

Branch: master ▾

Find file

Copy path

cs-35l / assignment3 / shuf.py



prithvikannan fixed formatting

b96812a on Oct 21

1 contributor

Raw Blame History



124 lines (93 sloc) 3.5 KB

● Code navigation is available for this repository but data for this commit does not exist.

[Learn more](#) or [give us feedback](#)

```
1  #!/usr/bin/python
2
3  # Prithvi Kannan
4  # UID: 405110096
5
6  import random, sys, argparse, string
7
8  from optparse import OptionParser
9
10 class shuf:
11
12     def __init__(self, inputs, num_lines, isRepeat):
13         self.inputs = inputs
14         self.isRepeat = isRepeat
15         self.num_lines = num_lines
16
17         random.shuffle(self.inputs)
18
19     def shuffleline(self):
20         if not len(self.inputs):
21             return
22
23         if self.isRepeat:
24             while self.num_lines > 0:
25                 sys.stdout.write(random.choice(self.inputs))
26                 self.num_lines = self.num_lines - 1
27         else:
28             for i in range(0, self.num_lines):
29                 sys.stdout.write(self.inputs[i])
30
31
32 def main():
33     version_msg = "%prog 2.0"
34     usage_msg = """%prog [OPTION]... FILE
35 or: %prog -i LO-HI [OPTION]...
36 Write a random permutation of the input lines to standard output."""
37
38     parser = OptionParser(version=version_msg, usage=usage_msg)
39
40     parser.add_option("-n", "--head-count",
41                     action="store", dest="num_lines", default=sys.maxsize,
42                     help="output at most count lines")
43
44     parser.add_option("-i", "--input-range",
45                     action="store", dest="inputRange", default="",
46                     help="treat each number LO through HI as an input line")
47
48     parser.add_option("-r", "--repeat",
49                     action="store_true", dest="isRepeat", default=False,
50                     help="output lines can be repeated")
```

```
51
52 options, args = parser.parse_args(sys.argv[1:])
53
54 try:
55     num_lines = int(options.num_lines)
56 except:
57     parser.error("invalid count: {}".format(options.num_lines))
58 if num_lines < 0:
59     parser.error("negative count {}".format(num_lines))
60
61 isRepeat = options.isRepeat
62
63 inputRange = options.inputRange
64 if len(inputRange) > 0:
65     if len(args) != 0:
66         parser.error("extra operand '{}'.format(args[0]))
67
68     try:
69         isDash = inputRange.index('-')
70     except ValueError as e:
71         parser.error("invalid input range: '{}'.format(options.inputRange))
72     if isDash == 0:
73         parser.error("invalid input range: '{}'.format(options.inputRange))
74
75     first, last = inputRange.split("-")
76
77     try:
78         firstNum = int(first)
79     except ValueError as e:
80         parser.error("invalid input range: '{}'.format(options.inputRange))
81
82     try:
83         lastNum = int(last)
84     except ValueError as e:
85         parser.error("invalid input range: '{}'.format(options.inputRange))
86
87     if first > last:
88         parser.error("invalid input range: '{}'.format(options.inputRange))
89
90     inputs = list(range(firstNum, lastNum+1))
91     for i in range(len(inputs)):
92         inputs[i] = str(inputs[i]) + "\n"
93
94 else:
95     if len(args) == 0:
96         if (len(args) == 1 and args[0] == "-"):
97             inputs = sys.stdin.readlines()
98
99     elif len(args) == 1:
100         try:
101             f = open(args[0], 'r')
102             inputs = f.readlines()
103             f.close()
104         except IOError as e:
105             errno, strerror = e.args
106             parser.error("I/O error({}): {}".format(errno, strerror))
107
108     else:
109         parser.error("extra operand '{}'.format(args[1]))
110
111
112
113
114
115
116
```

```
117     if num_lines > len(inputs) and not isRepeat:
118         num_lines = len(inputs)
119
120     shuffler = shuf(inputs, num_lines, isRepeat)
121     shuffler.shuffleline()
122
123 if __name__ == "__main__":
124     main()
```