7.9 JS Classes and Objects

professor owls - answered by Keith Aquino

- Use the keyword `class` to create a class.
- Always add a method named `constructor()`:

```
class ClassName {
  constructor() { ... }
}
```

- For example,

```
class Car {
  constructor(name, year) {
    this.name = name;
    this.year = year;
  }
}
```

  - This creates a class named `Car`.
  - The class has two initial properties: `name` and `year`.

Keyword(s)	Description	Sample code	Console output

`class`	• Used to define and create constructor functions for objects	`class Animal {`
		`  constructor(name) {`
		`  this.name = name;`
		`  }`
			`Mystery Animal makes a sound.`
			`Buddy the Golden Retriever barks.`

		`speak() { console.log(this.name + "makes a sound."); } } class Dog extends Animal { constructor(name, breed) { super(name); this.breed = breed; } speak() { console.log(this.name + "the" + this.breed + "barks."); } wagTail() { console.log(this.name + "wags its tail enthusiastically."); } } const generic_animal = new Animal('Mystery Animal'); const golden_retriever = new Dog('Buddy', 'Golden Retriever'); generic_animal.speak(); golden_retriever.speak(); golden_retriever.wagTail();`	`Buddy wags its tail enthusiastically.`
`extends`	• Used to create a child class that inherits properties and methods from a parent class		
`constructor`	• Used within a class to initialize object instances created from that class with specific initial values and configurations		
`this`	• Refers to the current execution context and allows access to the properties and methods of the object to which it is bound		
`super`	• Used to call and access the parent class's constructor and methods within a child class		

Using a Class

- When you have a class, you can use the class to create objects:

```
let myCar1 = new Car("Ford", 2014);
let myCar2 = new Car("Audi", 2019);
```

- The example uses the `Car` class to create two `Car` instances/objects.
- The constructor method is called automatically when a new object is created.

The Constructor Method

- The constructor method is a special method:
  - It has to have the exact name "constructor"
  - It is executed automatically when a new object is created
  - It is used to initialize object properties
- If you do not define a constructor method, JavaScript will add an empty constructor method.

Class Methods

- Use the keyword `class` to create a class.
  - Always add a `constructor()` method.
  - Then add any number of methods.

```
class ClassName {
  constructor() { ... }
  method_1() { ... }
  method_2() { ... }
  method_3() { ... }
}
```

  - Create a class method named `age`, that returns the Car `age`:

```
class Car {
  constructor(name, year) {
    this.name = name;
    this.year = year;
  }
  age() {
    let date = new Date();
    return date.getFullYear() - this.year;
  }
}

let myCar = new Car("Ford", 2014);
alert("My car is " + myCar.age() + " years old.");
```

- You can send parameters to class methods:

```
class Car {
  constructor(name, year) {
    this.name = name;
    this.year = year;
  }
  age(x) {
    return x - this.year;
  }
}

let date = new Date();
let year = date.getFullYear();

let myCar = new Car("Ford", 2014);
alert("My car is " + myCar.age(year) + " years old.");
```

The extend keyword

- The extends keyword is used to create a child class of another class (parent).
  - The child class inherits all the methods from another class.
  - Inheritance is useful for code reusability: reuse properties and methods of an existing class when you create a new class.
- The following creates a class named `Model` which will inherit the methods from the `Car` class.

```
class Car {
  constructor(brand) {
    this.carname = brand;
  }
  present() {
    return 'I have a ' + this.carname;
  }
}

class Model extends Car {
  constructor(brand, mod) {
    super(brand);
    this.model = mod;
  }
  show() {
    return this.present() + ', it is a ' + this.model;
  }
}
```

```
mycar = new Model("Ford", "Mustang");
document.getElementById("demo").innerHTML = mycar.show();
```

Additional Material

- **Learn more**
  - W3Schools