

CSC108H Lecture 18

Dan Zingaro

October 22, 2012

ConcepTest

Which of the following is best suited for a dictionary instead of a list?

- ▶ A. The order in which people finish a race
- ▶ B. The ingredients necessary for a recipe
- ▶ C. The names of world countries and their capital cities
- ▶ D. 50 random integers

Parallel Lists

Let's say we're bird-watching, and we want to keep track of the number of each type we've seen

kind	count
peregrine falcon	1
harrier falcon	5
red-tailed hawk	2
osprey	11

- ▶ One approach: parallel lists
- ▶ The element `kinds[i]` corresponds with `counts[i]`

```
kinds = ['peregrine falcon', 'harrier falcon',  
         'red-tailed hawk', 'osprey']  
counts = [1, 5, 2, 11]
```

ConceptTest

```
def new_sighting(kinds, counts, sighting):  
    '''(list of str, list of int, str) -> NoneType  
    Add new sighting to parallel lists kinds and counts.  
    '''  
  
    if sighting not in kinds:  
        kinds.append(sighting)  
        ... missing code  
    ind = kinds.index(sighting)  
    counts[ind] = counts[ind] + 1
```

What code should go in place of the missing code?

- ▶ A. `counts.append(0)`
- ▶ B. `counts.append(1)`
- ▶ C. `counts.append(kind)`
- ▶ D. No code necessary there

Dictionaries vs. Parallel Lists

```
bird_dict = {  
    'peregrine falcon':1, 'harrier falcon':5,  
    'red-tailed hawk':2, 'osprey':11}
```

```
def new_sighting(bird_dict, sighting):  
    if sighting not in bird_dict:  
        bird_dict[sighting] = 0  
    bird_dict[sighting] += 1
```

Compared to parallel lists:

- ▶ Only one dict (not two)
- ▶ No call to index that might search the whole list

Adding to Dictionaries

- ▶ Keys must be immutable
- ▶ Values can be mutable or immutable
- ▶ Use `d[k] = v` to add key `k` with value `v` to dictionary `d`
 - ▶ If `k` is already present, its value is overwritten
- ▶ To copy all key/value pairs from another dictionary, use the `update` method
- ▶ Use `dict` to (shallow) copy another dictionary
- ▶ `update` and `dict` also work with a sequence whose elements are all of length 2

Getting Values from Dictionaries

- ▶ Use `d[k]` to obtain the value associated with key `k` of dictionary `d`
 - ▶ If `k` does not exist, this causes an error
- ▶ The `get` method is similar, except it returns `None` instead of giving an error when the key does not exist
 - ▶ If a second parameter `v` is provided, `get` returns `v` instead of `None` when the key is not found

ConcepTest

What is dictionary `d` created by the following code?

```
d = {3:4}
d[5] = d.get(4, 8)
d[4] = d.get(3, 9)
```

- ▶ A. {3:4, 5:8, 4:9}
- ▶ B. {3:4, 5:8, 4:4}
- ▶ C. {3:4, 5:4, 4:3}
- ▶ D. Error caused by `get`

ConceptTest

What is dictionary `d` created by the following code?

```
d = {1:5}
d[2] = d.get(1, 6)
d[4] = d.get(3, 7)
```

- ▶ A. {1:5, 2:5, 4:7}
- ▶ B. {1:5, 2:6, 4:7}
- ▶ C. {1:5, 2:1, 4:2}
- ▶ D. Error caused by `get`