Day-14

JAVASCRIPT OBJECTS

What is an Object?

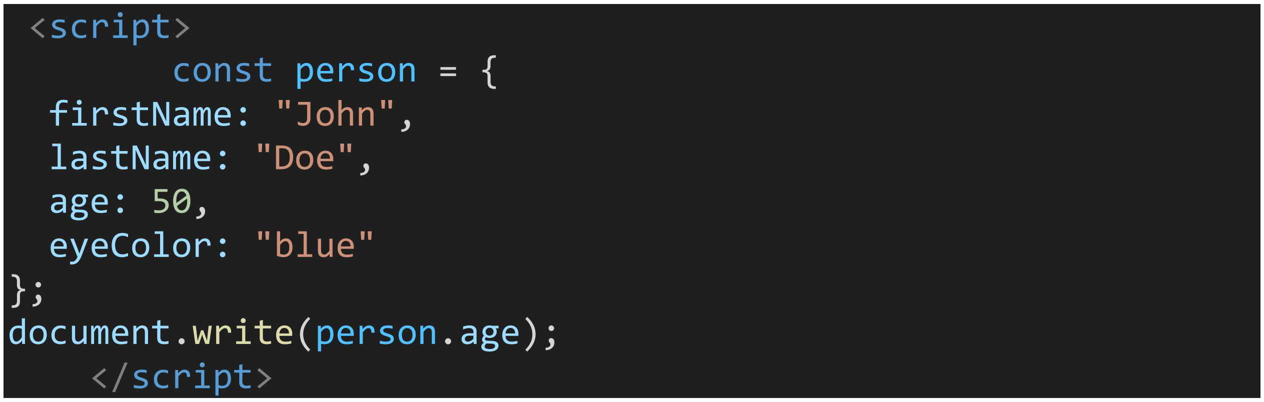
- An **object** is a variable that stores **multiple values**.
- Values are stored in **key : value** pairs.
- Keys = **properties**
- Values = can be **primitive**, **function**, or **another object**

Example:

const car = { type: "Fiat", model: "500", color: "white" };

How to Create an Object

Object Literal



```
 <script>
        const person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};
document.write(person.age);
    </script>
```

Using new Object()

(Not recommended but possible)

```
 <script>
     const person = new Object({
  firstName: "John",
  lastName: "Doe",
  age: 55
});

document.write(person.age);
    </script>
```

Accessing Object Properties

1. Dot Notation

person.lastName;

2. Bracket Notation

person["lastName"];

Example

```
<script>
     const person = new Object({
  firstName: "John",
  lastName: "Doe",
  age: 55
});

document.write(person.firstName + " is " + person.age + " 
years old.");
    </script>
```

Adding New Properties

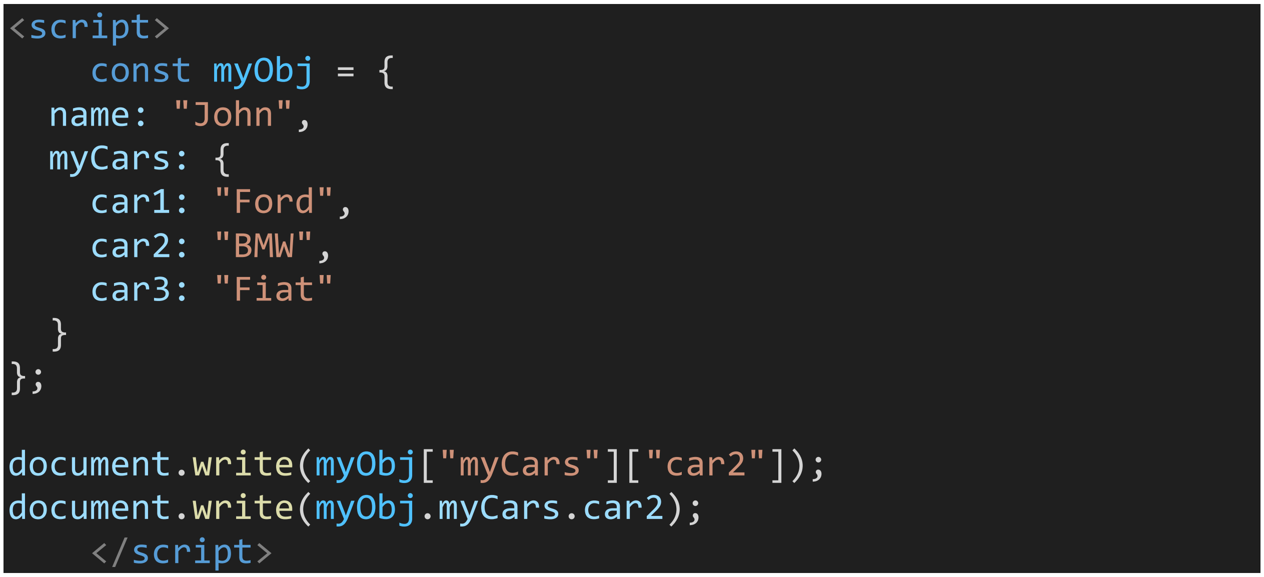
person.nationality = "English";

Deleting Properties

delete person.age;

Nested Objects

Objects inside objects:

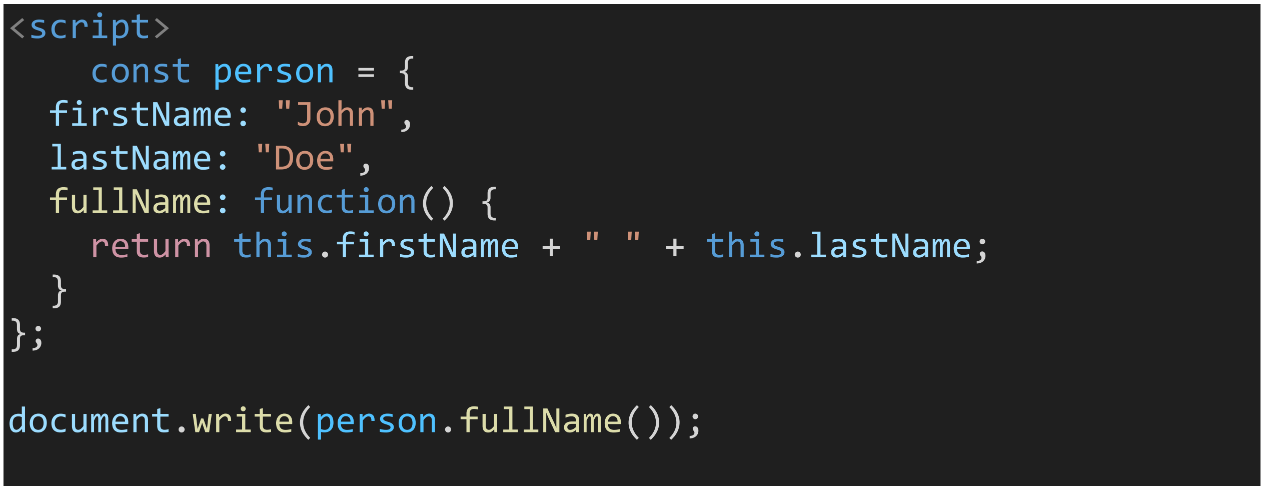


```
<script>
    const myObj = {
  name: "John",
  myCars: {
    car1: "Ford",
    car2: "BMW",
    car3: "Fiat"
  }
};

document.write(myObj["myCars"]["car2"]);
document.write(myObj.myCars.car2);
    </script>
```

JavaScript Object Methods

Method = function stored inside an object.

Example:



```
<script>
    const person = {
  firstName: "John",
  lastName: "Doe",
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
};

document.write(person.fullName());
```

```
    </script>
```

Constructor Function

Used when you want many similar objects.

Example:

```
<script>
    function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}

const father = new Person("John", "Doe", 50, "blue");
const mother = new Person("Sally", "Rally", 48, "green");
document.write(mother.firstName);

    </script>
```

JavaScript Date

1. new Date() → Current date & time

```
<script>
    const d = new Date();
console.log(d);

   </script>
```

2. new Date(dateString)

```
<script>
    const d1 = new Date("2022-03-25");
const d2 = new Date("October 13, 2014 11:13:00");
```

```
console.log(d1);
console.log(d2);

   </script>
```

Method	Returns
getFullYear()	Year
getMonth()	Month (0–11)
getDate()	Day (1–31)
getDay()	Weekday (0–6 → Sun=0)
getHours()	Hours
getMinutes()	Minutes
getSeconds()	Seconds
getMilliseconds()	Milliseconds
getTime()	Milliseconds since 1970

```
<script>
    const d = new Date();

console.log(d.getFullYear());
console.log(d.getMonth());
console.log(d.getDate());
console.log(d.getDay());
console.log(d.getHours());
console.log(d.getMinutes());
console.log(d.getSeconds());
console.log(d.getMilliseconds());
console.log(d.getTime());

   </script>
```

JavaScript Arrays

What is an Array?

An array is a **special object** used to store multiple values in one variable.

Features:

- Ordered values
- Zero–indexed
- Dynamic size
- Can store mixed data (numbers, strings, objects, functions, arrays)

Example:

const cars = ["Saab", "Volvo", "BMW"];

---

let car1 = "Saab";

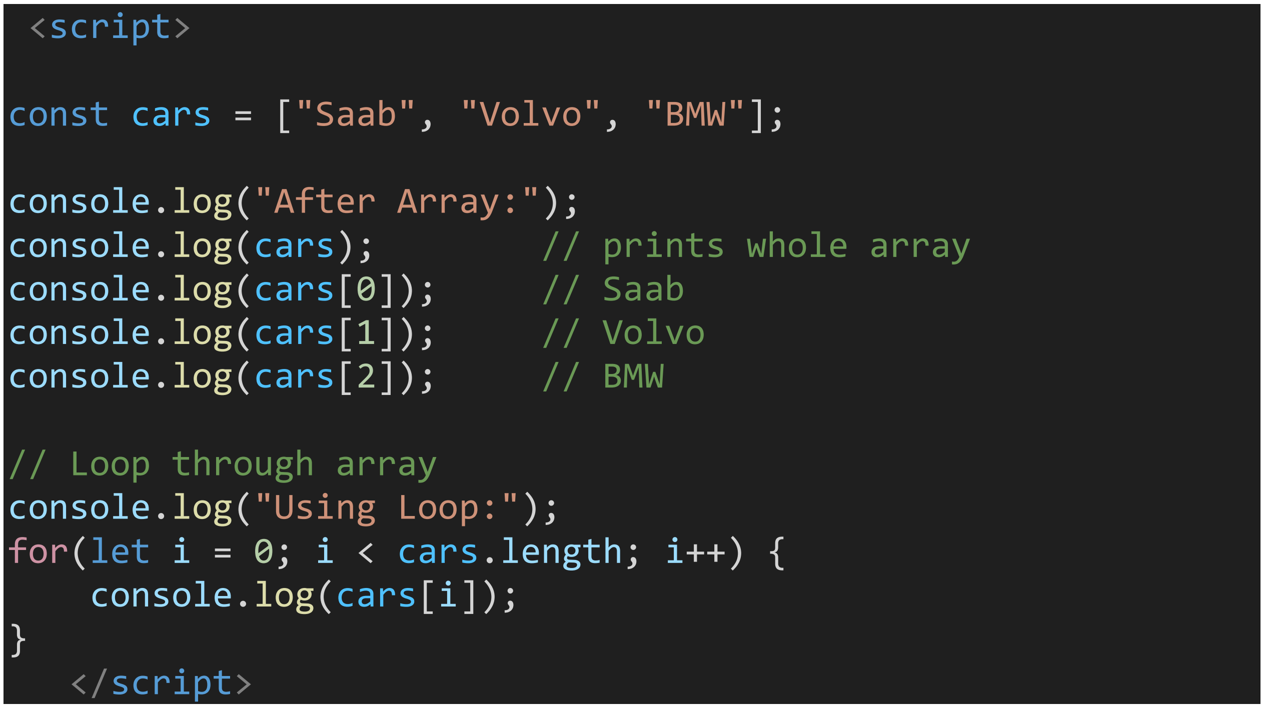
let car2 = "Volvo";

let car3 = "BMW";

Use:

const cars = ["Saab", "Volvo", "BMW"];

Easier to loop and manage.



```
 <script>

const cars = ["Saab", "Volvo", "BMW"];

console.log("After Array:");
console.log(cars);          // prints whole array
console.log(cars[0]);       // Saab
console.log(cars[1]);       // Volvo
console.log(cars[2]);       // BMW

// Loop through array
console.log("Using Loop:");
for(let i = 0; i < cars.length; i++) {
    console.log(cars[i]);
}
   </script>
```

Accessing Array Elements

By index:

const cars = ["Saab", "Volvo", "BMW"];

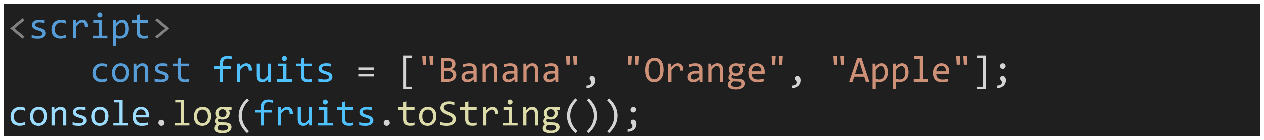
console.log(cars[0]); // Saab

Change element:

cars[0] = "Opel";

Array to String

toString()



```
<script>
    const fruits = ["Banana", "Orange", "Apple"];
console.log(fruits.toString());
```

```
  </script>
```

// Output: Banana,Orange,Apple

Arrays vs Objects

Array:

```
const person = ["John", "Doe", 46];

console.log(person[0]);  // John
```

Object:

```
const person = {firstName:"John", lastName:"Doe", age:46};

console.log(person.firstName); // John
```

8□Mixed Data in Arrays

```
const myArray = [

  Date.now,

  function () { return "Hello"; },

  ["Saab", "BMW"]

];
```

9□Array Properties & Methods

✓ Length

```
const fruits = ["Banana", "Orange", "Apple"];

console.log(fruits.length);  // 3
```

✔ First element

```
console.log(fruits[0]);
```

✔ Last element

```
console.log(fruits[fruits.length - 1]);
```

🔟 Looping Through Arrays

✔ For loop

```
const fruits = ["Banana", "Orange", "Apple"];


for (let i = 0; i < fruits.length; i++) {
  console.log(fruits[i]);
}
```

✔ forEach()

```
fruits.forEach(function(value) {
  console.log(value);
});
```

1️⃣1️⃣Adding Elements

✔ push()

```
fruits.push("Mango");
```

✔ Using length

```
fruits[fruits.length] = "Guava";
```

⚠ Incorrect Example (creates holes)

```
fruits[7] = "Kiwi";
```

---

12 Associative Arrays ✕ (JavaScript does NOT support)

✕ Wrong:

```
const person = [];
person["name"] = "John";  // becomes object
```

✓ Correct:

```
const person = {name:"John"};
```

---

13 new Array() Issues

```
const a = new Array(40);
// Creates array with 40 empty slots
```

But:

```
const a = [40];
// Array with one element
```

---

14 How to Check if Something is an Array

✓ Array.isArray()

```
console.log(Array.isArray(fruits));   // true
```

✓ instanceof

```
console.log(fruits instanceof Array); // true
```

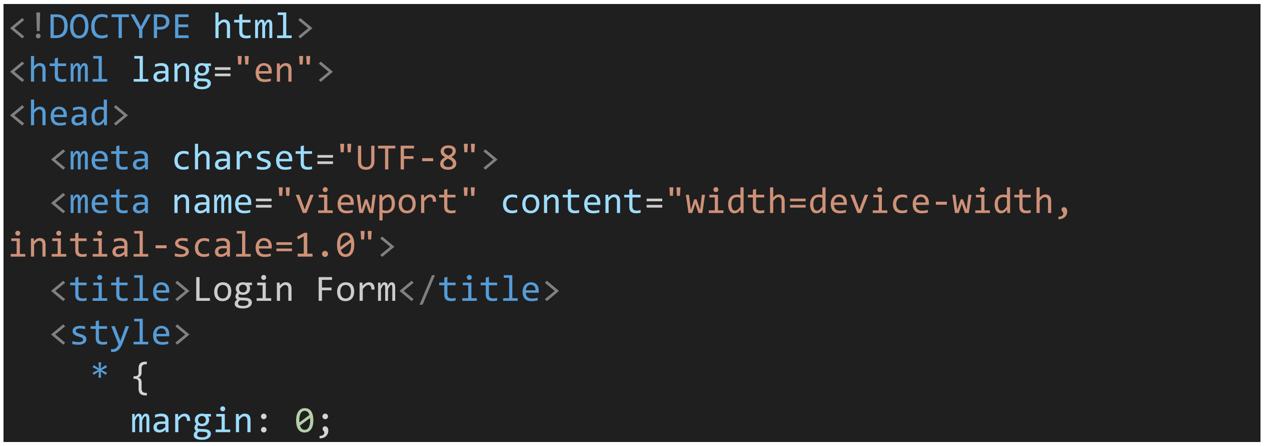
---

15 Nested Arrays and Objects

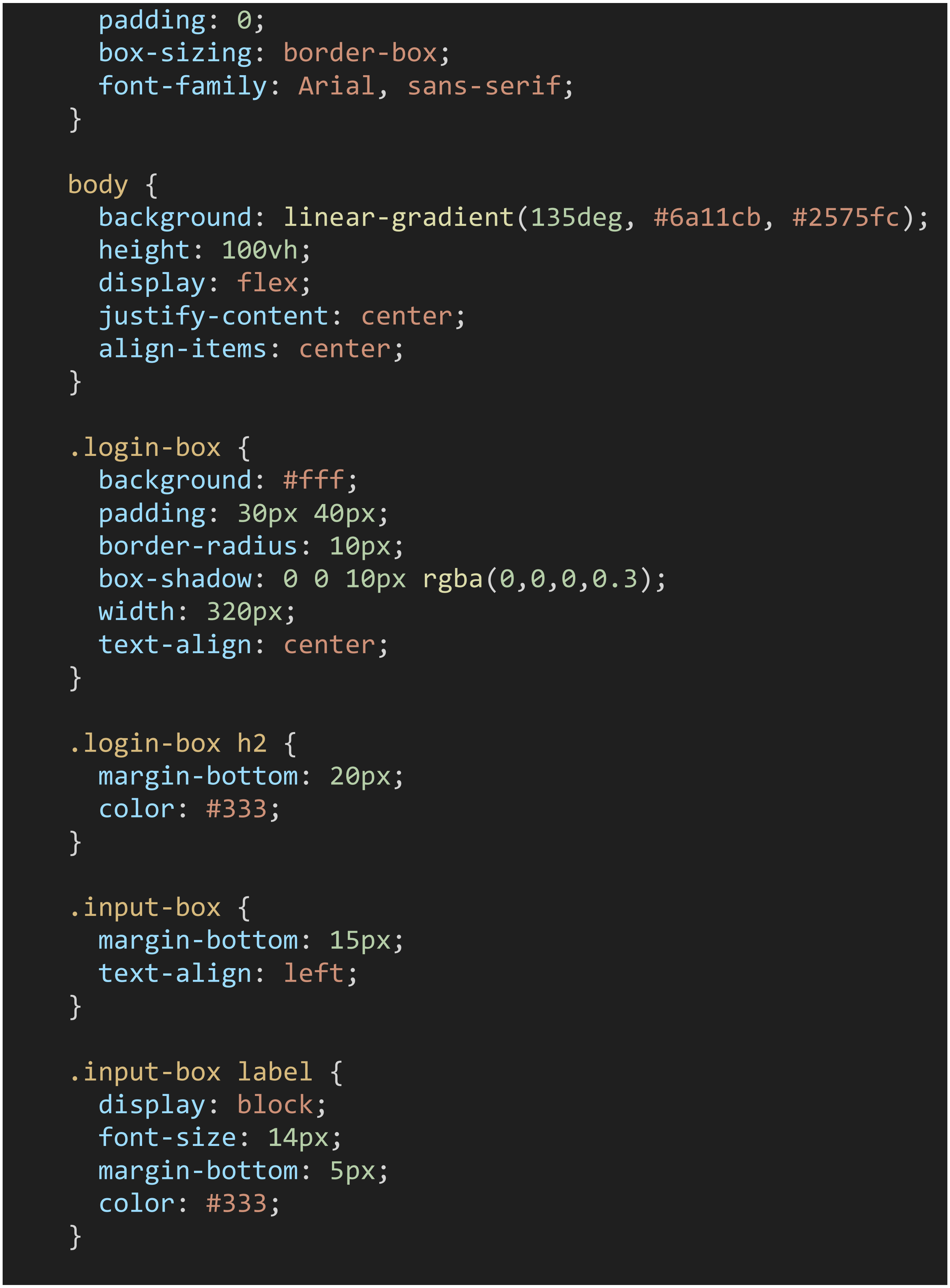
```
const myObj = {
  name: "John",
  age: 30,
  cars: [
    {name:"Ford", models:["Fiesta", "Focus"]},
    {name:"BMW", models:["320", "X5"]},
  ]
};
```

Access nested data:

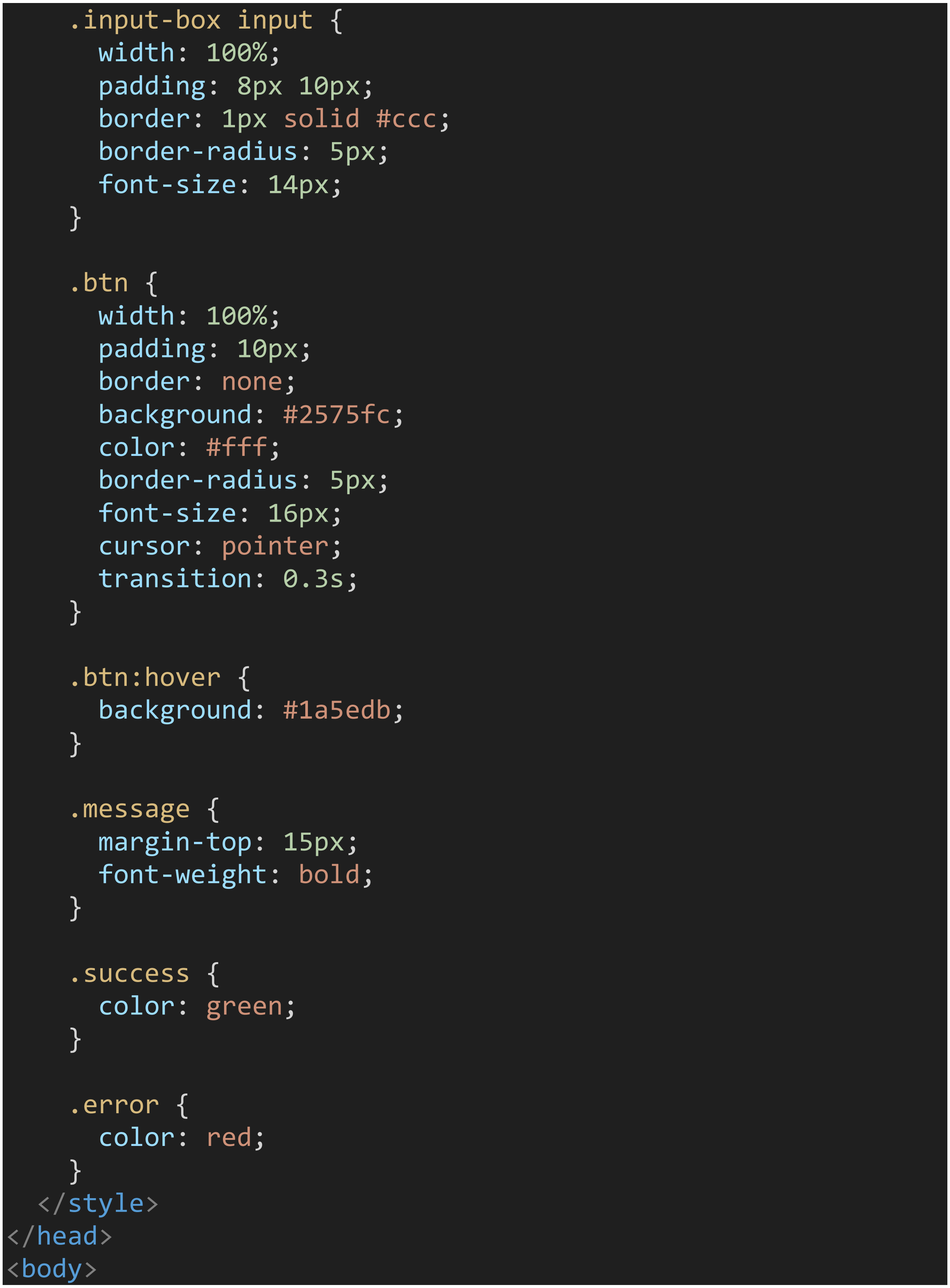
```
for (let i in myObj.cars) {
  console.log(myObj.cars[i].name);
  for (let j in myObj.cars[i].models) {
    console.log(" - " + myObj.cars[i].models[j]);
  }
}
```



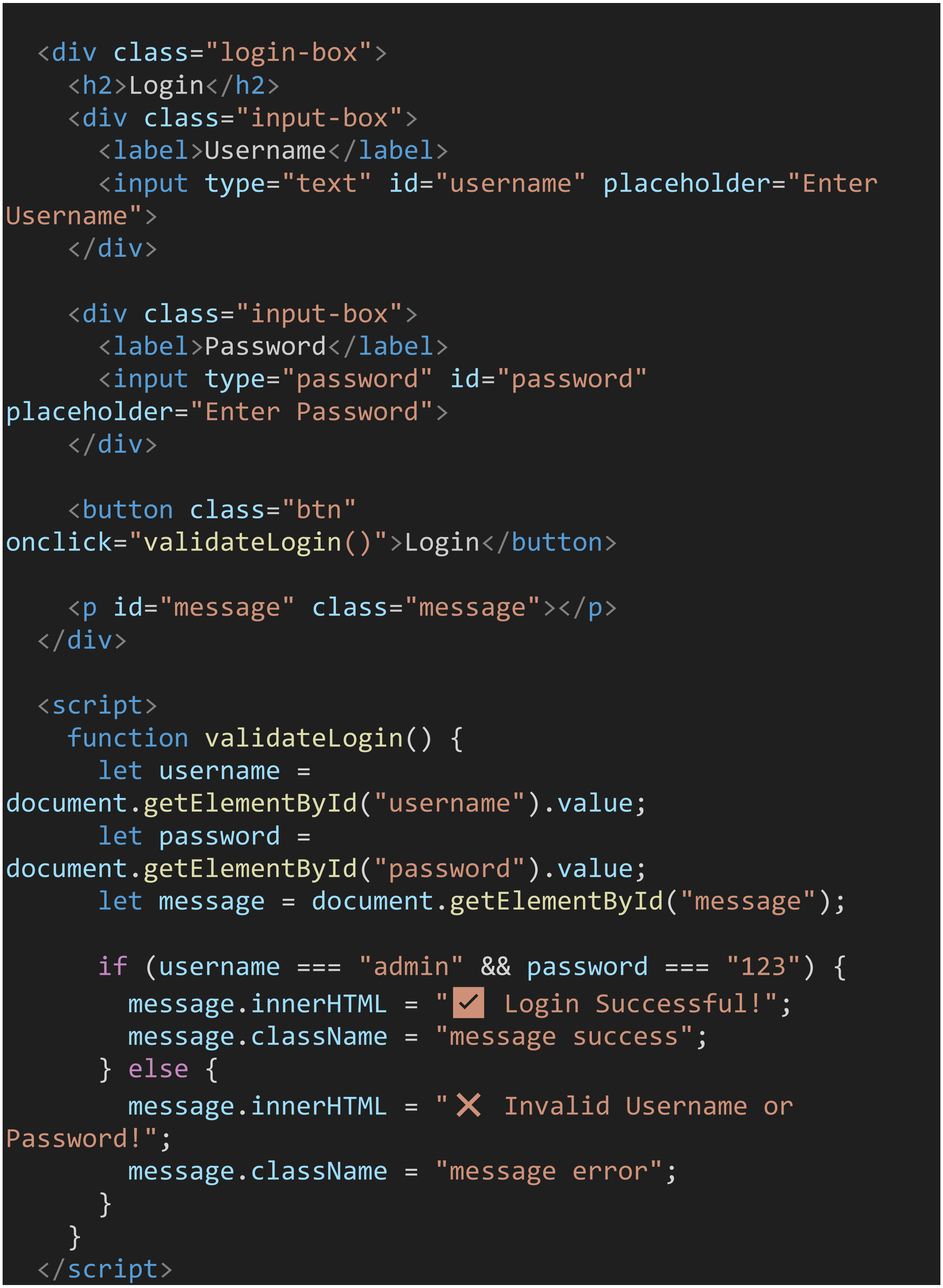
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Login Form</title>
  <style>
    * {
      margin: 0;
```



```css
      padding: 0;
      box-sizing: border-box;
      font-family: Arial, sans-serif;
    }

    body {
      background: linear-gradient(135deg, #6a11cb, #2575fc);
      height: 100vh;
      display: flex;
      justify-content: center;
      align-items: center;
    }

    .login-box {
      background: #fff;
      padding: 30px 40px;
      border-radius: 10px;
      box-shadow: 0 0 10px rgba(0,0,0,0.3);
      width: 320px;
      text-align: center;
    }

    .login-box h2 {
      margin-bottom: 20px;
      color: #333;
    }

    .input-box {
      margin-bottom: 15px;
      text-align: left;
    }

    .input-box label {
      display: block;
      font-size: 14px;
      margin-bottom: 5px;
      color: #333;
    }
```



```
    .input-box input {
      width: 100%;
      padding: 8px 10px;
      border: 1px solid #ccc;
      border-radius: 5px;
      font-size: 14px;
    }

    .btn {
      width: 100%;
      padding: 10px;
      border: none;
      background: #2575fc;
      color: #fff;
      border-radius: 5px;
      font-size: 16px;
      cursor: pointer;
      transition: 0.3s;
    }

    .btn:hover {
      background: #1a5edb;
    }

    .message {
      margin-top: 15px;
      font-weight: bold;
    }

    .success {
      color: green;
    }

    .error {
      color: red;
    }
  </style>
</head>
<body>
```



```
  <div class="login-box">
    <h2>Login</h2>
    <div class="input-box">
      <label>Username</label>
      <input type="text" id="username" placeholder="Enter
Username">
    </div>

    <div class="input-box">
      <label>Password</label>
      <input type="password" id="password"
placeholder="Enter Password">
    </div>

    <button class="btn"
onclick="validateLogin()">Login</button>

    <p id="message" class="message"></p>
  </div>

  <script>
    function validateLogin() {
      let username =
document.getElementById("username").value;
      let password =
document.getElementById("password").value;
      let message = document.getElementById("message");

      if (username === "admin" && password === "123") {
        message.innerHTML = "✅ Login Successful!";
        message.className = "message success";
      } else {
        message.innerHTML = "❌ Invalid Username or
Password!";
        message.className = "message error";
      }
    }
  </script>
```

```
</body>
</html>
```