

Lists, Indexing, and Slicing

Python Basics 4

Dr. Ryan Krauss

Grand Valley State University

Powerful Data Types

- ▶ lists and dictionaries are both powerful data types

Powerful Data Types

- ▶ lists and dictionaries are both powerful data types
- ▶ learning to use them well can greatly enhance what you can do with Python

Creating a List

- ▶ a list is simply a collection of items inside square brackets:

Creating a List

- ▶ a list is simply a collection of items inside square brackets:

Creating a List

- ▶ a list is simply a collection of items inside square brackets:

```
mylist = ['a', 'b', 'c', 'd']
```

- ▶ you can mix the data types of things in the list

Creating a List

- ▶ a list is simply a collection of items inside square brackets:

```
mylist = ['a', 'b', 'c', 'd']
```

- ▶ you can mix the data types of things in the list

Creating a List

- ▶ a list is simply a collection of items inside square brackets:

```
mylist = ['a', 'b', 'c', 'd']
```

- ▶ you can mix the data types of things in the list

```
mixedlist = [1, 2, 'a', 'ryan']
```

- ▶ you can also create a list of integers using the `range` function:

Creating a List

- ▶ a list is simply a collection of items inside square brackets:

```
mylist = ['a', 'b', 'c', 'd']
```

- ▶ you can mix the data types of things in the list

```
mixedlist = [1, 2, 'a', 'ryan']
```

- ▶ you can also create a list of integers using the range function:

Creating a List

- ▶ a list is simply a collection of items inside square brackets:

```
mylist = ['a', 'b', 'c', 'd']
```

- ▶ you can mix the data types of things in the list

```
mixedlist = [1, 2, 'a', 'ryan']
```

- ▶ you can also create a list of integers using the range function:

```
intlist = range(10)
```

Indexing

- ▶ grabbing one element out of a list is done with square brackets and an integer index:

Indexing

- ▶ grabbing one element out of a list is done with square brackets and an integer index:

Indexing

- ▶ grabbing one element out of a list is done with square brackets and an integer index:

```
mylist = ['a', 'b', 'c', 'd']  
item1 = mylist[1]
```

- ▶ what will `item1` be?

Indexing

- ▶ grabbing one element out of a list is done with square brackets and an integer index:

```
mylist = ['a', 'b', 'c', 'd']  
item1 = mylist[1]
```

- ▶ what will `item1` be?
 - ▶ try it out now

Parentheses vs. Square Brackets

- ▶ note that Python uses parentheses for calling functions and square brackets for indexing and slicing of lists:

Parentheses vs. Square Brackets

- ▶ note that Python uses parentheses for calling functions and square brackets for indexing and slicing of lists:

Parentheses vs. Square Brackets

- ▶ note that Python uses parentheses for calling functions and square brackets for indexing and slicing of lists:

```
myfunc(a,b)  
myitem = mylist[2]
```

Negative Indices

- ▶ negative indices also work on lists

Negative Indices

- ▶ negative indices also work on lists
- ▶ `mylist[-1]` refers to the last item in the list

Negative Indices

- ▶ negative indices also work on lists
- ▶ `mylist[-1]` refers to the last item in the list
 - ▶ `mylist[-2]` would refer to the second to the last item and so on

Slicing

- ▶ slicing refers to extracting a sublist from a list using a start and stop index separated by a colon:

Slicing

- ▶ slicing refers to extracting a sublist from a list using a start and stop index separated by a colon:

Slicing

- ▶ slicing refers to extracting a sublist from a list using a start and stop index separated by a colon:

```
mylist = ['a', 'b', 'c', 'd']  
myslice = mylist[0:2]
```

- ▶ note that like `range(2)`, the slice stopping index is one less than the second number

Slicing

- ▶ slicing refers to extracting a sublist from a list using a start and stop index separated by a colon:

```
mylist = ['a', 'b', 'c', 'd']  
myslice = mylist[0:2]
```

- ▶ note that like `range(2)`, the slice stopping index is one less than the second number
 - ▶ try out the code above

Slicing

- ▶ slicing refers to extracting a sublist from a list using a start and stop index separated by a colon:

```
mylist = ['a', 'b', 'c', 'd']  
myslice = mylist[0:2]
```

- ▶ note that like `range(2)`, the slice stopping index is one less than the second number
 - ▶ try out the code above
 - ▶ what is `myslice`?