

Java Script es un lenguaje multi-paradigma que permite aplicar los principios de la programación orientada a objetos, como la herencia, el polimorfismo y la abstracción, existen dos formas principales mediante las cuales se puede aplicar herencia en Java Script y es a través de las clases, introducidas con ES6 y a través del uso de funciones constructoras y prototipos. En ambos casos se puede utilizar y extender el código base, la herencia con clases se logra al utilizar la palabra reservada extends, esto hace que una clase hija herede métodos y propiedades de la clase padre, también se pueden escribir métodos heredados, estos se relacionan con el polimorfismo. Por otro lado utilizando la función constructora Prototype es posible aplicar la herencia de forma más detallada.

Sí mismo el polimorfismo se presenta cuando distintas clases definen un mismo método, pero con comportamientos diferentes como hablar en las clases Animal y Perro o por otro lado Java Script no soporta la sobrecarga de métodos tradicionales aunque se puede simular con funciones dentro de una sola función, mientras que en la abstracción que en otros lenguajes se logra con clases y métodos abstractos se puede simular en Java Script como por ejemplo, se puede lanzar un error si se intenta instanciar una clase directamente o si el método no proviene de una clase hija, esto obliga a los desarrolladores a definir métodos para ocultar la complejidad del código.

Clase auto 1

Clas Auto E

arrancar() E

console.log ("Arrancar Auto");

manjar() E

console.log ("Manjar Auto");

3

Encender() E

console.log ("Encender Auto");

3

Clase computadora 2

Class Computadora E

encender() E

console.log ("Encender computadora");

3

estudiar Datos() E

console.log ("Estudiar datos.");

3

apagar() E

console.log ("Apagar computadora");

3

3

Clase Celular 3

Class Celular {

 marcarChumtro() {

 console.log("Marcando " + numeros);

 }

 desbloquear() {

 console.log("Desbloquear celular");

 }

 grabarVideo() {

 console.log("Video grabado");

 }

}

Clas S Super heroe E

Constructor (Nombre, poder, ciudad, edad, universo) E

```
this.nombre = nombre;  
this.poder = poder;  
this.ciudad = ciudad;  
this.edad = edad;  
this.universo = universo;
```

3
3

```
const Superman = Superheroe("Superman", "Fuerza", "Metrópolis", 35, "DC");  
const Spiderman = Superheroe("Spiderman", "Agilidad", "Nueva York", 18, "Marvel");  
const Batman = Superheroe("Batman", "Inteligencia", "Gotham", 40, "DC");  
const Ironman = Superheroe("Iron man", "Tecnología", "Malibú", 38, "Marvel");  
const Hulk = Superheroe("Hulk", "Fuerza bruta", "Desconocido", 39, "Marvel");
```

Clas Pelicula E

Constructor (titulo, genero, director, año, duracion) E

```
this.titulo = titulo;  
this.genero = genero;  
this.director = Director;  
this.año = año;  
this.duracion = Duracion;
```

3
3

```
const Interstellar = new Pelicula("Interstellar", "Ciencia Ficción", "Christopher Nolan", 2014, "169 minutos");  
const Titanic = new Pelicula("Titanic", "Romance", "James Cameron", 1997, "195 minutos");  
const Avengers = new Pelicula("Avengers", "Acción", "Christopher Nolan", 2010, "148 minutos");  
const Inception = new Pelicula("Inception", "Thriller", "Joss Whedon", 2010, "143 minutos");  
const Coco = new Pelicula("Coco", "Animación", "Lee Unkrich", 2017, "105 minutos");
```

Class Edificio &

Constructor (nombre, altura, pisos, ubicación, uso) &

```
this.nombre = nombre;  
this.altura = altura;  
this.pisos = pisos;  
this.ubicacion = ubicacion;  
this.uso = uso;
```

3

3

```
const empireState = new Edificio("Empire State", "381", "102", "Nueva York",  
"Oficinas");  
const burjKhalifa = new Edificio("Burj Khalifa", "828", "163", "Dubái",  
"Residencial");  
const torreEiffel = new Edificio("Torre Eiffel", "300", "3", "París", "Turismo");  
const shard = new Edificio("Shard", "720", "95", "Londres", "Hotel");  
const latinoamericana = new Edificio("Latinoamericana", "182", "44", "CDMX",  
"Museo");
```