

const Car = function (make, speed){

    this.make = make;

    this.speed = speed;

    this.accelerate = function(){

        this.speed += 10;

        console.log(`${this.make} going at ${this.speed}km/h`);

    }

    this.brake = function(){

        this.speed -= 5;

        console.log(`${this.make} going at ${this.speed}km/h`);

    }

};

const bmw = new Car('BMW', 120);

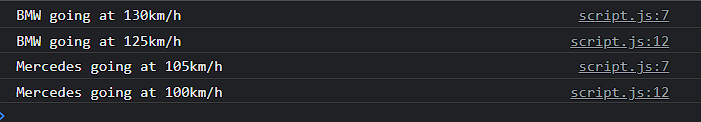
const mercedes = new Car('Mercedes', 95);

bmw.accelerate();

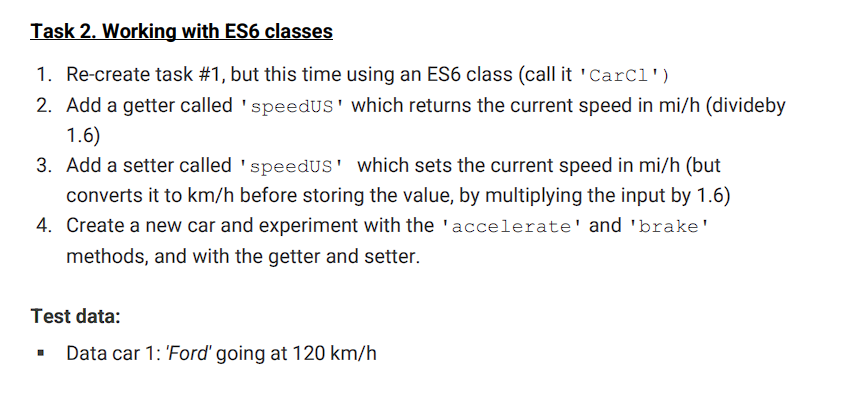
bmw.brake();

mercedes.accelerate();

mercedes.brake();



In this task I use constructor function, just class “Car” have parameters “make” and “speed”. This class have methods “accelerate” and “brake” that change the current speed of the car.



class CarCl{

    constructor(make, speed){

        this.make = make;

        this.speed = speed;

    }

    get speedUS(){

        return this.speed/1.6;

    }

    set speedUS(speed){

        this.speed = speed\*1.6;

    }

    accelerate(){

        this.speed += 10;

        console.log(`${this.make} going at ${this.speed}km/h`);

    }

    brake(){

        this.speed -= 5;

        console.log(`${this.make} going at ${this.speed}km/h`);

    }

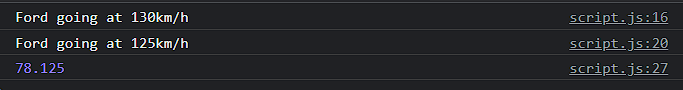
}

let ford = new CarCl('Ford', 120);

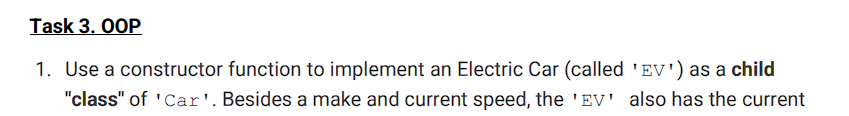
ford.accelerate();

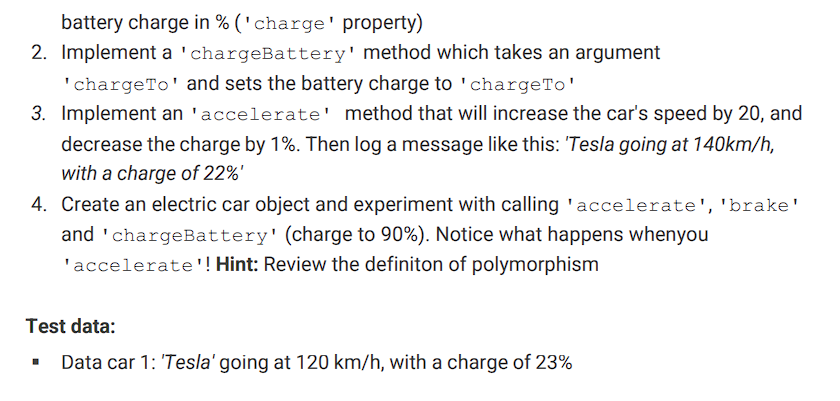
ford.brake();

console.log(ford.speedUS);



In this task also like a task #1 but in ES6, and have getters which can convert kilometers to miles,





function Car(make, speed){

    this.make = make;

    this.speed = speed;

}

Car.prototype.accelerate = function(){

    this.speed += 10;

    console.log(`${this.make} going at ${this.speed}km/h`);

}

Car.prototype.brake = function(){

    this.speed -= 5;

    console.log(`${this.make} going at ${this.speed}km/h`);

}

const EV = function(make, speed, charge){

    Car.call(this, make, speed);

    this.charge = charge;

}

EV.prototype.brake = function(){

    this.speed -= 5;

    console.log(`New speed is ${this.speed}`);

}

EV.prototype.accelerate = function(){

    this.speed += 20;

    this.charge -= 1;

    console.log(`${this.make} going at ${this.make}km/h, with charge of ${this.charge}%`);

}

EV.prototype.chargeBattery = function(chargeTo){

    this.charge = chargeTo;

}

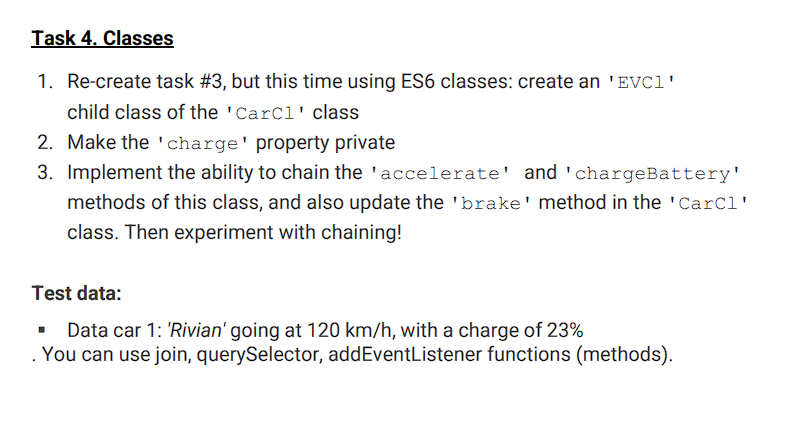
let tesla = new EV('Tesla', 120, 23);

tesla.accelerate();

tesla.brake();

tesla.chargeBattery();

This task is already a little more difficult and interesting. Many details are worked out with electric cars.



class EVCl extends CarCl{

    constructor(make, speed, charge){

        super(make, speed);

        this.charge = charge;

    }

    accelerate(){

        this.speed += 20;

        this.charge -= 1;

        console.log(`${this.make} going at ${this.speed}km/h, with charge of ${this.charge}%`);

    }

    brake(){

        this.speed -= 10;

        this.charge += 1;

        console.log(`${this.make} going at ${this.speed}km/h, with charge of ${this.charge}%`);

    }

    chargeBattery(chargeTo){

        this.charge = chargeTo;

    }

}

tesla = new EVCl('Rivian', 120, 23);

tesla.accelerate();

tesla.brake();

tesla.chargeBattery(90);

