Section Overview

What You Will Learn

- Lists
- Searching in Lists
- Exception Handling
- Slices
- Ranges
- For Loop
- While Loop

Lists

Lists

- A list is a data type that holds an ordered collection of items.
- The items can be of various data types.
- You can even have lists of lists!

Creating Lists

```
list_name = [item_1, item_2, item_N]

list_name = []

list_name[index]
```

om

```
animals = ['man', 'bear', 'pig']
print(animals[0])
print(animals[1])
print(animals[2])
```

```
man
bear
pig
```

om

```
animals = ['man', 'bear', 'pig']
print(animals[0])
animals[0] = 'cat'
print(animals[0])
```

```
man
cat
```

om

```
animals = ['man', 'bear', 'pig']
print(animals[-1])
print(animals[-2])
print(animals[-3])
```

```
pig
bear
man
```

```python
animals = ['man', 'bear', 'pig']
animals.append('cow')
print(animals[-1])
```

```
cow
```

```
animals = ['man', 'bear', 'pig']
animals.extend(['cow', 'duck'])
print(animals)

more_animals = ['horse', 'dog']
animals.extend(more_animals)
print(animals)
```

```
['man', 'bear', 'pig', 'cow', 'duck']
['man', 'bear', 'pig', 'cow', 'duck', 'horse', 'dog']
```

```
animals = ['man', 'bear', 'pig']
animals.insert(0, 'horse')
print(animals)

animals.insert(2, 'duck')
print(animals)
```

```
['horse', 'man', 'bear', 'pig']
['horse', 'man', 'duck', 'bear', 'pig']
```

om

Slices

Slices

```
list[index1:index2]


list[:index2]


list[index1:]
```

```
animals = ['man', 'bear', 'pig', 'cow', 'duck', 'horse']

some_animals = animals[1:4]
print('Some animals:      {}'.format(some_animals))

first_two = animals[0:2]
print('First two animals: {}'.format(first_two))

first_two_again = animals[:2]
print('First two animals: {}'.format(first_two_again))
```

```
Some animals:       ['bear', 'pig', 'cow']
First two animals: ['man', 'bear']
First two animals: ['man', 'bear']
```

```
animals = ['man', 'bear', 'pig', 'cow', 'duck', 'horse']

last_two = animals[4:6]
print('Last two animals:  {}'.format(last_two))

last_two_again = animals[-2:]
print('Last two animals:  {}'.format(last_two_again))
```

```
Last two animals:  ['duck', 'horse']
Last two animals:  ['duck', 'horse']
```

String Slices

```
part_of_a_horse = 'horse'[1:3]
print(part_of_a_horse)
```

```
or
```

om

Finding an item in a list.

```
animals = ['man', 'bear', 'pig']
bear_index = animals.index('bear')
print(bear_index)
```

```
1
```

om

Exceptions

```
animals = ['man', 'bear', 'pig']
cat_index = animals.index('cat')
print(cat_index)
```

```
Traceback (most recent call last):
  File "exception_example.py", line 2, in <module>
    cat_index = animals.index('cat')
ValueError: 'cat' is not in list
```

Exception Handling

Exception Handling

```
animals = ['man', 'bear', 'pig']
try:
    cat_index = animals.index('cat')
except:
    cat_index = 'No cats found.'
print(cat_index)
```

```
No cats found.
```

Loops

Looping through a list

```
for item_variable in list_name:
    # Code block
```

```
item_variable = list[0]
item_variable = list[1]
item_variable = list[N]
```

```
animals = ['man', 'bear', 'pig']
for animal in animals:
    print(animal.upper())
```

```
MAN
BEAR
PIG
```

om

While Loop

```
while condition:
    # Code block
```

```
animals = ['man', 'bear', 'pig', 'cow', 'duck', 'horse']

index = 0

while index < len(animals):
    print(animals[index])
    index += 1
```

```
man
bear
pig
cow
duck
horse
```

om

Sorting and Ranges

```
animals = ['man', 'bear', 'pig']
sorted_animals = sorted(animals)
print('Animals list:           {}'.format(animals))
print('Sorted animals list:    {}'.format(sorted_animals))
animals.sort()
print('Animals after sort method: {}'.format(animals))
```

```
Animals list:              ['man', 'bear', 'pig']
Sorted animals list:       ['bear', 'man', 'pig']
Animals after sort method: ['bear', 'man', 'pig']
```

om

```python
animals = ['man', 'bear', 'pig']
more_animals = ['cow', 'duck', 'horse']
all_animals = animals + more_animals
print(all_animals)
```

```
['man', 'bear', 'pig', 'cow', 'duck', 'horse']
```

om

```
animals = ['man', 'bear', 'pig']
print(len(animals))
animals.append('cow')
print(len(animals))
```

```
3
4
```

om

Ranges

```
for number in range(3):
    print(number)
```

```
0
1
2
```

om

```python
for number in range(1, 3):
    print(number)
```

```
1
2
```

om

```python
for number in range(1, 10, 2):
    print(number)
```

```
1
3
5
7
9
```

om

```
animals = ['man', 'bear', 'pig', 'cow',
'duck', 'horse', 'dog']
for number in range(0, len(animals), 2):
    print(animals[number])
```

```
man
pig
duck
dog
```

om

Section Summary

Summary

- Lists are created using comma separated values between square brackets. The format is:

```
list_name = [item_1, item_2, item_N]
```

Summary

- Items in a list can be accessed by index. List indices are zero based. The format is:

  `list_name[index]`

- Access items from the end of the list by using negative indices. The last item in a list is:

  `list_name[-1]`

Summary

- Add items to a list by using the `append()` or `extend()` list methods.
- Access a portion of a list using a slice. The format is: `list_name(start, stop)`
- The list `index()` method accepts a value as a parameter and returns the index of the first value in the list or an exception if the value is not in the list.

Summary

- Loop through a list using a for loop. The format is `for item_variable in list_name:` followed by a code block.
- The code block in a while loop executes as long as the condition evaluates to true. The format is `while condition:` followed by a code block.

Summary

- To sort a list, use the `sort()` list method or the built-in `sorted()` function.
- The built-in `range()` function generates a list of numbers. The format is: `range (start, stop, step)`
- Unhandled exceptions cause Python programs to terminate. Handle exceptions using `try/except` blocks.