14.7: CLASSES AS TYPES

Contributed by Chuck Severance
Clinical Associate Professor (School of Information) at University of Michigan

As we have seen, in Python, all variables have a type. And we can use the built-in `dir` function to examine the capabilities of a variable. We can use `type` and `dir` with the classes that we create.

CODE 14.7.1 (PYTHON):

```
%%python3

class PartyAnimal:
   x = 0

   def party(self) :
     self.x = self.x + 1
     print("So far",self.x)

an = PartyAnimal()
print ("Type", type(an))
print ("Dir ", dir(an))
print ("Type", type(an.x))
print ("Type", type(an.party))

# Code: http://www.py4e.com/code3/party3.py
```

run restart

When this program executes, it produces the following output:

```
Type <class '__main__.PartyAnimal'>
Dir  ['__class__', '__delattr__', ...
'__sizeof__', '__str__', '__subclasshook__',
'__weakref__', 'party', 'x']
Type <class 'int'>
Type <class 'method'>
```

You can see that using the `class` keyword, we have created a new type. From the `dir` output, you can see both the `x` integer attribute and the `party` method are available in the object.