

JAVASCRIPT BASICS

CLASSES, INHERITANCE



Object Methods

```
1 let user = {  
2   name: "John",  
3   age: 30  
4 };  
5  
6 user.sayHi = function() {  
7   alert("Hello!");  
8 };  
9  
10 user.sayHi(); // Hello!
```

```
user = {  
  sayHi: function() {  
    alert("Hello");  
  }  
};
```

this keyword

```
1 let user = {  
2   name: "John",  
3   age: 30,  
4  
5   sayHi() {  
6     // "this" is the "current object"  
7     alert(this.name);  
8   }  
9  
10 };  
11  
12 user.sayHi(); // John
```

Arrow functions have no “this”

```
4  
5 let user = {  
6   age: "John"  
7 };  
8  
9 user.sayHi = () => {  
10   alert(this.age);  
11 };  
12
```

Result will be undefined

Constructor, operator new

The regular `{...}` syntax allows us to create one object. But often we need to create many similar objects, like multiple users or menu items and so on.

That can be done using constructor functions and the `"new"` operator.

```
1 function User(name) {  
2   this.name = name;  
3   this.isAdmin = false;  
4 }  
5  
6 let user = new User("Jack");  
7  
8 alert(user.name); // Jack  
9 alert(user.isAdmin); // false
```

Optional chaining '?.'

```
1 let user = {}; // a user without "address" property
2
3 alert(user.address.street); // Error!
```

```
1 let user = {}; // user has no address
2
3 alert( user?.address?.street ); // undefined (no error)
```

Class

In object-oriented programming, a class is an extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behavior (member functions or methods).



Wikipedia

```
1 class MyClass {  
2     // class methods  
3     constructor() { ... }  
4     method1() { ... }  
5     method2() { ... }  
6     method3() { ... }  
7     ...  
8 }
```

```
1 class User {
2
3   constructor(name) {
4     this.name = name;
5   }
6
7   sayHi() {
8     alert(this.name);
9   }
10
11 }
12
13 // Usage:
14 let user = new User("John");
15 user.sayHi();
```

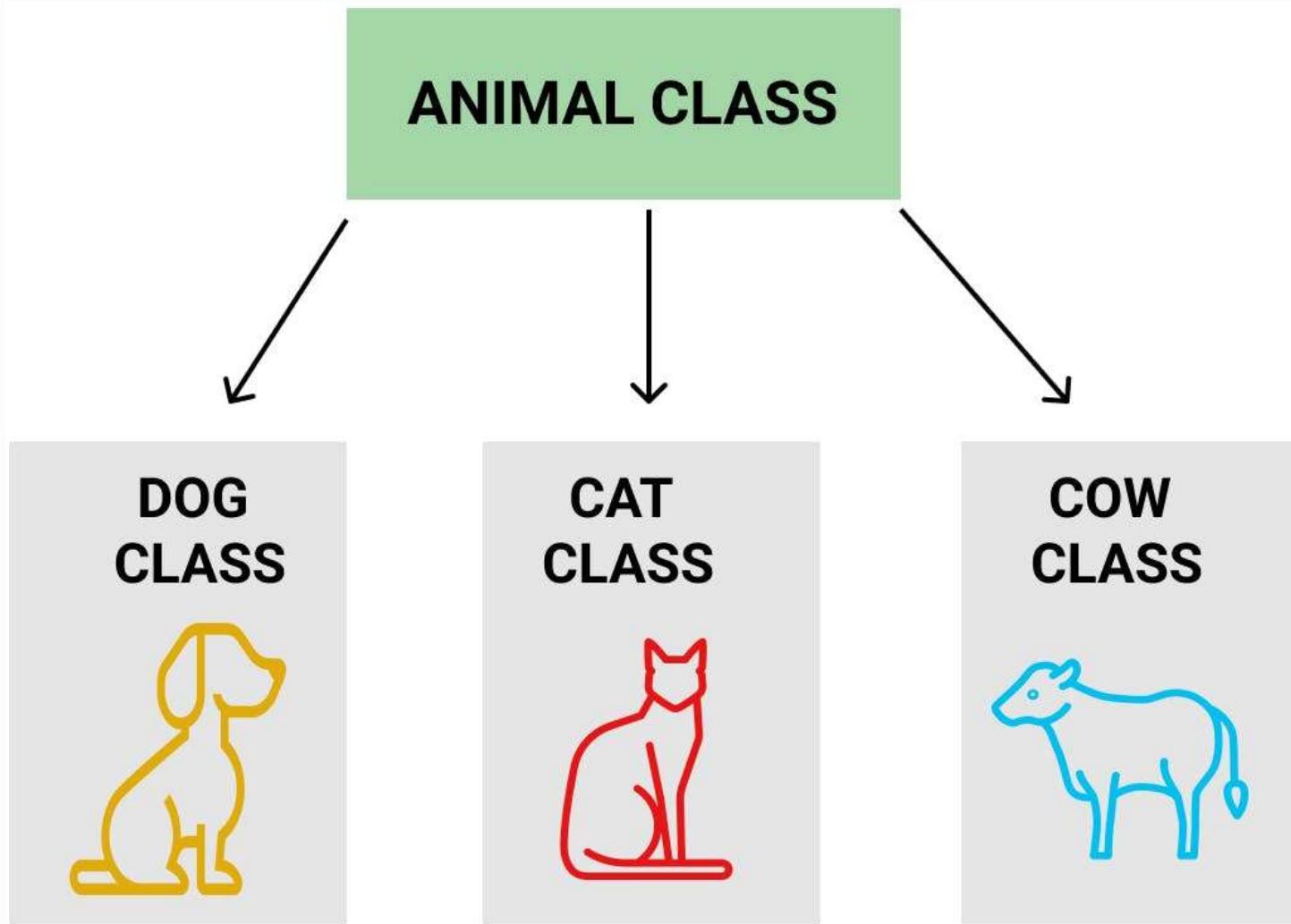
In JavaScript, a class is a kind of function.

```
1 class User {
2   constructor(name) { this.name = name; }
3   sayHi() { alert(this.name); }
4 }
5
6 // proof: User is a function
7 alert(typeof User); // function
```


Getters/Setters

```
1 class User {  
2  
3   constructor(name) {  
4     // invokes the setter  
5     this.name = name;  
6   }  
7  
8   get name() {  
9     return this._name;  
10  }  
11  
12  set name(value) {  
13    if (value.length < 4) {  
14      alert("Name is too short.");  
15      return;  
16    }  
17    this._name = value;  
18  }  
19  
20 }  
21  
22 let user = new User("John");  
23 alert(user.name); // John  
24  
25 user = new User(""); // Name is too short.
```


Class Inheritance



```
class Animal {  
  eat() {  
    console.log("I`m eating");  
  }  
}  
  
class Dog extends Animal {}  
class Cow extends Animal {}  
class Cat extends Animal {}  
  
const rex = new Dog();  
const tom = new Cat();  
const bagration = new Cow();  
  
console.log(rex.eat());  
console.log(tom.eat());  
console.log(bagration.eat());
```

super()

```
▼ class Animal {
▼   constructor(name) {
      this.name = name;
    }
▼   eat() {
      console.log(`I'm ${this.name} and I am eating`);
    }
  }

▼ class Dog extends Animal {
▼   constructor(name) {
      console.log(`My name is ${name}, nice to meet you`);
      super(name);
    }
  }
class Cow extends Animal {}
class Cat extends Animal {}

const rex = new Dog("Rex");
const tom = new Cat("Tom");
const bagration = new Cow("Bagration");
```

Tasks

1. Create a **BMW**, **Mercedes**, **Audi** classes extended from **Car** class
 - 1.2. All cars must have a **name** which is not changeable(Show warning message when someone will try to do it).
 - 1.3 Cars must have **color**, **width**, **bodyMass**, **maxSpeed** properties.
 - 1.4 All cars must have **getInfo()** method which will return all info about car.
 - 1.5 **BMW** class must have **`sportMode()`** method which will change **mode** of car to **sport**.
 - 1.6 **Mercedes** must have **`ecoMode()`** method which will change **mode** of car to **eco**.
 - 1.7 All Cars must have **drive()** method which will log info like **You are driving white Bmw which is now on eco mode and max speed is 260km/h** or **You are driving white Bmw which max speed is 180km/h**