Lists

Python for Ecologists

Jon Flaishans, Tom Purucker, Tao Hong, Marcia Snyder

Ecological Society of America Workshop
Minneapolis, MN

purucker.tom@gmail.com

August 3, 2013

Lists, Tuples, and Dictionaries

Type	Create Empty	Mutable?	Order
List	my_list = []	Mutable	Yes
Tuple	my_tuple = ()	Immutable	No
Dictionary	my_dictionary = {}	Mutable	No

- Mutable here means you can append, change, subtract, etc.

Lists

```
species_names = []

species_names.append("Geospiza␣fuliginosa") # small

species_names.extend(
  [’Geospiza␣fortis’,’Geospiza␣magnirostris’]
  ) # medium, large

sorted(species_names) #produces copy of list, sorted

species_names.sort() #sorts existing list (mutable)
```

Lists

- Index, Value pairs
- Lists can be nested
- Slices, Length, and Index
- Tuples can use any immutable type as an index (not just integers).....more on this later

Some list functions

- Lists can mix types

```
some_list = [23, 23., ’Frog’, None, True]
#None and Boolean types
```

- Lists have similar methods as strings

```
some_list[0]
len(some_list)
[1,2] + [3,4]
```

- We can easily loop over the list elements

```
for thing in some_list:
  print thing
```

- And check to see if elements are in the list

```
’Frog’ in some_list
’Bird’ in some_list
```

Deleting list elements

- Getting rid of list elements:

```
some_list = ["list␣item1","list␣item2","list␣item3"]

some_list.pop(0)
  # removes last item from list & returns its value

some_list.insert(0,"list␣item1") # re—add the item

del some_list[2] # must be int (index value)

del some_list

some_list.remove("list␣item1")
  # removes only 1st occurance in list
```

Tuples

- Tuples are immutable objects that cannot be altered
- Use parentheses () instead of square brackets []

```
some_tuple = [
    ("Frog1",23.2),
    ("Frog1",20.8),
    ]
```

- Tuples and lists can both be sliced

```
some_tuple[0:2]
some_list[0:2]
```

- ....and sorted

```
sorted(some_tuple, key=lambda grams:grams[1])
```

Exercise 3- Run the script exer03_lists.py

```
class TestLists(unittest.TestCase):
    def test_lists(self):
        """
        A basic introduction to lists
        """
        # Create the variable ‘‘bird_list‘‘ and assign to an empty list
        # **************************************************

        self.assertEquals(bird_list, [])

        # Append ’American redstart’ and ’Arctic tern’ to ‘‘bird_list‘‘
        # **************************************************

        self.assertEquals(bird_list, [’American␣redstart’,’Arctic␣tern’ ])

        # Sort ‘‘bird_list‘‘
        # **************************************************

        self.assertEquals(bird_list,  [’Arctic␣tern’, ’American␣redstart’])

        # ‘‘extend‘‘ the list ‘‘bird_list‘‘ with [’Northern parula’, ’george’]
        # **************************************************

        self.assertEquals(bird_list, [’Arctic␣tern’, ’American␣redstart’, ...])

        # create a variable ‘‘warbler_id‘‘ with the index of ’Hooded warbler’ in
        # ‘‘bird_list‘‘ using list methods.
        # **************************************************

        self.assertEquals(warbler_id, 3)
```