

Introduction to classes in Python

We'll explore classes in Python in this chapter.

Everything in Python is an object. That's a very vague statement unless you've taken a computer programming class or two. What this means is that every *thing* in Python has methods and values. The reason is that everything is based on a class. A class is the blueprint of an object. Let's take a look at what I mean:

```
x = "Mike"
attrs = dir(x)
print(attrs)

# ['__add__', '__class__', '__contains__', '__delattr__', '__doc__', '__eq__',
#  '__format__', '__ge__', '__getattr__', '__getitem__', '__getnewargs__',
#  '__getslice__', '__gt__', '__hash__', '__init__', '__le__', '__len__', '__lt__',
#  '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
#  '__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__',
#  '_formatter_field_name_split', '_formatter_parser', 'capitalize', 'center', 'count',
#  'decode', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'index', 'isalnum',
#  'isalpha', 'isdigit', 'islower', 'isspace', 'istitle', 'isupper', 'join', 'ljust',
#  'lower', 'lstrip', 'partition', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition',
#  'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title',
#  'translate', 'upper', 'zfill']
```



Here we have a string assigned to the variable **x**. It might not look like much, but that string has a lot of methods. If you use Python's **dir** keyword, you can get a list of all the methods you can call on your string. There are 71 methods here! Technically we're not supposed to call the methods that start with underscores directly, so that reduces the total to 38, but that's still a lot of methods! What does this mean though? It means that a string is based on a class and **x** is an **instance** of that class!

In Python we can create our own classes. Let's get started!

