

Apr 03, 12 9:24

FileVisit.py

Page 1/1

```

'''
Created on Mar 29, 2011

@author: ola
'''
5  import os

10 def bigfiles(dirname,min_size):
    large = []
    #print dirname
    for sub in os.listdir(dirname):
        path = os.path.join(dirname,sub)
15     if os.path.isdir(path):
        large.extend(bigfiles(path,min_size))
    else:
        size = os.path.getsize(path)
        if size > min_size:
20         large.append((path,size))
    return large

25 def ftree(indent,dirname,depth):
    sub_depth = indent.count("+")
    if sub_depth >= depth:
        return
    print indent,dirname
    contents = os.listdir(dirname)
30     for sub in contents:
        path = os.path.join(dirname,sub)
        if os.path.isdir(path):
            ftree(indent+"----",path,depth)
        else:
35         print indent,path

if __name__ == "__main__":
    #ftree("+","/Users/ola/Desktop",3)
    bigs = bigfiles("/Users/ola/Desktop/courses/6python",10000)
40     for f in bigs:
        print f

```

Apr 03, 12 9:24

Koch.py

Page 1/1

```

'''
Created on Apr 4, 2011

@author: ola
'''
5  import turtle

def draw(iters):
    flake = "FRFRF"
10     for steps in range(iters):
        flake = flake.replace("F", "FLFRFLF")

    for move in flake:
        if move == "F":
            turtle.forward(100.0/3**(iters-1))
15         elif move == "L":
            turtle.left(60)
        elif move == "R":
            turtle.right(120)

20 draw(2)
x = raw_input()

```

Apr 03, 12 9:24

SimpleGrammar.py

Page 1/1

```

'''
Created on Apr 4, 2011

@author: ola
5 '''

import random

def create_content():
10     """
    return a dictionary used for generating random sentences
    """
    adjectives = [ "<color>", "slimy", "wonderful", "beautiful", "obese", "teeny", "<adj>", "<adj>" ]
    colors = [ "green", "red", "yellow", "blue", "maroon" ]
15     rules = { "<color>": colors, "<adj>": adjectives }

    return rules

20 def expand(sentence, rules):
    """
    expand sentence using rules as source of meta-words
    """
    sent = ""
25     for w in sentence.split():
        if w.startswith("<"):
            chosen = random.choice(rules[w])
            sent += expand(chosen, rules) + " "
        else:
30             sent += w + " "
    return sent.strip()

def create():
    rules = create_content()
35     print expand("the <adj> dog ate the <adj> bone", rules)
    print expand("the <color> house was a <adj> edifice", rules)

if __name__ == "__main__":
    create()

```