

# An Introduction to STEM Programming with Python 3 – Chapter 4 Lists and Tuples

By  
James M. Reneau, Ph.D.  
[jreneau@shawnee.edu](mailto:jreneau@shawnee.edu)

Version 2020-03-03a



Shawnee State  
University

© 2020 James M. Reneau  
This work is licensed under a Creative  
Commons Attribution-NonCommercial-  
ShareAlike 4.0 International License.

# In this video we will cover:

- ✓ Defining lists and tuples
- ✓ Creating a tuple with values
- ✓ Creating a list with braces and the list() object
- ✓ Accessing values and slices from a list and tuple using the indexing operator
- ✓ Changing values in a list
- ✓ Deleting items in a list

# What is a Tuple?

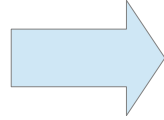
- Tuples and lists are built-in data types that allow you to store and use a group of values, as if it were a single value.
- Tuple is a read-only collection of values.
  - Once you create a tuple you may not change or add values
  - Defined inside parenthesis.
  - `t = (1, 2, 3, 4, 5)`
    - creates a variable `t` with the numbers 1-5 in a tuple
  - `x = ("something",)`
    - You need an extra comma if defining a tuple of one value so that python doesn't think the parentheses are part of an expression.

# What is a List?

- Built in data type that allows you to create a group of values that you may modify (add, delete, change, sort...)
- Created using square braces [ ]
- `animals = ['cat', 'dog', 'snake']` and `m = [1, 2, 3, 4]`
- `x = []` or `z = list()` are empty lists
- Comma separated values
- Do not need to be all the same type
- Lists may contain any types of items, even other lists

# Creating a List

```
1 list1 = []
2 list2 = list()
3 list3 = [1, 2, 3, 4, 5, 6]
4 pets = ["cat", "dog", "fish",
          "snake"]
5
6 print(list1)
7 print(list2)
8 print(list3)
9 print(pets)
10
11 listoflists = [list1, list2,
                 list3, pets]
12 print(listoflists)
```



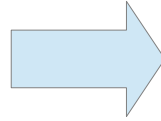
```
[]
>[]
[1, 2, 3, 4, 5, 6]
['cat', 'dog', 'fish',
'snake']
[[], [], [1, 2, 3, 4, 5, 6],
['cat', 'dog', 'fish',
'snake']]
```

# Indexing Operator to get a Single Value

- The indexing operator [ ] with lists and tuples works like it does with strings.
- Lists and tuples are zero indexed (the first item is 0)
- Remember you get the value in the list at that location.

# Indexing Operator to get a Single Value

```
1 list1 = []
2 list2 = list()
3 list3 = [1,2,3,4,5,6]
4 pets =
  ["cat","dog","fish","snake"]
5 listoflists = [list1, list2,
  list3, pets]
6
7 print(pets[1])
8 print(listoflists[2])
9
10 print(list3[-2])
```



```
dog
[1, 2, 3, 4, 5, 6]
5
```

# Slicing a List

- Using the indexing operator to get a sub-list from a list or tuple is called slicing.
- It works like getting a sub-string from a string but always returns a list or tuple.
- `list[start : end]`
- Extracts the items from start to one less than end.
- Start may be omitted
  - Start at character 0
- End may be omitted
  - Extract to the end



# Slicing a List

```
1 people = ['amy', 'bob',  
            'charlie', 'danielle',  
            'evan', 'franky', 'george']
```

```
2
```

```
3 print(people)
```

```
4 print()
```

```
5
```

```
6 print(people[1:3])
```

```
7 print()
```

```
8
```

```
9 print(people[:4])
```

```
10 print()
```

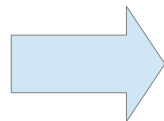
```
11
```

```
12 print(people[5:])
```

```
13 print()
```

```
14
```

```
15 print(people[-2:])
```



```
['amy', 'bob', 'charlie',  
'danielle', 'evan',  
'franky', 'george']
```

```
['bob', 'charlie']
```

```
['amy', 'bob', 'charlie',  
'danielle']
```

```
['franky', 'george']
```

```
['franky', 'george']
```

# Updating a List

- With lists, you may use the indexing operator in an assignment statement to change an item or group of items in a list.
- You can't do this with tuples (read only)

# Updating a List

```
1 people = ['amy', 'bob',  
            'charlie', 'danielle',  
            'evan', 'franky', 'george']
```

```
2
```

```
3 print(people)  
4 print()
```

```
5
```

```
6 people[1] = "betty"
```

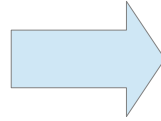
```
7 print(people)  
8 print()
```

```
9
```

```
10
```

```
11 people[3:5] = ["danny",  
                 "sue", "edna"]
```

```
12 print(people)
```



```
['amy', 'bob', 'charlie',  
 'danielle', 'evan',  
 'franky', 'george']
```

```
['amy', 'betty', 'charlie',  
 'danielle', 'evan',  
 'franky', 'george']
```

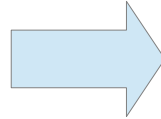
```
['amy', 'betty', 'charlie',  
 'danny', 'sue', 'edna',  
 'franky', 'george']
```

# Delete an Item

- In addition to changing items or slices of items, you can simply delete an item.
- `del variable[location]`

# Delete an Item from a List

```
1 people = ['amy', 'bob',  
            'charlie', 'danielle',  
            'evan', 'franky', 'george']  
2  
3 print(people)  
4 print()  
5  
6 del people[1]  
7 print(people)
```



```
['amy', 'bob', 'charlie',  
 'danielle', 'evan',  
 'franky', 'george']  
  
['amy', 'charlie',  
 'danielle', 'evan',  
 'franky', 'george']
```

# Thank you

- This presentation is Copyright 2020 by James M. Reneau PhD.
- Contact me at [jreneau@shawnee.edu](mailto:jreneau@shawnee.edu)
- This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

