<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

The “this” keyword

The this keyword refers to the context where a piece of code, such as a function's body, is supposed to run. Typically, it is used in object methods, where this refers to the object that the method is attached to, thus allowing the same method to be reused on different objects.

The value of this in JavaScript depends on how a function is invoked (runtime binding), not how it is defined. When a regular function is invoked as a method of an object (obj.method()), this points to that object. When invoked as a standalone function (not attached to an object: func()), this typically refers to the global object (in non-strict mode).

Eg. this.job = person1_data.job, where job is a property of person1_data.job object

The “this” keyword

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.

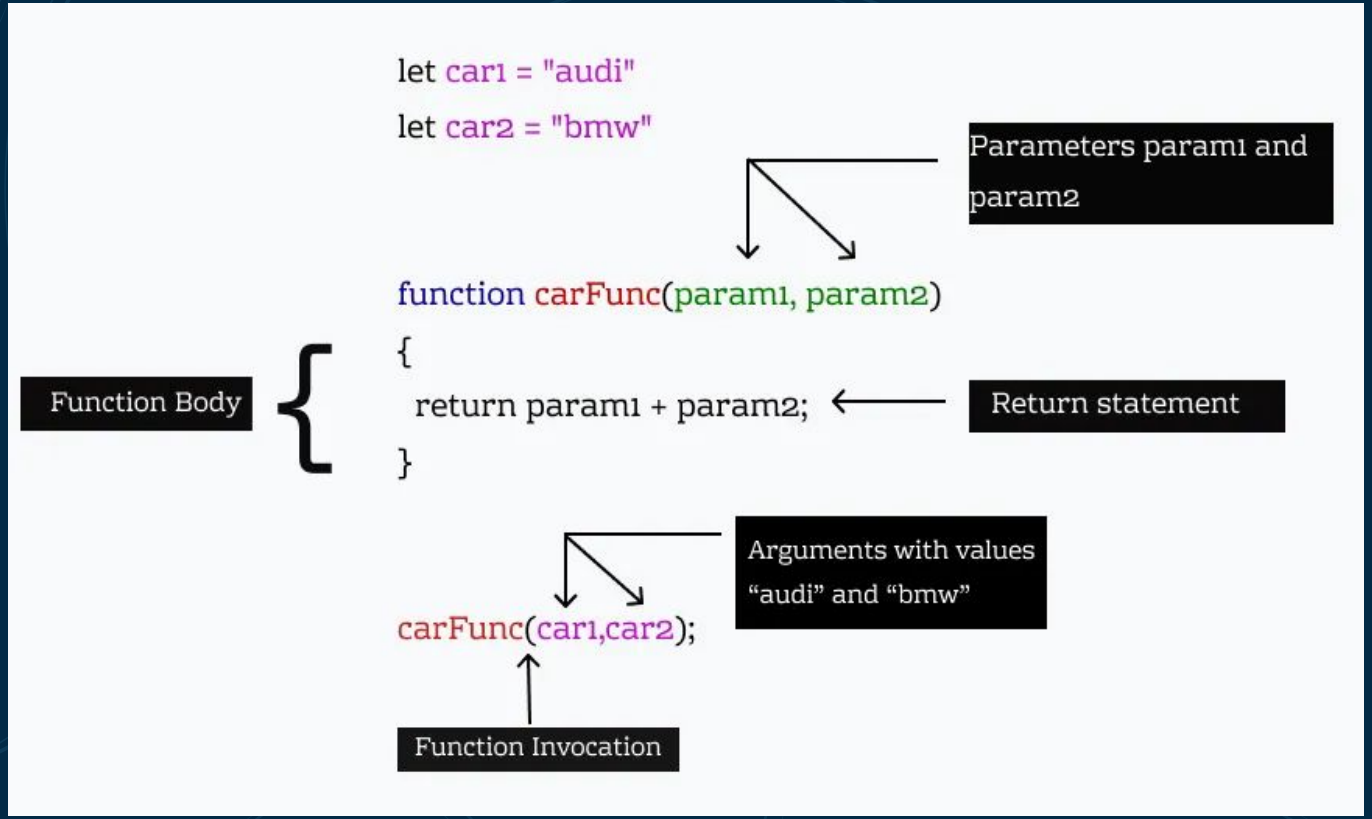
JavaScript Functions

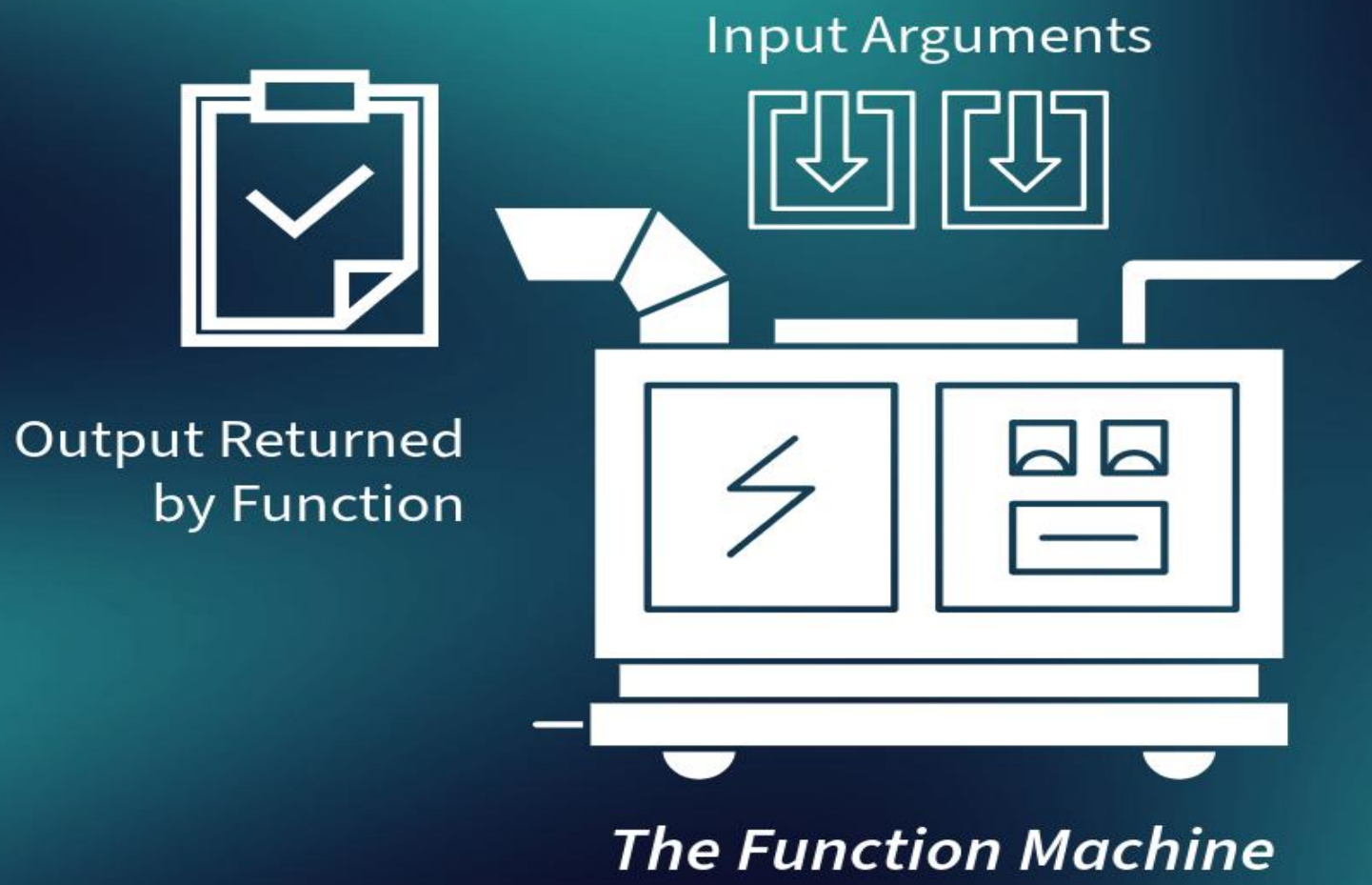
- A JavaScript function/method is a block of code designed to perform a particular task.A JavaScript function is executed when "something" invokes it (calls it).
- **There are some predefined functions in JavaScript, and your can create your own custom functions also.**
- A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ().
- Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).
- The parentheses may include parameter names separated by commas: (*parameter1, parameter2, ...*) The code to be executed, by the function, is placed inside curly brackets: {}
- function *name*(*parameter1, parameter2, parameter3*) {

  // code to be executed

  }
- **Custom Functions are first defined then called when you want to execute it**

Structure of a JavaScript Function




Input Arguments
Output Returned by Function
The Function Machine

Arrow Functions

An arrow function is a shorter syntax for writing functions in JavaScript. Introduced in ES6, arrow functions allow for a more concise and readable code, especially in cases of small functions.

- Arrow functions are written with the => symbol, which makes them compact.
- They don’t have their own this. They inherit this from the surrounding context.
- For functions with a single expression, the return is implicit, making the code more concise.
- Arrow functions do not have access to the arguments object, which is available in regular functions.

Eg:
```
const add = (a, b) => a + b;
console.log(add(5, 3));
```


#46
How to use
Arrow Functions
JS

Homework: Continue the HTML + CSS portion of your LCA2 project.

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
8. JavaScript Arrow Function - W3Schools
9. Arrow functions in JavaScript | GeeksforGeeks

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)