

Web Programming

JavaScript – DOM and Events

Instructor: Jin Woo Choi

School of Computing, Gachon University



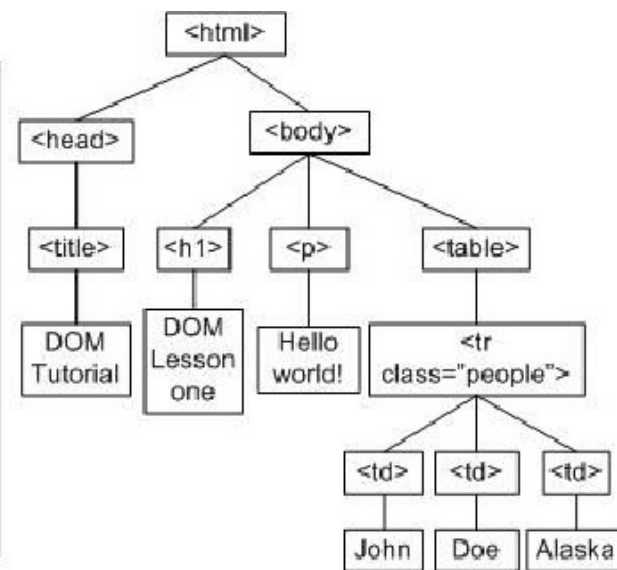
Document Object Model (DOM)

● HTML DOM

- When a web page is loaded, the browser creates a *document* object of the page
- The object is a tree of HTML elements
- Using DOM, JavaScript can:
 - Change HTML elements in the page
 - Change HTML attributes in the page
 - Change CSS styles in the page
 - React to all the events in the page

```
<html>
<head>
  <title>DOM Tutorial</title>
</head>
<body>
  <h1>DOM Lesson one</h1>
  <p>Hello world!</p>
  <table>
    <tr class="people">
      <td>John</td>
      <td>Doe</td>
      <td>Alaska</td>
    </tr>
  </table>
</body>
</html>
```

test.html



DOM Tree

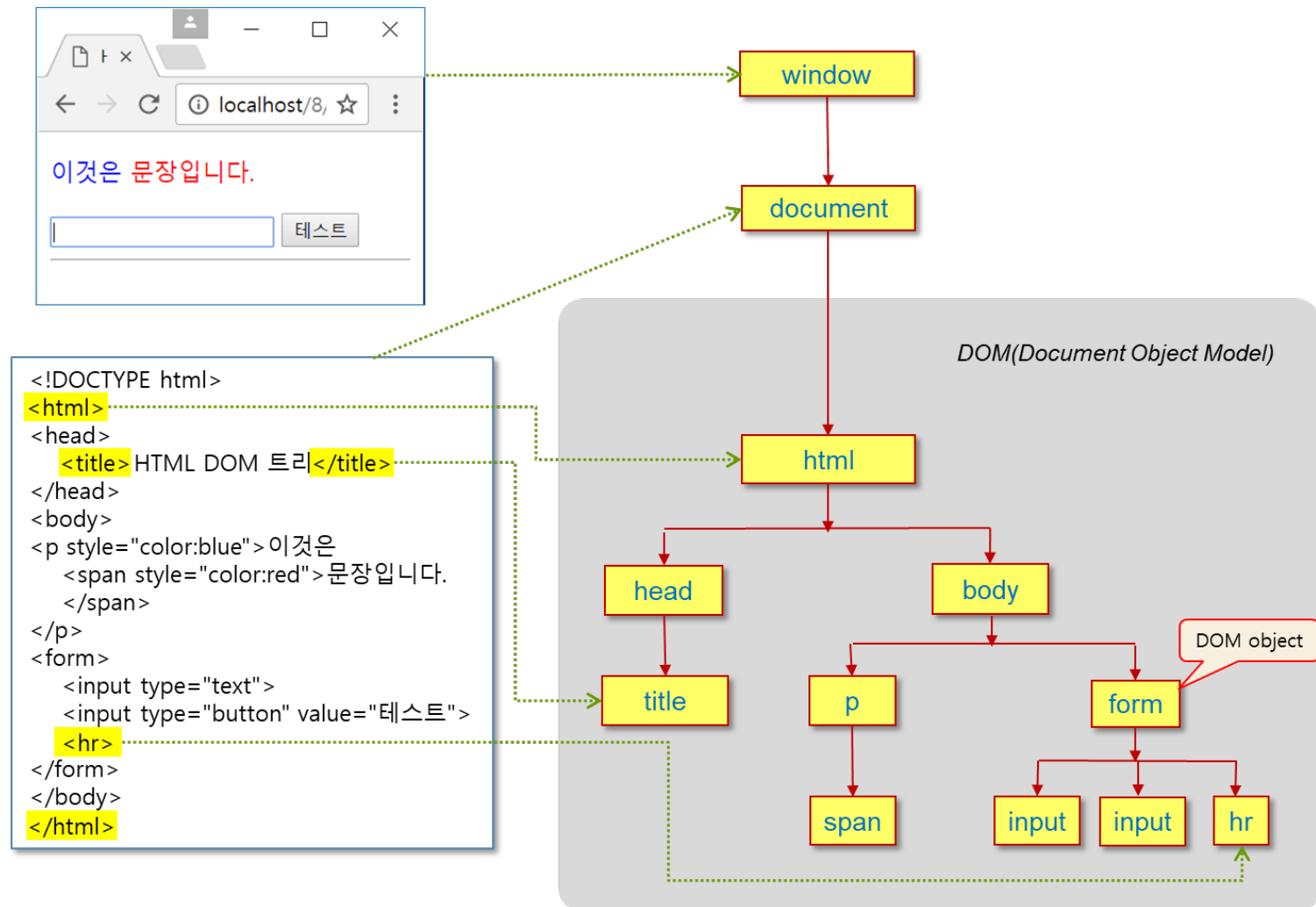
Document Object Model (DOM) – [cont'd]

● HTML DOM

- A standard object model and programming interface for HTML documents
- DOM defines
 - HTML *elements* as objects
 - *Properties* of all HTML elements
 - *Methods* to access all HTML elements
 - *Events* for all HTML elements

Document Object Model (DOM) – [cont'd]

● HTML DOM tree

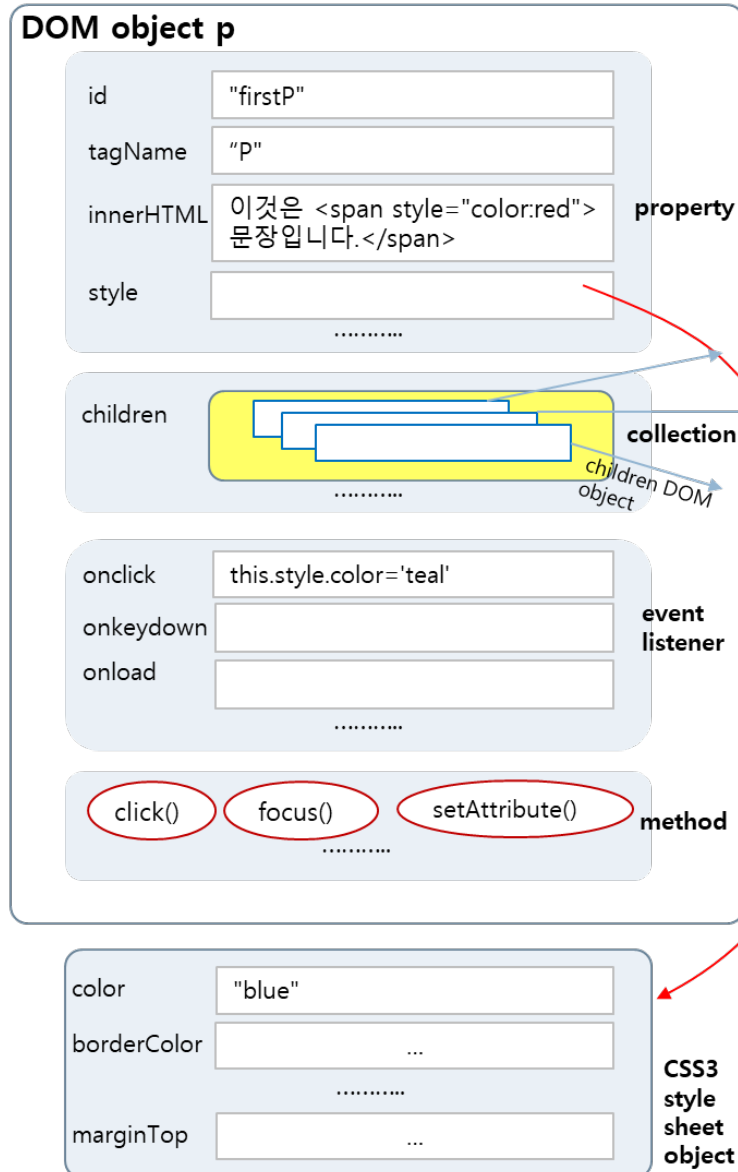


Document Object Model (DOM) – [cont'd]

● HTML DOM object example

```
<p id="firstP"
  style="color:blue"
  onclick="this.style.color='teal'">
  이것은
  <span style="color:red">
    문장입니다.
  </span>
</p>
```

<p>...</p> tag



What can we do with HTML DOM?

- **Get an HTML element as an object**
 - e.g., `x=document.getElementById("ID_name");`
- **Change the contents of HTML elements**
 - e.g., `x.innerHTML = "Change text!";`
- **Change the design of HTML elements using CSS**
 - e.g., `x.style.color = "blue";`
- **React to events**
 - e.g., `<button type="button" onclick="x.style.color='red'">`

What can we do with HTML DOM?

```

<!DOCTYPE html>
<html>

<body>
  <h1>The Element Object</h1>
  <h2>The innerHTML Property</h2>

  <p id="demo" onclick="myFunction()">Click me to change</p>

  <script>
    function myFunction() {
      x = document.getElementById("demo");
      x.innerHTML = "I have changed!";
      x.style.color = "blue";
    }
  </script>

</body>

</html>

```

The Element Object

The innerHTML Property

Click me to change



The Element Object

The innerHTML Property

I have changed!

Document

● The Document Object

- When an HTML document is loaded into a web browser, it becomes a document object.
- The document object is the root node of the HTML document.
- The document object is a property of the window object.
- The document object is accessed with:
 - window.document or just document

https://www.w3schools.com/jsref/dom_obj_document.asp

Get a DOM Element (1/4)

- **By ID**

- e.g., `x=document.getElementById("ID_name");`

- **By tag name**

- e.g., `y=x.getElementsByTagName("p");`

- **By class name**

- e.g., `x=document.getElementsByClassName("CLASS_name");`

- **HTML object collections**

- e.g., `var a=document.forms["form1"];`

- **Query Selectors**

- e.g., `var a=document.querySelector('#query');`
- e.g., `var a=document.querySelectorAll('.query');`

Get a DOM Element (2/4)

● by ID

```
<p id="intro">Hello World!</p>
<p>This example demonstrates <em>getElementById</em> method.</p>
<script>
  x=document.getElementById("intro");
  // If the element is found, return it as an object; null, otherwise
  document.write("<p>The text from intro paragraph is: "
    + x.innerHTML + "</p>");
  // innerHTML: returns/sets the content of an element
</script>
```



Hello World!

This example demonstrates *getElementById* method.

The text from intro paragraph is: Hello World!

Get a DOM Element (3/4)

● by tag name

```
<p>Hello World!</p>
<div id="main">
  <p>DOM is very useful.</p>
  <p>This example demonstrates <em>getElementsByTagName</em> method.</p>
</div>
<script>
  var x=document.getElementById("main");
  var y=x.getElementsByTagName("p");
  // Finds <p> elements inside "main" element and returns as an array
  document.write("<p>First &lt;p>&gt; element inside \"main\" is: "
    + y[0].innerHTML + "</p>");
</script>
```



Hello World!

DOM is very useful.

This example demonstrates *getElementsByTagName* method.

First <p> element inside "main" is: DOM is very useful.

Get a DOM Element (4/4)

● by class name

```
<div class="example">
  <p>This is a paragraph.</p>
</div>
<br>
<div class="example color">
  <p>This is a paragraph.</p>
</div>
<br>
<div class="example color">
  <p>This is a paragraph.</p>
</div>

<script>
var collection = document.getElementsByClassName("example color");
collection[0].style.backgroundColor = "red";
</script>
```

This is a paragraph.

This is a paragraph.

This is a paragraph.

Change DOM Elements

- **Change element content**

- e.g., `document.getElementById(id).innerHTML = "new text!"`;

- **Change element attribute**

- e.g., `document.getElementById("image").src = "landscape.jpg"`;

Change Element Content & Attribute (1/2)

```

<p id="p1">Hello World!</p>

<script>
    var x = document.getElementById("p1");
    x.innerHTML="Nice to See You!";
    // Use innerHTML property to change element content
    var x = document.getElementById("image");
    x.src="world.jpg"; x.width=300;
</script>
<p>The paragraph above was changed by JavaScript.<br/>
    The image was apple.gif, but was changed to world.jpg.</p>

```

Change Element Content & Attribute (2/2)

Hello World!



The paragraph above was changed by JavaScript.
The image was apple.gif, but was changed to world.jpg.



Nice to See You!



The paragraph above was changed by JavaScript.
The image was apple.gif, but was changed to world.jpg.

Change CSS Style (1/2)

```
<p id="p1">Hello World!</p>
<p id="p2">Hello World!</p>
<script>
  var x1=document.getElementById("p1");
  var x2=document.getElementById("p2");
  x1.style.color="red"; x1.style.textDecoration="underline";
  x2.style.fontFamily="Helvetica"; x2.style.fontSize="xx-large";
</script>
<p>The style of paragraphs above was changed by JavaScript.</p>
```



Hello World!

Hello World!

The style of paragraphs above was changed by JavaScript.

Change CSS Style (2/2)

- **Naming conventions are a little different**
 - Cannot use hyphen (-) in JavaScript naming
 - CSS 'background-color' is JS 'backgroundColor'
 - CSS 'font-size' is JS 'fontSize'

Exercise 1

- Add the following in the `<body>` tag of your web page

```
<h2 id="heading">JS EX</h2>
<h3 id="ex1">EX 1</h3><a id="a1" name="EX1.1"/>
<p>Change the following image to a grape whose width is 300.</p>

<script>
    // Add your code here.. (Exercise 1.1)
</script>
<h3 id="ex2">Exercise 1.2</h3>
<p>Go to <a href="#EX1">Exercise 1.1</a>.<br/>
    The name of anchor "a1" is: <span id="sp1"></span></p>
<script>
    // Add your code here.. (Exercise 1.2)
</script>
```

Exercise 1 [cont'd]

● Write code (exercise 1.1)

- Change the content of id="heading" element to "JavaScript Exercise 1.1~2"
- Change the content of id="ex1" element to "Exercise 1.1"
- Change the *apple* image to a *banana* and set its width=300
 - Specify a URL for the image

● Write code (exercise 1.2)

- Change the content of id="sp1" element as the name of anchor "a1"
- For the content of id="ex2" element, change its *color* to "red" and *fontStyle* to "italic"

Exercise 1 [cont'd]

● Result should look like:

JS EX

EX 1

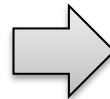
Change the following image to a grape whose width is 300.



Exercise 1.2

Go to [Exercise 1.1](#).

The name of anchor "a1" is:



JavaScript Exercise 1

Exercise 1.1

Change the following image to a grape whose width is 300.



Exercise 1.2

Go to [Exercise 1.1](#).

The name of anchor "a1" is: EX1.1

Exercise 1 [cont'd]

● solution

```
<body>
  <h2 id="heading">JS EX</h2>
  <h3 id="ex1">EX 1</h3><a id="a1" name="EX1.1" />
  <p>Change the following image to a grape whose width is 300.</p>
  
  <script>
    // Add your code here.. (Exercise 1.1)
    document.getElementById("heading").innerHTML = "JavaScript Exercise 1";
    document.getElementById("ex1").innerHTML = "Exercise 1.1";
    document.getElementById("img1").src = "banana.png";
    document.getElementById("img1").width = 300;
  </script>
  <h3 id="ex2">Exercise 1.2</h3>
  <p>Go to <a href="#EX1.1">Exercise 1.1</a>.<br />
    The name of anchor "a1" is: <span id="sp1"></span></p>
  <script>
    // Add your code here.. (Exercise 1.2)
    document.getElementById("sp1").innerHTML = document.getElementById("a1").name;
    document.getElementById("ex2").style.color = "red";
    document.getElementById("ex2").style.fontStyle = "italic";
  </script>
</body>
```

Exercise 1 [cont'd]

● better solution

```
<body>
  <h2 id="heading">JS EX</h2>
  <h3 id="ex1">EX 1</h3><a id="a1" name="EX1.1" />
  <p>Change the following image to a grape whose width is 300.</p>
  
  <script>
    // Add your code here.. (Exercise 1.1)
    document.getElementById("heading").innerHTML = "JavaScript Exercise 1";
    document.getElementById("ex1").innerHTML = "Exercise 1.1";
    var img1 = document.getElementById("img1");
    img1.src = "banana.png";
    img1.width = 300;
  </script>
  <h3 id="ex2">Exercise 1.2</h3>
  <p>Go to <a href="#EX1.1">Exercise 1.1</a>.<br />
    The name of anchor "a1" is: <span id="sp1"></span></p>
  <script>
    // Add your code here.. (Exercise 1.2)
    document.getElementById("sp1").innerHTML = document.getElementById("a1").name;
    var ex2 = document.getElementById("ex2");
    ex2.style.color = "red";
    ex2.style.fontStyle = "italic";
  </script>
</body>
```

DOM Events

- **You can react to HTML events**

- When a user clicks the mouse
- When a web page has been loaded
- When an image has been loaded
- When the mouse moves over an element
- When an input field is changed
- When an HTML form is submitted
- When a user strikes a key

Event Handling (1/2)

- You can specify what to do when an event happens, as a value of the **onclick** attribute or a **handler** function

```
<p onclick="this.innerHTML='You clicked this paragraph!'">
  Click this paragraph.</p>
<p onclick="changeText(this)">This is another paragraph to click.</p>
<script>
  function changeText(element) {
    element.innerHTML="This paragraph is also clicked!";
  }
</script>
```



Event Handling (2/2)

- Assigning an event handler

```
<script>
    function viewDate() {
        var now = new Date();
        document.getElementById("sp1").innerHTML = now.getDate();
    }
</script>
<p>Current date is: <span id="sp1"></span></p>
<input type="button" id="b1" value="Get date" onclick="viewDate()"/>
```

Mouse Events

Event	Description (the event occurs when ...)
onclick	the user clicks on an element
oncontextmenu	the user right-clicks on an element to open a context menu
ondblclick	the user double-clicks on an element
onmousedown	a user presses a mouse button over an element
onmouseenter	the pointer is moved onto an element
onmouseleave	the pointer a user moves the mouse pointer out of an element
onmousemove	the pointer is moving while it is over an element
onmouseover	the pointer is moved onto an element, or onto one of its children
onmouseout	a user moves the mouse pointer out of an element or one of its children
onmouseup	a user releases a mouse button over an element

Example 1

```
<!DOCTYPE HTML>
<HTML>
<head>
  <meta charset="utf-8">
  <title> JavaScript code in script tag </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>
```

The 'obj' argument points to the image of the img tag that is delivered

mouse over the image



mouse over the image



'this' is currently the img tag JavaScript keywords point to

Example 2

```
<head>
<style>
  div { background-color:#D94A38; color:white; width:120px; height:20px; padding:40px; }
</style>
<script>
  function mouseOver(elem) { elem.innerHTML="Thank you!"; }
  function mouseOut(elem) { elem.innerHTML="Move mouse here"; }
</script>
</head>
<body>
  <div onmouseover="mouseOver(this)" onmouseout="mouseOut(this)">Move mouse here</div>
</body>
```



Example 3

- Make the heading red when the mouse moves over it, and navy when the mouse moves away
- Show a grape image when it is clicked, and an apple when it is released

```
<h3 onmouseover="style.color='red'" onmouseout="style.color='navy'">
  Move mouse over here.</h3>
<h3>Click the image below.</h3>

```

Move mouse over here.

Click the image below.



Move mouse over here.

Click the image below.



Move mouse over here.

Click the image below.



HTML DOM Events

- Mouse, keyboard, form, frame/object events
- Media, drag, clipboard, print events
- See: http://www.w3schools.com/jsref/dom_obj_event.asp

Exercise 2

- **Modify your web page for Exercise 1 as follows**
 - Add two buttons labeled “Exercise 2.1” and “Exercise 2.2”
 - Run Exercise 1.1 code when “Exercise 2.1” button is clicked
 - Do the same for Exercise 1.2

- **Write event handlers for mouse events**
 - Add a div element
 - `backgroundColor="#d94a38", text="Press here"`
 - For *onmousedown* event, change div element
 - `backgroundColor="#1ec5e5", text="Release mouse"`
 - For *onmouseup* event, change to original state

Exercise 2 – [cont'd]

● Result should look like:

JS EX

EX 1

Change the following image to a grape whose width is 300.



Exercise 1.2

Go to [Exercise 1.1](#).

The name of anchor "a1" is:

Exercise 2.1

Exercise 2.2

Press Here



JavaScript Exercise 1

Exercise 1.1

Change the following image to a grape whose width is 300.



Exercise 1.2

Go to [Exercise 1.1](#).

The name of anchor "a1" is: EX1.1

Exercise 2.1

Exercise 2.2

Release mouse

Exercise 2 – [cont'd]

```
<button onclick="exercise2_1()">Exercise 2.1</button>
<button onclick="exercise2_2()">Exercise 2.2</button>
<br><br>
<div onmousedown="mouseDown(this)" onmouseup="mouseUp(this)">
  Press Here
</div>
```

● solution

```
<script>
  function exercise2_1() {
    document.getElementById("heading").innerHTML = "JavaScript Exercise 1";
    document.getElementById("ex1").innerHTML = "Exercise 1.1";
    var img1 = document.getElementById("img1");
    img1.src = "banana.png";
    img1.width = 300;
  }

  function exercise2_2() {
    document.getElementById("sp1").innerHTML = document.getElementById("a1").name;
    var ex2 = document.getElementById("ex2");
    ex2.style.color = "red";
    ex2.style.fontStyle = "italic";
  }

  function mouseDown(elem) {
    elem.style.backgroundColor = "#1ec5e5";
    elem.innerHTML = "Release mouse";
  }

  function mouseUp(elem) {
    elem.style.backgroundColor = "#d94a38";
    elem.innerHTML = "Press here";
  }
</script>
```

DOM Collections

- *forms* - returns an array of all forms in a document
- *images* - returns an array of all `` elements in a document
- *links* - returns an array of all links (`` elements) in a document

Example 4

```
<a name="google" href="http://www.google.com">Google</a>
<a href="http://www.naver.com">Naver</a>
<p>
<script>
    document.write("Number of links: " + document.links.length + "<br>");
    document.write("First link is: " + document.links[0] + "<br>");
    document.write("Second link's html is: " + document.links[1].innerHTML + "<br>");
</script>
```



[Google](#) [Naver](#)

Number of links: 2

First link is: http://www.google.com/

Second link's html is: Naver

Timing Events

● Timing events

- `window.setInterval()` - executes a function, over and over again, at specified time intervals (in milliseconds)
- `window.setTimeout()` - executes a function, once, after waiting a specified number of milliseconds

● Stop execution

- `window.clearInterval()` - stops further executions of the function specified in `setInterval()`
- `window.clearTimeout()` - stops the execution of the function specified in `setTimeout()`

Example 5

```
<body onload="setDate()">
<!-- Runs setDate() once when the document has been loaded -->
<script>
    var myDate;      // an interval variable; needed when stopping
    function setDate() {
        var now = new Date();
        document.getElementById("sp1").innerHTML = now;
    }
</script>
<p>Current time: <span id="sp1"></span></p>
<button onclick="myDate = setInterval('setDate()', 500)">Start</button>
<!-- Runs setDate() every 500 milliseconds (0.5 seconds) -->
<button onclick="clearInterval(myDate)">Stop</button>
<!-- Stops executions of setDate() function -->
</body>
```



Current time: Mon May 23 2022 20:21:55 GMT+0900 (한국 표준시)

Start Stop

Exercise 3

● Write a JavaScript code to do the following:

- Show a text input and two buttons labeled *Start* and *Stop*
- When Start button is pressed, the counter from 0 is displayed in the text input at the interval of 0.5 second
- When Stop button is pressed, the counter stops
- Pressing Start button more than once does not re-initiate the counter until Stop button is pressed

Counter:

Exercise 3

● solution

```
<body>
  Counter: <input type="text" id="myText">
  <br><br>
  <button onclick="myStart()">Start</button>
  <button onclick="myStop()">Stop</button>
  <script>
    var num = 0;
    var myCounter;

    function myStart() {
      myCounter = setInterval(function () {
        document.getElementById("myText").value = num++;
      }, 500);
    }

    function myStop() {
      clearInterval(myCounter);
    }
  </script>
</body>
```

End of Class

