dict.sorted Documentation

Release 1.0.0-rc1

Luke Sneeringer

October 20, 2015

Contents

This is dict.sorted, which provides subclasses to Python's built-in dictionary which order its keys.

It provides two such subclasses: one for arbitrary sorting based on a method you provide, and one for sorting in alphabetical order.

CHAPTER 1

Python Support

dict.sorted is compatible with Python 2.6, Python 2.7, and Python 3.3 or later.

It makes use of six to provide compatibility with Python 2 and Python 3, and therefore requires six as a dependency.

Note: All output shown in the examples below is generated on Python 3.3. Python 2 output will look slightly different (in particular, the unicode keys will have the leading `u` character).

CHAPTER 2

Getting Help

If you think you've found a bug in dict.sorted itself, please post an issue on the Issue Tracker.

For usage help, you're free to e-mail the author, who will provide help (on a best effort basis) if possible.

CHAPTER 3

AlphaSortedDict

`AlphaSortedDict` is a dictionary subclass that generally behaves exactly like a dictionary, but will always render and iterate over its keys in alphabetical order, rather than arbitrary order of the built-in `dict` type.

The `AlphaSortedDict` constructor accepts a dictionary, any iterable that can coerce to a dictionary (e.g. a tuple of two-tuples), or keyword arguments:

```
>>> from sdict import AlphaSortedDict
>>> d = AlphaSortedDict({ 'x': 1 })
>>> d['a'] = 2
>>> d
{'a': 2, 'x': 1}
```

The `AlphaSortedDict` class is also aliased to `adict` on the same module, for shorter invocation. This example is identical to the one above:

```
>>> from sdict import adict
>>> d = adict({ 'x': 1 })
>>> d['a'] = 2
>>> d
{'a': 2, 'x': 1}
```

Keyword arguments are also acceptable:

```
>>> from sdict import adict
>>> d = adict(x=1, a=2)
>>> d
{'a': 2, 'x': 1}
```

Note: Because `AlphaSortedDict` sorts its keys in alphabetical order, it will coerce all keys to text (`str` in Python 3, `unicode` in Python 2) upon insertion into the dictionary.

CHAPTER 4

SortedDict

`SortedDict` is a dictionary subclass that behaves like a dictionary, except that keys are sorted based on a function you provide.

The `SortedDict` constructor requires, as its first positional argument, a function, which accepts one argument and returns back a value used for ordering (any value that properly answers to comparison operators is fine).

Here is an example of a dictionary that will yield keys in reverse-alphabetical order:

```
>>> from sdict import SortedDict
>>> import six
>>> cmp_func = lambda x: [-ord(i) for i in six.text_type(x)]
>>> d = SortedDict(cmp_func, { 'a': 5, 'b': 10 })
>>> d
{'b': 10, 'a': 5}
```

The `SortedDict` class is aliased to `sdict` on the same module, for shorter invocation.

Note that you are responsible for having a comparison function robust enough to accept and know what to do with any key value that you send.

CHAPTER 5

Running the Tests

To run the tests, clone the package from GitHub and invoke `python test.py`. To run the tests against every supported Python version, assuming you have the appropriate Python binaries on your machine, first install and then run `tox`.

CHAPTER 6

License

New BSD.