Introduction to Python

Lecture 5: Classes and Objects

Daniel Kadyrov

Objects and Classes

- Python is an object-oriented programming language.
- Almost everything in Python is an object, with its properties and methods.
- A Class is like an object constructor, or a "blueprint" for creating objects.
- A Class is defined using the `class` keyword.
- The `class` keyword is followed by the class name, parentheses `()`, and a colon `:`.
- The `self` parameter is a reference to the current instance of the class, and is used to access variables that belong to the class.
- It does not have to be named `self`, you can call it whatever you like, but it has to be the first parameter of any function in the class.

Objects and Classes

- Use the `init()` function to assign values to object properties, or other operations that are necessary to do when the object is being created.
- The `init()` function is called automatically every time the class is being used to create a new object.
- All classes have a function called `init()`, which is always executed when the class is being initiated.
- Use the `self` parameter to refer to the current instance of the class, and access variables that belongs to the class.
- It does not have to be named `self`, you can call it whatever you like, but it has to be the first parameter of any function in the class.

Objects and Classes

Example

```
1  class MyClass:
2      x = 5
3      def __init__(self, name, age):
4          self.name = name
5          self.age = age
6      def myfunc(self):
7          print("Hello my name is " + self.name)
8      def myfunc2(self):
9          print("Hello my age is " + str(self.age))
```