
dk Documentation

Release 0.7.8

Bjorn Pettersen

January 15, 2017

1	Project information	3
2	dk.ttcab	5
3	dk.fstr	7
4	dk.collections	9
5	dk.html	13
6	dk.text	15
7	Indices and tables	17
	Python Module Index	19

Installation:

```
pip install dk
```

Contents:

Project information

The project home page is at:

<http://thebjorn.github.io/dk/>

The github directory is at:

<https://github.com/thebjorn/dk>

Report any issues at:

<https://github.com/thebjorn/dk/issues>

dk.ttcal

Date classes (originally from TikTok).

`dk.ttcal.from_idtag(idtag)`

Return a class from idtag.

dk.fstr

class dk.fstr.fstr

String sub-class with a split() method that splits a given indexes.

Usage:

```
>>> r = fstr('D2008022002')
>>> print r.split(1, 5, 7, 9)
['D', '2008', '02', '20', '02']
>>> _, year, _ = r.split(1,5)
>>> year
'2008'
```

dk.collections

Inversible dictionary.

class dk.collections.invdict.**invdict**

Inversible dict:

```
>>> -invdict({'key': 'val'}) == {'val': 'key'}
```

class dk.collections.mmap.**mmap** (**attrs)

Multi Map class, ie. a key/value collection where each key can occur multiple times. Implemented as a list of key/value tuples.

add (key, val)

append (kv)

Ordered Set. Items can only be added once, further additions have no effect. The iterator iterates over the items in insertion order.

class dk.collections.OrderedSet.**oset** (iterable=())

Ordered Set. Items can only be added once, further additions have no effect. The iterator iterates over the items in insertion order.

add (item)

Mapping classes.

class dk.collections.pset.**defset** (defval)

pset with default value.

class dk.collections.pset.**keyval**

keyval(key, val)

key

Alias for field number 0

val

Alias for field number 1

class dk.collections.pset.**pset** (items=(), **attrs)

This code is placed in the Public Domain, or released under the wtfpl (<http://sam.zoy.org/wtfpl/COPYING>) wherever PD is problematic.

Property Set class. A property set is an object where values are attached to attributes, but can still be iterated over as key/value pairs. The order of assignment is maintained during iteration. Only one value allowed per key.

```
>>> x = pset()
>>> x.a = 42
>>> x.b = 'foo'
>>> x.a = 314
>>> x
pset(a=314, b='foo')
```

apply (*fn*)
Apply function *fn* to all values in self.

items ()

keys ()

pprint (*indent=0, tab=' ', seen=None*)
Pretty print the pset, indented.

remove (*key*)
Remove key from client vars.

values ()

class dk.collections.pset.**record** (*items=()*, ***attrs*)
A property set with commit, rollback, and encoding translation.

changed ()
Return list of fields that have changed since last commit.

commit ()
Copy current state to *self._history*

decode (*encoding*)
Decode using *encoding*.

encode (*encoding*)
Encode using *encoding*.

fields
Verbose name of all fields.

rollback ()
Copy snapshot from *self._history* into self.

strvals (*empty='', none='NULL', encoding='u8'*)
Return a list of all values, formatted for human consumption.

trans (*source='iso-8859-1', dest='utf-8'*)
Translate encoding.

dk.collections.pset.**test_pset** ()
Unit tests...

```
>>> request = pset(REQUEST={}, META={}, path='/', user=None, session={}, method='GET',
...               COOKIES={}, LANGUAGE_CODE='no')
>>> p = page(request)
>>> p.forms = 'fruit'
>>> p.forms.foo = 'bar'
>>> print p.forms.foo
bar
>>> p.forms.fob = 'baz'
>>> print p.forms.fob
baz
>>> x = pset()
```

```

>>> x.a
Traceback (most recent call last):
...
AttributeError: a
>>> y = pset(a=1, b=2, c=3)
>>> y.a
1
>>> y.b
2
>>> y.c
3
>>> z = pset()
>>> z.a = 1
>>> z.b = 2
>>> z.c = 3
>>> z[1]
2
>>> z
pset(a=1, b=2, c=3)
>>> class Point(pset): pass
>>> p = Point(x=11, y=22)
>>> p
Point(y=22, x=11)

```

`dk.collections.pset.xmlrepr(v, toplevel=False)`
Return `v` as xml tag-soup.

class `dk.collections.sdict.sdict(**attrs)`
Sorted Dictionary class. Iterating over the sdict will give back the key/value pairs in order of insertion. A key can only be in a sdict once.

keys()

values()

`dk.collections.xmlrec.Boolean(s)`

`dk.collections.xmlrec.Date(s)`

`dk.collections.xmlrec.Datetime(s)`

`dk.collections.xmlrec.NOK(s)`

class `dk.collections.xmlrec.xmlrec(soup, **types)`

convert = {'date': <function Date at 0x7fea219c72a8>, 'int': <type 'int'>, 'NOK': <function NOK at 0x7fea219c71b8>,

Attempt at defining themes from Python (probably better to do this elsewhere).

```
dk.html.theme.mkpalette(_source, base, colorlist)
    Convenience function to define a palette.
```

```
class dk.html.theme.palette
    Color palettes from http://colormatch.dk. Colors are named with a lower case x in front of the nearest defined colorvalue.
```

```
class LightSkyBlue2
```

```
    Gray7 = '#121212'
```

```
    Gray93 = '#ededed'
```

```
    LightSkyBlue2 = '#a4d3ee'
```

```
    get = ['#a4d3ee', '#6f8ea1', '#edd7a4', '#a1896f', '#121212', '#ededed']
```

```
    xLemonChiffon4 = '#a1896f'
```

```
    xLightSlateGray = '#6f8ea1'
```

```
    xNavajoWhite2 = '#edd7a4'
```

```
class palette.xForrestGreen
```

```
    Gray48 = '#7a7a7a'
```

```
    Gray52 = '#858585'
```

```
    get = ['#257b24', '#3cc73a', '#5f7a23', '#9ac73a', '#858585', '#7a7a7a']
```

```
    xForrestGreen = '#257b24'
```

```
    xLimeGreen = '#3cc73a'
```

```
    xOliveDrab4 = '#5f7a23'
```

```
    xYellowGreen = '#9ac73a'
```

```
class dk.html.css.css (**attrs)
```

dk.text

`dk.text.u(obj)`

Return `obj` as a unicode string. If `obj` is a (non-)unicode string, then first try to decode it as utf-8, then as iso-8859-1.

`dk.text.u8(obj)`

Return a utf-8 representation of `obj`.

`dk.text.unicode_repr(obj)`

Return `obj` as a unicode string. If `obj` is a (non-)unicode string, then first try to decode it as utf-8, then as iso-8859-1.

`dk.text.utf8(obj)`

Return a utf-8 representation of `obj`.

Indices and tables

- *genindex*
- *modindex*
- *search*

d

- `dk.collections.invdict`, 9
- `dk.collections.mmap`, 9
- `dk.collections.OrderedSet`, 9
- `dk.collections.pset`, 9
- `dk.collections.sdict`, 11
- `dk.collections.xmlrec`, 11
- `dk.fstr`, 7
- `dk.html`, 13
 - `css`, 13
 - `theme`, 13
- `dk.text`, 15
- `dk.ttcalf`, 5

A

add() (dk.collections.mmap.mmap method), 9
add() (dk.collections.OrderedSet.oset method), 9
append() (dk.collections.mmap.mmap method), 9
apply() (dk.collections.pset.pset method), 10

B

Boolean() (in module dk.collections.xmlrec), 11

C

changed() (dk.collections.pset.record method), 10
commit() (dk.collections.pset.record method), 10
convert (dk.collections.xmlrec.xmlrec attribute), 11
css (class in dk.html.css), 13

D

Date() (in module dk.collections.xmlrec), 11
Datetime() (in module dk.collections.xmlrec), 11
decode() (dk.collections.pset.record method), 10
defset (class in dk.collections.pset), 9
dk.collections.invdict (module), 9
dk.collections.mmap (module), 9
dk.collections.OrderedSet (module), 9
dk.collections.pset (module), 9
dk.collections.sdict (module), 11
dk.collections.xmlrec (module), 11
dk.fstr (module), 7
dk.html (module), 13
dk.html.css (module), 13
dk.html.theme (module), 13
dk.text (module), 15
dk.ttcalf (module), 5

E

encode() (dk.collections.pset.record method), 10

F

fields (dk.collections.pset.record attribute), 10
from_idtag() (in module dk.ttcalf), 5
fstr (class in dk.fstr), 7

G

get (dk.html.theme.palette.LightSkyBlue2 attribute), 13
get (dk.html.theme.palette.xForrestGreen attribute), 13
Gray48 (dk.html.theme.palette.xForrestGreen attribute), 13
Gray52 (dk.html.theme.palette.xForrestGreen attribute), 13
Gray7 (dk.html.theme.palette.LightSkyBlue2 attribute), 13
Gray93 (dk.html.theme.palette.LightSkyBlue2 attribute), 13

I

invdict (class in dk.collections.invdict), 9
items() (dk.collections.pset.pset method), 10

K

key (dk.collections.pset.keyval attribute), 9
keys() (dk.collections.pset.pset method), 10
keys() (dk.collections.sdict.sdict method), 11
keyval (class in dk.collections.pset), 9

L

LightSkyBlue2 (dk.html.theme.palette.LightSkyBlue2 attribute), 13

M

mkpalette() (in module dk.html.theme), 13
mmap (class in dk.collections.mmap), 9

N

NOK() (in module dk.collections.xmlrec), 11

O

oset (class in dk.collections.OrderedSet), 9

P

palette (class in dk.html.theme), 13
palette.LightSkyBlue2 (class in dk.html.theme), 13
palette.xForrestGreen (class in dk.html.theme), 13

`pprint()` (dk.collections.pset.pset method), 10
`pset` (class in dk.collections.pset), 9

R

`record` (class in dk.collections.pset), 10
`remove()` (dk.collections.pset.pset method), 10
`rollback()` (dk.collections.pset.record method), 10

S

`sdict` (class in dk.collections.sdict), 11
`strvals()` (dk.collections.pset.record method), 10

T

`test_pset()` (in module dk.collections.pset), 10
`trans()` (dk.collections.pset.record method), 10

U

`u()` (in module dk.text), 15
`u8()` (in module dk.text), 15
`unicode_repr()` (in module dk.text), 15
`utf8()` (in module dk.text), 15

V

`val` (dk.collections.pset.keyval attribute), 9
`values()` (dk.collections.pset.pset method), 10
`values()` (dk.collections.sdict.sdict method), 11

X

`xForrestGreen` (dk.html.theme.palette.xForrestGreen attribute), 13
`xLemonChiffon4` (dk.html.theme.palette.LightSkyBlue2 attribute), 13
`xLightSlateGray` (dk.html.theme.palette.LightSkyBlue2 attribute), 13
`xLimeGreen` (dk.html.theme.palette.xForrestGreen attribute), 13
`xmlrec` (class in dk.collections.xmlrec), 11
`xmlrepr()` (in module dk.collections.pset), 11
`xNavajoWhite2` (dk.html.theme.palette.LightSkyBlue2 attribute), 13
`xOliveDrab4` (dk.html.theme.palette.xForrestGreen attribute), 13
`xYellowGreen` (dk.html.theme.palette.xForrestGreen attribute), 13