# examples of using lists in python
x = [0, 2, 4, 6, 8]
y = [1, 3, 5, 7, 9]

# index starting at 0; getting slices from a list
print x[0], x[1], x[2], x[3], x[4], x[-1]
print y[0:5], y[2:3], y[0:5:2], y[5:0:-1]

# the length function and the concatenation operator
print len(x)
print x + y

# what does the following do?
head, tail = x[0], x[1:]
print head, tail

# a function to reverse a list
def rev(x):
    if len(x) == 0:
        return x
    else:
        head, tail = x[0], x[1:]
        return rev(tail) + [head]
print "The reverse of", x, "is", rev(x), "."

# examples of using tuples in python; in general, we prefer
# the elements in a tuple are of different types, while
# the elements in a list are of the same type
a = ("a", 3, 0.2)
b = ()
c = (True,)
print a + b + c, (a, b, c)
u, v, w = a
print u, v, w

# the following are binary search trees. why?
left = (((()), "clueless", ()), "complexify", (((), "jazzed", ())
right = (((), "phat", ()), "poset", (((), "sheafify", ()))
tree = (left, "macchiato", right)

print tree

# a function to search a binary search tree
def search (t, x):
    if len(x) == 0:
        return False
    else:
        left, root, right = x
        if t == root:
            return True
        elif t < root:
            return search (t, left)
        else:
            return search (t, right)
print search ("phat", tree)
print search ("orange", tree)  
print search ("phat", ())  

# examples of using sets in python;  
basket = ["apple", "orange", "apple", "pear", "orange", "banana"]  
fruit = set(basket)  
taiwanfruit = {"banana", "pineapple", "mango"}  

print fruit, taiwanfruit  
print fruit | taiwanfruit  
print fruit & taiwanfruit  
print fruit - taiwanfruit  
print "mango" in taiwanfruit  
print "mango" in fruit  
print taiwanfruit == {"pineapple", "mango", "banana"}  

# examples of list and set comprehensions  
for e in x:  
    print e*e  
print [ e*e for e in x ]  

for e in fruit:  
    print (e, e)  
print { (e, e) for e in fruit }  

print [ e for e in x if e > 5]  
print { e for e in fruit if e in taiwanfruit }  

# nested comprehensions  
print [ (u, v) for u in x for v in y if u < v]  
print { (u, v) for u in fruit for v in taiwanfruit }  

# more nested comprehensions  
this = { (u, v) for u in basket for v in x }  
that = [ (u, v) for u in fruit for v in x ]  
print this  
print that  
print this == set(that)