

# Ling 555 — Programming for Linguists

Python - List and Tuples

Robert Albert Felty

Speech Research Laboratory  
Indiana University

Sep. 17, 2008

# Indexing

## List basics

indexing

slicing

lol

mutable

operations

len

## List methods

## tuples

## Zero comes first

Python starts all indices with 0.

## Practice

- 1 How do you print just the first letter of “hello”?  

```
print('hello'[0])
```
- 2 How do you print just the last letter of “hello”?  

```
print('hello'[-1])
```
- 3 Create a list *foo*, with the following values: 25, 68, “bar”, 89.45, 789, “spam”, 0, “last item”  

```
foo = [25, 68, 'bar', 89.45, 789, 'spam', 0, 'last item']
```

# Slicing

## List basics

indexing

slicing

lol

mutable

operations

len

## List methods

tuples

## Definition

A list slice takes part of a list. A slice `list[i:j]` starts at the  $i_{th}$  index, and goes up to (**but does not include**) the  $j_{th}$  index.

# Slicing practice

## List basics

indexing

slicing

lol

mutable

operations

len

## List methods

## tuples

## Practice

HINT: remember that indices start at 0

- 1 Print the 1st to 3rd item in the list *foo*  
`print(foo[:3])`
- 2 Print the 3rd to last item in the list *foo*  
`print(foo[2:])`
- 3 Print the 2nd to the 2nd to last item in the list *foo*  
`print(foo[1:-1])`
- 4 Copy the entire *foo* list to a new list named *bar*  
`bar=foo[:]`

# Lists can contain other lists

## List basics

indexing

slicing

lol

mutable

operations

len

## List methods

## tuples

### practice

- 1 Create a new list *lol* which contains the lists *foo* and *bar*  

```
lol=list()  
lol.append(foo)  
lol.append(bar)
```
- 2 Print the first item of *lol*  

```
print(lol[0])
```
- 3 Print the second item of the first list in *lol*  

```
print(lol[0][1])
```
- 4 Print the third item to the last item of the second list in *lol*  

```
print(lol[1][2:])
```

# Lists are mutable

## List basics

indexing

slicing

lol

mutable

operations

len

## List methods

## tuples

### Definition

If you perform an operation on a list, it changes the list. In contrast, tuples and strings are immutable.

### Example

Try the following:

```
newfoo = foo
```

```
newfoo[0] = 'new value'
```

## List basics

indexing

slicing

lol

mutable

operations

len

## List methods

## tuples

## Practice

- 1 Change the first item in the *foo* list to 12  
`foo[0]=12`
- 2 Now multiply the first item in the *foo* list by 2  
`foo[0]*2`
- 3 Test whether “ham” is in the list *foo*  
`'ham' in foo`

# Len, min, and max

## List basics

indexing

slicing

lol

mutable

operations

len

## List methods

## tuples

### practice

- 1 How many items does *foo* contain?  
`len(foo)`
- 2 What does `min(foo)` return?
- 3 What does `max(foo)` return?  
Is that what you expected?



# Queues and stacks

List basics

List methods

pop

sort

tuples

## FIFO and LIFO

LIFO Last in, first out (stack)

FIFO First in, first out (queue)

# Popping, appending, etc.

List basics

List methods

pop

sort

tuples

## practice

- 1 Append the value 24 to the list *foo*  
`foo.append(24)`
- 2 Insert the value “twenty” to the list *foo* as the 4th item  
`foo.insert(3, 'twenty')`
- 3 Find the index of “spam” in the list *foo*  
`foo.index('spam')`
- 4 remove the last item from *foo*, and store it as a new variable  
`lastfoo=foo.pop()`

# More list methods

List basics

List methods

pop

sort

tuples

## Practice

- 1 Append the following values to *foo*: 89, 23.4, 1  

```
foo.extend([89, 23.4, 1])  
foo.extend((89, 23.4, 1))
```
- 2 Create a new list *fooSorted* with the same contents as *foo*, but sorted  

```
fooSorted=foo.sort()
```

# Tuples

List basics

List methods

tuples

## Definition

Tuples are very similar to lists but are **immutable**

- Indexing and slicing work with tuples just as with lists.
- Tuples do not support methods such as sorting.
- You can create them with parentheses:  
`mytuple=(10,50,'foo')`