



# 1420-7001

By

Dr. Abdul Majeed (조교수)

1<sup>st</sup> Semester, 2025

웹 프로그래밍

Web Programming



IT융합대학 컴퓨터공학부(컴퓨터공학전공)

---

# Part II

# Project Work

# Development Projects-Ideas

1. Word cloud generation
2. Crud application development for class
3. Visual application for data analysis
4. Event website generation (festival)
5. Web-based Grading system
6. Corona possibility calculation
7. Age calculator
8. Shopping website [some data from DB and some hard coded]
9. Sentiment analysis
10. Math quiz game

<https://www.interviewbit.com/blog/web-development-projects/>

# Development Projects-Instructions

- Chose a relevant topic for project from internet or by yourself
- Submit short summary of the project
- Work on the project for about a month
- Present the work done in the last class
  - ▣ Presentation
  - ▣ Demonstration
- Submit the project report on final exam day

# Class Presentations-Instructions

- Chose a relevant topic presentation from the below List
- Prepare Presentation and Demo
- Teach in the class with Example
- Share the material with other students
  - ▣ Presentation
  - ▣ Code
  - ▣ Demonstration Video (if Any)
- Submit the project report on final exam day

# Topics for Class Presentation/Discussion

1. Data transfer/storage and retrieval from the MongoDB
2. A website development and illustration of its key modules
3. JSON data processing and visualization in graphs
4. MVC [Design Patterns] based small-scale website development
5. Sharing a best website/portal and explaining the development procedures
6. Any other useful web programming-related task/demo
7. Developing a website/prototype using ChatGPT
8. Data validation before transferring it to the database
9. Development of a small-scale CRUD application



# Part III

## Course Outline [2<sup>nd</sup> Half of Semester]

# Syllabus for Rest of the Semester-2<sup>nd</sup> Half

---

JS

Basics

Web Programming related  
[visualization, dynamic]

Data validation

Functions and classes etc.

PhP

Basics

Web Programming related [Data  
send, Data receive]  
Database details etc.

AngularJS Overview  
&  
Other related  
concepts

Database

Data organization  
GUI Components etc.

Model view controller discussion.  
Smart app development following  
MVC

Project demonstration  
& Report submission



---

# Part IV

## Regular Lesson

# Summary of the Previous Lesson

## Part I

### Working with forms

- ☐ Simple forms
- ☐ Elements of forms
- ☐ Input element
- ☐ Etc.

## Part II

### Working with JS Basics

- ☐ External
- ☐ Internal
- ☐ Inline



**Note:** Please execute all codes at least once on your computers.



# **JavaScript**

World's most popular  
programming language!!!

# JavaScript Concepts-Why Study JavaScript?

⌘ JavaScript is one of the 3 languages all web developers must learn:

1. HTML to define the content of web pages.
2. CSS to specify the layout of web pages.
3. JavaScript to program the behaviour of web pages.

# JavaScript Concepts-JS Functions & Events

- ⌘ A JavaScript **function** is a block of JavaScript code, that can be executed when "called" for.
- ⌘ For example, a function can be called when an **event** occurs, like when the user clicks a button.
- ⌘ Event can be a click, mouseover, page open, page close, etc.
- ⌘ You will learn much more about functions and events in later lectures.



# Displaying Output in JavaScript

# JavaScript Concepts-Displaying Outputs/Data

JavaScript can "display" data in different ways:

- ⌘ Writing into an HTML element, using `innerHTML`.
- ⌘ Writing into the HTML output using `document.write()`.
- ⌘ Writing into an alert box, using `window.alert()`.
- ⌘ Writing into the browser console, using `console.log()`.

# JavaScript Coding Examples-Displaying Outputs/Data- Inner HTML

- ⌘ To access an HTML element, JavaScript can use the `document.getElementById(id)` method.
- ⌘ The `id` attribute defines the HTML element. The `innerHTML` property defines the HTML content

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

<h2>My First Web Page</h2>
<p>My First Paragraph.</p>

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

<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>

</body>
</html>
```



Output of the code

## My First Web Page

My First Paragraph.

11

Changing the `innerHTML` property of an HTML element is a common way to display data in HTML.



# JavaScript Coding Examples-Displaying Outputs/Data- Document Write

⌘ For testing purposes, it is convenient to use document.write().

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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
document.write(5 + 6);
</script>

</body>
</html>
```



Output of the code

**My First Web Page**

My first paragraph.

11

# JavaScript Coding Examples-Displaying Outputs/Data- Document Write

For testing purposes, it is convenient to use document.write()

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

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<button type="button" onclick="document.write(5 + 6)">Try it</button>

</body>
</html>
```

Output of the code

**My First Web Page**

My first paragraph.

Try it



The document.write() method should only be used for testing.

Using document.write() after an HTML document is loaded, will delete all existing HTML:

# JavaScript Coding Examples-Displaying Outputs/Data- Window Alert

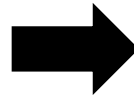
⌘ Alert box can be used to display JavaScript data.

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

<h2>My First Web Page</h2>
<p>My first paragraph.</p>

<script>
window.alert(5 + 6);
</script>

</body>
</html>
```



Output of the code

www.w3schools.com says  
11

OK



**My First Web Page**

My first paragraph.

You can skip the **window** keyword.

# JavaScript Coding Examples-Displaying Outputs/Data- Skip Window Alert

- ⌘ In JavaScript, the window object is the global scope object. This means that variables, properties, and methods by default belong to the window object. This also means that specifying the `window` keyword is optional:

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

<h1>My First Web Page</h1>
<p>My first paragraph.</p>

<script>
alert(5 + 6);
</script>

</body>
</html>
```



Output of the code

www.w3schools.com says  
11

OK



**My First Web Page**

My first paragraph.

You can skip the `window` keyword.

# JavaScript Coding Examples-Displaying Outputs/Data- Console Log

⌘ For debugging purposes, you can call the `console.log()` method in the browser to display data

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

<h2>Activate Debugging</h2>

<p>F12 on your keyboard will activate debugging.</p>
<p>Then select "Console" in the debugger menu.</p>
<p>Then click Run again.</p>

<script>
console.log(5 + 6);
</script>

</body>
</html>
```

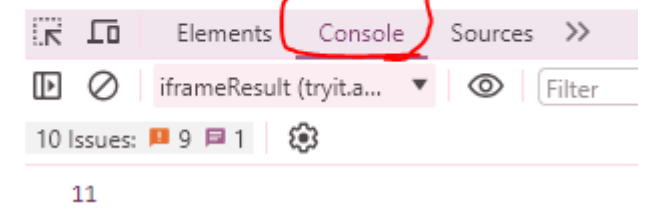
Output of the code

## Activate Debugging

F12 on your keyboard will activate debugging.  
Then select "Console" in the debugger menu.  
Then click Run again.

Inspect

2



# JavaScript Coding Examples-Displaying Outputs/Data- JS Print

- ⌘ JavaScript does not have any print object or print methods.
- ⌘ You cannot access output devices from JavaScript.
- ⌘ The only exception is that you can call the `window.print()` method in the browser to print the content of the current window.

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

<h2>The window.print() Method</h2>

<p>Click the button to print the current page.</p>

<button onclick="window.print()">Print this page</button>

</body>
</html>
```



## The window.print() Method

Click the button to print the current page.

Print this page



4/13/24, 11:14 PM

W3Schools Tryit Editor

### The window.print() Method

Click the button to print the current page.

Print this page

Print

1 page

Destination

Save as PDF

Pages

All

Layout

Portrait

More settings

# JavaScript Coding Examples- Displaying output

- ✓ Writing into an HTML element, using `innerHTML`.
- ✓ Writing into the HTML output using `document.write()`.
- ✓ Writing into an alert box, using `window.alert()`.
- ✓ Writing into the browser console, using `console.log()`.
- ✓ You can call the `window.print()` method in the browser to print the content of the current window.

# JavaScript Coding Examples- Simple Example

- ⌘ Semicolons separate JavaScript statements.
- ⌘ Add a semicolon at the end of each executable statement:

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

<h2>JavaScript Statements</h2>

<p>JavaScript statements are separated by semicolons.</p>

<p id="demo1"></p>

<script>
let a, b, c;
a = 5;
b = 6;
c = a + b;
document.getElementById("demo1").innerHTML = c;
</script>

</body>
</html>
```


Output of the code

## JavaScript Statements

JavaScript statements are separated by semicolons.

11





# Coding Examples

## JavaScript

# JavaScript Coding Examples- Variables

- ⌘ In a programming language, **variables** are used to **store** data values.
- ⌘ JavaScript uses the keywords **var**, **let** and **const** to **declare** variables.
- ⌘ An **equal sign** is used to **assign values** to variables.
- ⌘ In this example, x is defined as a variable. Then, x is assigned (given) the value 6:

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

<h2>JavaScript Variables</h2>

<p>In this example, x is defined as a variable.
Then, x is assigned the value of 6:</p>

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

<script>
let x;
x = 6;
document.getElementById("demo").innerHTML = x;
</script>

</body>
</html>
```

## JavaScript Variables

In this example, x is defined as a variable. Then, x is assigned the value of 6:

6

# JavaScript Coding Examples- Operators

⌘ JavaScript uses arithmetic operators ( + - \* / ) to compute values:

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

<h2>JavaScript Operators</h2>

<p>JavaScript uses arithmetic operators to compute values (just like algebra).</p>

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

<script>
document.getElementById("demo").innerHTML = (5 + 6) * 10;
</script>

</body>
</html>
```

## JavaScript Operators

JavaScript uses arithmetic operators to compute values (just like algebra).

110

Q: What will be the output when we remove brackets in the above code?

# JavaScript Coding Examples- Value Assignment

⌘ JavaScript uses an assignment operator ( = ) to assign values to variables

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>Assigning JavaScript Values</h2>
```

```
<p>In JavaScript the = operator is used to assign values to variables.</p>
```

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

```
<script>
```

```
let x, y;
```

```
x = 5;
```

```
y = 6;
```

```
document.getElementById("demo").innerHTML = x + y;
```

```
</script>
```

```
</body>
```

```
</html>
```

## Assigning JavaScript Values

In JavaScript the = operator is used to assign values to variables.

# JavaScript Coding Examples- Expressions

- ⌘ An expression is a combination of values, variables, and operators, which computes to a value.
- ⌘ The computation is called an evaluation.
- ⌘ For example,  $5 * 10$  evaluates to 50

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

<h2>JavaScript Expressions</h2>

<p>Expressions compute to values.</p>

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

<script>
document.getElementById("demo").innerHTML = 5 * 10;
</script>

</body>
</html>
```

## JavaScript Expressions

Expressions compute to values.

50

# JavaScript Coding Examples- Expressions

- ⌘ An expression is a combination of values, variables, and operators, which computes to a value.
- ⌘ The computation is called an evaluation.
- ⌘ For example,  $5 * 10$  evaluates to 50

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

<h2>JavaScript Expressions</h2>

<p>Expressions compute to values.</p>

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

<script>
var x;
x = 5;
document.getElementById("demo").innerHTML = x * 10;
</script>

</body>
</html>
```

## JavaScript Expressions

Expressions compute to values.

50

# JavaScript Coding Examples- Expressions

- ⌘ The values can be of various types, such as numbers and strings.
- ⌘ For example, "John" + " " + "Doe", evaluates to "John Doe".

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<h2>JavaScript Expressions</h2>
```

```
<p>Expressions compute to values.</p>
```

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

```
<script>
```

```
document.getElementById("demo").innerHTML = "John" + " " + "Doe";
```

```
</script>
```

```
</body>
```

```
</html>
```

## JavaScript Expressions

Expressions compute to values.

John Doe

# JavaScript Coding Examples- Keywords

- ⌘ JavaScript **keywords** are used to identify actions to be performed.
- ⌘ The **let** keyword tells the browser to create variables

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

<h2>The <b>let</b> Keyword Creates Variables</h2>

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

<script>
  let x, y;
  x = 5 + 6;
  y = x * 10;
  document.getElementById("demo").innerHTML = y;
</script>

</body>
</html>
```

**The let Keyword Creates Variables**

110



# JavaScript Coding Examples- Keywords

- ⌘ JavaScript **keywords** are used to identify actions to be performed.
- ⌘ The **var** keyword tells the browser to create variables

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

<h2>The var Keyword Creates Variables</h2>

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

<script>
var x, y;
x = 5 + 6;
y = x * 10;
document.getElementById("demo").innerHTML = y;
</script>

</body>
</html>
```

**The var Keyword Creates Variables**

110

In these examples, using **var** or **let** will produce the same result.

# JavaScript Coding Examples- Case Sensitive

- ⌘ All JavaScript identifiers are **case sensitive**.
- ⌘ The variables **lastName** and **lastname**, are two different variables

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

<h2>JavaScript is Case Sensitive</h2>

<p>Try to change lastName to lastname.</p>

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

<script>
let lastname, lastName;
lastName = "Doe";
lastname = "Peterson";
document.getElementById("demo").innerHTML = lastname;
</script>

</body>
</html>
```

## JavaScript is Case Sensitive

Try to change lastName to lastname.

Doe

## JavaScript is Case Sensitive

Try to change lastName to lastname.

Peterson

JavaScript does not interpret **LET** or **Let** as the keyword **let**.

# JavaScript Coding Examples- Comments

- ⌘ JavaScript comments can be used to explain JavaScript code, and to make it more readable.
- ⌘ JavaScript comments can also be used to prevent execution, when testing alternative code.

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

<h1 id="myH"></h1>
<p id="myP"></p>

<script>
// Change heading:
document.getElementById("myH").innerHTML = "JavaScript Comments";
// Change paragraph:
document.getElementById("myP").innerHTML = "My first paragraph.";
</script>

</body>
</html>
```

**JavaScript Comments**

My first paragraph.

# JavaScript Coding Examples- Multi-line Comments

- ⌘ JavaScript comments can be used to explain JavaScript code, and to make it more readable.
- ⌘ JavaScript comments can also be used to prevent execution, when testing alternative code.

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

<h1 id="myH"></h1>
<p id="myP"></p>

<script>
/*
The code below will change
the heading with id = "myH"
and the paragraph with id = "myP"
*/
document.getElementById("myH").innerHTML = "JavaScript Comments";
document.getElementById("myP").innerHTML = "My first paragraph.";
</script>

</body>
</html>
```

- It is most common to use single line comments.
- Block comments are often used for formal documentation.

**JavaScript Comments**

My first paragraph.

# JavaScript Coding Concepts- Variables Declaration

When to Use var, let, or const?

1. Always declare variables
2. Always use `const` if the value should not be changed
3. Always use `const` if the type should not be changed (Arrays and Objects)
4. Only use `let` if you can't use `const`
5. Only use `var` if you MUST support old browsers.

# Summary of the Today's Lesson

## Part I

### Exam Solutions

- ☐ MCQs
- ☐ Short Questions
- ☐ Etc.

## Part II

### Project work

- ☐ Ideas collection
- ☐ Project presentation

### Course contents

- ☐ JS
- ☐ PHP
- ☐ Databases.



## Part III

### Displaying JS Outputs

- ☐ Window alert
- ☐ Inner Html
- ☐ Console.log
- ☐ Print
- ☐ Alert

## Part IV

### Basics of JS

- ☐ Variables
- ☐ Operators
- ☐ Keywords
- ☐ Comments
- ☐ Etc.

**Note:** Please execute all codes at least once on your computers.