Course Module 1: Course Unit 6: JavaScript (Objects and Functions)

Week 8

JavaScript

JavaScript

JavaScript is an Object Oriented Programming (OOP) language.

4 basic capabilities:

- Encapsulation
- Aggregation
- Inheritance
- Polymorphism

JavaScript Objects


00-020-MM

Object	Properties	Methods
	car.name = Fiat	car.start()
	car.model = 500	car.drive()
	car.weight = 850kg	car.brake()
	car.color = white	car.stop()

JavaScript Objects

> Objects are variables too. But objects can contain many values.
Object values are written as name: value pairs (name and value separated by a colon).
The named values, in JavaScript objects, are called properties.

> JavaScript objects are containers for named values, called properties and methods.
Methods are actions that can be performed on objects.

JavaScript Objects

- It is just a collection of named values. These named values are usually referred to as properties of the object.
- It is an entity having state and behavior (properties and method).

JavaScript Objects

In JavaScript, almost "everything" is an object.

- Booleans can be objects (if defined with the new keyword)
- Numbers can be objects (if defined with the new keyword)
- Strings can be objects (if defined with the new keyword)
- Dates are always objects
- Maths are always objects
- Regular expressions are always objects
- Arrays are always objects
- Functions are always objects
- Objects are always objects
- All JavaScript values, except primitives, are objects.

JavaScript Primitives

JavaScript Primitives

A primitive value is a value that has no properties or methods.
A primitive data type is data that has a primitive value.
JavaScript defines 7 types of primitive data types:

- string
- number
- boolean
- null
- undefined
- symbol
- bigint

Primitive values are immutable (they are hardcoded and cannot be changed).

JavaScript Primitives

Value	Type	Comment
"Hello"	string	"Hello" is always "Hello"
3.14	number	3.14 is always 3.14
true	boolean	true is always true
false	boolean	false is always false
null	null (object)	null is always null
undefined	undefined	undefined is always undefined

Creating a JavaScript Object

Creating a JavaScript Object

There are different ways to create new objects:

- Create a single object, using an object literal.
- Create a single object, with the keyword new.
- Define an object constructor, and then create objects of the constructed type.
- Create an object using Object.create().

using an object literal

Using an object literal, you both define and create an object in one statement.
An object literal is a list of name:value pairs (like age:50) inside curly braces {}.

Example

```
const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

using an object literal

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Objects</h2>
<p>Creating a JavaScript Object:</p>

<p id="demo"></p>

<script>
const person = {firstName:"John", lastName:"Doe",
age:50,eyeColor:"blue"};

document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years
old.";
</script>

</body>
</html>
```

creates a new JavaScript object with four properties

JavaScript Objects

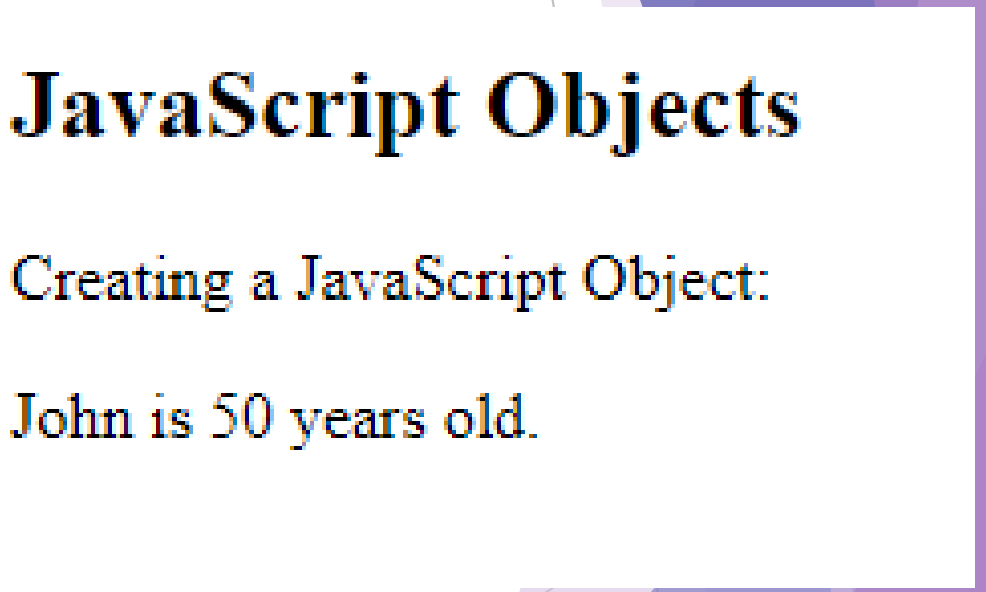
Creating a JavaScript Object:

John is 50 years old.

using an object literal

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Objects</h2>
<p>Creating a JavaScript Object:</p>

<p id="demo"></p>

<script>
const person = {
  firstName: "John",
  lastName: "Doe",
  age: 50,
  eyeColor: "blue"
};

document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years
old.";
</script>

</body>
</html>
```

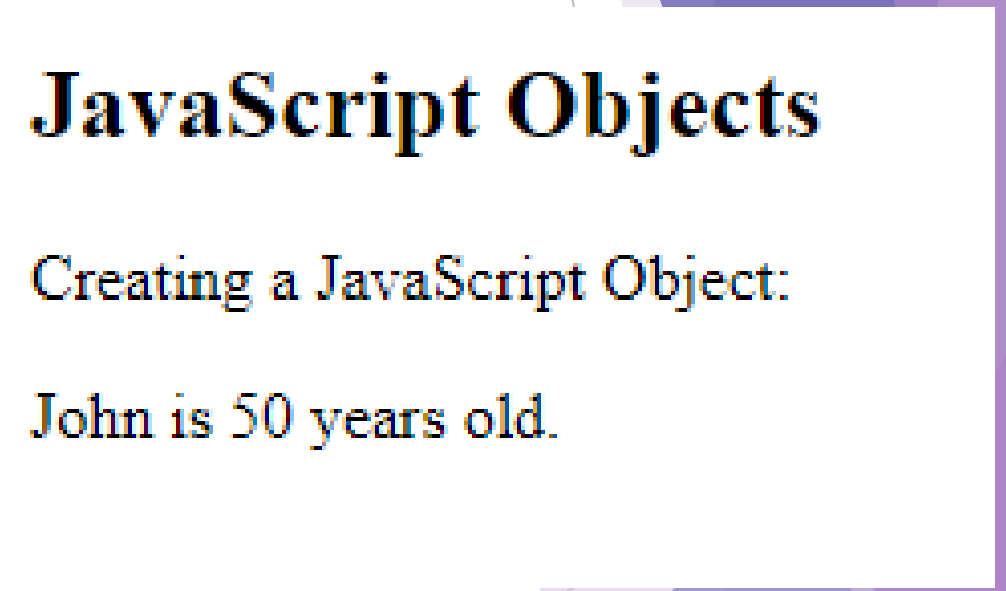
*Spaces and line breaks are not important.
An object definition can span multiple lines*



using an object literal

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Objects</h2>
<p>Creating a JavaScript Object:</p>

<p id="demo"></p>

<script>
const person = {};
person.firstName = "John";
person.lastName = "Doe";
person.age = 50;
person.eyeColor = "blue";

document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years
old.";
</script>

</body>
</html>
```

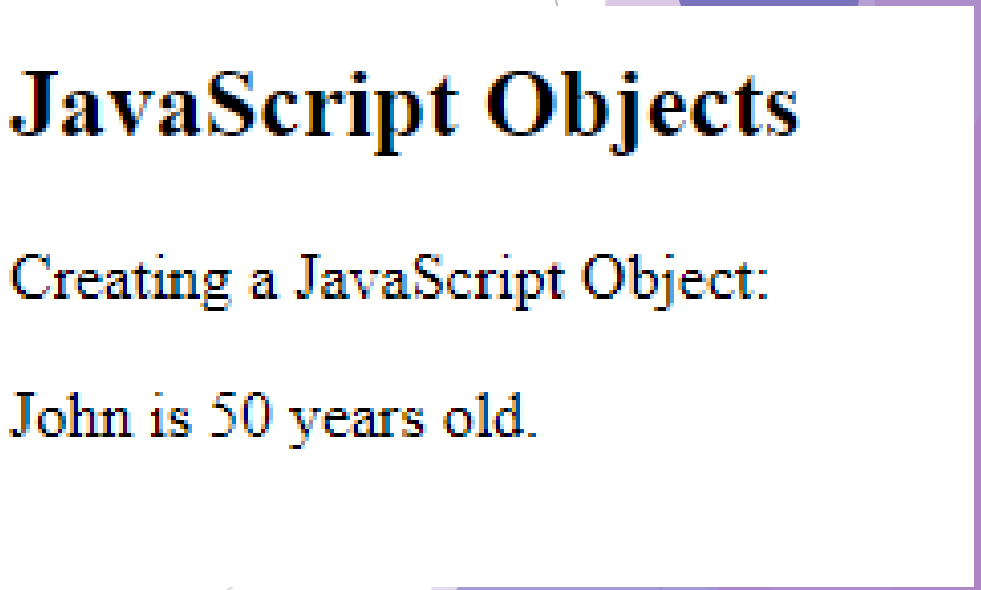
creates an empty JavaScript object, and then adds 4 properties



Using the JavaScript Keyword new

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Objects</h2>
<p>Creating a JavaScript Object:</p>

<p id="demo"></p>

<script>
const person = new Object();
person.firstName = "John";
person.lastName = "Doe";
person.age = 50;
person.eyeColor = "blue";

document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years
old.";
</script>

</body>
</html>
```

create a new JavaScript object using `new Object()`, and then adds 4 properties:



JavaScript Objects are Mutable

Objects are mutable: They are addressed by reference, not by value.

If person is an object, the following statement will not create a copy of person:

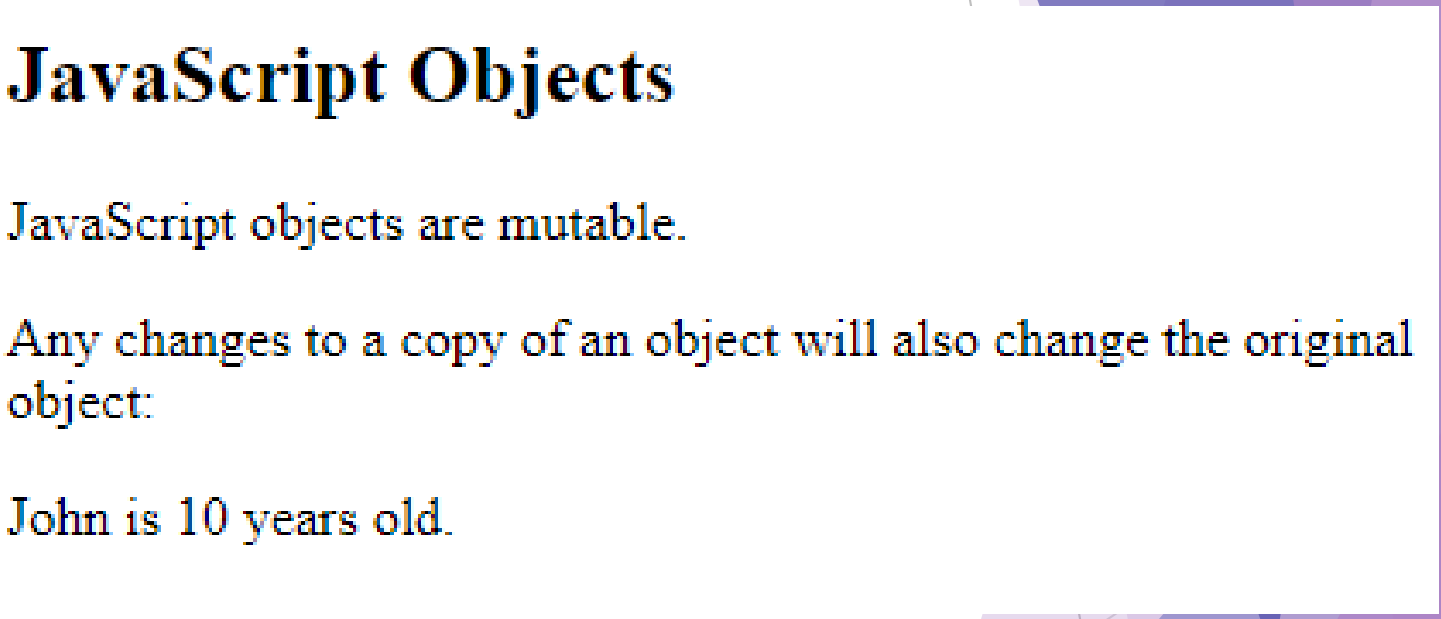
```
const x = person;  // Will not create a copy of person.
```

The object x is **not a copy** of person. It **is** person. Both x and person are the same object.

Any changes to x will also change person, because x and person are the same object.

JavaScript Objects are Mutable

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Objects</h2>
<p>JavaScript objects are mutable.</p>
<p>Any changes to a copy of an object will also change
the original object:</p>

<p id="demo"></p>

<script>
const person = {
  firstName: "John",
  lastName: "Doe",
  age:50,
  eyeColor: "blue"
};

const x = person;
x.age = 10;

document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years
old.";
</script>

</body>
</html>
```



Applications???

Any questions???

REFERENCES:

- JavaScript basics. (2022). https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics
- JavaScript Tutorial. (2022). https://www.javascripttutorial.net/
- JavaScript Tutorial. (n.d.). https://www.javatpoint.com/javascript-tutorial
- JavaScript Tutorial. (n.d.). https://www.tutorialrepublic.com/javascript-tutorial/
- JavaScript Tutorial. (n.d.). https://www.tutorialspoint.com/javascript/index.htm
- JavaScript Tutorial. (n.d.). https://www.w3schools.com/js/default.asp