

Defensive_Programming

May 8, 2023

```
[3]: numbers = [1.5, 2.3, 0.7, 0.001, 4.4]
total = 0.0
for num in numbers:
    assert num > 0.0, 'Data should only contain positive values'
    total += num
print('total is:', total)
```

total is: 8.901

```
[11]: def normalize_rectangle(rect):
        """Normalizes a rectangle so that it is a the origin and 1.0 units long on
        →its longest axis. input should be of the format (x0th, y0, x1, y1).
        (x0, y0) and (x1, y1) define the lower left and upper right corners of the
        →rectangle repsectively."""
        assert len(rect) == 4, 'Rectangles must contain 4 coordinates'
        x0, y0, x1, y1 = rect
        assert x0 < x1, 'Invalid X coordinates'
        assert y0 < y1, 'Invalid Y coordinates'

        dx = x1 - x0
        dy = y1 - y0
        if dx > dy:
            scaled = dy / dx
            upper_x, upper_y = 1.0, scaled
        else:
            scaled = dx / dy
            upper_x, upper_y = scaled, 1.0

        assert 0 < upper_x <= 1.0, 'Calculated upper x coordinate invalid'
        assert 0 < upper_y <= 1.0, 'Calculated upper y coordinate invalid'

        return (0, 0, upper_x, upper_y)
```

```
[6]: print(normalize_rectangle(0.0, 1.0, 2.0))
```

↳ -----

TypeError Traceback (most recent call↳
↳last)

<ipython-input-6-07e1bfd52af7> in <module>
----> 1 print(normalize_rectangle(0.0, 1.0, 2.0))

TypeError: normalize_rectangle() takes 1 positional argument but 3 were↳
↳given

```
[7]: print(normalize_rectangle( (4.0, 2.0, 1.0, 5.0) ))
```

↳ -----

AssertionError Traceback (most recent call↳
↳last)

<ipython-input-7-f7e0d48bdfd0> in <module>
----> 1 print(normalize_rectangle((4.0, 2.0, 1.0, 5.0)))

<ipython-input-5-e9eddaa78003> in normalize_rectangle(rect)
4 assert len(rect) == 4, 'Rectangles must contain 4 coordinates'
5 x0, y0, x1, y1 = rect
----> 6 assert x0 < x1, 'Invalid X coordinates'
7 assert y0 < y1, 'Invalid Y coordinates'
8

AssertionError: Invalid X coordinates

```
[8]: print(normalize_rectangle( (0.0, 0.0, 1.0, 5.0) ))
```

(0, 0, 0.2, 1.0)

```
[12]: print(normalize_rectangle( (0.0, 0.0, 5.0, 1.0) ))
```

(0, 0, 1.0, 0.2)

```
[ ]:
```