Front-End Lecture (JavaScript) - Session 8

Introduction to JavaScript

Introduction to JavaScript

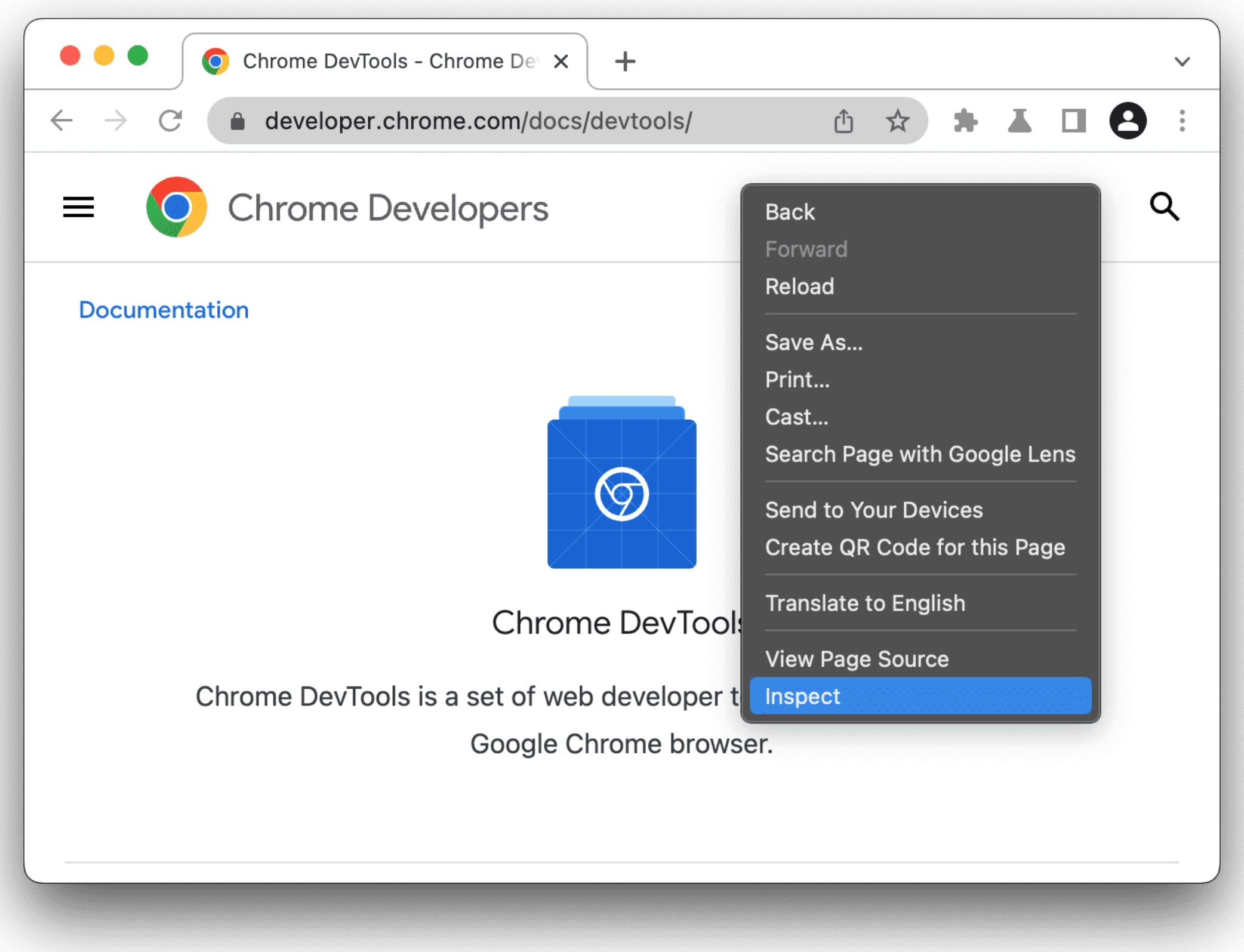
Introduction And What Is JavaScript?

JavaScript is a scripting language used to create dynamic and interactive web pages.

```
console.log("Hello, World!");
```

Work With Chrome Developer Tools

Use Chrome DevTools to debug and test JavaScript code.



Where To Put The Code

JavaScript can be placed in `<script>` tags in HTML or in external `.js` files.

```
<header>
    <script src="script.js"></script>
</header>
<body>
    <script src="script.js"></script>
</body>
```

Comments

Use `//` for single-line comments and `/* */` for multi-line comments. Avoid bad practices like inline scripts.

```
// This is a single-line comment
/* This is a
   multi-line comment */
```

Output To Screen

Use `document.write()`, `alert()`, or `console.log()` to output data.

```
window.alert("This is an alert");
document.write("Hello, World!");
console.log("Hello, World!");
```

Console Methods And Styling And Web API

Use `console.log()`, `console.error()`, and `console.warn()` for debugging. Apply CSS styling in the console.

```
console.log("%cStyled Message", "color: blue; font-size: 20px;");
```

What Is ECMAScript?

ECMAScript is the standard behind JavaScript. ES6 (ES2015) introduced many modern features.

Data Types And typeof Operator

JavaScript has primitive data types like `string`, `number`, `boolean`, and `object`. Use `typeof` to check the type.

```
console.log(typeof "Hello"); // "string"
console.log(typeof 42); // "number"
console.log(typeof true); // "boolean"
console.log(typeof { name: "Saja" }); // Array => "object"
console.log(typeof undefined); // "undefined"
console.log(typeof null); // "object"
console.log(typeof BigInt("9999999999999999")); // "bigint"
console.log(typeof Symbol()); // "symbol"
```

Variables Introduction

Variables store data. Use `var`, `let`, or `const` to declare them.

```
  keyword  variableName  assignmentOperator  VariableValue;
    let         name              =              "Saja"   ;
```

Identifiers Name Conventions And Rules

Variable names can starting with a letter, `_`, or `$`. Use camelCase for naming.

```
let userName = "SajaMalek";
```

Var, Let, Const Compare

`var` is function-scoped, `let` and `const` are block-scoped. `const` cannot be reassigned.

```
var x = 10;
let y = 20;
const z = 30;
```

String Syntax And Character Escape Sequences

Strings can be enclosed in `""` or `''`. Use escape sequences like `\n` for new lines.

```
let str = "Hello\nWorld";
```

Concatenation

Combine strings using the `+` operator.

```
let fullName = "Saja" + " " + "Malek";
```

Template Literals (Template Strings)

Use backticks (` `` `) for template literals to embed expressions.

```
let name = "Saja";
console.log(`Hello, ${name}!`);
```

Variables And Concatenation Challenge

```
                     [1] Create 3 Variables [Title, Description, Date]
                     -- All In One Statement
                     -- Variable Name Must Be Two Words
                     -- Title Content Is "Elzero"
                     -- Description Content Is "Elzero Web School"
                     -- Date Content Is "25/10"
                     [2] Create Variable Contains Div And This Div Contains
                     -- H3 For Title
                     -- P For Paragraph
                     -- Span For Time
                     [3] Add This Card To Page 4 Times ( You can use "Repeat" )
                     [4] Use Template Literals For Concatenate
```

```
let titleCont = "Elzero", pargrapghCont = "Elzero Web School", dateCont = "25/10";

let card = `
    <div class="card">
        <h3>Hello ${titleCont}</h3>
        <p>${pargrapghCont}</p>
        <span>${dateCont}</span>
    </div>
`;

document.write(card.repeat(4));
```

Arithmetic Operators

Use `+`, `-`, `*`, `/`, and `%` for arithmetic operations.

```
let sum = 10 + 5;
```

Unary Plus And Negation Operators

The unary `+` converts a string to a number, and `-` negates a value.

```
let num = +"10"; // converts to number
let neg = -num; // negates the value
```

Type Coercion

JavaScript automatically converts types in certain operations.

```
let result = "5" + 2; // "52"
```

Assignment Operators

Use `=`, `+=`, `-=`, `*=`, and `/=` for assignments.

```
let x = 10;
x += 5; // x is now 15 => x = x + 5
```

Operators Challenges

```
                     /* Challenge 1 */
                     let a = 10;
                     let b = "20";
                     let c = 80;

                     console.log(++a + +b++ + +c++ - +a++);
                     console.log(++a + -b + +c++ - -a++ + +a);
                     console.log(--c + +b + --a * +b++ - +b * a + --a - +true);

                     /*
                         [++a] [+]
                         [++a]
                         - Value:
                         - Explain:
                         [+]
                         - Explain:
                     */


                     /* Challenge 2 */
                     let d = "-100";
                     let e = "20";
                     let f = 30;
                     let g = true;

                     // Only Use Variables Value
                     // Do Not Use Variable Twice

                     console.log(); // 2000
                     console.log(); // 173
```

```
 /* Challenge 1 */
    let a = 10;
    let b = "20";
    let c = 80;
    console.log(++a + +b++ + +c++ - +a++);
    /*
        [++a]
            Value = 11
            Explain = pre-increment operator

        [+] = "addition operater"

        [+b++]
            Value = 20;
            Explain = first unary plus operator convert the value from string to number,
            then post-increment increase the value by 1 to be 21 but not used

        [+] = "addition operater"

        [+c++]
            Value = 80;
            Explain = unary plus operator has not effect becuase the date type already number,
            then post-increment increase the value by 1 to be 81 but not used

        [-] = subtraction operater

        [+a++]
            Value = 11;
            Explain = Unary plus operator has not effect becuase the date type already number,
            then post-increment increase the value by 1 to be 12 but not used


        so, the total value = 11 + 20 + 80 - 11 = 100

    */


    /*Challenge 2 */
    let d = "-100";
    let e = "20";
    let f = 30;
    let g = true;
    console.log(-d * +e); //2000
    console.log(++e * ++g + -d + ++f); //173
```


Powered by: Saja Malek