Using dictionaries

DATA TYPES IN PYTHON

Jason Myers
Instructor

datacamp

Creating and looping through dictionaries

- Hold data in key/value pairs
- Nestable (use a dictionary as the value of a key within a dictionary)
- Iterable
- Created by `dict()` or `{}`

```
art_galleries = {}

for name, zip_code in galleries:
    art_galleries[name] = zip_code
```

Printing in the loop

```
for name in sorted(art_galleries)[-5:]:
    print(name)
```

```
Zwirner David Gallery
Zwirner & Wirth
Zito Studio Gallery
Zetterquist Galleries
Zarre Andre Gallery
```

Safely finding by key

```
art_galleries['Louvre']
```

```
|---------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-1-4f51c265f287> in <module>()
--> 1 art_galleries['Louvre']

KeyError: 'Louvre'
```

- Getting a value from a dictionary is done using the key as an index
- If you ask for a key that does not exist that will stop your program from running in a KeyError

Safely finding by key (cont.)

- `.get()` method allows you to safely access a key without error or exception handling
- If a key is not in the dictionary, `.get()` returns `None` by default or you can supply a value to return

```
art_galleries.get('Louvre', 'Not Found')
```

```
'Not Found'
```

```
art_galleries.get('Zarre Andre Gallery')
```

```
'10011'
```

Let's practice!

DATA TYPES IN PYTHON

Altering dictionaries

DATA TYPES IN PYTHON

Jason Myers
Instructor

Adding and extending dictionaries

- Assignment to add a new key/value to a dictionary
- `.update()` method to update a dictionary from another dictionary, tuples or keywords

```
print(galleries_10007)
```

```
{'Nyabinghi Africian Gift Shop': '(212) 566-3336'}
```

```
art_galleries['10007'] = galleries_10007
```

Updating a dictionary

```
galleries_11234 = [
  ('A J ARTS LTD', '(718) 763-5473'),
  ('Doug Meyer Fine Art', '(718) 375-8006'),
  ('Portrait Gallery', '(718) 377-8762')]
art_galleries['11234'].update(galleries_11234)
print(art_galleries['11234'])
```

```
{'Portrait Gallery': '(718) 377-8762',
'A J ARTS LTD': '(718) 763-5473',
'Doug Meyer Fine Art': '(718) 375-8006'}
```

Popping and deleting from dictionaries

- `del` instruction deletes a key/value
- `.pop()` method safely removes a key/value from a dictionary.

```
del art_galleries['11234']
galleries_10310 = art_galleries.pop('10310')
print(galleries_10310)
```

```
{'New Dorp Village Antiques Ltd': '(718) 815-2526'}
```

Let's practice!

DATA TYPES IN PYTHON

Pythonically using dictionaries

DATA TYPES IN PYTHON

Jason Myers
Instructor

datacamp

Working with dictionaries more pythonically

- `.items()` method returns an object we can iterate over

```
for gallery, phone_num in art_galleries.items():
    print(gallery)
    print(phone_num)
```

```
'Miakey Art Gallery'
'(718) 686-0788'
'Morning Star Gallery Ltd'
'(212) 334-9330'}
'New York Art Expo Inc'
'(212) 363-8280'
```

Checking dictionaries for data

- `.get()` does a lot of work to check for a key
- `in` operator is much more efficient and clearer

```
'11234' in art_galleries
```

```
False
```

```
if '10010' in art_galleries:
    print('I found: %s' % art_galleries['10010'])
else:
    print('No galleries found.')
```

```
I found: {'Nyabinghi Africian Gift Shop': '(212) 566-3336'}
```

Let's practice!

DATA TYPES IN PYTHON

Mixed data types in dictionaries

DATA TYPES IN PYTHON

Jason Myers
Instructor

datacamp

Working with nested dictionaries

```
art_galleries.keys()
```

```
dict_keys(['10021', '10013', '10001', '10009', '10011',
   ...: '10022', '10027', '10019', '11106', '10128'])
```

```
print(art_galleries['10027'])
```

```
{"Paige's Art Gallery": '(212) 531-1577',
 'Triple Candie': '(212) 865-0783',
 'Africart Motherland Inc': '(212) 368-6802',
 'Inner City Art Gallery Inc': '(212) 368-4941'}
```

- The `.keys()` method shows the keys for a given dictionary

Accessing nested data

```
art_galleries['10027']['Inner City Art Gallery Inc']
```

```
'(212) 368-4941'
```

- Common way to deal with repeating data structures
- Can be accessed using multiple indices or the `.get()` method

Let's practice!

DATA TYPES IN PYTHON