Lesson 7.3 - Getters, Setters, and Base Class Access

A couple of interesting things we can do with classes is defining getters and setters. It looks like this:

```
class Car {
  constructor(make) {
    this._make = make
  }

  get make() {
    return this._make
  }

  set make(newMake) {
    this._make = newMake
  }
}
```

This is a pretty simple concept and allows us to have some control over the inner workings of our class at the instance level. Let's look at it in action:

```
> var car = new Car('Toyota')
> console.log(car.make)
-> Toyota
> car.make = 'Dodge'
> console.log(car.make)
-> 'Dodge'
```

Cool! Let's check out how we can access the superclass when using inheritance.

Imagine we have the `Truck` class from the last lesson, and we want to add the `drive` variable to the `print` method. We can do that pretty easily like this:

```
print() {
  return `${super.print()} with ${this.drive}`
}
```

What we did there was access the superclass with the `super` keyword, and then call the `print` method from the superclass. We used string interpolation to add this to a new string that includes whether the vehicle is 2WD or 4WD. That’s pretty cool!