python™

Bootcamp

# Session 5

ALPHA
CLOUD SERVICES

alphacloudone.com

python

Bootcamp

## Class Object

- Another object in Python is the Class object.
- The use of classes allows programmers to create their own objects that have custom methods and common attributes.
- Similar to how other types of objects, such as lists and dictionaries have built-in methods, our Class objects can have their own custom built-in methods
- In large programs, Classes give us the opportunity to maintain and use objects repeatedly

ALPHA
CLOUD SERVICES

python™

Bootcamp

## Class Object

Let’s start with a conceptual view of what a Class Object might look like

For this example, our Class is going to be an Automobile

For this class, we will define three attributes

- Engine
- Doors
- tires

ALPHA
CLOUD SERVICES

alphacloudone.com

python™

Bootcamp

## Class Object

Let’s create an instance of this Class

We’ll call it HondaAccord

And we’ll set the following attributes

- Engine- v6
- Doors- 4
- Tires- Goodyear

Now I can pass around this object, and retrieve it’s attributes at any time

ALPHA
CLOUD SERVICES

python™

Bootcamp

## Class Object

Let’s create another instance of this Class

We’ll call it ChevySiverado

And we’ll set the following attributes

- Engine- v8
- Doors- 2
- Tires- Continental

Now I can pass around this object, with it’s specific attributes

ALPHA
CLOUD SERVICES

python

Bootcamp

## Class Object

- Now that we have our new class called Automobile, we can create some custom methods for it
- For example, we could have one called change_oil, that checks on the specs for the object and determines which type of oil it needs
- Or one that is called sell_car, that passes ownership of the object.
- These are conceptual views of what we can do with Classes, but you would approach coding a new class object in the same way.
- Figure out what you want to do with your new class, what attributes it will hold, and the functions you want to perform on it/with it before you start coding.

ALPHA CLOUD SERVICES

python

Bootcamp

## Coding a new Class Object

```
class NewClassName():
    def __init__(self, parameter1, parameter2)
    self.param1 = parameter1
    self.param2 = parameter2


    def new_method_name(self):
```

ALPHA
CLOUD SERVICES

python

Bootcamp

## Coding a new Class

```
class NewClassName():
      def __init__(self, parameter1, parameter2)
          self.param1 = parameter1
          self.param2 = parameter2


   def new_method_name(self, input1, input2):
```

Name of the Class- you will call this whenever creating a new class object. Use camel case.

Required function- this defines the structure of the class object, including it's parameters

Custom methods you create. Must always include 'self' in method definitions.

ALPHA CLOUD SERVICES

python™

Bootcamp

## User Input

In Python, we can prompt for user input

```
input_text = input(“Pease type in some text”)
```

The user input is then saved as our variable input_text

ALPHA
CLOUD SERVICES

alphacloudone.com

python™

Bootcamp

## User Input

- Input is always text
- Can check for proper response, and loop until the expected input is received
- Can use the **int** method to convert a string number to int
    - “8” to int 8

```
int(input_variable)
```

ALPHA
CLOUD SERVICES

alphacloudone.com

python™

Bootcamp

## User Input

- When you run the input method, it waits for the input
    - Wont proceed until the user responds
- If you run the cell twice, it will get stuck

ALPHA
CLOUD SERVICES

python™

Bootcamp

## User Input

**Validation Loop**

```
Valid_input = False
while valid_input  == False
    Choice = input(“enter input”)
    If choice != ‘Y’ or choice != ‘N’
        Print(“that is not valid, try again”)
    else
        valid_input = True
```

ALPHA
CLOUD SERVICES

alphacloudone.com

python™

Bootcamp

## Scope

- Set of rules that determines visibility of your variable in other areas of your code
- LEGB
  - Local- assigned in a function or lambda
  - Enclosing function locals- names in the local scope of any and all enclosing functions from inner to outer
  - Global- assigned at the top-evel of a module file, or declared global
  - Built-in – Names preassigned in the built-in names module- list, open, range, etc

ALPHA
CLOUD SERVICES

python™

Bootcamp

## Scope

- Python checks in order
    - Local
    - Enclosed local
    - Global
- Variables assigned globally are accessible anywhere
- Variables assigned in a function, accessible inside nested functions
- Variables assigned in a function aren't accessible outside of the function
- Don't assign Global variables in functions
- Be careful not to overwrite built-in names

ALPHA
CLOUD SERVICES