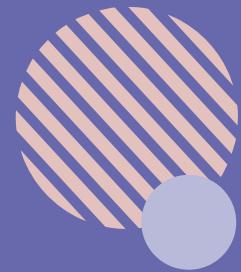


Web Programming

JavaScript Overview

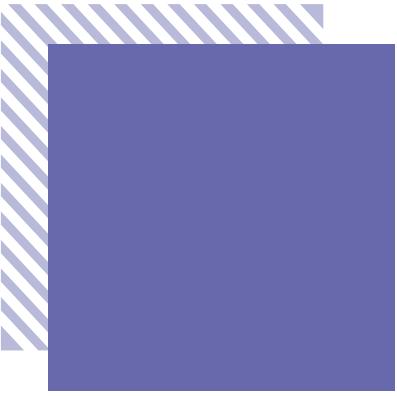
Instructor: Prof. SoYeop Yoo
School of Computing, Gachon University



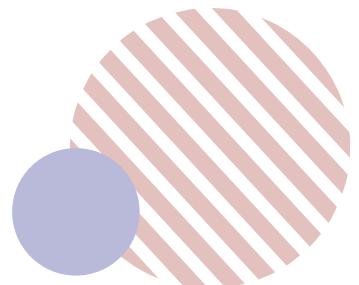
CONTENTS

01 Overview

02 Basics



Overview





JavaScript and the Web (1/2)

■ HTML

- Tells the browser how a document should appear
- Supports viewing (no interaction)

■ JavaScript

- Makes Web pages (documents) dynamic and interactive
- Changes contents of documents, provides forms and controls, animation, control window, etc.



JavaScript and the Web (2/2)

■ Scripting Language

- Executed by an interpreter contained in the Web browser (scripting host)

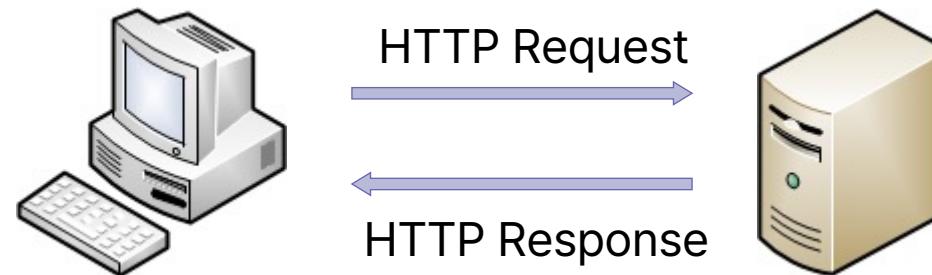
■ Interpreter

- Uses a scripting engine
- Converts code to executable format each time it runs
- Converts when browser loads a Web document



HTML and JavaScript

■ HTML and JavaScript



HTML file with embedded script

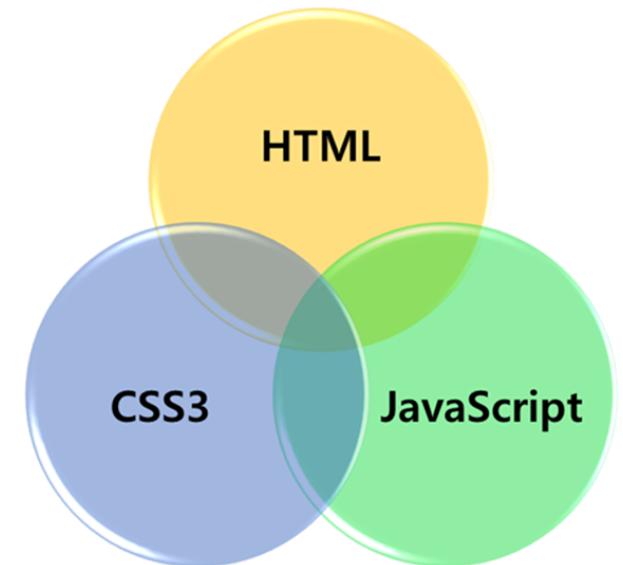
```
<html>
  <head>
    <script> ... </script>
  </head>
  <body> ... </body>
</html>
```



Programming Languages Used in Most Websites

■ Front End (client, browser)

- HTML (HyperText Markup Language)
- CSS (Cascading Style Sheets)
- JavaScript, TypeScript
- React, Angular, Vue, ...



■ Back End (server) (to study in Web Programming 2)

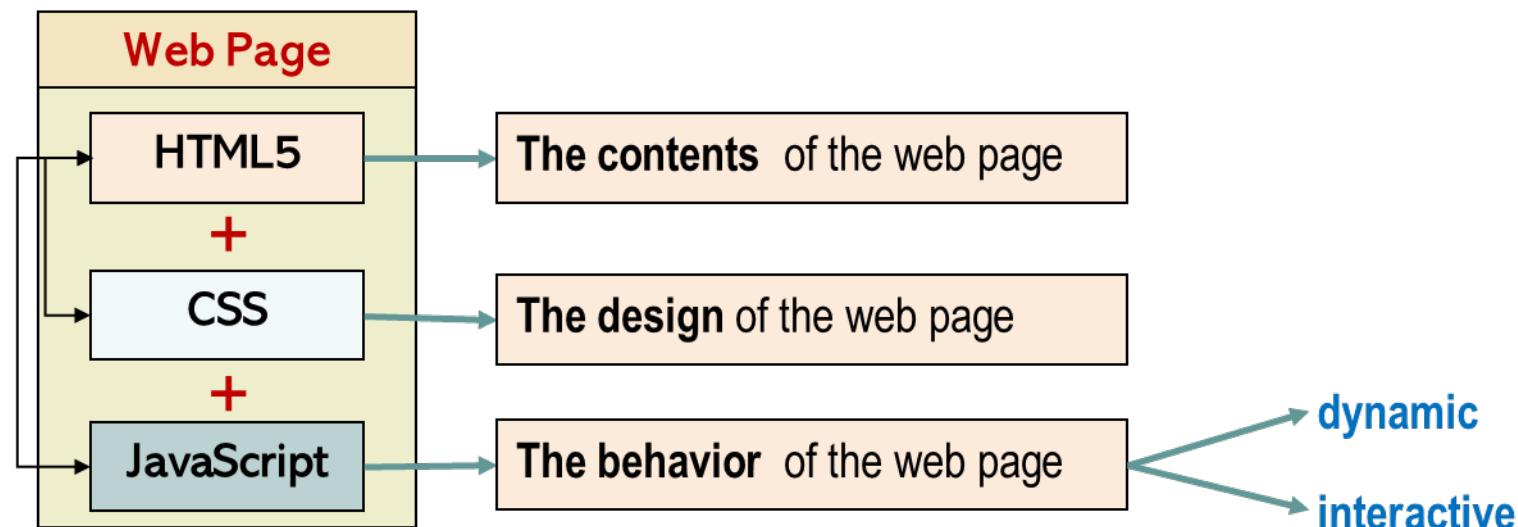
- Java, Python, PHP, GoLang, Kotlin, etc.
- NodeJS, expressJS, NextJS, Spring, Django, ...



HTML vs. CSS vs. JavaScript

■ HTML vs. CSS vs. JavaScript

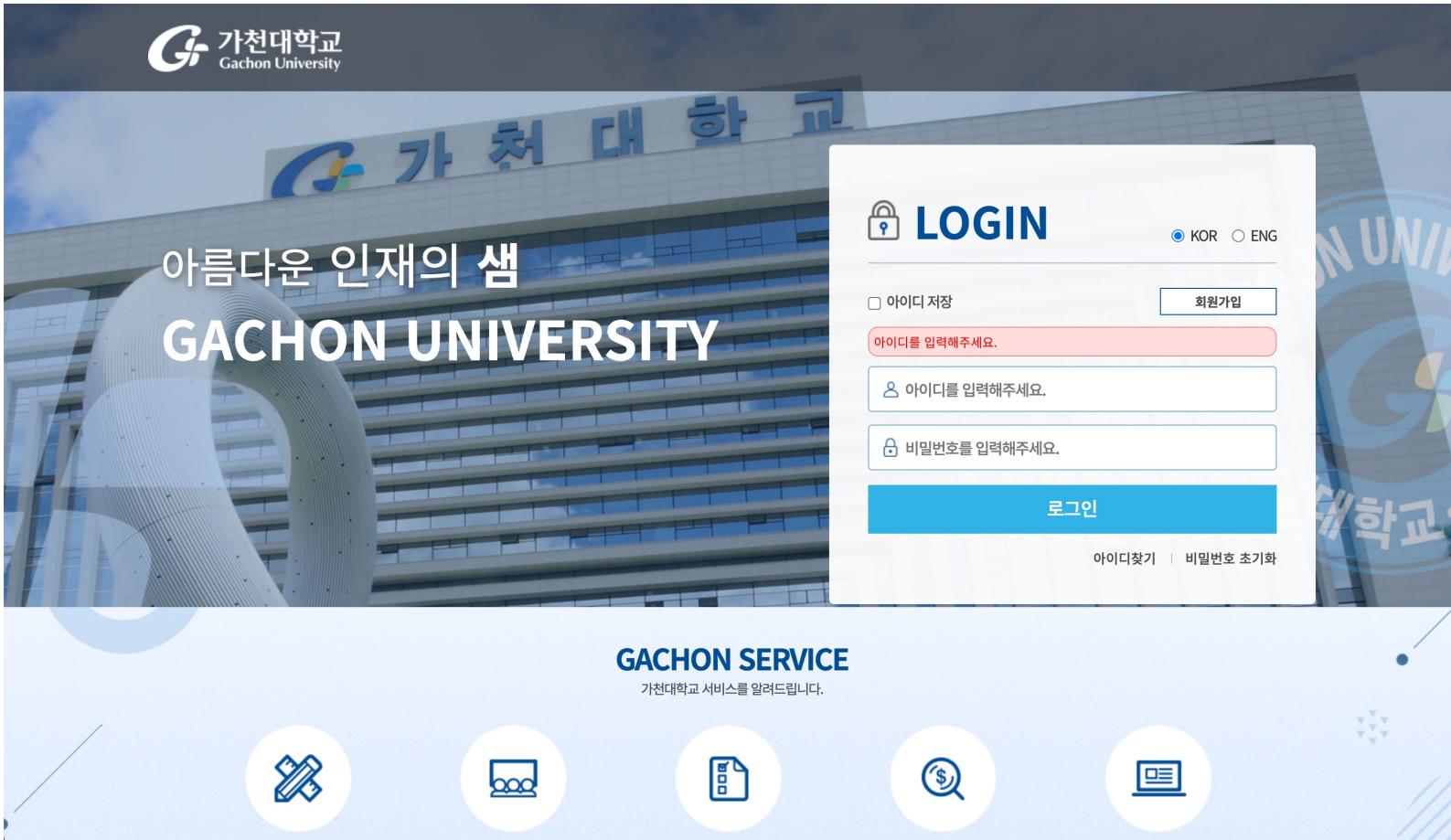
- HTML5, CSS3, and JavaScript work together to form the front-end design of a website by applying information that affects content, style and interactivity of a website
- HTML is a markup language; CSS is a design language; JavaScript is a programming language



Common Uses of JavaScript (1/2)

■ Example

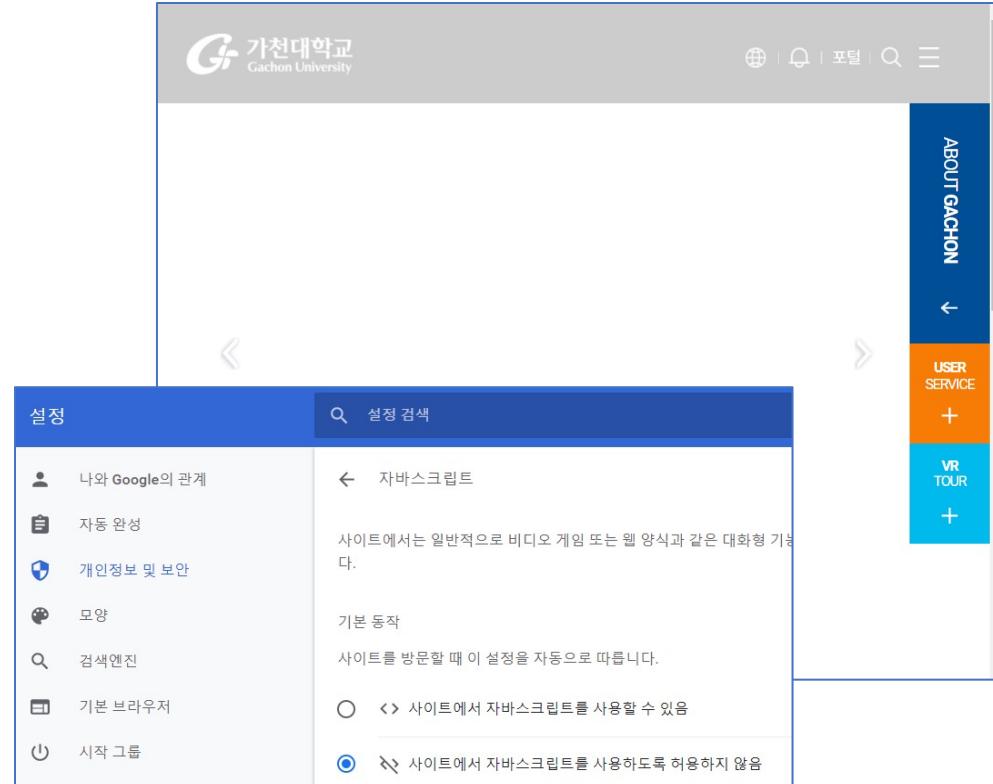
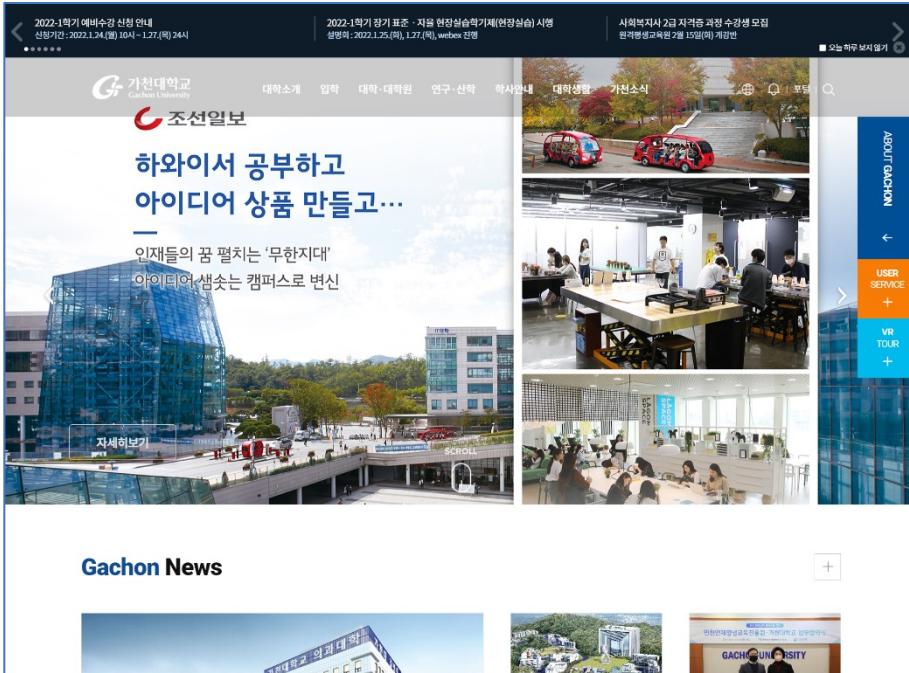
- Website login form



Common Uses of JavaScript (2/2)

■ Example

- Website using JavaScript vs. Website not using JavaScript



- ✓ Disable JavaScript Settings in Chrome
- [개인정보및보안]-[사이트설정]-[콘텐츠]-[자바스크립트]



Front End vs. Back End

Front End

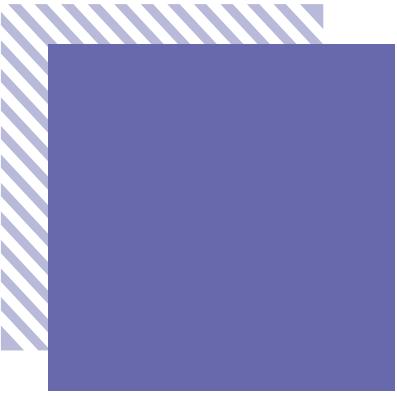
- = client (Web browser)

Back End

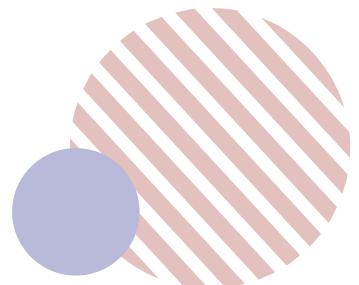
- = server



Server-Side and Client-Side Scripting



Basics

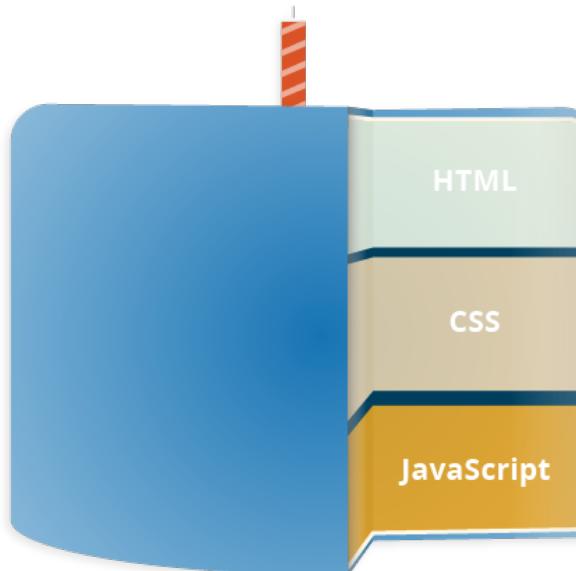




JavaScript

■ JavaScript

- Most widely used Web script language
- Light-weight programming language
- Inserted into HTML pages
- Executed by all modern browsers
- Easy to learn (syntax similar to C, Java, PHP, etc.)





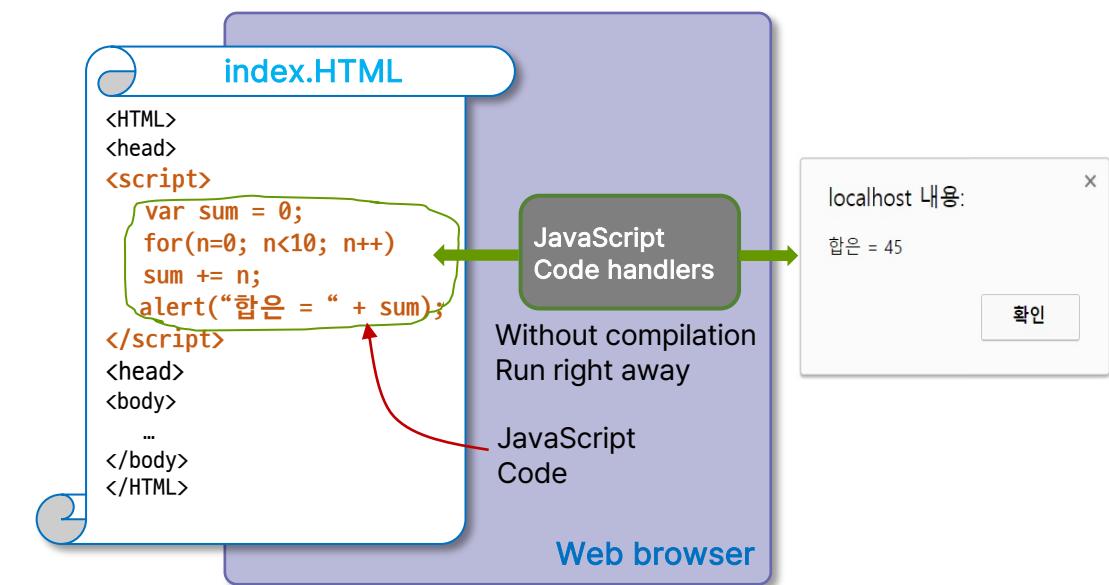
JavaScript Language

■ Start

- 1995 Netscape development
 - Netscape Navigator 2.0 installed for the first time in the browser
 - Web programming concepts

■ Features

- Build into HTML documents
 - Fragment source code
- Script language
 - Running interpreters
 - No compilation required
- Simple
 - Borrowing of C-language structures
 - Easy to learn





Inserting a JavaScript into a HTML page

Example

```
<!DOCTYPE html>
<html>
  <head>
    <title>First JavaScript Program</title>
  </head>
  <body>
    <script type="text/javascript" src="your_source.js"></script>
    <script type="text/javascript">
      document.write("Hello, World Wide Web!<br/>");
      // Prints the string on the HTML page
      // Same as writing in <body> element
    </script>
  </body>
</html>
```

Hello, World Wide Web!



Script Tag (1/2)

■ <script> Tag

- Can be placed within the <head> or <body> of an HTML document
- Internal JavaScript
 - `<script type="text/javascript"> ... </script>`
 - `<script> ... </script>`
- External JavaScript
 - `<script type="text/javascript" src="your_source.js"></script>`
 - If "src" attribute presents, <script> element should be empty



Script Tag (2/2)

Attribute

- **src**
 - specifies the location of an external script
- **type**
 - specifies the scripting language of the script
 - **text/javascript** (default)



Use of Semi-colons

Semi-colon

- Semi-colon(;) at the end of a line is not mandatory in JavaScript
 - ex) document.write("Hello, World Wide Web")
- However, if you are writing two or more statements in one line, then you need the semi-colon
 - ex) x=10; y=5; z=0;
- So, let's just put the semi-colon for our habit's sake
 - Example
 - To distinguish between sentences, use semi-colon(;)

```
i = i + 1
// (0) If there is only one sentence in a line, you can omit the semicolon.

j = j + 1;           // (0)
k = k + 1; m = m + 1; // (0) Multiple sentences on a single line
n = n + 1 p = p + 1;
// (x) A semicolon is required at the end of the first sentence
```



Comments

Comment

```
// One line comment. Commenting to the end of the line
/*
    Multiple line comments
*/
```



JavaScript Output

■ **document.write()**

- The write() method (or function) writes HTML expressions or JavaScript code to a document
 - `document.write()`
 - ex) `document.write("<h3>Welcome!</h3>");`
 - `document.writeln()`
 - writeln() adds '\n' to the text and outputs it



JavaScript Output (cont'd)

■ Illustration

b.js

```
document.write("(b.js) Hello, World Wide Web!<br/>");
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>First JavaScript Program</title>
  </head>
  <body>
    <script type="text/javascript" src="b.js"></script>
    <script type="text/javascript">
      document.write("Hello, World Wide Web!<br>");
    </script>
  </body>
</html>
```

(b.js) Hello, World Wide Web!
Hello, World Wide Web!



JavaScript Output (cont'd)

■ Example 1

- Add text

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example 1</title>
  </head>
  <body>
    <script>
      document.write("Hi~ Have a good time~");
    </script>
  </body>
</html>
```

Object

Method

"Text displayed on the screen"

Hi~ Have a good time~

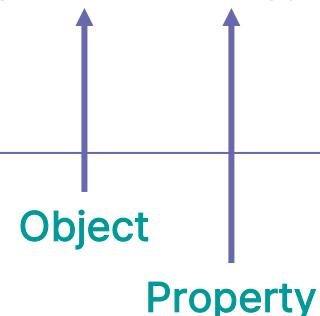


JavaScript Output (cont'd)

■ Example 2

- Add script code

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example 2</title>
  </head>
  <body>
    <script>
      document.write(document.title);
    </script>
  </body>
</html>
```



Example 2



JavaScript Output (cont'd)

■ Example 3

- Add an HTML tag

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example 3</title>
  </head>
  <body>
    <script>
      document.write("<h1>Hi~ Have a good day~</h1>");
    </script>
  </body>
</html>
```

Hi~ Have a good day~



JavaScript Output (cont'd)

■ Example 4

- Add multiple scripts

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example 4</title>
  </head>
  <body>
    <script>
      document.write("<h1>Hi~</h1>");
    </script>
    <hr />
    <script>
      document.write("<h1>Have a good day~</h1>");
    </script>
  </body>
</html>
```

Hi~

Have a good day~



JavaScript Output (cont'd)

Example 5

```
<!DOCTYPE html>
<html>
  <head>
    <title>Example 5</title>
  </head>
  <body>
    <h3>document.write()</h3>
    <hr />
    <script>
      document.write("<h3>Welcome!</h3>");
      document.write("2 + 5 is <br>");
      document.write("<mark>7.</mark>");
    </script>
  </body>
</html>
```

document.write()

Welcome!

2 + 5 is
7.



External JavaScript

■ External JS

- Save JavaScript code in a file
 - Save to an file_name.js file
 - Save only JavaScript code as a file
- The saved file can be accessed from multiple web pages
 - The file is the value of the src attribute of the <script> tag

```
<script src="file_name.js">  
    // The script element should be empty.  
</script>
```



External JavaScript (cont'd)

Example

- In the example code below, save the JavaScript code contained in the `<script>` tag to the `lib.js` file
- Ignore the details of the code; we will study it later in the course

```
<!DOCTYPE html>
<html>
  <head>
    <title>External JavaScript</title>
  <script>
    function over(obj) {
      obj.src = "media/banana.png";
    }
    function out(obj) {
      obj.src = "media/apple.png";
    }
  </script>
  </head>
  <body>
    <h3>Mouse over the image</h3>
    <hr />
    
  </body>
</html>
```



External JavaScript (cont'd)

Example (Result)

lib.js

```
function over(obj) {  
    obj.src = "media/banana.png";  
}  
function out(obj) {  
    obj.src = "media/apple.png";  
}
```

Mouse over the image



Mouse over the image



```
<!DOCTYPE html>  
<html>  
    <head>  
        <title>External JavaScript</title>  
        <script src="lib.js"></script>  
    </head>  
    <body>  
        <h3>Mouse over the image</h3>  
        <hr />  
          
    </body>  
</html>
```



Debugging

■ Debugging JavaScript

- Browser dependent!
 - Maybe different options for different browser versions
 - Look up for JavaScript debugging!
- MS Internet Explorer
 - Tools → Internet Options → Advanced
 - Clear “Disable script debugging” check box
 - 스크립트 디버깅 사용 안 함(Internet Explorer)
 - 스크립트 디버깅 사용 안 함(기타)
- Google Chrome
 - <https://developer.chrome.com/devtools/docs/javascript-debugging>

The screenshot shows the 'Documentation' section of the Chrome Developers website. It features the Chrome logo and the word 'Developers'. Below this, there's a large blue icon of a cube with a circular interface on its top face. The text 'Chrome DevTools' is displayed next to the icon. A descriptive paragraph explains that DevTools are built directly into the Google Chrome browser. At the bottom, there's a link to 'Overview' and a brief introduction to getting started with the tools.

Documentation

Chrome DevTools

Chrome DevTools is a set of web developer tools built directly into the Google Chrome browser.

Overview

Get started with Google Chrome's built-in web developer tools.



JavaScript Pros and Cons

■ Pros

- Improves the user interface of Web sites
- Makes Web sites easier to navigate
- Easily creates pop-up alerts and windows
- Replaces images on a page without reloading it
- Validates forms

■ Cons

- Cannot access file or database systems
- Cannot do networking



JavaScript vs. Java (applet)

■ JavaScript vs. Java Applet

- (Don't worry about this. This is just for reference after you learn Java)
- Both execute on the client
- JavaScript is interpreted; Java is compiled
- JavaScript embedded in an HTML page; Java applet is a separate download
- JavaScript is simpler than Java
- JavaScript is specific to Web pages; Java is for general purpose



JavaScript Dialogue

■ prompt()

- `prompt("Message", "Default Input")` function
- Allows the user to enter a response
- Get a string from the user and return it
- Can provide a default value to avoid displaying "undefined"
- All responses are String type values (even if user enters number)

```
<!DOCTYPE html>
<html>
  <head>
    <title>prompt() Method</title>
  </head>
  <body>
    <h3>prompt() method</h3>
    <hr />
    <script type="text/javascript">
      y = prompt("What is your name?");
      z = prompt("How old are you?", "20");
    </script>
  </body>
</html>
```

The image shows two separate browser windows side-by-side. Both windows have a light gray header bar with the text '이 페이지 내용:' and a question below it. The first window has the question 'What is your name?' and contains an input field with a blue border. The second window has the question 'How old are you?' and contains an input field with a blue border containing the number '20'. Each window has a small blue button labeled '취소' (Cancel) and a larger blue button labeled '확인' (Confirm).

이 페이지 내용:
What is your name?

이 페이지 내용:
How old are you?



Verification Dialogue

■ confirm()

- confirm("Message") function
- Outputs messages
- Dialog output with 'OK/CANCEL' button
- If you press the 'OK' button, return 'true', if you press the 'CANCEL' button, return 'false'

```
<!DOCTYPE html>
<html>
  <head>
    <title>confirm() Method</title>
  </head>
  <body>
    <h3>confirm() method</h3>
    <hr />
    <script type="text/javascript">
      x = confirm("Are you OK?");
    </script>
  </body>
</html>
```



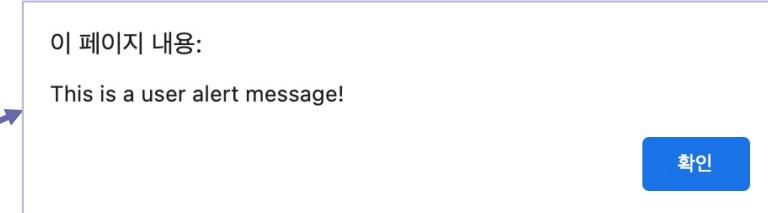


Warning Dialogue

■ alert()

- alert("Message") function
- Displays messages in the dialog box
- An alert box is often used if you want to make sure information comes through to the user
- Click the OK button to exit

```
<!DOCTYPE html>
<html>
  <head>
    <title>alert() Method</title>
  </head>
  <body>
    <h3>alert() method</h3>
    <hr />
    <script type="text/javascript">
      alert("This is a user alert message!");
    </script>
  </body>
</html>
```



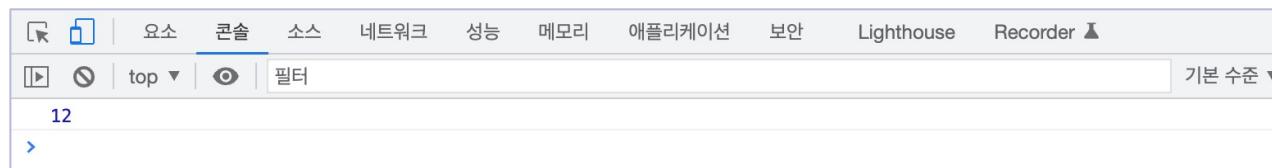


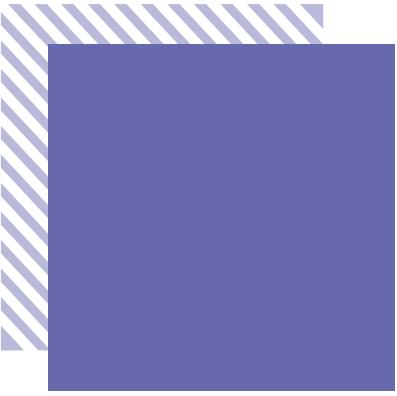
console.log()

■ console.log() Method

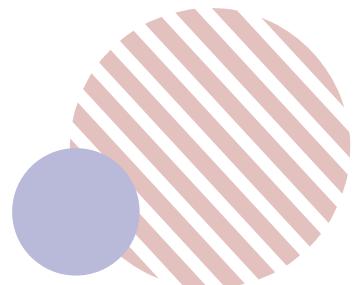
- Display JavaScript values in the debugger window
- Debugger is activated by pressing F12

```
<!DOCTYPE html>
<html>
  <head>
    <title>console.log()</title>
  </head>
  <body>
    <h3>console.log() method</h3>
    <hr />
    <script type="text/javascript">
      a = 5;
      b = 7;
      c = a + b;
      console.log(c);
    </script>
  </body>
</html>
```





Exercises





Exercise 1

Exercise 1

- Create the following web page, using JavaScript document.write()

Web Programming Exercise 1

This is an h2 heading.

This is a paragraph.

This is another paragraph.

<h1> tag

<h2> tag

<p> tag

<p> tag



Exercise 2

Exercise 2

- For each of the following, evaluate using document.write() function
 - $\text{value} = (19 \% 4) / 1 - 1$
 - $\text{value} = 3 / 1 - 1$
 - $\text{value} = 3 - 1$
 - $\text{value} = 1$

Exercise 2

$(19 \% 4) / 1 - 1 = 2$
 $3 / 1 - 1 = 2$
 $3 - 1 = 2$
 $\text{value} = 1$



Exercise 2 (cont'd)

Hint

```
<!DOCTYPE html>
<html>
  <head>
    <title>Exercise 2</title>
  </head>
  <body>
    <h3>Exercise 2</h3>
    <hr />
    <script>
      var value = (19 % 4) / 1 - 1;
      document.write("(19 % 4) / 1 - 1 = " + value + "<br/>");
    </script>
  </body>
</html>
```

Exercise 2

$$(19 \% 4) / 1 - 1 = 2$$



End of Class

