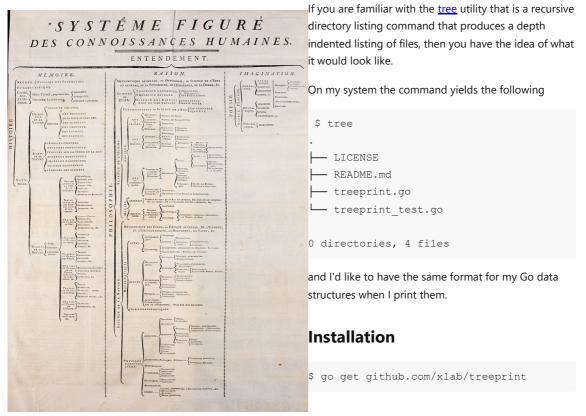
reference coverage 68.6%

treeprint

Package treeprint provides a simple ASCII tree composing tool.



Concept of work

The general idea is that you initialise a new tree with <code>treeprint.New()</code> and then add nodes and branches into it. Use <code>AddNode()</code> when you want add a node on the same level as the target or use <code>AddBranch()</code> when you want to go a level deeper. So <code>tree.AddBranch().AddNode().AddNode()</code> would create a new level with two distinct nodes on it. So <code>tree.AddNode().AddNode()</code> is a flat thing and <code>tree.AddBranch().AddBranch().AddBranch()</code> is a high thing. Use <code>String()</code> or <code>Bytes()</code> on a branch to render a subtree, or use it on the root to print the whole tree.

The utility will yield Unicode-friendly trees. The output is predictable and there is no platform-dependent exceptions, so if you have issues with displaying the tree in the console, all platform-related transformations can be done after the tree has been rendered: <u>an example</u> for Asian locales.

Use cases

When you want to render a complex data structure:

```
func main() {
   tree := treeprint.New()

// create a new branch in the root
```

```
one := tree.AddBranch("one")

// add some nodes
one.AddNode("subnode1").AddNode("subnode2")

// create a new sub-branch
one.AddBranch("two").
    AddNode("subnode1").AddNode("subnode2"). // add some nodes
    AddBranch("three"). // add a new sub-branch
    AddNode("subnode1").AddNode("subnode2") // add some nodes too

// add one more node that should surround the inner branch
one.AddNode("subnode3")

// add a new node to the root
tree.AddNode("outernode")

fmt.Println(tree.String())
}
```

Will give you:

```
- one
- subnode1
- subnode2
- two
- subnode1
- subnode2
- subnode2
- subnode2
- subnode1
- subnode1
- subnode3
- outernode
```

Another case, when you have to make a tree where any leaf may have some meta-data (as tree is capable of it):

```
func main {
    tree := treeprint.New()

    tree.AddNode("Dockerfile")
    tree.AddNode("Makefile")
    tree.AddNode("aws.sh")
    tree.AddMetaBranch(" 204", "bin").
        AddNode("dbmaker").AddNode("someserver").AddNode("testtool")
    tree.AddMetaBranch(" 374", "deploy").
        AddNode("Makefile").AddNode("bootstrap.sh")
    tree.AddMetaNode("122K", "testtool.a")

fmt.Println(tree.String())
}
```

Output:

```
Dockerfile

Makefile

aws.sh

[204] bin

button

button

contact the same server

contact the same server

button

contact the same
```

Yay! So it works.

License

MIT