Static and Class Methods

November 18, 2022

Static Methods

```
class Calculator:

    # create addNumbers static method
    @staticmethod
    def addNumbers(x, y):
        return x + y
```

```
>>> print('Product:', Calculator.addNumbers(15, 110))
125
```

- `@staticmethod` is a **decorator**
- Static methods have nothing to do with the instances or the class, but are useful within the class.

Class method

```
class Fruit:
    name = 'Fruitas'

    @classmethod
    def printName(cls):
        print('The name is:', cls.name)
```

```
>>> Fruit.printName()
Fruitas
apple = Fruit()
berry = Fruit()
>>> apple.printName()
Fruitas
>>> berry.printName()
Fruitas
```

classmethod *vs* staticmathod

- Neither need an object to be instantiated
- Static method doesn't know about the class.
- Class method the class is always the first parameter.
- Static method might as well be a function call, but placing it in the class controls the namespace.
- Class method knows about all the class attributes and methods.