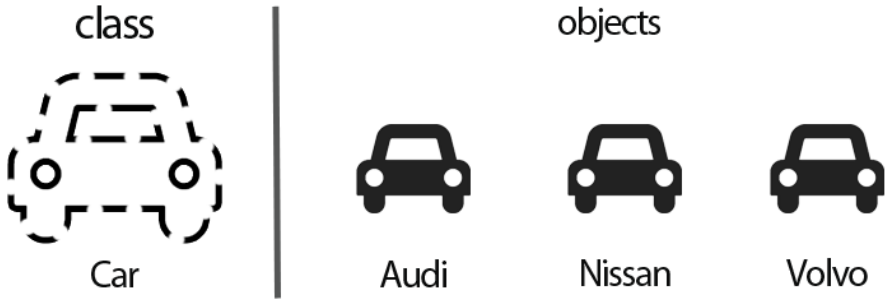
JS-V

Clases

En informática, una **clase** es una plantilla para la creación de objetos de datos según un modelo predefinido. Las clases se utilizan para representar entidades o conceptos, como los sustantivos en el lenguaje.



Class e instanciación pseudo-clásica

```
 1 function Gato(nombre) {
 2     // El nuevo operador crea un objeto, "this"
 3     this.nombre = nombre;
 4     this.maullar = function() {
 5         return 'Mi nombre es ' + this.nombre + ' ... Meow!';
 6     }
 7     // Devuelve el objeto "this"
 8 }
 9
10 var sam = new Gato('Sam');
11 var kitty = new Gato('Kitty');
12 console.log(sam.maullar()); // 'Mi nombre es Sam ... Meow!'
13 console.log(kitty.maullar()); // 'Mi nombre es Kitty ... Meow!'
```

Prototipos

Las clases tienen una forma única de establecer un método una vez y dar acceso a cada objeto de esa clase a esos métodos. Esto se llama el prototype. Cada clase tiene una propiedad *prototype*, que luego podemos establecer en métodos:

```
function Usuario(nombre, email) {
    this.nombre = nombre;
    this.email = email;
}

Usuario.prototype.presentacion = function(){
    return 'Mi nombre es ' + this.nombre + ', mi email es ' + this.email + '.';
}

let juan = new Usuario('Juan', 'juanperez@mail.com');
let antonio = new Usuario('Antonio', 'anton@mail.com');

console.log(juan.presentacion()); // Mi nombre es Juan, mi email es juanperez@mail.com.
console.log(antonio.presentacion()); // Mi nombre es Antonio, mi email es anton@mail.com.
```

Object.create

El método create de los objetos nos permite crear un nuevo objeto a partir de un prototype especificado.

```
1 // creamos un objecto con un objeto vacio como proto
2 var obj = Object.create({})
3
4 console.log(obj) // Object {}
5
6 // creamos un objeto a partir de un proto de Objeto
7 var obj = Object.create(Object.prototype)
8 // que es lo mismo que crear un objeto vacio literal
9 var obj = {}
```

Object.assign

Nos permite agregar propiedades a un objeto pasado por parámetro:

```
var obj = {}

// No es necesario guardar el resultado porque los objetos
Object.assign(obj, {nombre:'Emi', apellido:'Chequer'})

obj.nombre // 'Emi'
```

Herencia Clásica

```
1  function Persona(nombre, apellido, ciudad) {
2    this.nombre = nombre;
3    this.apellido = apellido;
4    this.ciudad = ciudad;
5  }
6
7  Persona.prototype.saludar = function() {
8    console.log('Soy '+ this.nombre +' de '+ this.ciudad);
9  }
10
11 var Emi = new Persona('Emi', 'Chequer', 'Buenos Aires');
12
13 Emi.saludar(); // 'Soy Emi de Buenos Aires'
```

```
1 function Alumno(nombre, apellido, ciudad, curso) {
2   // podríamos copiar las mismas propiedades de Persona acá adentro
3   this.nombre = nombre;
4   this.apellido = apellido;
5   this.ciudad = ciudad;
6   this.curso = curso
7 }
```