

Dictionaries

Oliver W. Layton

CS151: Computational Thinking: Visual Media

Lecture 34, Fall 2020

Friday November 13

Project 10 overview

- Make your own game! Either a more sophisticated Asteroids or any game you like.
- If mechanics similar to Asteroids (events, enemies, etc), **use inheritance** to minimize code duplication.
- Design your game **before** writing code. This design should be included in report.
- Make demo video **no longer than 90 seconds**, include in submitted .ZIP file.
- On last day of class (Nov 24), you will demo your games for the class and talk about how you designed game (at high level)!
- **Let me know in advance if you think your internet connection will be unstable.**
Most likely, I will show your demo video that you submitted and you can talk over it.

Dictionaries

Dictionaries vs lists

In certain cases, downside of lists: need **numeric index** to access data:

```
students = ['Paul', 'Joel', 'Kim', 'Rose', 'Victoria']  
grades = [99, 93, 88, 80, 95]  
# Get Kim's grade  
grades[2] # Not intuitive! Want: grades['Kim']
```

- A dictionary (also called an **associative array** or **map**) works like a list, but you don't access data with an index. It works more like a address book (or dictionary!).
- Using the dictionary metaphor: We look up a definition using a word.
- Another way to say this: We look up definition/data (**value**) with a **key** (word).

Creating dictionaries

New dictionary:

```
myDictionary = {}
```

Creating dictionaries with data

Initialize dictionaries with some keys and values, separating each key-value pair with a colon and multiple pairs with a comma (just like lists)

```
students = ['Paul', 'Joel', 'Kim', 'Rose', 'Victoria']
```

```
grades = [99, 93, 88, 80, 95]
```

```
myDictionary = {'Paul': 99,  
                'Joel': 93,  
                'Kim': 88,  
                'Rose': 80,  
                'Victoria': 95}
```

```
myDictionary['Kim'] # 88
```

Values in dictionaries can be any data

```
myDictionary = { 'shapes' : [ 'Square' , 'Circle' ],  
                 'gpa' : 4.0 ,  
                 'name' : 'bob' }
```

- Keys should be strings, values can be any type of data.

Accessing/modifying data in dictionaries

Access data with square brackets, with a key of course:

```
>>> myDictionary['shapes']  
['Square', 'Circle']
```

Modify data with square brackets associated with a key:

```
>>> myDictionary['gpa'] = 3.8
```

Adding new key-value pair to dictionary:

```
>>> myDictionary['favorite color'] = 'blue'
```

Keys must be strings, but can be variables

```
myDict = {}
names = ['Bill', 'Carol', 'Ben']
colors = ['Blue', 'Magenta', 'Purple']
for i in range(len(names)):
    myDict[names[i]] = colors[i]
    # 'Bill' -> 'Blue'
    # 'Carol' -> 'Magenta'
    # 'Ben' -> 'Purple'
```

Checking membership and printing a dictionary

Similar to lists:

```
key = 'favorite color'
if key in myDictionary:
    print('My favorite color is', myDictionary[key])
```