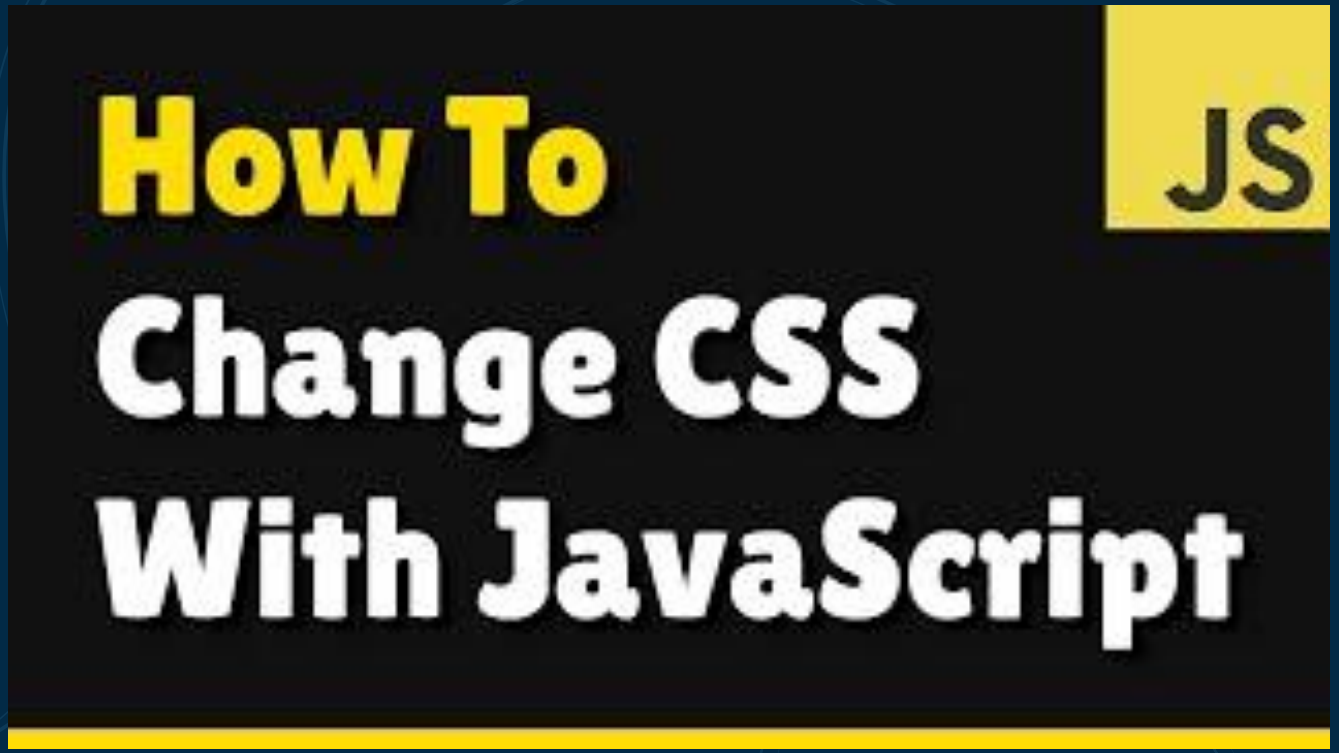
<Semester II>

</Programming for Interactive Interfaces/>

Week 9

Changing Style through JavaScript



JavaScript Objects

- **An object in JavaScript is a data structure used to store related data collections.**
- It stores data as key-value pairs, where each key is a unique identifier for the associated value.
- Objects are dynamic, which means the properties can be added, modified, or deleted at runtime.
- An object in JavaScript is a collection of key-value pairs, where keys are strings (properties) and values can be any data type.
- Objects can be created using object literals, constructors, or classes. Properties are defined with key-value pairs, and methods are functions defined within the object

JavaScript Objects

There are two primary ways to create an object in JavaScript: Object Literal and Object Constructor.

1. Creation Using Object Literal - The object literal syntax allows you to define and initialize an object with curly braces {}, setting properties as key-value pairs.

```
let obj = {
    name: "Sourav",
    age: 23,
    job: "Developer"
};
  console.log(obj);
```

JavaScript Objects

2. Creation Using new Object() Constructor

```
let obj = new Object();
obj.name= "Sourav",
obj.age= 23,
obj.job= "Developer"
console.log(obj);
```

You can access an object’s properties using either dot notation or bracket notation.

```
let obj = { name: "Sourav", age: 23 };
// Using Dot Notation
console.log(obj.name);
// Using Bracket Notation
console.log(obj["age"]);
```

JavaScript Objects

You can dynamically add new properties to an object using dot or bracket notation.

The delete operator removes properties from an object.

Syntax: delete obj.color;

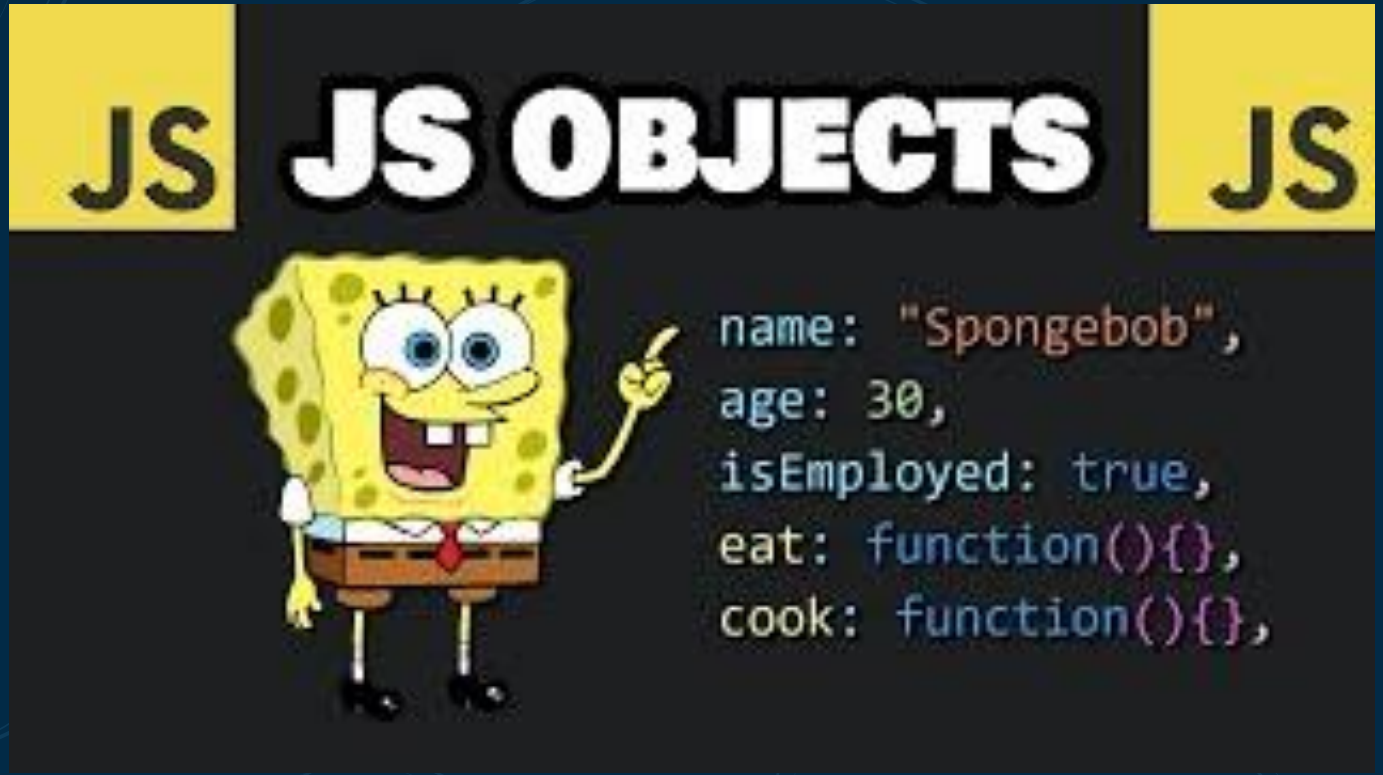
You can check if an object has a property using the in operator or hasOwnProperty() method.

```
let obj = { model: "Tesla" };

console.log("color" in obj);

  console.log(obj.hasOwnProperty("model"));
```

Use for…in loop to iterate through the properties of an object.

```
for (let key in obj) {

  }
```

JavaScript Objects



JavaScript Events

- JavaScript Events are actions or occurrences that happen in the browser. They can be triggered by various user interactions or by the browser itself. When JavaScript is used in HTML pages, JavaScript can "react" on these events.
- <button onclick="myFun()">Click me</button>
- There many types of events in JavaScript: Browser level events and DOM level events.
- Browser level events: load, resize, scroll.
- DOM level events: Click, mouseover, drag, form-based events
- The change in the state of an object is known as an Event.
- This process of reacting over the events is called Event Handling. js handles the HTML events via **Event Handlers.**

JavaScript Mouse Event Handlers

Event Performed	Event Handler	Description
click	onclick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

Homework: Continue the HTML + CSS portion of your LCA2 project. If design is still left, complete it and create a working prototype/storyboard.

W3Schools JavaScript Tutorial

JavaScript for Web | MDN

Reference

1. JavaScript Objects - W3Schools
2. Objects in Javascript - Syntax and Operations | GeeksforGeeks
3. Introduction to Events -| MDN
4. JavaScript Events | GeeksforGeeks
5. Event Handlers in JavaScript - Tpoint Tech
6. Event handling (overview) - Event reference | MDN
7. Mouse Events - W3Schools

Fonts & colors used

This presentation has been made using the following fonts:

Blinker
(https://fonts.google.com/specimen/Blinker)

Inconsolata
(https://fonts.google.com/specimen/Inconsolata)