)`, Python calls what magic method on `x` *before* printing?

# Arithmetic Operator Overloads

+	<code>__add__(self, other)</code>
-	<code>__sub__(self, other)</code>
*	<code>__mul__(self, other)</code>
/	<code>__truediv__(self, other)</code>
**	<code>__pow__(self, other)</code>
%	<code>__mod__(self, other)</code>

# Comparison Operator Overloads

<	<code>__lt__(self, other)</code>
>	<code>__gt__(self, other)</code>
<=	<code>__le__(self, other)</code>
>=	<code>__ge__(self, other)</code>
==	<code>__eq__(self, other)</code>
!=	<code>__ne__(self, other)</code>

For each magic method call, what is self and (if applicable) what is other?

<code>str(a)</code>	<code>__str__(self)</code>
<code>a + b</code>	<code>__add__(self, other)</code>
<code>a - b</code>	<code>__sub__(self, other)</code>
<code>a * b</code>	<code>__mul__(self, other)</code>
<code>a &lt; b</code>	<code>__lt__(self, other)</code>
<code>a == b</code>	<code>__eq__(self, other)</code>

# Diagramming

```
1 from __future__ import annotations
2
3 class ShoppingGuide:
4     groceries: list[str]
5     budget: float
6     store: str
7
8
9     def __init__(self, groceries: list[str], budget: float, store: str):
10         self.groceries = groceries
11         self.budget = budget
12         self.store = store
13
14     def __add__(self, more_money: float) -> ShoppingGuide:
15         return ShoppingGuide(self.groceries, self.budget + more_money, self.store)
16
17 my_plan: ShoppingGuide = ShoppingGuide(["apples", "kiwi"], 5.55, "Food Lion")
18 AJs_plan: ShoppingGuide = my_plan + 2.12
```

## Extra Challenge

- Write a `__str__` magic method that gives me all the information of a `ShoppingGuide` object
- Change the `__add__` magic method to add a list of more groceries instead of adding money to the budget. (Note that it still shouldn't modify self!)