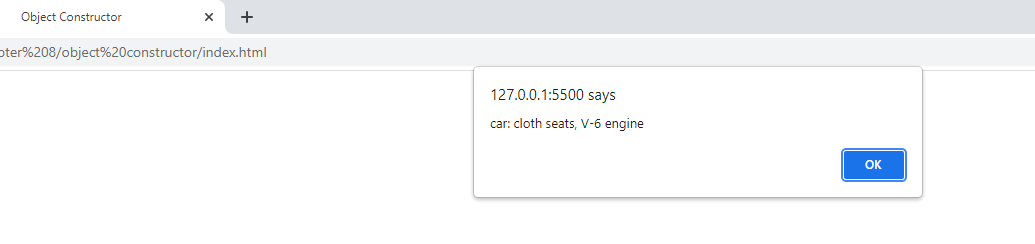
Jesus Nolazco

Chapter 8

**Page 197 Object constructor**



<!DOCTYPE html>

<html>

<head>

    <title>Object Constructor</title>

</head>

<body>

    <script src="obconjs.js"></script>

</body>

</html>

let car = new Object();

car.seats = "cloth";

car.engine = "V-6";

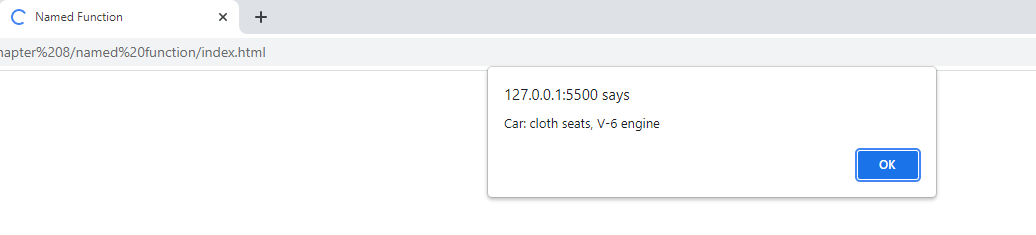
car.show\_features = function() {

    window.alert("car: " + car.seats + " seats, " + car.engine + " engine");

};

car.show\_features();

**Page 197 Named function as method**

****

<!DOCTYPE html>

<html>

<head>

    <title>Named Function</title>

</head>

<body>

    <script src="namefuncjs.js"></script>

</body>

</html>

let car = new Object();

car.seats = "cloth";

car.engine = "V-6";

function my\_alert() {

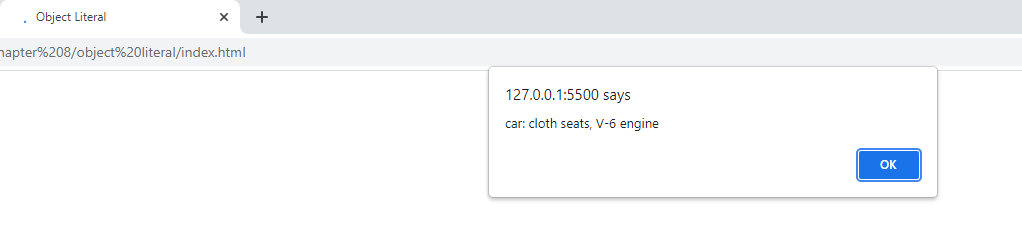
    window.alert("Car: " + car.seats + " seats, " + car.engine + " engine");

}

car.show\_features = my\_alert;

car.show\_features();

**Page 198 Object literal**

****

<!DOCTYPE html>

<html>

<head>

    <title>Object Literal</title>

</head>

<body>

    <script src="objlitjs.js"></script>

</body>

</html>

let car = {

    seats: "cloth",

    engine: "V-6",

    show\_features: function() {

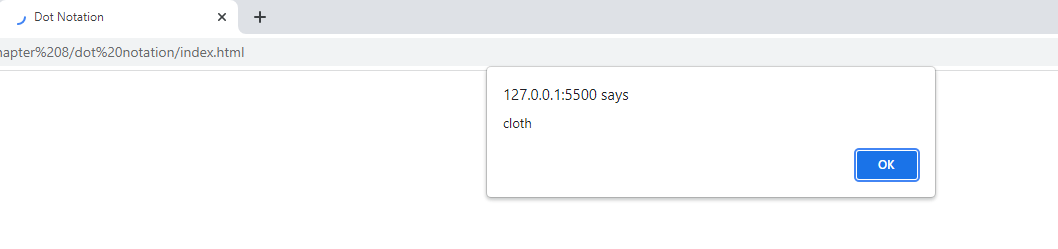
        window.alert("car: " + car.seats + " seats, " + car.engine + " engine");

    }

};

car.show\_features();

**Page 198 Dot notation**

****

<!DOCTYPE html>

<html>

<head>

    <title>Dot Notation</title>

</head>

<body>

    <script src="dotjs.js"></script>

</body>

</html>

let car = {

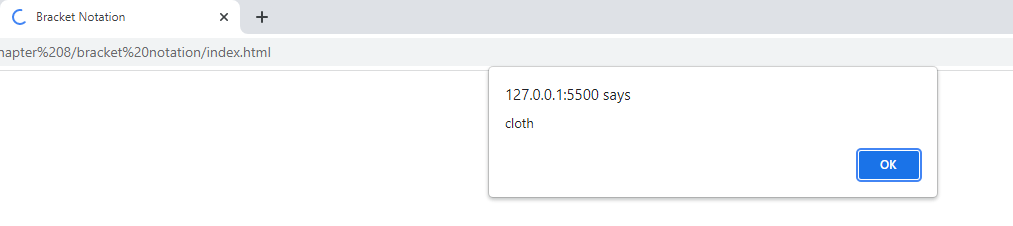
    seats: "cloth",

    engine: "V-6"

};

window.alert(car.seats);

**Page 199 Bracket notation**

****

<!DOCTYPE html>

<html>

<head>

    <title>Bracket Notation</title>

</head>

<body>

    <script src="bracketjs.js"></script>

</body>

</html>

let car = {

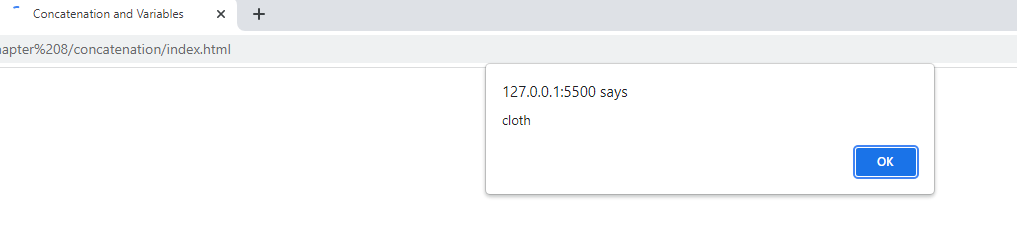
    seats: "cloth",

    engine: "V-6"

};

window.alert(car["seats"]);

**Page 199 Concatenation and variables**

****

<!DOCTYPE html>

<html>

<head>

    <title>Concatenation and Variables</title>

</head>

<body>

    <script src="convarjs.js"></script>

</body>

</html>

let car = {

    seats: "cloth",

    engine: "V-6"

};

let s = "seats";

let start = "sea";

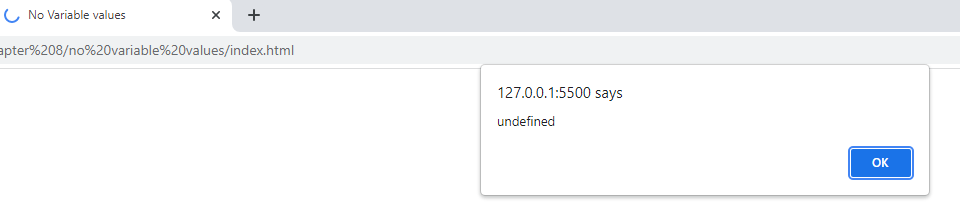
window.alert(car["seats"]);

window.alert(car["sea" + "ts"]);

window.alert(car[s]);

window.alert(car[start + "ts"]);

**Page 199 No variable values**

****

<!DOCTYPE html>

<html>

<head>

    <title>No Variable values</title>

</head>

<body>

    <script src="novarjs.js"></script>

</body>

</html>

let car = {

    seats: "cloth",

    engine: "V-6"

};

let s = "seats";

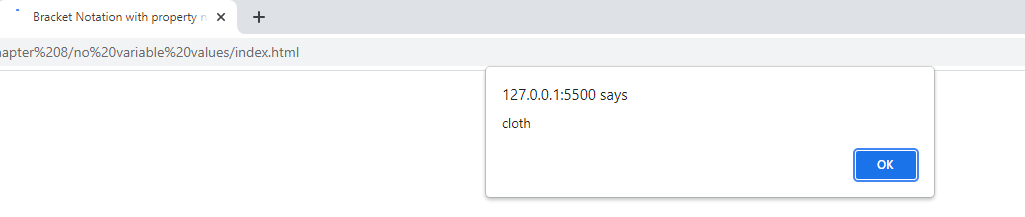
function show\_seat\_type(sts) {

    window.alert(car.sts);

}

show\_seat\_type(s);

**Page 200 Bracket notation with property name**

****

<!DOCTYPE html>

<html>

<head>

    <title>Bracket Notation with property name</title>

</head>

<body>

    <script src="novarjs.js"></script>

</body>

</html>

let car = {

    seats: "cloth",

    engine: "V-6"

};

let s = "seats";

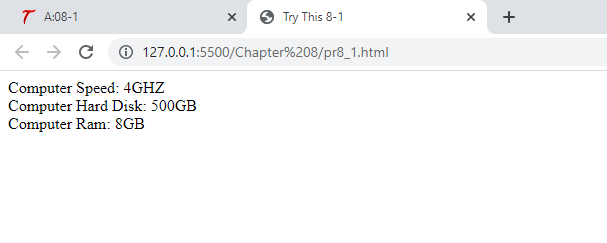
function show\_seat\_type(sts) {

    window.alert(car[sts]);

}

show\_seat\_type(s);

**Try This 8-1**

****

<!DOCTYPE html>

<html>

<head>

    <title>Try This 8-1</title>

</head>

<body>

    <script src="prjs8\_1.js"></script>

</body>

</html>

let computer = {

    speed: "4GHZ",

    hd: "500GB",

    ram: "8GB"

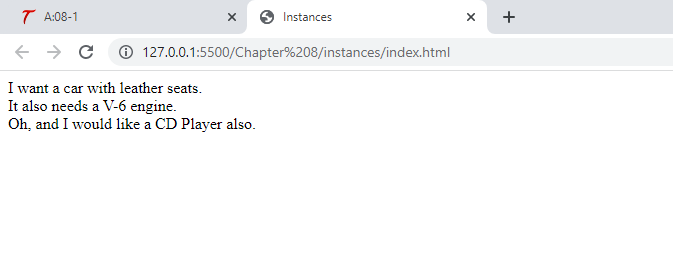
};

document.write("Computer Speed: " + computer.speed + "<br>");

document.write("Computer Hard Disk: " + computer.hd + "<br>");

document.write("Computer Ram: " + computer.ram);

**Page 203 Instances**

****

<!DOCTYPE html>

<html>

<head>

    <title>Instances</title>

</head>

<body>

    <script src="instancesjs.js"></script>

</body>

</html>

function Car(seats, engine, radio) {

    this.seats = seats;

    this.engine = engine;

    this.radio = radio;

}

let work\_car = new Car("cloth", "V-6", "Tape Deck");

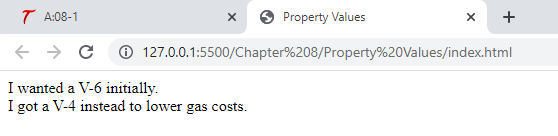
let fun\_car = new Car("leather", "V-8", "CD Player");

document.write("I want a car with " + fun\_car.seats + " seats.<br>");

document.write("It also needs a " + work\_car.engine + " engine.<br>");

document.write("Oh, and I would like a " + fun\_car.radio + " also.");

**Page 204 Property Values**

****

<!DOCTYPE html>

<html>

<head>

    <title>Property Values</title>

</head>

<body>

    <script src="propvaljs.js"></script>

</body>

</html>

function Car(seats, engine, radio) {

    this.seats = seats;

    this.engine = engine;

    this.radio = radio;

}

let work\_car = new Car("cloth", "V-6", "Tape Deck");

let fun\_car = new Car("leather", "V-8", "CD Player");

let original\_work\_car\_engine = work\_car.engine;

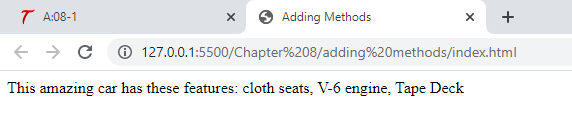
work\_car.engine = "V-4";

let new\_work\_car\_engine = work\_car.engine;

document.write("I wanted a " + original\_work\_car\_engine + " initially.<br>");

document.write("I got a " + new\_work\_car\_engine + " instead to lower gas costs.");

**Page 205 Adding methods**

****

<!DOCTYPE html>

<html>

<head>

    <title>Adding Methods</title>

</head>

<script src="methodjs.js"></script>

</html>

function Car(seats, engine, radio) {

    this.seats = seats;

    this.engine = engine;

    this.radio = radio;

    this.describe = function() {

        document.write("This amazing car has these features: ");

        document.write(this.seats + " seats, " + this.engine + " engine, ");

        document.write(this.radio);

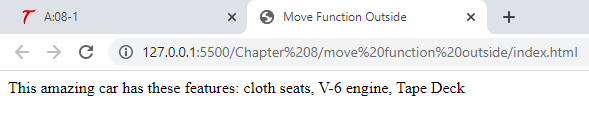
    };

}

let work\_car = new Car("cloth", "V-6", "Tape Deck");

work\_car.describe();

**Page 206 Move function outside**

****

<!DOCTYPE html>

<html>

<head>

    <title>Move Function Outside</title>

</head>

<body>

    <script src="movefuncjs.js"></script>

</body>

</html>

function Car(seats, engine, radio) {

    this.seats = seats;

    this.engine = engine;

    this.radio = radio;

    this.describe = describe\_car;

}

function describe\_car() {

    document.write("This amazing car has these features: ");

    document.write(this.seats + " seats, " + this.engine + " engine, ");

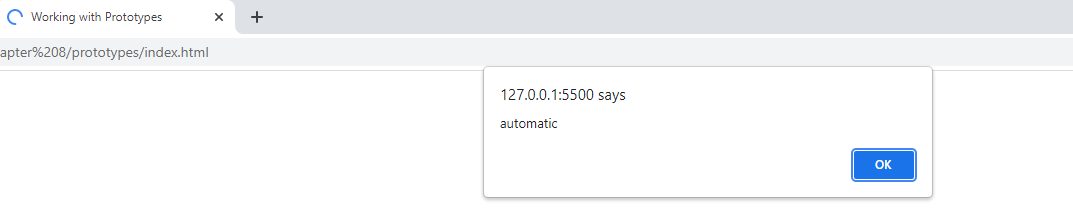
    document.write(this.radio);

}

let work\_car = new Car("cloth", "V-6", "Tape Deck");

work\_car.describe();

**Page 207 Working with prototypes**

****

<!DOCTYPE html>

<html>

<head>

    <title>Working with Prototypes</title>

</head>

<body>

    <script src="protojs.js"></script>

</body>

</html>

function Car(seats, engine, radio) {

    this.seats = seats;

    this.engine = engine;

    this.radio = radio;

}

Car.prototype.locks = "automatic";

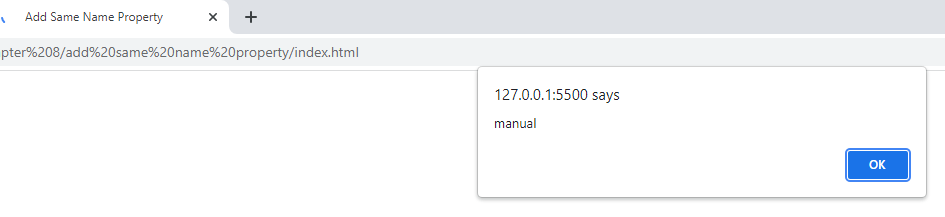
let work\_car = new Car("cloth", "V-6", "Tape Deck");

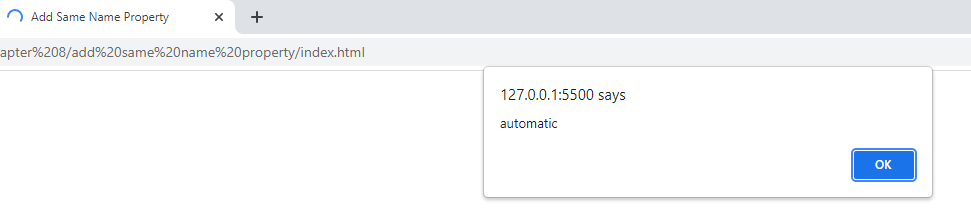
let family\_car = new Car("cloth", "V-4", "CD Changer");

window.alert(work\_car.locks);

window.alert(family\_car.locks);

**Page 208 Adding same name property**

****

****

<!DOCTYPE html>

<html>

<head>

    <title>Add Same Name Property</title>

</head>

<body>

    <script src="samenamejs.js"></script>

</body>

</html>

function Car(seats, engine, radio) {

    this.seats = seats;

    this.engine = engine;

    this.radio = radio;

}

Car.prototype.locks = "automatic";

let work\_car = new Car("cloth", "V-6", "Tape Deck");

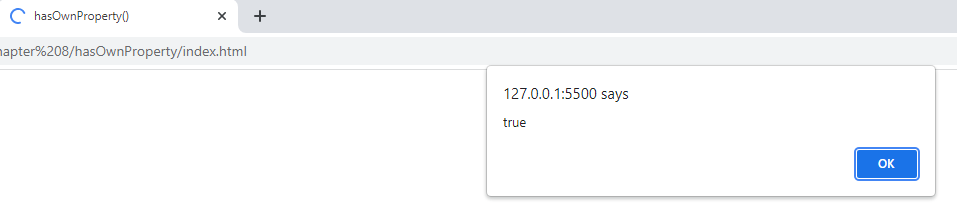
let family\_car = new Car("cloth", "V-4", "CD Changer");

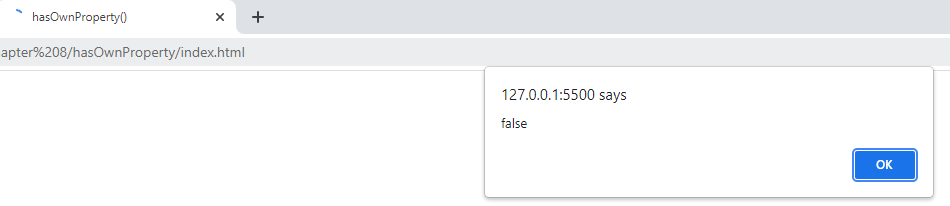
work\_car.locks = "manual";

window.alert(work\_car.locks);

window.alert(family\_car.locks);

**Page 208 hasOwnProperty()**

****

****

<!DOCTYPE html>

<html>

<head>

    <title>hasOwnProperty()</title>

</head>

<body>

    <script src="hasownjs.js"></script>

</body>

</html>

function Car(seats, engine, radio) {

    this.seats = seats;

    this.engine = engine;

    this.radio = radio;

}

Car.prototype.locks = "automatic";

let work\_car = new Car("cloth", "V-6", "Tape Deck");

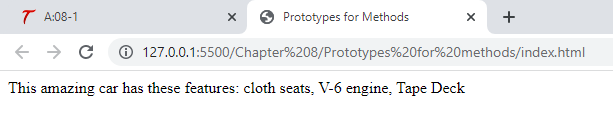
let family\_car = new Car("cloth", "V-4", "CD Changer");

work\_car.locks = "manual";

window.alert(work\_car.hasOwnProperty("locks"));

window.alert(family\_car.hasOwnProperty("locks"));

**Page 209 Prototypes for methods**

****

<!DOCTYPE html>

<html>

<head>

    <title>Prototypes for Methods</title>

</head>

<body>

    <script src="protojs.js"></script>

</body>

</html>

function Car(seats, engine, radio) {

    this.seats = seats;

    this.engine = engine;

    this.radio = radio;

}

Car.prototype.describe = function() {

    document.write("This amazing car has these features: ");

    document.write(this.seats + " seats, " + this.engine + " engine, ");

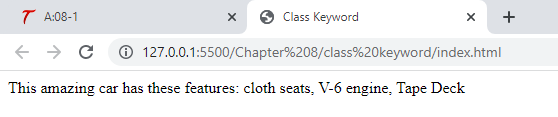
    document.write(this.radio);

};

let work\_car = new Car("cloth", "V-6", "Tape Deck");

work\_car.describe();

**Page 209 Class keyword**

****

<!DOCTYPE html>

<html>

<head>

    <title>Class Keyword</title>

</head>

<body>

    <script src="classjs.js"></script>

</body>

</html>

class Car {

    constructor(seats, engine, radio) {

        this.seats = seats;

        this.engine = engine;

        this.radio = radio;

    }

    describe() {

        document.write("This amazing car has these features: ");

        document.write(this.seats + " seats, " + this.engine + " engine, ");

        document.write(this.radio);

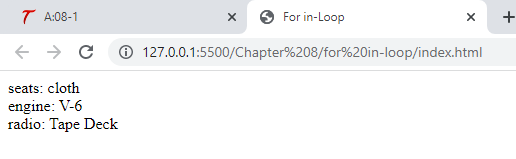
    }

}

let work\_car = new Car("cloth", "V-6", "Tape Deck");

work\_car.describe();

**Page 211 for-in loop**

****

<!DOCTYPE html>

<html>

<head>

    <title>For in-Loop</title>

</head>

<body>

    <script src="inloopjs.js"></script>

</body>

</html>

function Car(seats, engine, radio) {

    this.seats = seats;

    this.engine = engine;

    this.radio = radio;

}

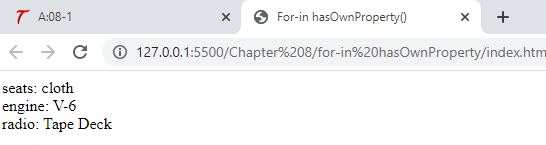
let work\_car = new Car("cloth", "V-6", "Tape Deck");

for (let prop\_name in work\_car) {

    document.write(prop\_name + ": " + work\_car[prop\_name] + "<br>");

}

**Page 211 for-in / hasOwnProperty()**

****

<!DOCTYPE html>

<html>

<head>

    <title>For-in hasOwnProperty()</title>

</head>

<body>

    <script src="inhasownjs.js"></script>

</body>

</html>

function Car(seats, engine, radio) {

    this.seats = seats;

    this.engine = engine;

    this.radio = radio;

}

let work\_car = new Car("cloth", "V-6", "Tape Deck");

for (let prop\_name in work\_car) {

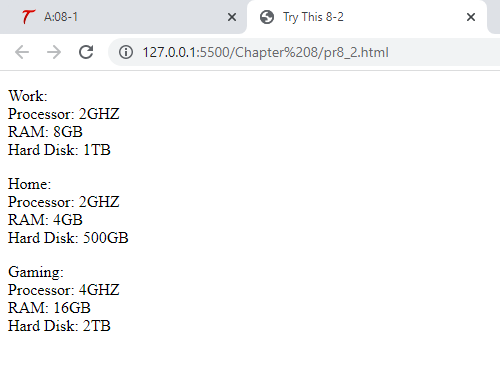
    if (work\_car.hasOwnProperty(prop\_name)) {

        document.write(prop\_name + ": " + work\_car[prop\_name] + "<br>");

    }

}

**Page 212 Try This 8-2**

****

<!DOCTYPE html>

<html>

<head>

    <title>Try This 8-2</title>

</head>

<body>

    <script src="prjs8\_2.js"></script>

</body>

</html>

function Computer(type, processor, ram, hd) {

    this.type = type;

    this.processor = processor;

    this.ram = ram;

    this.hd = hd;

}

Computer.prototype.describe = function() {

    document.write("<p>" + this.type + ":<br>");

    document.write("Processor: " + this.processor + "<br>");

    document.write("RAM: " + this.ram + "<br>");

    document.write("Hard Disk: " + this.hd + "</p>");

};

let work\_computer = new Computer("Work", "2GHZ", "8GB", "1TB");

let home\_computer = new Computer("Home", "2GHZ", "4GB", "500GB");

let gaming\_computer = new Computer("Gaming", "4GHZ", "16GB", "2TB");

work\_computer.describe();

home\_computer.describe();

gaming\_computer.describe();