

While inside of a class, functions are referred to as 'methods'. If you hear someone mention methods, they're speaking about classes. Methods are essentially functions, but only callable on the instances of a class.

Creating

```
In [ ]: ▶ class ShoppingBag():
        """
        The ShoppingBag Class will have handles, capacity and items
        to place inside.

        Attributes for the class:
        - handles: expected to be an integer
        - capacity: expected to be a string OR an integer
        - items: expected to be a list
        """

        def __init__(self, handles, capacity, items):
            self.handles = handles
            self.capacity = capacity
            self.items = items

        # Method that shows the shoppingBag items
        def showShoppingBag(self):
            print("You Have items in your bag!!!")
            for item in self.items:
                print(item)

        # Show capacity of shoppingBag
        def showCapacity(self):
            print(f'Your Capacity is: {self.capacity}')
```

```
        # Add Item(s) to the items list for the shoppingBag
        def addToShoppingBag(self):
            products = input('What would you like to add? ')
            self.items.append(products)

        # Change the capacity of the shoppingBag (dynamically)
        def changeBagCapacity(self, capacity):
            self.capacity = capacity

        # Increase Capacity of the shoppingBag by 10 -- default
        def increaseCapacity(self, changed_capacity = 10):
            if self.capacity == isinstance(self.capacity, str):
                print("We can't do that here boss...")
            else:
                self.capacity += changed_capacity
```

Calling

```

In [ ]: ▶ # See Above
# See Above
# Created and instance of the ShoppingBag Class
wholeFoods_bag = ShoppingBag(2,10,[])

# Create Function to run the shoppingBag(wholeFoods_bag) methods

def run():
    while True:
        response = input('What do you want to do? add/show/ or quit ')

        if response.lower() == 'quit':
            wholeFoods_bag.showShoppingBag()
            print('Thanks for shopping')
            break
        elif response.lower() == 'add':
            wholeFoods_bag.addToShoppingBag()

        elif response.lower() == 'show':
            wholeFoods_bag.showShoppingBag()
        else:
            print('Try another command')

run()

```

Modifying an Attribute's Value Through a Method

```

In [ ]: ▶ # Show Capacity
wholeFoods_bag.showCapacity()
print('Capacity AFTER the change...')
wholeFoods_bag.changeBagCapacity(40)
wholeFoods_bag.showCapacity()

```

Incrementing an Attribute's Value Through a Method

```

In [ ]: ▶ wholeFoods_bag.showCapacity()
print('After Increase')
wholeFoods_bag.increaseCapacity()
wholeFoods_bag.showCapacity()

```

In-Class Exercise #3 - Add a method that takes in three parameters of year, doors and seats and prints out a formatted print statement with make, model, year, seats, and doors