

1. Create an object animal with properties to store family, number of paws and how it moves and three objects, cat, spider and rabbit that inheritate from animal. Initialize every animal with proper values and iterate through them showing their values

```
let animal = {
  familia: "",
  numeroPatas: 0,
  seMueve: "",
}

let gato = {

}

let spider = {

}

let conejo = {

}

Object.setPrototypeOf(gato, animal);
Object.setPrototypeOf(spider, animal);
Object.setPrototypeOf(conejo, animal);

gato.familia = "felino";
gato.numeroPatas = 4;
gato.seMueve = "andando";

spider.familia = "aracnidos";
spider.numeroPatas = 8;
spider.seMueve = "trepan";

conejo.familia = "leporidos";
conejo.numeroPatas = 4;
conejo.seMueve = "saltando";

for (let valor in conejo){
  console.log(valor + ": " + conejo[valor]);
}

for (let valor in spider){
  console.log(valor + ": " + spider[valor]);
}
```

```
}

for (let valor in gato){
  console.log(valor + ": " + gato[valor]);
}
```

2. Given the following code, write your own code to make informatico and ordenanza inherit from persona and to check its operation. If needed, make the changes you consider in order to properly work.

```
let persona = {
  bolsillo: ,
  listaTareas: [],
  cobraSueldo(dinero) {
    this.bolsillo=dinero;
  },
  anyadeTarea(tarea){
    this.listaTareas.push(tarea);
  }
};
let informatico = {
};
```

```
let ordenanza = {
};
```

```
let persona = {
  bolsillo: 0,
  listaTareas: [],
  cobraSueldo(dinero) {
    this.bolsillo=dinero;
  },
  anyadeTarea(tarea) {
    if (this.listaTareas.length == 0) {
      this.listaTareas = [];
    }
    this.listaTareas.push(tarea);
  },
};

let informatico = {

};

let ordenanza = {
```

```
};

Object.setPrototypeOf(informatico, persona);
Object.setPrototypeOf(ordenanza, persona);

informatico.cobraSueldo(3000);
informatico.anyadeTarea("Programa en javaScript");
console.log(informatico.bolsillo);
console.log(informatico.listaTareas);

ordenanza.cobraSueldo(1500);
ordenanza.anyadeTarea("Organizar grupos trabajo");
console.log(ordenanza.bolsillo);
console.log(ordenanza.listaTareas);
```

3. Take the former code and add a method to remove tasks from listaTareas once informatico or ordenanza have finished them.

```
let persona = {
  bolsillo: 0,
  listaTareas: [],
  cobraSueldo(dinero) {
    this.bolsillo=dinero;
  },
  anyadeTarea(tarea) {
    if (this.listaTareas.length == 0) {
      this.listaTareas = [];
    }

    this.listaTareas.push(tarea);
  },
  quitarTarea(tarea) {
    for (let i = 0; i < this.listaTareas.length; i++) {
      if (tarea == this.listaTareas[i]) {
        this.listaTareas.splice(i, 1);
      }
    }
  }
};

let informatico = {

};
```

```
let ordenanza = {  
  
};  
  
Object.setPrototypeOf(informatico, persona);  
Object.setPrototypeOf(ordenanza, persona);  
  
informatico.cobraSueldo(3000);  
informatico.anyadeTarea("Programa en javaScript");  
informatico.anyadeTarea("Suma.js");  
console.log(informatico.bolsillo);  
console.log(informatico.listaTareas);  
informatico.quitarTarea("Suma.js");  
console.log(informatico.listaTareas);  
  
ordenanza.cobraSueldo(1500);  
ordenanza.anyadeTarea("Organizar grupos trabajo");  
console.log(ordenanza.bolsillo);  
console.log(ordenanza.listaTareas);
```

4. Using objects and inheritance, create the needed structure to store information regarding vehicles. Besides Each object should have information about type of vehicle, number of wheels and type of engine, and the methods accelerate, brake, stop. Add two more properties and two more methods.

```
let vehiculos = {  
  tipo: "",  
  numeroRuedas: 0,  
  tipoMotor: "",  
  velocidad: 0,  
  kilometros: 0,  
  acelerar(velocidad) {  
    this.velocidad += velocidad;  
  },  
  parar(velocidad) {  
    this.velocidad -= velocidad;  
  },  
  frenar() {  
    this.velocidad = 0;  
  },  
  añadirVelocidad(nuevaVelocidad) {  
    this.velocidad = nuevaVelocidad;  
  }  
}
```

```
    },  
    kilometrosActuales(nuevoKm){  
        this.kilometros = nuevoKm;  
    }  
}  
  
let yamaha_MT09 = {  
  
}  
  
let cupra_Formentor = {  
  
}  
  
Object.setPrototypeOf(yamaha_MT09, vehiculos);  
Object.setPrototypeOf(cupra_Formentor, vehiculos);  
  
yamaha_MT09.tipo = "Moto";  
yamaha_MT09.numeroRuedas = 2;  
yamaha_MT09.tipoMotor = "890 cc";  
yamaha_MT09.añadirVelocidad(60);  
yamaha_MT09.kilometrosActuales(20000);  
console.log(yamaha_MT09.velocidad);  
console.log(yamaha_MT09.kilometros);  
yamaha_MT09.acelerar(20);  
console.log(yamaha_MT09.velocidad);  
yamaha_MT09.parar(10);  
console.log(yamaha_MT09.velocidad);  
yamaha_MT09.frenar();  
console.log(yamaha_MT09.velocidad);  
  
cupra_Formentor.tipo = "Coche";  
cupra_Formentor.numeroRuedas = 4;  
cupra_Formentor.tipoMotor = "350 cc";  
cupra_Formentor.añadirVelocidad(80);  
cupra_Formentor.kilometrosActuales(15000);  
console.log(cupra_Formentor.velocidad);  
console.log(cupra_Formentor.kilometros);  
cupra_Formentor.acelerar(20);  
console.log(cupra_Formentor.velocidad);  
cupra_Formentor.parar(10);  
console.log(cupra_Formentor.velocidad);
```

```
cupra_Formentor.frenar();  
console.log(cupra_Formentor.velocidad);
```

5. Using objects and inheritance, model something you may think of. You have to use chained inheritance and double inheritance.

```
let animal = {  
  nombre: "",  
  comer() {  
    console.log(`El ${this.nombre} esta comiendo`);  
  },  
  dormir() {  
    console.log(`El ${this.nombre} esta durmiendo`);  
  }  
}  
  
let mamifero = {  
  patas: "",  
  especie: "",  
  seAlimenta(comida) {  
    console.log(`El ${this.nombre} se alimenta de ${comida}`);  
  }  
}  
  
Object.setPrototypeOf(mamifero, animal);  
  
let perro = {  
  
}  
  
Object.setPrototypeOf(perro, mamifero);  
  
perro.nombre = "Pastor Aleman";  
perro.especie = "Canidos";  
perro.seAlimenta("pienso");  
perro.comer();  
perro.dormir();
```