Introduction to classes

JavaScript is an *object-oriented programming* (OOP) language we can use to model real-world items. So, it is very important to learn how to make classes.

Classes are a tool that developers use to quickly produce similar objects.

```
class Dog {
  constructor(name) {
    this._name = name;
    this._behavior = 0;
  }

  get name() {
    return this._name;
  }
  get behavior() {
    return this._behavior;
  }

  incrementBehavior() {
    this._behavior ++;
  }
}

const halley = new Dog('Halley');
```

Constructor, Instance and Methods

```
class Surgeon {
  constructor(name, department) {
    this._name = name;
    this._department = department;
    this._remainingVacationDays = 20;
  }

  get name() {
    return this._name;
  }

  get remainingVacationDays() {
    return this._remainingVacationDays;
  }

  takeVacationDays(daysOff) {
    this._remainingVacationDays -= daysOff;
  }
}

const surgeonRomero = new Surgeon('Francisco Romero',
'Cardiovascular');
surgeonRomero.takeVacationDays(5);
console.log(surgeonRomero.remainingVacationDays);
```

Inheritance

```
class Animal {
  constructor(name) {
    this._name = name;
    this._behavior = 0;
  }

  get name() {
    return this._name;
  }

  get behavior() {
    return this._behavior;
  }

  incrementBehavior() {
    this._behavior++;
  }
}
```

```
class Cat extends Animal {
  constructor(name, usesLitter) {
    super(name);
    this._usesLitter = usesLitter;
  }
}
```

Static methods

Sometimes you will want a class to have methods that aren't available in individual instances, but that you can call directly from the class.

Take the `Date` class, for example — you can both create `Date` instances to represent whatever date you want, and call *static* methods, like `Date.now()` which returns the current date, directly from the class. The `.now()` method is static, so you can call it directly from the class, but not from an instance of the class.

```
class Animal {
  constructor(name) {
    this._name = name;
    this._behavior = 0;
  }

  static generateName() {
    const names = ['Angel', 'Spike', 'Buffy', 'Willow',
'Tara'];
    const randomNumber = Math.floor(Math.random()*5);
    return names[randomNumber];
  }
}
```

```
console.log(Animal.generateName()); // returns a name
```

Activity