Python Tree Data

Release 1.0.4

c0fec0de

Mar 13, 2017

Contents

1 Installation 3

2 API 5

3 Export to DOT 7

4 Getting started 9

Simple, lightweight and extensible Tree data structure.

CHAPTER 1

Installation

To install the *anytree* module run:

```
pip install anytree
```

If you do not have write-permissions to the python installation, try:

```
pip install anytree --user
```

CHAPTER 2

API

CHAPTER 3

Export to DOT

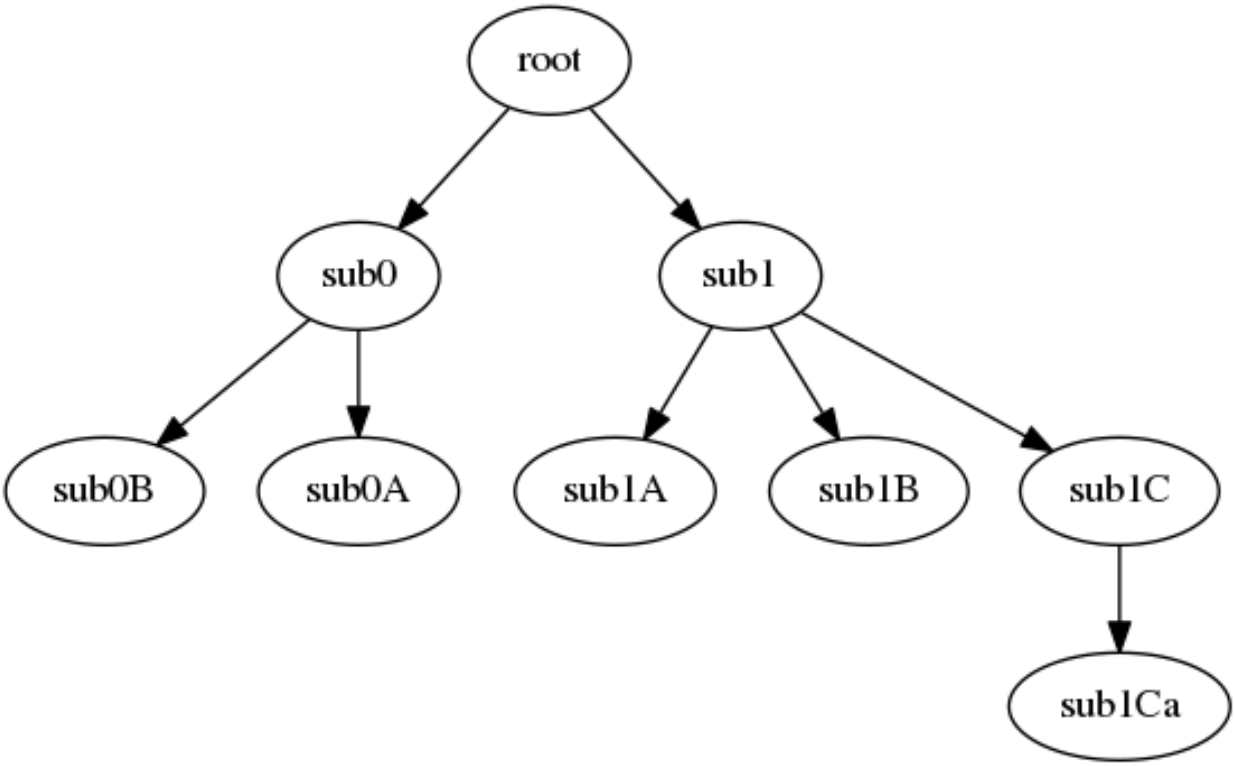
Any `anytree` graph can be converted to a graphviz graph.

This tree:

```
>>> from anytree import Node
>>> root = Node("root")
>>> s0 = Node("sub0", parent=root)
>>> s0b = Node("sub0B", parent=s0)
>>> s0a = Node("sub0A", parent=s0)
>>> s1 = Node("sub1", parent=root)
>>> s1a = Node("sub1A", parent=s1)
>>> s1b = Node("sub1B", parent=s1)
>>> s1c = Node("sub1C", parent=s1)
>>> s1ca = Node("sub1Ca", parent=s1c)
```

Can be rendered to a tree by `RenderTreeGraph`:

```
>>> from anytree.dotexport import RenderTreeGraph
>>> RenderTreeGraph(root).to_picture("tree.png")
```


root
sub0
sub1
sub0B
sub0A
sub1A
sub1B
sub1C
sub1Ca

CHAPTER 4

Getting started

Usage is simple.

Construction

```
>>> from anytree import Node, RenderTree
>>> udo = Node("Udo")
>>> marc = Node("Marc", parent=udo)
>>> lian = Node("Lian", parent=marc)
>>> dan = Node("Dan", parent=udo)
>>> jet = Node("Jet", parent=dan)
>>> jan = Node("Jan", parent=dan)
>>> joe = Node("Joe", parent=dan)
```

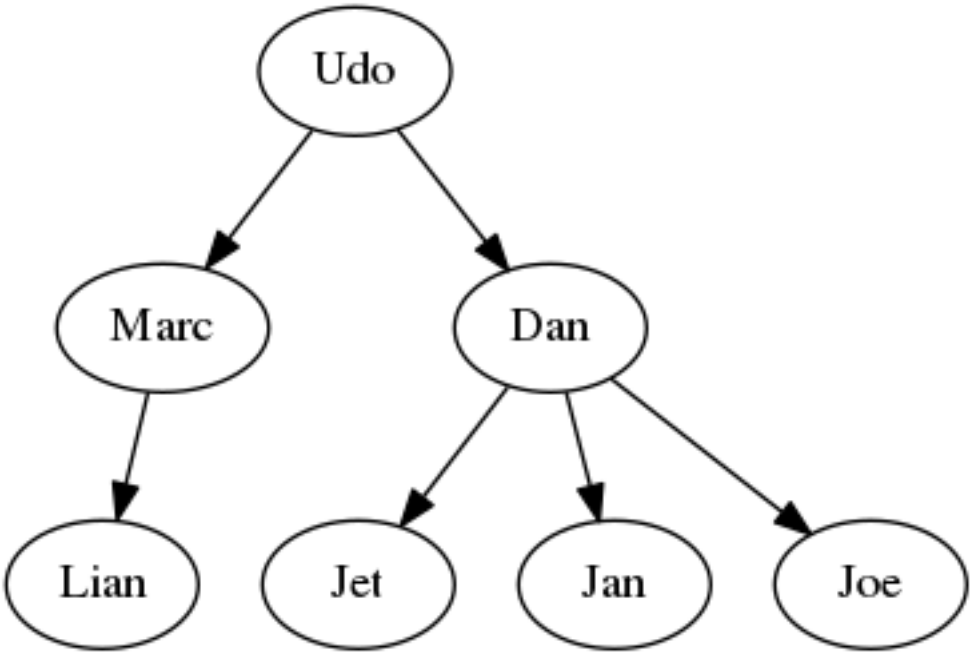
Node

```
>>> print(udo)
Node('Udo')
>>> print(joe)
Node('Udo/Dan/Joe')
```

Tree

```
>>> for pre, fill, node in RenderTree(udo):
...     print("%s%s" % (pre, node.name))
Udo
- Marc
|   - Lian
- Dan
    - Jet
    - Jan
    - Joe
```

```
>>> from anytree.dotexport import RenderTreeGraph
>>> # graphviz needs to be installed for the next line!
>>> RenderTreeGraph(root).to_picture("udo.png")
```



Manipulation

A second tree:

```
>>> mary = Node("Mary")
>>> urs = Node("Urs", parent=mary)
>>> chris = Node("Chris", parent=mary)
>>> marta = Node("Marta", parent=mary)
>>> print(RenderTree(mary))
Node('Mary')
- Node('Mary/Urs')
- Node('Mary/Chris')
- Node('Mary/Marta')
```

Append:

```
>>> udo.parent = mary
>>> print(RenderTree(mary))
Node('Mary')
- Node('Mary/Urs')
- Node('Mary/Chris')
- Node('Mary/Marta')
- Node('Mary/Udo')
    - Node('Mary/Udo/Marc')
    |   - Node('Mary/Udo/Marc/Lian')
    - Node('Mary/Udo/Dan')
        - Node('Mary/Udo/Dan/Jet')
        - Node('Mary/Udo/Dan/Jan')
        - Node('Mary/Udo/Dan/Joe')
```

Subtree rendering:

```
>>> print(RenderTree(marc))
Node('Mary/Udo/Marc')
- Node('Mary/Udo/Marc/Lian')
```

Cut:

```
>>> dan.parent = None
>>> print(RenderTree(dan))
Node('Dan')
- Node('Dan/Jet')
- Node('Dan/Jan')
- Node('Dan/Joe')
```