
kutils Documentation

Release 0.0.1

Kyle Isom

Jun 28, 2017

Contents:

1	Introduction	3
2	Dictionaries	5
3	API reference: Dictionaries	7
4	Indices and tables	9
	Python Module Index	11

These are various data structures and tools that I think are useful and want to re-use across projects.

CHAPTER 1

Introduction

Most programmers will have their own collection of useful code they'd like to reuse across projects. This is mine. It follows in spirit with my [Lisp kutils package](#) and the [Go analogue](#). Note that these three libraries don't really have much overlap, and reflect the things I tend to use each programming language for.

CHAPTER 2

Dictionaries

There are a few utility dictionaries available. All of them inherit from the base `dict` class, and can be used as drop-ins for the standard `dict` type. There are a few “primitive” base types that are then composed to more complex types. To use these types, import `kutils.dicts`.

- `AttrDict`: this dictionary provides access to the keys as attributes.
- `NoneDict`: this dictionary returns `None` for non-existent keys. This means it will not throw a `KeyError` for top-level keys; note that nested dictionaries may not be `NoneDict`'s.
- `StrDict`: this dictionary returns an empty string for non-existent keys.
- `DictDict`: this dictionary returns an empty `StrDict` for non-existent keys. This is mostly useful for values that are nested dictionaries of keys.
- `AttrNoneDict`: this is a composition of the `AttrDict` and the `NoneDict`: it provides access to its keys as attributes, and returns `None` for missing keys or attributes; this means missing top-level keys and attributes will not throw `KeyError` or `AttributeError`.
- `AttrDictDict`: this is a composition of the `AttrDict` and `DictDict`: it provides access to its keys as attributes, and returns an empty `StrDict` for missing keys or attributes; this means missing top-level keys and attributes will not throw `KeyError` or `AttributeError`.

API reference: Dictionaries

The `kutils.dicts` module provides several utility dictionaries. `dicts` contains a number of special-case dictionaries.

class `kutils.dicts.AttrDict(*args, **kwargs)`

`AttrDict` represents a dictionary where the keys are represented as attributes.

```
>>> d = kd.AttrDict(foo='bar')
>>> d.foo
'bar'
>>> d['foo']
'bar'
```

class `kutils.dicts.AttrDictDict(*args, **kwargs)`

`AttrDictDict` is a dictionary that returns an empty `StrDict` if a key is unavailable. This is meant for a dictionary of dictionaries of string values.

```
>>> d = kd.AttrDictDict()
>>> d.foo
{}
>>> d['foo']
{}
>>> d.foo.bar
''
>>> d['foo']['bar']
''
```

class `kutils.dicts.AttrNoneDict(*args, **kwargs)`

`AttrNoneDict` returns an `AttrDict` that returns `None` if a key isn't present.

```
>>> d = kd.AttrNoneDict(foo='bar')
>>> d.foo
'bar'
>>> d['foo']
'bar'
>>> d.bar is None
```

```
True
>>> d['bar'] is None
True
```

class `kutils.dicts.AttrStrDict (*args, **kwargs)`

`AttrStrDict` returns an `AttrDict` that returns an empty string if a key isn't present.

```
>>> d = kd.AttrStrDict(foo='bar')
>>> d.foo
'bar'
>>> d['foo']
'bar'
>>> d.bar
''
>>> d['bar']
''
```

class `kutils.dicts.DictDict (*args, **kwargs)`

`DictDict` is a dictionary that returns an empty `AttrStrDict` if a key is unavailable. This is meant for a dictionary of dictionaries of string values.

```
>>> d = kd.DictDict()
>>> d['foo']
{}
>>> type(d['foo'])
<class 'kutils.dicts.AttrStrDict'>
>>> d.foo.bar
''
```

class `kutils.dicts.NoneDict (*args, **kwargs)`

`NoneDict` is a dictionary that returns `None` if a key is unavailable.

```
>>> d = kd.NoneDict(foo='bar')
>>> d['foo']
'bar'
>>> d['baz'] is None
True
```

class `kutils.dicts.StrDict (*args, **kwargs)`

`StrDict` is a dictionary that returns an empty string if a key is unavailable.

```
>>> d = kd.StrDict(foo='bar')
>>> d['foo']
'bar'
>>> d['baz']
''
```

CHAPTER 4

Indices and tables

- `genindex`
- `modindex`
- `search`

k

`kutils.dicts`, 7

A

`AttrDict` (class in `kutils.dicts`), [7](#)
`AttrDictDict` (class in `kutils.dicts`), [7](#)
`AttrNoneDict` (class in `kutils.dicts`), [7](#)
`AttrStrDict` (class in `kutils.dicts`), [8](#)

D

`DictDict` (class in `kutils.dicts`), [8](#)

K

`kutils.dicts` (module), [7](#)

N

`NoneDict` (class in `kutils.dicts`), [8](#)

S

`StrDict` (class in `kutils.dicts`), [8](#)