* In JS, all objects have a hidden [[Prototype]] property that’s either another object or null.
* We can use obj.\_\_proto\_\_ to access it (historical getter/setter)
* The object REFERENCED by [[Prototype]] is called a prototype.
  + Referenced meaning it does not modify the original object
* Reading a property of an obj
  + Search first in the obj itself
  + Then goes upward from its inherited prototypes (up and up to the prototypal chain until it found the property or if not found, returns null)
* Write/delete operations act direct on the object
* Calling obj.method()
  + “this” still references obj even if the method is from its prototype
* For…in loop
  + Iterates both its own and inherited properties
  + All other Obj key/value getting methods only operate on the object itself
    - Returns only own properties