



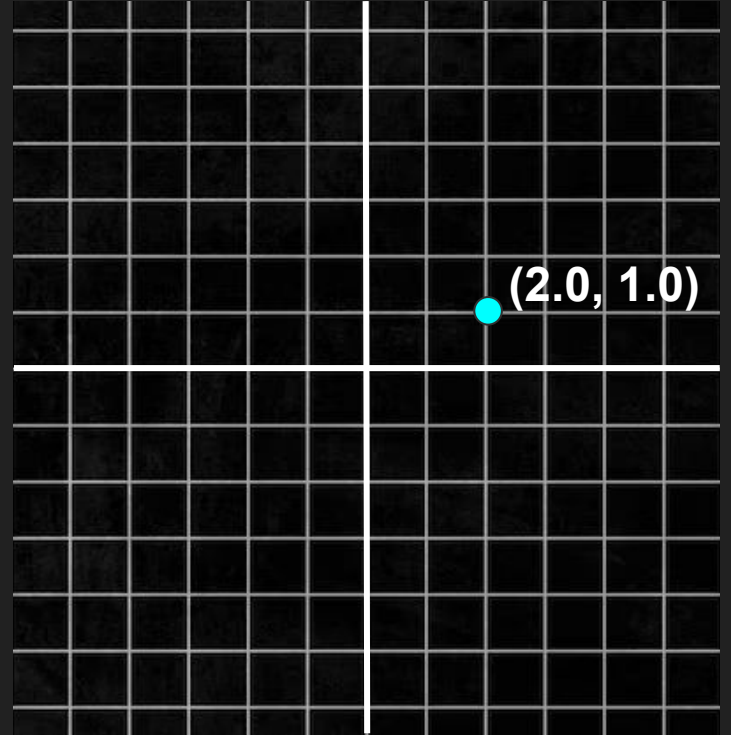
🪄 Magic Methods 🪄

Announcements

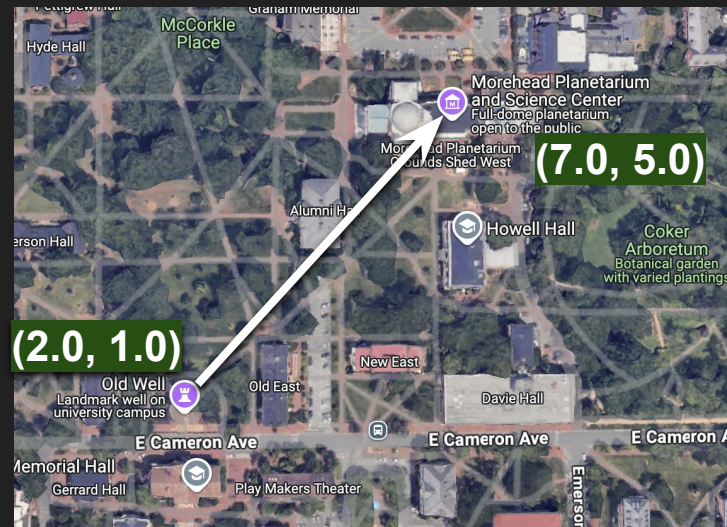
EX06: Dictionary Unit Tests due today at 11:59pm!

Recall this Point class

```
0 class Point:
1     x: float
2     y: float
3
4     def __init__(self, x: float, y: float):
5         self.x = x
6         self.y = y
7
8     def dist_from_origin(self) -> float:
9         return (self.x**2 + self.y**2) ** 0.5
10
11    def translate_x(self, dx: float) -> None:
12        self.x += dx
13
14    def translate_y(self, dy: float) -> None:
15        self.y += dy
16
17 pt: Point = Point(2.0, 1.0)
```



```
1-16 class Point: ... # collapsed for space
17
18 class Line:
19     start: Point
20     end: Point
21
22     def __init__(self, start: Point, end: Point):
23         self.start = start
24         self.end = end
25
26     def get_length(self) -> float:
27         x_diffs: float = self.end.x - self.start.x
28         y_diffs: float = self.end.y - self.start.y
29         return (x_diffs**2 + y_diffs**2) ** 0.5
30
31     def get_slope(self) -> float:
32         x_diffs: float = self.end.x - self.start.x
33         y_diffs: float = self.end.y - self.start.y
34         return y_diffs / x_diffs
```



Warm-up: Create a Line object and print the distance from the Old Well to the Planetarium:



```
old_well: Point = Point(2.0, 1.0)
planetarium: Point = Point(7.0, 5.0)
ow_to_p: Line = Line(old_well, planetarium)
print(ow_to_p.get_length())
```

On your own: try printing a Point or Line object!

What happens?

```
0 class Point:
1     x: float
2     y: float
3
4     def __init__(self, x: float, y: float):
5         self.x = x
6         self.y = y
7
8     def dist_from_origin(self) -> float:
9         return (self.x**2 + self.y**2) ** 0.5
10
11    def translate_x(self, dx: float) -> None:
12        self.x += dx
13
14    def translate_y(self, dy: float) -> None:
15        self.y += dy
16
17 pt: Point = Point(2.0, 1.0)
```

On your own: try printing a Point or Line object!

```
0 class Point:
1     x: float
2     y: float
3
4     def __init__(self, x: float, y: float):
5         self.x = x
6         self.y = y
7
8     def dist_from_origin(self) -> float:
9         return (self.x**2 + self.y**2) ** 0.5
10
11    def translate_x(self, dx: float) -> None:
12        self.x += dx
13
14    def translate_y(self, dy: float) -> None:
15        self.y += dy
16
17 pt: Point = Point(2.0, 1.0)
```

What happens?

<__main__.Point object at
0xffff9506d9a0>

(this is the address where your
computer is storing the Point
object!)

If you could decide what is printed
when you print a Point object,
what would you print?

`__str__` method

```
def __str__(self) -> str:
```

```
    """Return a string representation for people."""
```

```
    return _____
```

Let's implement some magic in VS Code!



__repr__ method

```
def __repr__(self) -> str:  
    """Return a string representation for Python evaluation."""  
    return _____
```

Practice: Consider this `Dog` class

```
1 class Dog:
2     name: str
3     breed: str
4     age: int
5
6     def __init__(self, name: str, breed: str, age: int):
7         self.name = name
8         self.breed = breed
9         self.age = age
10
```

With a partner:

Step 1: Write a `__str__` magic method.

Step 2: Write a `__repr__` magic method.

Practice: Consider this **Dog** class

```
1 class Dog:
2     name: str
3     breed: str
4     age: int
5
6     def __init__(self, name: str, breed: str, age: int):
7         self.name = name
8         self.breed = breed
9         self.age = age
10
11     def __str__(self) -> str:
12         """Returns a string representation (for humans)."""
13         return _____
14
15     def __repr__(self) -> str:
16         """Returns a string representation (for debugging)."""
17         return _____
```

Practice: Consider this `Dog` class

```
1 class Dog:
2     name: str
3     breed: str
4     age: int
5
6     def __init__(self, name: str, breed: str, age: int):
7         self.name = name
8         self.breed = breed
9         self.age = age
10
11     def __str__(self) -> str:
12         """Returns a string representation (for humans)."""
13         return _____
14
15     def __repr__(self) -> str:
16         """Returns a string representation (for debugging)."""
17         return _____
```