Test of DocOnce support for LaTeX code block environments

HPL

Feb 1, 2015

1 Demo 1

Suppose we have some data in a file:

```
#     A          B          C          D          E
   -0.5253    -0.9315    -0.3427    -0.1613    -0.8472
   -0.9740    -0.2558    -0.5622    -0.7635    -0.0914
    0.9216     0.7702    -0.4818     0.2155     0.2967
    0.6217     0.6100    -0.3846    -0.7904     0.9166
    0.1006    -0.3162     0.3841     0.5241    -0.6530
    0.6207    -0.9299     0.4837     0.5755    -0.6024
    0.4278    -0.0014     0.8184     0.9382    -0.1449
   -0.9178     0.2612    -0.7532     0.3901    -0.0075
    0.2134     0.6217     0.0545     0.6980    -0.2172
   -0.9529     0.8989    -0.1969    -0.3079     0.0389
    0.8311     0.0145     0.4215    -0.5451    -0.3415
```

This program (which breaks a page) reads the data and performs analysis:

```
#!/usr/bin/env python

import numpy as np

def readfile(filename):
    """Read tabular data from file and return as numpy array."""
    f = open(filename, 'r')
    data = []  # list of rows in table
    for line in f:
        if line.startswith('#'):
            continue   # drop comment lines
        numbers = [float(w) for w in line.split()]
        data.append(numbers)
    return np.array(data)

def analyze(data):
    """Return statistical measures of an array data."""
    return np.mean(data), \
           np.std(data), \
           np.corrcoef(data)
```

```
if __name__ == '__main__':
    data = readfile('mydat.txt')
    # Treat each column as a variable
    m, s, c = analyze(data.transpose())
    print """
mean=%f
st.dev=%f
correlation matrix:
%s
""" % (m, s, c)
```

The output becomes

```
Terminal> python fileread.py

mean=-0.006005
st.dev=0.583542
correlation matrix:
[[ 1.          0.0509676   0.52406366  0.20964645  0.1574504 ]
 [ 0.0509676   1.         -0.30920845 -0.12129049  0.7611538 ]
 [ 0.52406366 -0.30920845  1.          0.49355806 -0.42263817]
 [ 0.20964645 -0.12129049  0.49355806  1.         -0.38286589]
 [ 0.1574504   0.7611538  -0.42263817 -0.38286589  1.        ]]
```

2 Demo 2

The file `mypro.py` contains the program

```
#!/usr/bin/env python

def run(program):
    import os
    failure = os.system(os.path.join(os.curdir, program))
    if failure:
        raise OSError('Could not run Fortran program')

run('hw')
```

The program `hw` is defined in `hw.f`:

```
      program hw
      call print_msg()
      end
```

This program must be linked with the definition of `print_msg` in a file `routines.f`:

```
      subroutine print_msg()
      write(*,*) 'Hello, World!'
      end
```

The Fortran files can be compiled by

```
Terminal> gfortran -o hw hw.f routines.f
```

Finally, we can run our `mypro.py` program:

```
Terminal> python mypro.py
Hello, World!
```