File - D:\googledrive\Year4\006BigProject\Python\ast1\read2.py

```
1  import ast
2  ##common = { 'col_offset', 'lineno'}
3  # Python Language Dictionary = PLD refers to Python
   version - 3.5.2
4  from PLD2 import PLD, PLDLIST
5  from textfile import text as text
6  tree = ast.parse(text)
7
8  def add_to_list(base, sep, bits):
9      return [sep.join([base, bit]) for bit in bits]
10
11
12 def tie_in(endlist):
13     for row in sorted(endlist):
14         rowl = len(str(row[1]))
15         print(' '*(60-rowl),row[1],' ',eval(row[1]))
16
17
18 def get_children_of(item):
19     (count, parent_full_name) = item
20     child_list = [(count,parent_full_name)]
21     print('starting analysis of ',parent_full_name)
22     parent_is_list = eval('isinstance({0}, list)'.format(
   parent_full_name))
23     if parent_is_list:
24         child_list_len = len(eval(parent_full_name))
25         list_of_children = list(map(lambda x : '[{0}]'.
   format(x),range(child_list_len)))
26         child_list.extend(add_to_list(parent_full_name, ''
   , list_of_children))
27         print('children from list ', child_list)
28     else:
29         parent = eval('type({0})'.format(parent_full_name)
   )
30         parent_value = str(parent)
31         parent_value2 = (parent_value)[13:-2] if
   parent_value.startswith("<class '_ast.") else parent_value
   [8:-2]
32         if parent_value2 in PLD:
33             sub_list = PLD[parent_value2]
34             for item in sub_list:
35                 child_list.append('.'.join([
   parent_full_name, item]))
36         else:
37             print('not found in PLD ',parent_value2)
```

```
38          print('returning child_list for {0} = {1}'.format(
   parent_full_name, child_list))
39      return child_list
40
41
42  def main():
43      count = 0
44      stacker = [(count, 'tree')]
45      endlist=[]
46      parent_array ={   }
47      child_array ={ }
48      while len(stacker) > 0:
49          parent_sent = stacker[0]
50          current_nodes_children = get_children_of(
   parent_sent)
51          for child in current_nodes_children:
52              current_count = count
53              if child == parent_sent:
54                  print(child, ' has returned')
55                  endlist.append(parent_sent)
56                  current_parent = parent_sent[0]
57                  stacker.remove(child)
58              else:
59                  print(child,' is new for stacker')
60                  count += 1
61                  stacker.append((count, child))
62                  parent_array[count] = current_parent
63                  child_array.setdefault(current_parent, [])
   .append(count)
64          print('stack currently holds : ',stacker)
65          print('endlist currently holds : ', endlist)
66          print('count is at ', count)
67      print('end result = ', endlist)
68      print(parent_array)
69      print(child_array)
70      goget = tie_in(endlist)
71
72  if __name__=='__main__':
73      main()
74
```