# Classes/Objects

**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.**

# Objects

**In Python, *everything is an object*. Remember from previous lectures we can use type() to check the type of object.**

In [1]:

```
print(type(1))
print(type([]))
print(type(()))
print(type({}))
```

```
<class 'int'>
<class 'list'>
<class 'tuple'>
<class 'dict'>
```

**So we know all these things are objects, so how can we create our own Object types? That is where the `class` keyword comes in.**

# class

**User defined objects are created using the `class` keyword. The class is a blueprint that defines the nature of a future object. From classes we can construct instances. An instance is a specific object created from a particular class. For example, above we created the object `lst` which was an instance of a list object.**

**Let see how we can use `class`:**

In [2]:

```
# Create a new object type called Sample
class Sample:
    pass

# Instance of Sample
x = Sample()

print(type(x))
```

```
<class '__main__.Sample'>
```

**By convention we give classes a name that starts with a capital letter. Note how `x` is now the reference to our new instance of a Sample class. In other words, we instantiate the Sample class.**

**Inside of the class we currently just have pass. But we can define class attributes and methods.**

In [ ]:

```

```