

Dictionary Views & Iteration

Dictionary Views & Iteration

Here's a Python 3 dictionary:

```
>>> calories = {  
...     "apple": 95,  
...     "slice of bacon": 43,  
...     "cheddar cheese": 113,  
...     "ice cream": 15, # You wish!  
... }  
>>> items = calories.items()  
>>> type(items)  
<class 'dict_items'>  
>>> hasattr(items, '__next__')  
False  
>>> hasattr(items, '__iter__')  
True
```

What is returned by `.items()`?

A dictionary view object.

Quacks like a dictionary view if it supports three things:

- `len(view)` returns the number of items
- `view` is iterable
- `(key, value) in view` returns `True` if that pair is in the dictionary; else, `False`.

Iterable Views

A view is iterable, so you can use it in a for loop:

```
>>> for food, count in calories.items():  
...     print("{: <20s} {: <d} cal".format(food, count))  
...  
ice cream..... 15 cal  
slice of bacon..... 43 cal  
apple..... 95 cal  
cheddar cheese..... 113 cal
```

Dynamically updates

A view dynamically updates, even if the source dictionary changes:

```
>>> items = calories.items()
>>> len(items)
4
>>> calories['orange'] = 50
>>> len(items)
5
>>> ('orange', 50) in items
True
```

Other methods

There are two other methods on dictionaries, called `.keys()` and `.values()`. They also return views.

```
>>> foods = calories.keys()
>>> counts = calories.values()
>>> 'yogurt' in foods
False
>>> 100 in counts
False
>>> calories['yogurt'] = 100
>>> 'yogurt' in foods
True
>>> 100 in counts
True
```

Benefits

Views improve over regular iterators:

- Are iterable, so can spawn multiple iterators
- Let you pass dict contents to caller and know it won't be modified
- Support extra services, like `len()` and `(key, val) in view`

And of course, views are more scalable & performant than a list of (key, value) pairs.

What about Python 2?

All the above was for Python 3. Here's how it works in 2:

- `calories.items()` returns a list of (key, value) tuples.
 - So if it has 100,000 entries...
- `iteritems()`: returns an iterator over the key-value tuples
- `viewitems()`: which returned a view

`iteritems` is basically obsoleted by `viewitems`, but most people don't realize this yet.

Obsolete methods

In Python 3, what used to be called `viewitems()` was renamed `items()`, and the old `items()` and `iteritems()` went away.

If you still need an actual list in Python 3, you can just say
`list(calories.items())`