

Nested Structures

Housekeeping

- Assignment 4 due on August 1!
- No OH today :(

Today

- **Recap Dictionaries**
 - **Key value pairs**
 - **How to add/remove**
- Talk about nested structures

Why is this so fast?

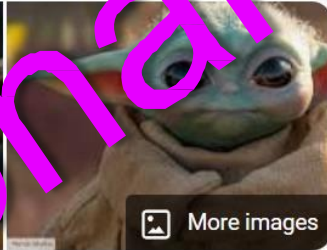
Google yoda

Images Videos News Shopping Maps Books Flights Finance

About 95,200,000 results (0.39 seconds)

 **Yoda**
Star Wars character

Overview Quotes Playlists Appearances Videos

    [More images](#)

 **Fandom**
<https://starwars.fandom.com/wiki/Yoda>

Yoda - Wookieepedia - Fandom

Yoda was a legendary Jedi Master who led the Jedi Order through the time of the High Republic, and in the years leading to its destruction by the Sith ...

Apprentices: Kantam Sy (Padawan)Lula T... Died: 4 ABY, Dagobah
Born: 896 BBY Height: 0.66 meters

Yoda · Yoda's homeworld · Yoda's lightsaber · Yoda's old Master



About

Yoda was a legendary Jedi Master and stood in his connection with the Force. Small in powerful, he trained Jedi for over 800 years in roles in the Clone Wars, the instruction of and unlocking the path to immortality. Star

Affiliations: Jedi Order
Location: Yoda's Hut
Vehicles: Yoda's Fighter
Weapons: Lightsaber, Yoda's Lightsaber
Creator: George Lucas

Actor
Who played yoda in the empire strikes ba



Core Data Structures

Many forms of data can be represented by a combination of:

- dictionaries
- lists
- strings
- floats
- ints
- bools

But sometimes you just need to combine them in funky ways!

Common Use of Dictionary

1. Make an empty dictionary (or dict, for short):

```
phonebook = {}
```

2. Add key/value pairs to the dict

```
name = line[0]      # key from line in file
number = line[1]    # value from line in file
# Creates new key/value pair if key is not in dict.
# Overwrites old value for key if key is in dict.
phonebook[name] = number
```

3. Look up keys in the dict

```
name = "frankie"
if name in phonebook: # see if key exists
    # gets the value associated with the key
    print(name + ": " + phonebook[name])
```

Adding Elements to Dictionary

- Can add pairs to a dictionary:

`phone = {}`

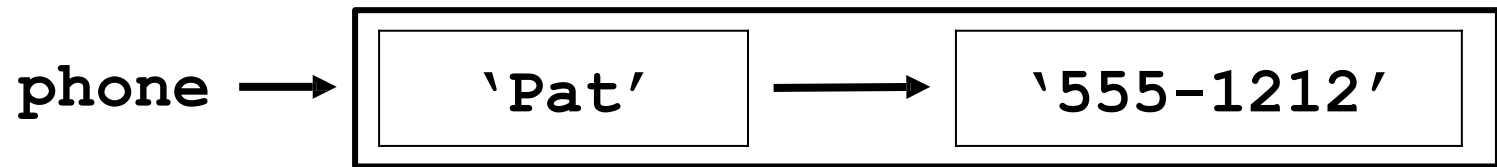
`phone` $\longrightarrow$ *Empty dictionary*

Adding Elements to Dictionary

- Can add pairs to a dictionary:

```
phone = {}
```

```
phone['Pat'] = '555-1212'
```



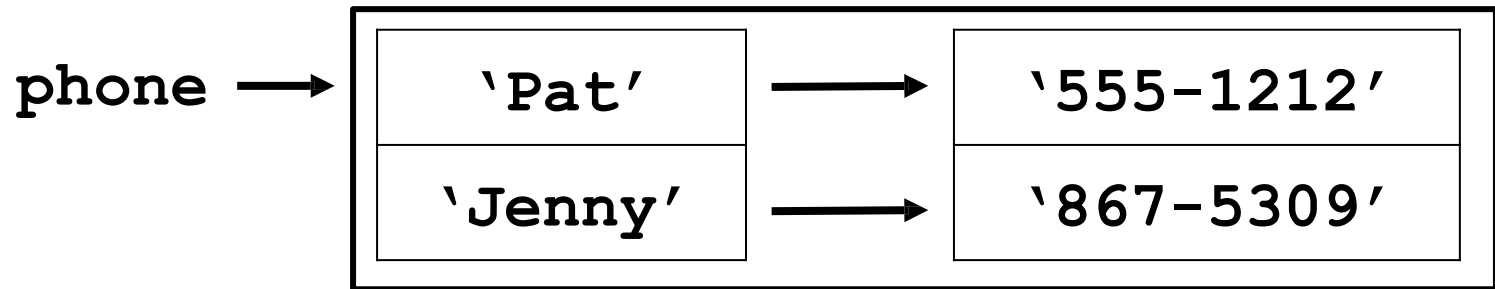
Adding Elements to Dictionary

- Can add pairs to a dictionary:

```
phone = {}
```

```
phone[ 'Pat' ] = '555-1212'
```

```
phone[ 'Jenny' ] = '867-5309'
```



Adding Elements to Dictionary

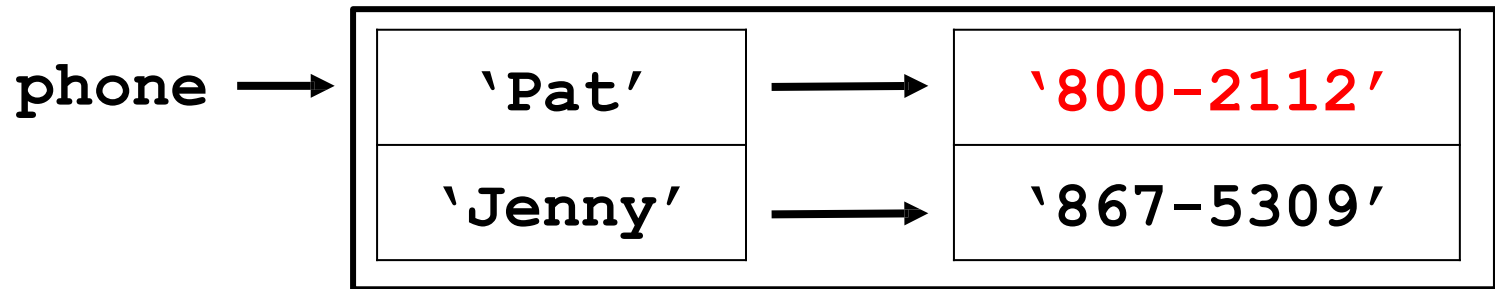
- Can add pairs to a dictionary:

```
phone = {}
```

```
phone['Pat'] = '555-1212'
```

```
phone['Jenny'] = '867-5309'
```

```
phone['Pat'] = '800-2112'
```



Key Idea!

Each key only has one value

If you “put a key in twice”, you are really just changing the value associated with that key

Dictionaries are one way!

In dictionaries you can only look up values by keys.

You can't look up keys by value.



`phonebook [ ' Pat ' ]`



`' 800-2112 '`



`KeyError`



`phonebook [ ' 800-2112 ' ]`



The ultimate 106A problem

Reversing a dictionary

Reversing a dictionary:

Normal dict:

```
{  
    key1: value1,  
    key2: value2  
}
```

Reversed dict:

```
{  
    value1: key1,  
    value2: key2  
}
```

What's the
issue?

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}  
  
reversed = {  
    1970: ['Mehran Sahami', 'Queen Latifah',  
    'Jennifer Connelly'],  
    1988: ['Chris Piech'],  
    2003: ['Pat Jones', 'Jamie Chang']  
}
```

Ultimate CS106A: Reverse a Dict

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        # grab each value in original  
        value = original[key]  
        # make value a key in reversed ?  
        reversed[value] = key ?
```

Ultimate CS106A: Reverse a Dict

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        # grab each value in original  
        value = original[key]  
        #make value a key in reversed if new  
        if value not in reversed:  
            reversed[value] = ??
```

Ultimate CS106A: Reverse a Dict

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        # grab each value in original  
        value = original[key]  
        #make value a key in reversed if new  
        if value not in reversed:  
            reversed[value] = []  
        #values in reversed are lists!  
        reversed[value].append(key)  
  
    return reversed
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
  
        reversed[value].append(key)  
  
    return reversed
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
    return reversed
```

```
reversed = {}
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
reversed = {}
```

```
key = 'Mehran Sahami'
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
    return reversed
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
reversed = {}  
key = 'Mehran Sahami'  
value = 1970
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
    return reversed
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
reversed = {}  
key = 'Mehran Sahami'  
value = 1970
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
  
        reversed[value].append(key)  
  
    return reversed
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
reversed = {  
    1970: []  
}  
key = 'Mehran Sahami'  
value = 1970
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
    return reversed
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami']  
}  
key = 'Mehran Sahami'  
value = 1970
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
  
        reversed[value].append(key)  
  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami']  
}  
key = 'Chris Piech'  
value = 1970
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami']  
}  
key = 'Chris Piech'  
value = 1988
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
  
        reversed[value].append(key)  
  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami']  
}  
key = 'Chris Piech'  
value = 1988
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami']  
    1988: []  
}  
key = 'Chris Piech'  
value = 1988
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
            reversed[value].append(key)  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami']  
    1988: ['Chris Piech']  
}  
key = 'Chris Piech'  
value = 1988
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
  
        reversed[value].append(key)  
  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami']  
    1988: ['Chris Piech']  
}  
  
key = 'Queen Latifah'  
value = 1988
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami']  
    1988: ['Chris Piech']  
}  
key = 'Queen Latifah'  
value = 1970
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
  
        reversed[value].append(key)  
  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami']  
    1988: ['Chris Piech']  
}  
  
key = 'Queen Latifah'  
value = 1970
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
            reversed[value].append(key)  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami', 'Queen Latifah']  
    1988: ['Chris Piech']  
}  
key = 'Queen Latifah'  
value = 1970
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami', 'Queen Latifah']  
    1988: ['Chris Piech']  
    2003: ['Jamie Chang']  
}  
  
key = 'Jamie Chang'  
value = 2003
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami', 'Queen Latifah', 'Jennifer Connelly']  
    1988: ['Chris Piech']  
    2003: ['Jamie Chang']  
}  
key = 'Jennifer Connelly'  
value = 1970
```

Ultimate CS106A: Reverse a Dict

```
birth_year = {  
    'Mehran Sahami': 1970,  
    'Chris Piech': 1988,  
    'Queen Latifah': 1970,  
    'Jamie Chang': 2003,  
    'Jennifer Connelly': 1970,  
    'Pat Jones': 2003  
}
```

```
def reverse(original):  
    reversed = {}  
    for key in original:  
        value = original[key]  
        if value not in reversed:  
            reversed[value] = []  
        reversed[value].append(key)  
    return reversed
```

```
reversed = {  
    1970: ['Mehran Sahami', 'Queen Latifah', 'Jennifer Connelly']  
    1988: ['Chris Piech']  
    2003: ['Jamie Chang', 'Pat Jones']  
}  
key = 'Pat Jones'  
value = 2003
```

Better Phonebook: Multiple Phones Per Person

- In a real contact list, you can store multiple phone numbers for the same person
 - Different numbers denoted by different phone types
 - E.g. home, work, mobile, etc.
- How might we design a data structure for this?

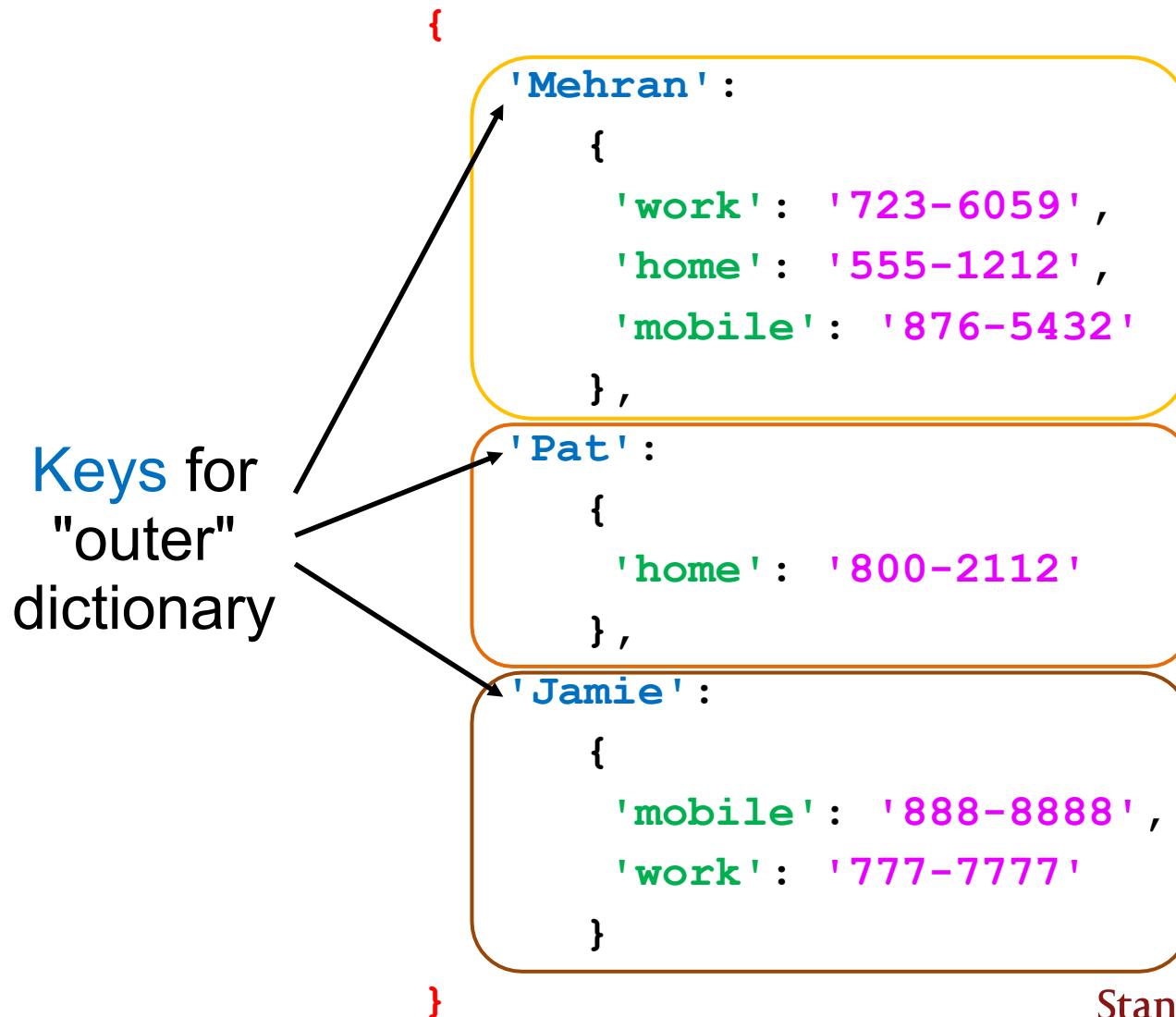
Structuring the Data

```
{  
  'Mehran':  
    {  
      'work': '723-6059',  
      'home': '555-1212',  
      'mobile': '876-5432'  
    },  
  'Pat':  
    {  
      'home': '800-2112'  
    },  
  'Jamie':  
    {  
      'mobile': '888-8888',  
      'work': '777-7777'  
    }  
}
```

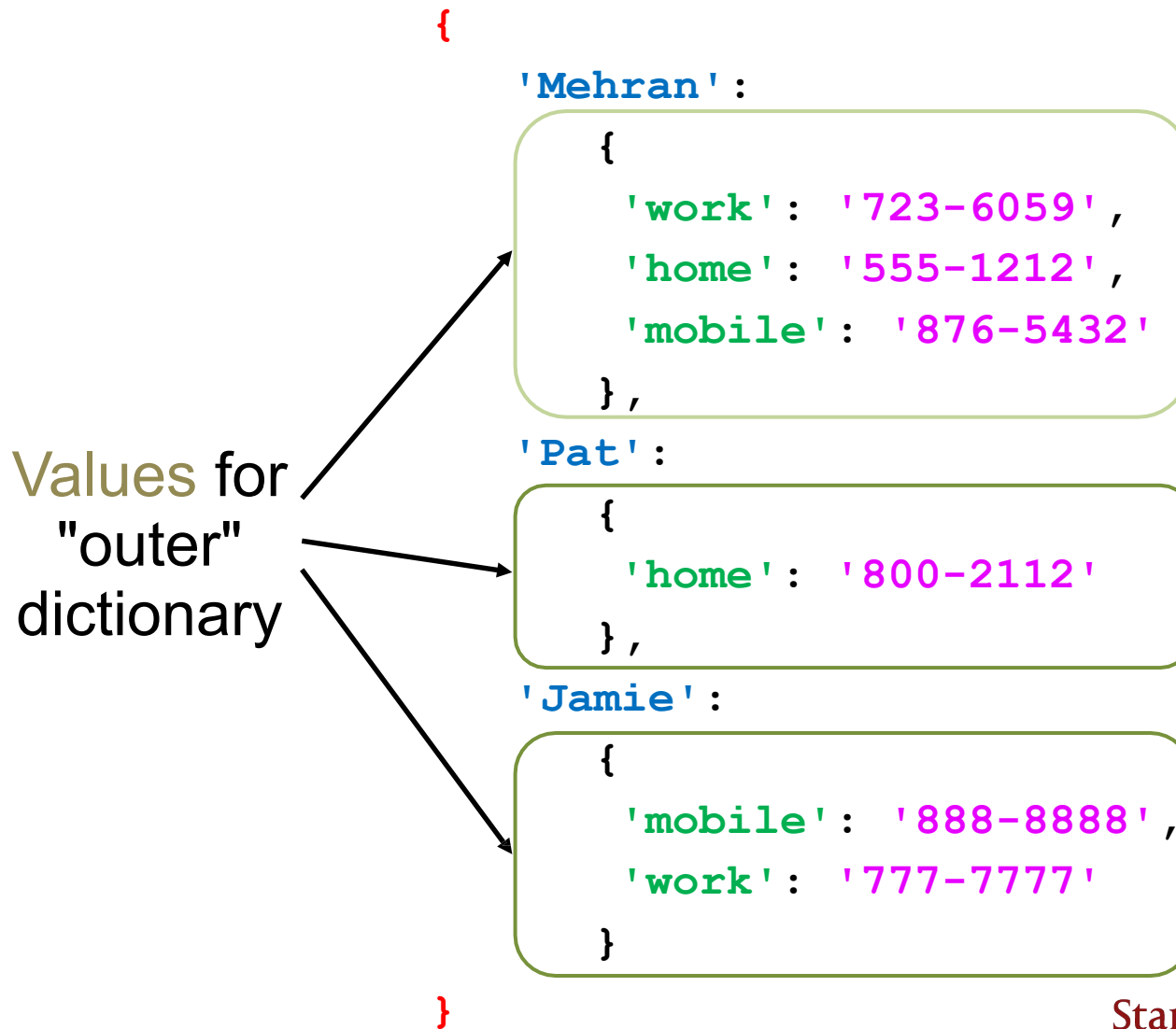
Structuring the Data

```
{  
  'Mehran':  
    {  
      'work': '723-6059',  
      'home': '555-1212',  
      'mobile': '876-5432'  
    },  
  'Pat':  
    {  
      'home': '800-2112'  
    },  
  'Jamie':  
    {  
      'mobile': '888-8888',  
      'work': '777-7777'  
    }  
}
```

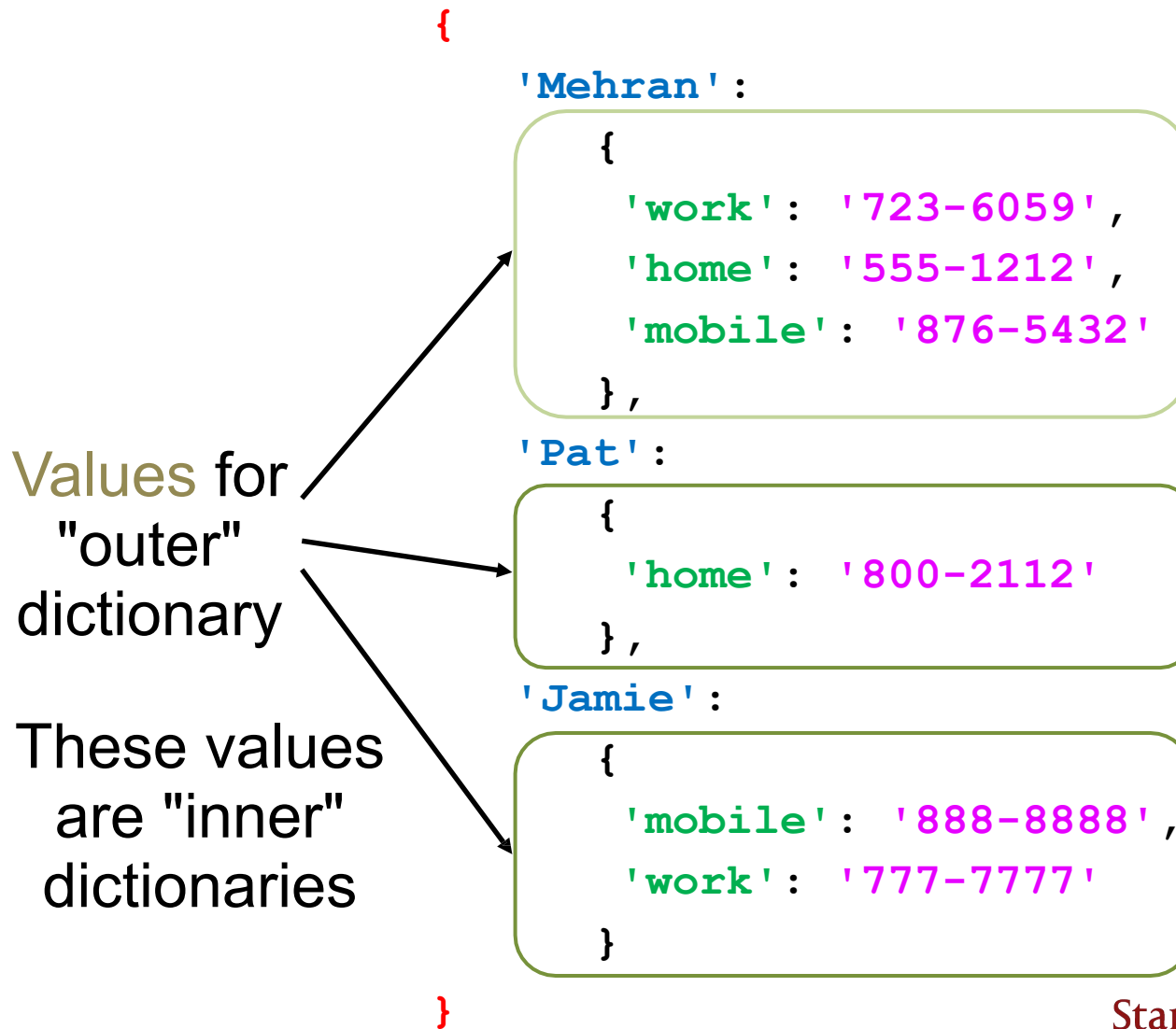
Structuring the Data



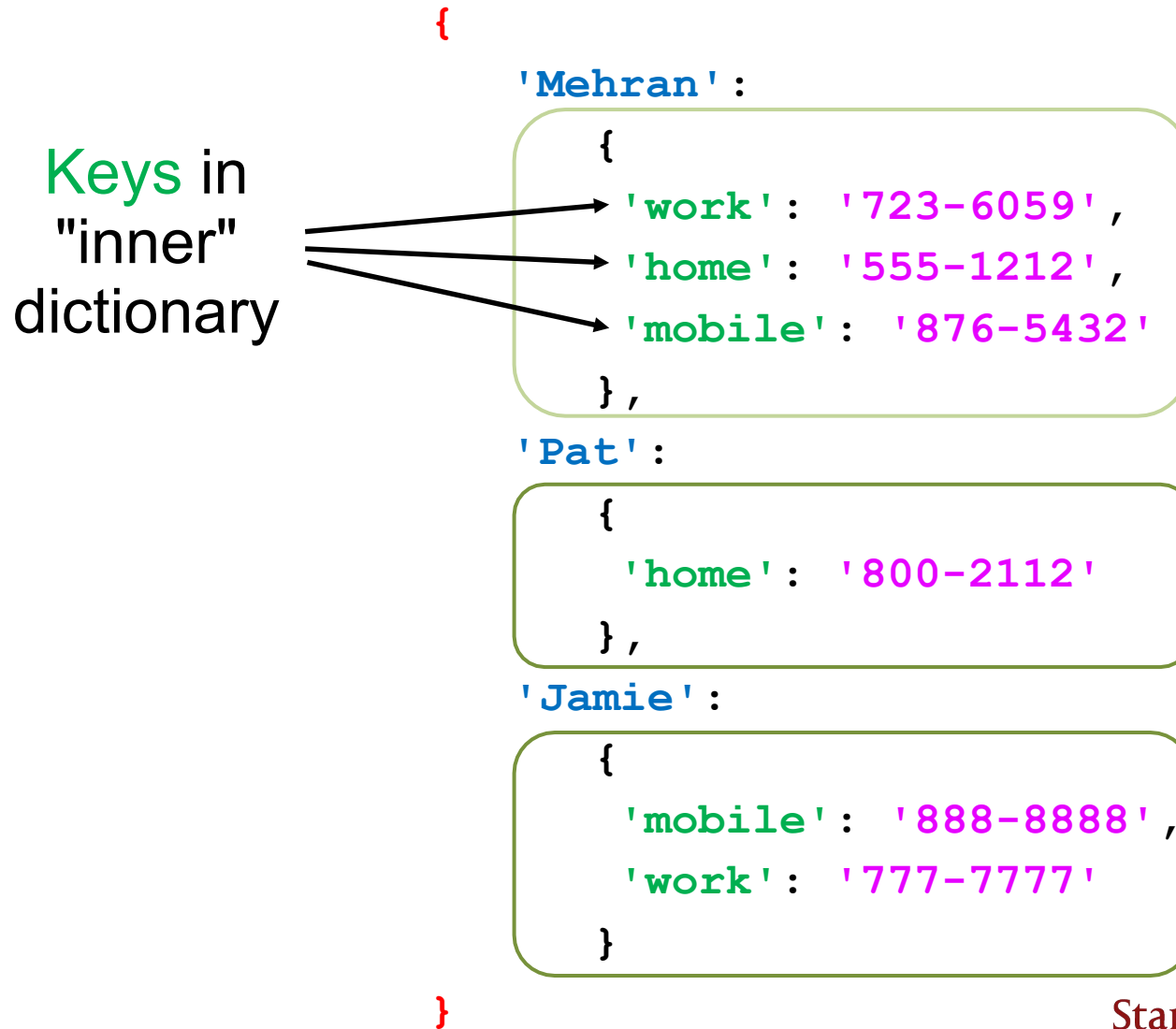
Structuring the Data



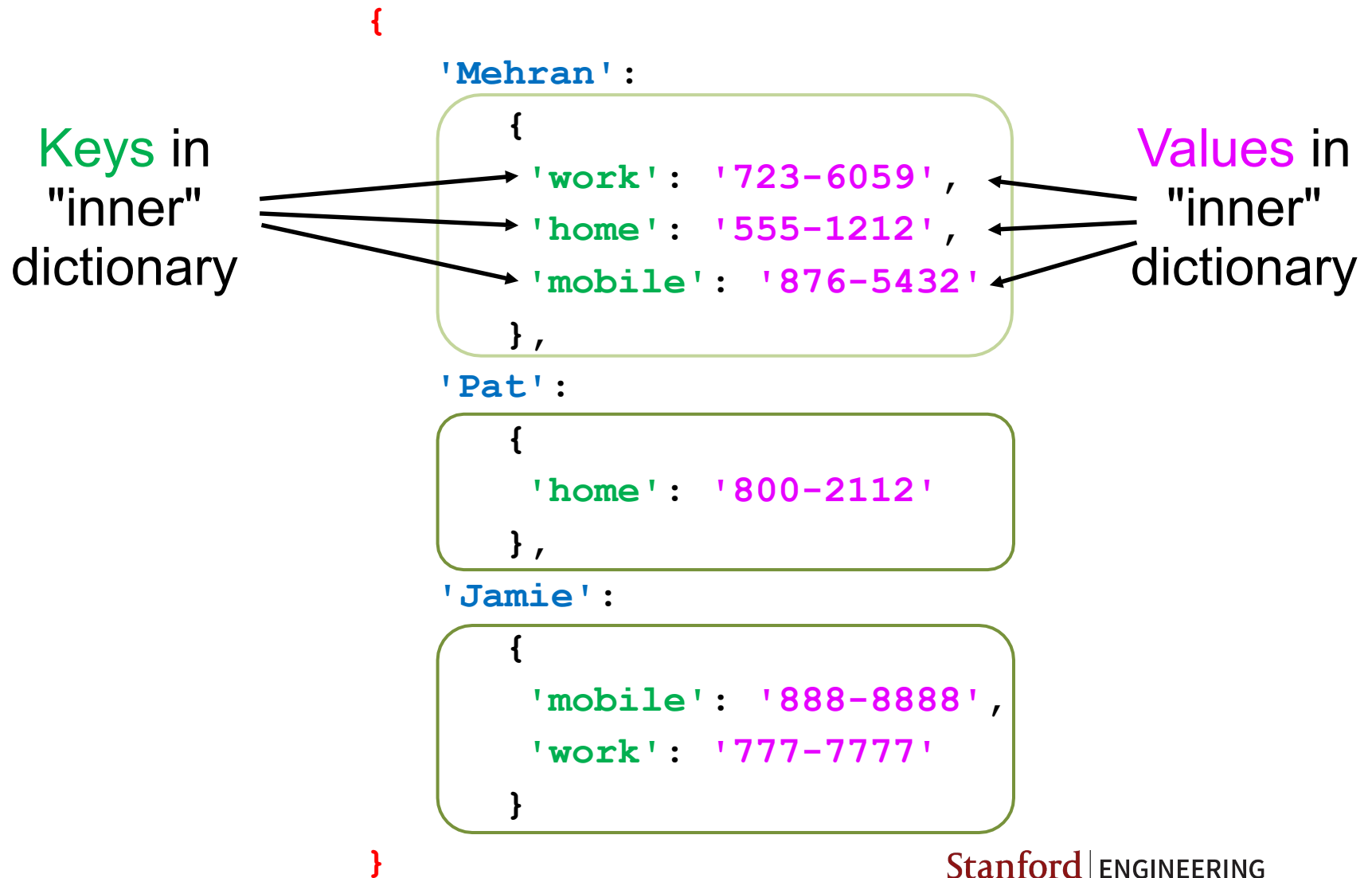
Structuring the Data



Structuring the Data



Structuring the Data



better_phonebook.py

- Write a program that reads in a csv with this format:
`name1 , type1 , phone_number1 , type2 , number2...`
`name2 , type , phone_number...`
...
- And stores the data in a dictionary structured like so:

```
{  
    'name1' : {  
        'type' : 'phone_number' ,  
        'type' : 'phone_number'  
    }  
    'name2' : {  
        ...  
    }  
}
```
- And prints this dictionary!

Pycharm: better_phonebook.py

Recap

- Dictionaries sometimes need to store lists
- Or other dictionaries!
- Practice combining all your data structures in new interesting ways!