DATA TYPES FOR DATA SCIENCE

Introduction and lists

Jason Myers
Instructor

Data types

- Data type system sets the stage for the capabilities of the language
- Understanding data types empowers you as a data scientist

Container sequences

- Hold other types of data
- Used for aggregation, sorting, and more
- Can be mutable (list, set) or immutable (tuple)
- Iterable

Lists

- Hold data in order it was added
- Mutable
- Index

```
In [1]: cookies = ['chocolate chip', 'peanut butter', 'oatmeal', 'sugar']

In [2]: cookies.append('Tirggel')

In [3]: print(cookies)
['chocolate chip', 'peanut butter', 'oatmeal', 'sugar', 'Tirggel']

In [4]: print(cookies[2])
oatmeal
```

Combining Lists

- Using operators, you can combine two lists into a new one

```
In [1]: cakes = ['strawberry', 'vanilla']

In [2]: desserts = cookies + cakes

In [3]: print(desserts)
['chocolate chip', 'peanut butter', 'oatmeal', 'sugar', 'Tirggel',
'strawberry', 'vanilla']
```

- .extend() method merges a list into another list at the end

Finding and Removing Elements in a List

- .index() method locates the position of a data element in a list

```
In [1]: position = cookies.index('sugar')

In [2]: print(position)
3

In [3]: cookies[3]
'sugar'
```

- .pop() method removes an item from a list and allows you to save it

```
In [1]: name = cookies.pop(position)

In [2]: print(name)
sugar

In [3]: print(cookies)
['chocolate chip', 'peanut butter', 'oatmeal', 'Tirggel',
'Biscotti', 'digestive', 'fortune']
```

Iterating and Sorting

- for loops are the most common way of interating over a list

```
In [1]: for cookie in cookies:
   ...:     print(cookie)
chocolate chip
peanut butter
oatmeal
Tirggel
Biscotti
digestive
fortune
```

- sorted() function sorts data in numerical or alphabetical order and returns a new list

```
In [1]: print(cookies)
['chocolate chip', 'oatmeal', 'Tirggel', 'Biscotti', 'digestive', 'fortune']

In [2]: sorted_cookies = sorted(cookies)

In [3]: print(sorted_cookies)
['Biscotti', 'Tirggel', 'chocolate chip', 'digestive', 'fortune', 'oatmeal']
```

DATA TYPES FOR DATA SCIENCE

Let's practice!

DATA TYPES FOR DATA SCIENCE

Meet the Tuples

Jason Myers
Instructor

Tuple, Tuple

- Hold data in order
- Index
- *Immutable*
- Pairing
- Unpackable

Zipping and Unpacking

- Tuples are commonly created by zipping lists together with zip()
- Two lists: us_cookies, in_cookies

```
In [1]: top_pairs = zip(us_cookies, in_cookies)

In [2]: print(top_pairs)
[('Chocolate Chip', 'Punjabi'), ('Brownies', 'Fruit Cake Rusk'),
('Peanut Butter', 'Marble Cookies'), ('Oreos', 'Kaju Pista Cookies'),
('Oatmeal Raisin', 'Almond Cookies')]
```

- Unpacking tuples is a very expressive way for working with data

```
In [1]: us_num_1, in_num_1 = top_pairs[0]

In [2]: print(us_num_1)
Chocolate Chip

In [3]: print(in_num_1)
Punjabi
```

More Unpacking in Loops

- Unpacking is especially powerful in loops

```
In [1]: for us_cookie, in_cookie in top_pairs:
   ...:     print(in_cookie)
   ...:     print(us_cookie)
Punjabi
Chocolate Chip
Fruit Cake Rusk
Brownies
# ..etc..
```

Enumerating positions

- Another useful tuple creation method is the enumerate() function
- Enumeration is used in loops to return the position and the data in that position while looping

```
In [1]: for idx, item in enumerate(top_pairs):
   ...:     us_cookie, in_cookie = item
   ...:     print(idx, us_cookie, in_cookie)
(0, 'Chocolate Chip', 'Punjabi')
(1, 'Brownies', 'Fruit Cake Rusk')
# ..etc..
```

Be careful when making tuples

- Use zip(), enumerate(), or () to make tuples

```
In [1]: item = ('vanilla', 'chocolate')

In [2]: print(item)
('vanilla', 'chocolate')
```

- Beware of tailing commas!

```
In [1]: item2 = 'butter',

In [2]: print(item2)
('butter',)
```

DATA TYPES FOR DATA SCIENCE

Let's practice!

DATA TYPES FOR DATA SCIENCE

Sets for unordered and unique data

Jason Myers
Instructor

Set

- Unique
- Unordered
- Mutable
- Python's implementation of Set Theory from Mathematics

Creating Sets

- Sets are created from a list

```
In [1]: cookies_eaten_today = ['chocolate chip', 'peanut butter',
   ...: 'chocolate chip', 'oatmeal cream', 'chocolate chip']

In [2]: types_of_cookies_eaten = set(cookies_eaten_today)

In [3]: print(types_of_cookies_eaten)
set(['chocolate chip', 'oatmeal cream', 'peanut butter'])
```

Modifying Sets

- .add() adds single elements
- .update() merges in another set or list

```
In [1]: types_of_cookies_eaten.add('biscotti')

In [2]: types_of_cookies_eaten.add('chocolate chip')

In [3]: print(types_of_cookies_eaten)
set(['chocolate chip', 'oatmeal cream', 'peanut butter', 'biscotti'])

In [4]: cookies_hugo_ate = ['chocolate chip', 'anzac']

In [5]: types_of_cookies_eaten.update(cookies_hugo_ate)

In [6]: print(types_of_cookies_eaten)
set(['chocolate chip', 'anzac', 'oatmeal cream', 'peanut butter',
'biscotti'])
```

Removing data from sets

- .discard() safely removes an element from the set by value
- .pop() removes and returns an arbitrary element from the set

  (KeyError when empty)

```
In [1]: types_of_cookies_eaten.discard('biscotti')

In [2]: print(types_of_cookies_eaten)
set(['chocolate chip', 'anzac', 'oatmeal cream', 'peanut butter',
'biscotti'])

In [3]: types_of_cookies_eaten.pop()
'chocolate chip'

In [4]:types_of_cookies_eaten.pop()
'anzac'
```

Set Operations - Similarities

- .union() set method returns a set of all the names (|)
- .intersection() method identifies overlapping data (&)

```
In [1]: cookies_jason_ate = set(['chocolate chip', 'oatmeal cream',
   ...: 'peanut butter'])

In [2]: cookies_hugo_ate = set(['chocolate chip', 'anzac'])

In [3]: cookies_jason_ate.union(cookies_hugo_ate)
set(['chocolate chip', 'anzac', 'oatmeal cream', 'peanut butter'])

In [4]: cookies_jason_ate.intersection(cookies_hugo_ate)
set(['chocolate chip'])
```

Set Operations - Differences

- .difference() method identifies data present in the set on which the method was used that is not in the arguments (-)
- Target is important!

```
In [1]: cookies_jason_ate.difference(cookies_hugo_ate)
set(['oatmeal cream', 'peanut butter'])

In [2]: cookies_hugo_ate.difference(cookies_jason_ate)
set(['anzac'])
```

DATA TYPES FOR DATA SCIENCE

Let's practice!