|  |  |
| --- | --- |
| **class** | Tell Python to make a new type of thing. |
| **object** | Two meanings: the most basic type of thing, and any instance of some thing. |
| **instance** | What you get when you tell Python to create a class. |
| **def** | How you define a function inside a class. |
| **self** | Inside the functions in a class, self is a variable for the instance/object being accessed. |
| **inheritance** | The concept that one class can inherit traits from another class, much like you and your  parents. |
| **composition** | The concept that a class can be composed of other classes as parts, much like how a car has  wheels. |
| **attribute** | A property classes have that are from composition and are usually variables. |
| **is-a** | A phrase to say that something inherits from another, as in a ”salmon” is-a ”fish.” |
| **has-a** | A phrase to say that something is composed of other things or has a trait, as in ”a salmon has-a  mouth.” |

|  |  |
| --- | --- |
| **class X(Y)** | ”Make a class named X that is-a Y.” |
| **class X(object): def \_\_init\_\_(self, J)** | ”class X has-a \_\_init\_\_ that takes self and J parameters.” |
| **class X(object): def M(self, J)** | ”class X has-a function named M that takes self and J parameters.” |
| **foo = X()** | ”Set foo to an instance of class X.” |
| **foo.M(J)** | ”From foo get the M function, and call it with parameters self, J.” |
| **foo.K = Q** | ”From foo get the K attribute and set it to Q.” |