Counting made easy

DATA TYPES FOR DATA SCIENCE IN PYTHON

Jason Myers
Instructor

Collections Module

- Part of Standard Library
- Advanced data containers

Counter

- Special dictionary used for counting data, measuring frequency

```
from collections import Counter
nyc_eatery_count_by_types = Counter(nyc_eatery_types)
print(nyc_eatery_count_by_type)
```

```
Counter({'Mobile Food Truck': 114, 'Food Cart': 74, 'Snack Bar': 24,
'Specialty Cart': 18, 'Restaurant': 15, 'Fruit & Vegetable Cart': 4}
```

```
print(nyc_eatery_count_by_types['Restaurant'])
```

```
15
```

Counter to find the most common

- `.most_common()` method returns the counter values in descending order

```
print(nyc_eatery_count_by_types.most_common(3))
```

```
[('Mobile Food Truck', 114), ('Food Cart', 74), ('Snack Bar', 24)]
```

Let's practice!

DATA TYPES FOR DATA SCIENCE IN PYTHON

Dictionaries of unknown structure - defaultdict

DATA TYPES FOR DATA SCIENCE IN PYTHON

Jason Myers
Instructor

Dictionary Handling

```
for park_id, name in nyc_eateries_parks:
    if park_id not in eateries_by_park:
       eateries_by_park[park_id] = []
    eateries_by_park[park_id].append(name)

print(eateries_by_park['M010'])
```

```
{'MOHAMMAD MATIN','PRODUCTS CORP.', 'Loeb Boathouse Restaurant',
'Nandita Inc.', 'SALIM AHAMED', 'THE NY PICNIC COMPANY',
'THE NEW YORK PICNIC COMPANY, INC.', 'NANDITA, INC.',
'JANANI FOOD SERVICE, INC.'}
```

Using defaultdict

- Pass it a default type that every key will have even if it doesn't currently exist
- Works exactly like a dictionary

```
from collections import defaultdict
eateries_by_park = defaultdict(list)
for park_id, name in nyc_eateries_parks:
    eateries_by_park[park_id].append(name)
print(eateries_by_park['M010'])
```

```
{'MOHAMMAD MATIN','PRODUCTS CORP.', 'Loeb Boathouse Restaurant',
'Nandita Inc.', 'SALIM AHAMED', 'THE NY PICNIC COMPANY',
'THE NEW YORK PICNIC COMPANY, INC.', 'NANDITA, INC.', ...}
```

defaultdict (cont.)

```
from collections import defaultdict

eatery_contact_types = defaultdict(int)

for eatery in nyc_eateries:
    if eatery.get('phone'):
        eatery_contact_types['phones'] += 1
    if eatery.get('website'):
        eatery_contact_types['websites'] += 1

print(eatery_contact_types)
```

```
defaultdict(<class 'int'>, {'phones': 28, 'websites': 31})
```

Let's practice!

DATA TYPES FOR DATA SCIENCE IN PYTHON

Maintaining Dictionary Order with OrderedDict

DATA TYPES FOR DATA SCIENCE IN PYTHON

Jason Myers
Instructor

Order in Python dictionaries

- Python version < 3.6 NOT ordered
- Python version > 3.6 ordered

Getting started with OrderedDict

```
from collections import OrderedDict

nyc_eatery_permits = OrderedDict()

for eatery in nyc_eateries:
    nyc_eatery_permits[eatery['end_date']] = eatery

print(list(nyc_eatery_permits.items())[:3]
```

```
('2029-04-28', {'name': 'Union Square Seasonal Cafe',
'location': 'Union Square Park', 'park_id': 'M089',
'start_date': '2014-04-29', 'end_date': '2029-04-28',
'description': None, 'permit_number': 'M89-SB-R', ...})
```

OrderedDict power feature

- `.popitem()` method returns items in reverse insertion order

```
print(nyc_eatery_permits.popitem())
```

```
('2029-04-28', {'name': 'Union Square Seasonal Cafe',
'location': 'Union Square Park', 'park_id': 'M089',
'start_date': '2014-04-29', 'end_date': '2029-04-28',
'description': None, 'permit_number': 'M89-SB-R', ...})
```

```
print(nyc_eatery_permits.popitem())
```

```
('2027-03-31', {'name': 'Dyckman Marina Restaurant',
'location': 'Dyckman Marina Restaurant', 'park_id': 'M028',
'start_date': '2012-04-01', 'end_date': '2027-03-31', ...})
```

OrderedDict power feature (2)

- You can use the `last=False` keyword argument to return the items in insertion order

```
print(nyc_eatery_permits.popitem(last=False))
```

```
('2012-12-07', {'name': 'Mapes Avenue Ballfields Mobile Food Truck',
'location': 'Prospect Avenue, E. 181st Street', 'park_id': 'X289',
'start_date': '2009-07-01', 'end_date': '2012-12-07',
'description': None, 'permit_number': 'X289-MT', 'phone': None,
'website': None, 'type_name': 'Mobile Food Truck'})
```

Let's practice!

DATA TYPES FOR DATA SCIENCE IN PYTHON

namedtuple

DATA TYPES FOR DATA SCIENCE IN PYTHON

Jason Myers
Instructor

What is a namedtuple?

- A tuple where each position (column) has a name
- Ensure each one has the same properties
- Alternative to a `pandas` DataFrame row

Creating a namedtuple

- Pass a name and a list of fields

```
from collections import namedtuple
Eatery = namedtuple('Eatery', ['name', 'location', 'park_id
   ...: 'type_name'])
eateries = []
for eatery in nyc_eateries:
    details = Eatery(eatery['name'],
                     eatery['location'],
                     eatery['park_id'],
                     eatery['type_name'])
    eateries.append(details)
```

Print the first element

```
print(eateries[0])
```

```
Eatery(name='Mapes Avenue Ballfields Mobile Food Truck',
location='Prospect Avenue, E. 181st Street',
park_id='X289', type_name='Mobile Food Truck')
```

Leveraging namedtuples

- Each field is available as an attribute of the namedtuple

```
for eatery in eateries[:3]:
    print(eatery.name)
    print(eatery.park_id)
    print(eatery.location)
```

```
Mapes Avenue Ballfields Mobile Food Truck
X289
Prospect Avenue, E. 181st Street

Claremont Park Mobile Food Truck
X008
East 172 Street between Teller & Morris avenues ...
```

Let's practice!

DATA TYPES FOR DATA SCIENCE IN PYTHON