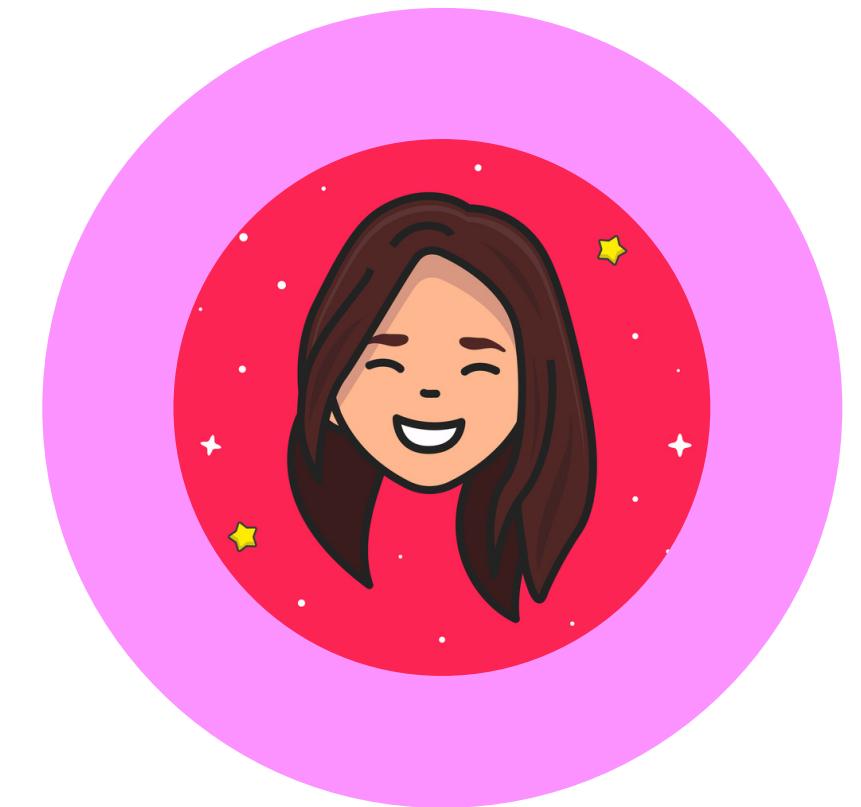


# **Useful collections in Python**

that you should know



@laysauchoa

# Agenda

collections

namedtuple()

deque

Counter

ChainMap

OrderedDict

defaultdict

@laysauchoa

# self.intro()



**Laysa Uchoa**

 @laysauchoa

 [github.com/laysauchoa](https://github.com/laysauchoa)



@laysauchoa

# Python containers



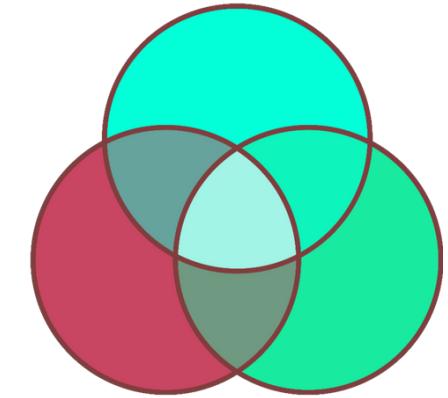
lists



dictionaries



tuples



sets

# What are containers?



source: amazon.de

@laysauchoa

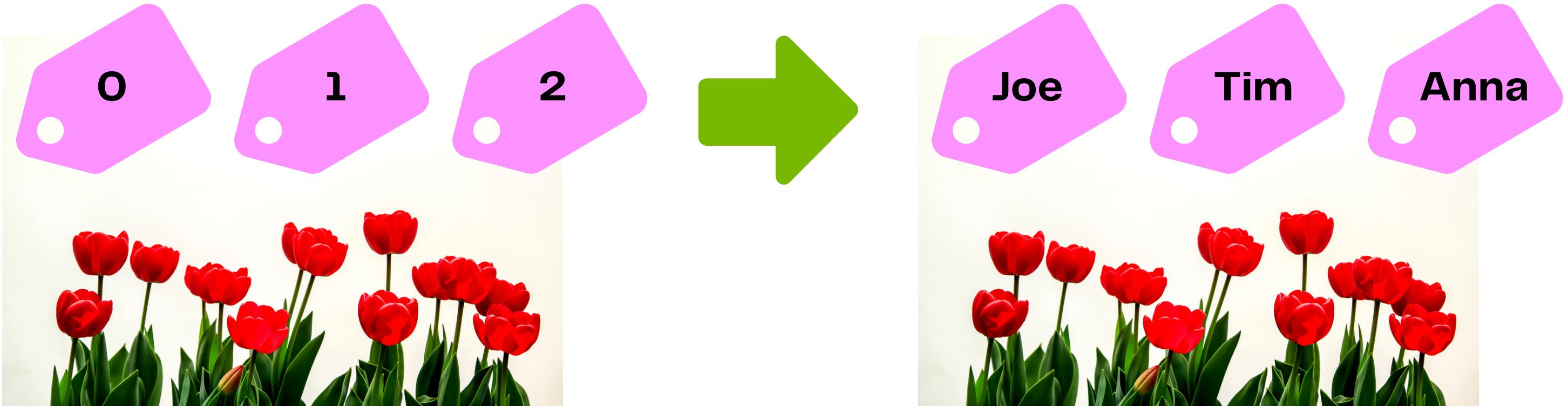
# Collections in Python



source: amazon.de

@laysauchoa

# namedtuple



@laysauchoa

# Without and with namedtuple



```
movie_data = ('The Dark Knight', 2008)

movie_data[0] #The Dark Knight
movie_data[1] #2008
```

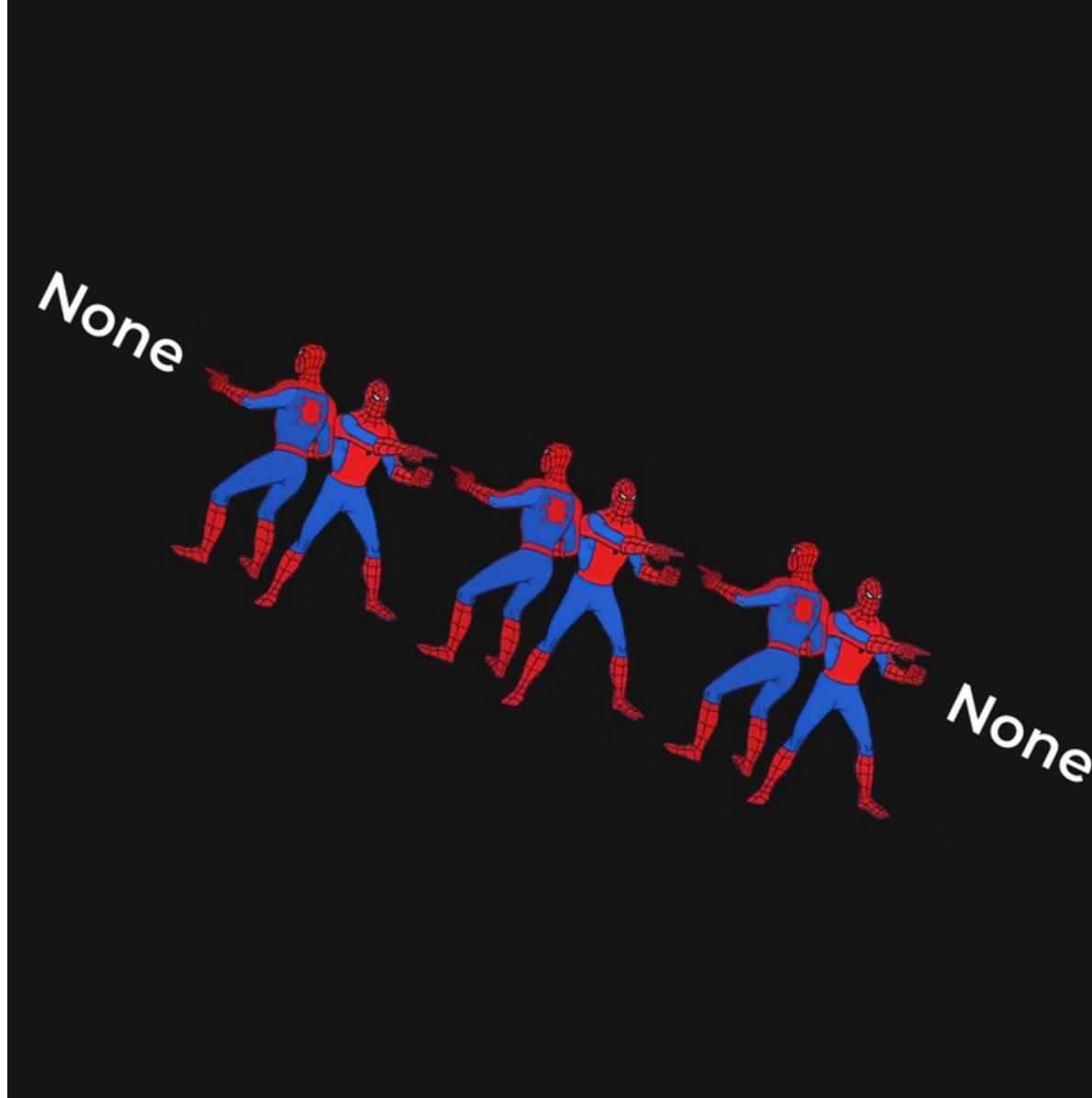
```
from collections import namedtuple

MovieData = namedtuple("MovieData", ["name",
"release_date"])

movie_data = MovieData("The Dark Knight", 2008)

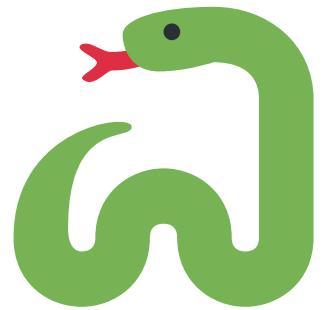
movie_data.name
movie_data.release_date
```

# deque



source: @JSreddit

@laysauchoa



**If you need to talk to both ends of the snake, consider a deque.**

Quora: Kaiser Leib

@laysauchoa

# list x deque

## left operations

```
○ ○ ○
```

```
my_lst = [4, 5, 6]
value = 3
```

```
# my_lst = [3, 4, 5, 6]
%timeit my_lst.insert(0, value)
```

```
>>> 234 µs
```

```
○ ○ ○
```

```
from collections import deque
```

```
my_deque = deque([4, 5, 6])
value = 3
```

```
# deque([3, 4, 5, 6])
%timeit my_deque.appendleft(value)
```

```
>>> 0.0614 µs
```

# **Where are my keys?**

How to handle missing keys



@laysauchoa

# defaultdict

○ ○ ○

```
prices = {"mocha": 5.10, "latte": 4.50,  
"juice": 4.15, "americano": 3.55}  
  
print(prices["smoothie"])  
# KeyError: 'smoothie'
```

○ ○ ○

```
from collections import defaultdict  
  
product_price = defaultdict(lambda: "No Price Assigned")  
product_price["mocha"] = 5.10  
product_price["latte"] = 4.50  
product_price["juice"] = 4.15  
product_price["americano"] = 3.55  
  
product_price["smoothie"]  
  
# No Price Assigned
```

# How many orange cats? Time to count



source: <https://www.lifegate.com/aoshima-island-japan-cats>

@laysauchoa

# Counter

○ ○ ○

```
from collections import Counter

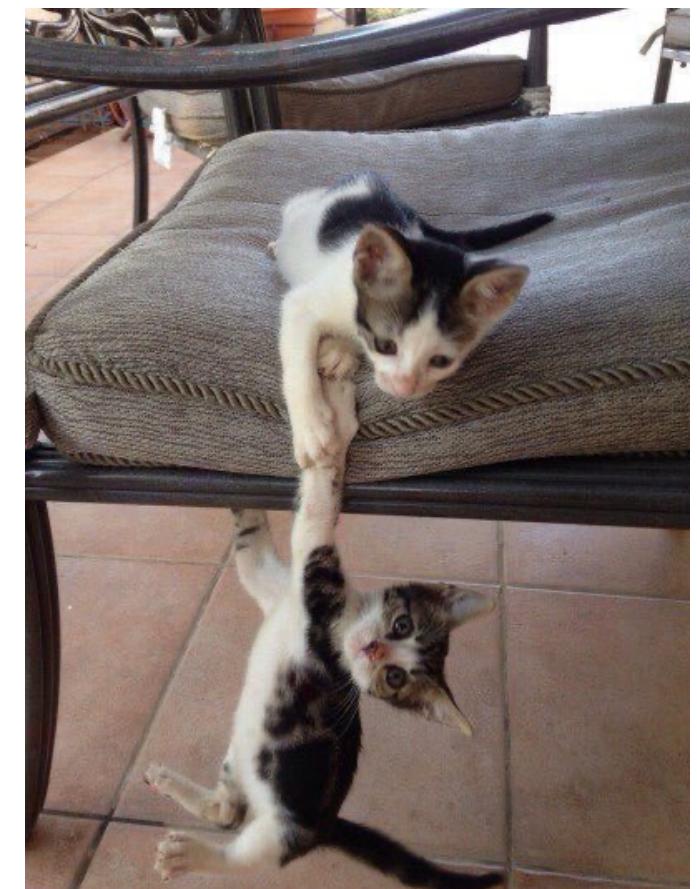
cat_colors = ["grey", "white", "orange",
"orange", "orange", "orange", "orange",
"orange", "white", "white", "orange"]

c = Counter(cat_colors)
# Counter({'grey': 1, 'white': 3, 'orange': 7})

c.most_common(1)
# [('orange', 7)]
```

@laysauchoa

# How can I merge two Python dictionaries?



@TodoDiaGatinhos

# ChainMap

```
○ ○ ○
```

```
a = {"a": 1}  
b = {"b": 2}  
c = **a, **b}
```

```
# {'a': 1, 'b': 2}
```

```
a["a"] = 100
```

```
# {'a': 1, 'b': 2}
```

```
○ ○ ○
```

```
a = {"a": 1}  
b = {"b": 2}  
c = ChainMap(a, b)
```

```
# ChainMap({'a': 1}, {'b': 2})
```

```
a["a"] = 100
```

```
# ChainMap({'a': 100}, {'b': 2})
```

# **Is everything in order with your dictionary?**

Depends on the Python version



source: canva

# OrderedDict

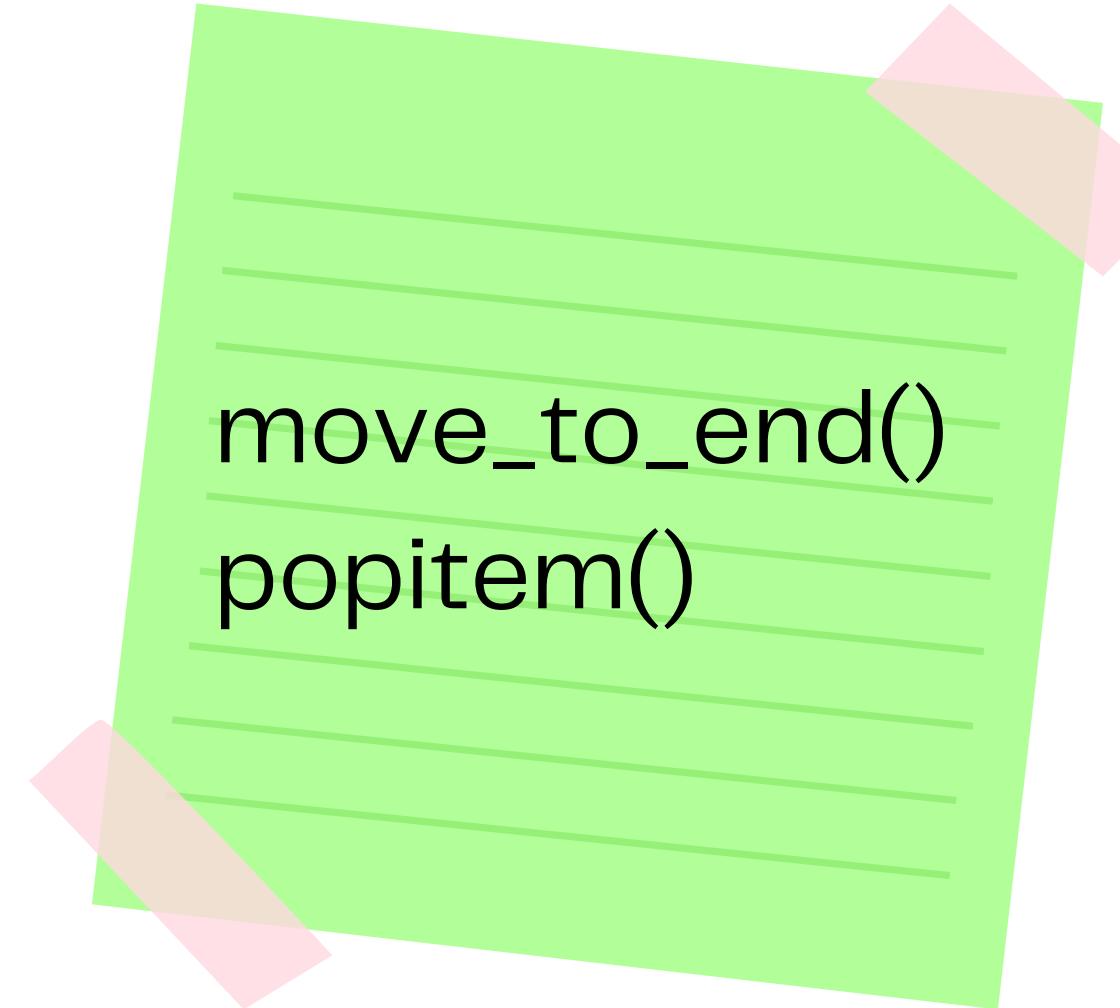
```
from collections import OrderedDict

numbers = OrderedDict()

numbers["one"] = 1
numbers["two"] = 2
numbers["three"] = 3

numbers
# OrderedDict([('one', 1), ('two', 2), ('three', 3)])
```

# dict x OrderedDict



move\_to\_end()  
popitem()

# OrderedDict

```
○ ○ ○  
from collections import OrderedDict  
dict_b = OrderedDict({1: "a", 2: "b"})  
# OrderedDict([(1, 'a'), (2, 'b')])  
  
dict_b.move_to_end(1)  
  
# OrderedDict([(2, 'b'), (1, 'a')])
```

```
○ ○ ○  
from collections import OrderedDict  
  
# pop item from the beginning  
dict_a = OrderedDict({1: "a", 2: "b"})  
  
dict_a.popitem(last=False)  
(1, 'a')  
  
# OrderedDict([(2, 'b')])
```

# Why use OrderedDict?

intent signaling

control over the order of items

backward compatibility

# Questions

If you want to know anything more **Collections** want to share your ideas,  
**feel free to talk to me @laysauchoa**

---



**Thank you!**

---