

삼성 청년 SW 아카데미

AJAX & Bootstrap

Confidential

<알림>

본 강의는 삼성 청년 SW아카데미의 컨텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

AJAX(Aynchronous Javascript And XML)

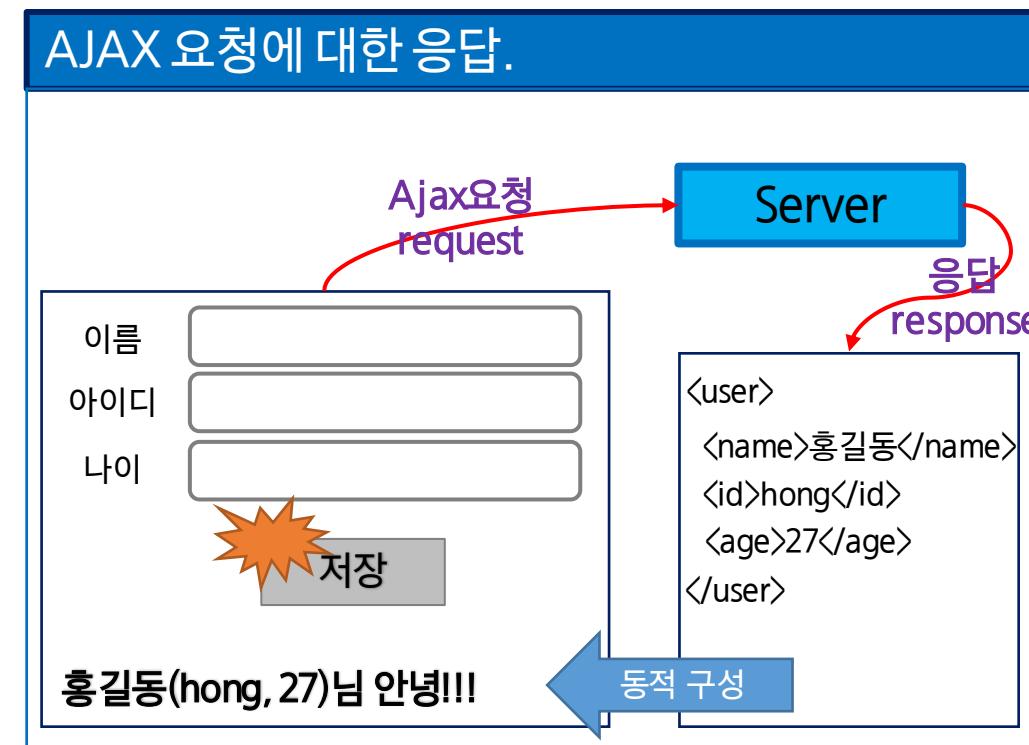
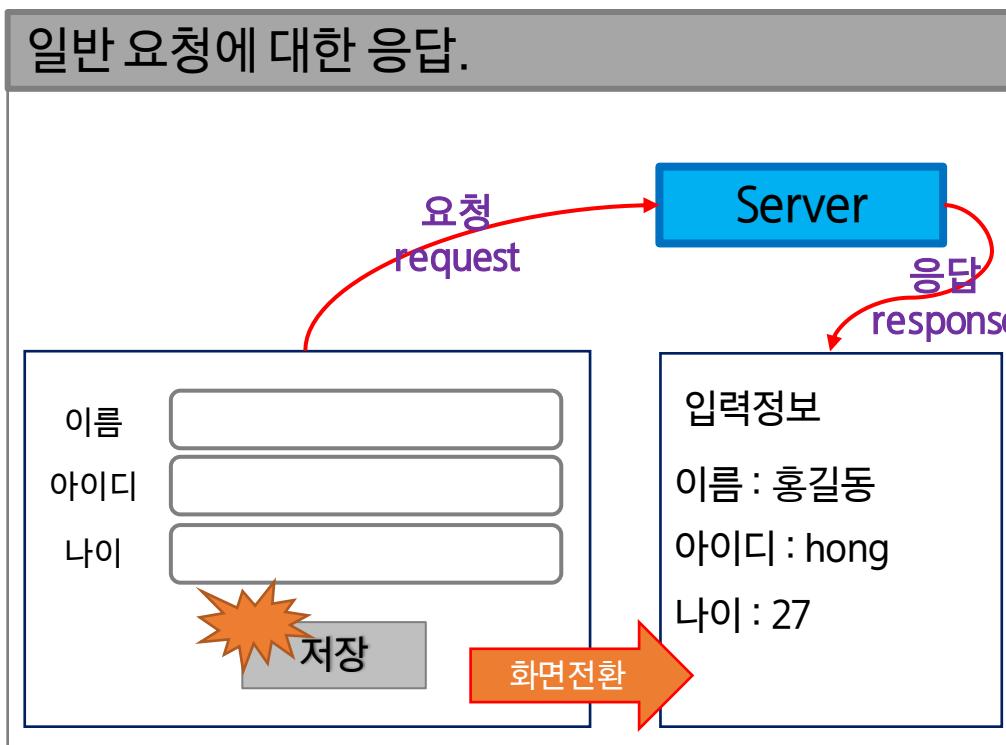
AJAX

Ajax (Asynchronous Javascript And XML).

1-1. Ajax 소개 (1/2).

Confidential

- ✓ Ajax (Asynchronous Javascript and XML)는 언어나 프레임워크가 아닌 구현하는 방식을 의미.
- ✓ Ajax는 웹(Web)에서 화면을 갱신하지 않고 데이터를 서버로부터 가져와 처리하는 방법을 의미.
- ✓ JavaScript의 XMLHttpRequest(XHR) 객체로 데이터를 전달하고 비동기 방식으로 결과를 조회.
- ✓ 화면 갱신이 없으므로 사용자 입장에서는 편리하지만, 동적으로 DOM을 구성해야 하므로 구현이 복잡.



Ajax (Asynchronous Javascript And XML).

1-1. Ajax 소개 (2/2).

Confidential

- ✓ 일반 요청에 대한 응답.
 - data를 입력 후 event 발생.
 - Ajax를 적용하지 않은 요청은 서버에서 data를 이용하여 logic 처리.
 - logic 처리에 대한 결과에 따라 응답 page를 생성하고 client에 전송(화면 전환이 일어남).

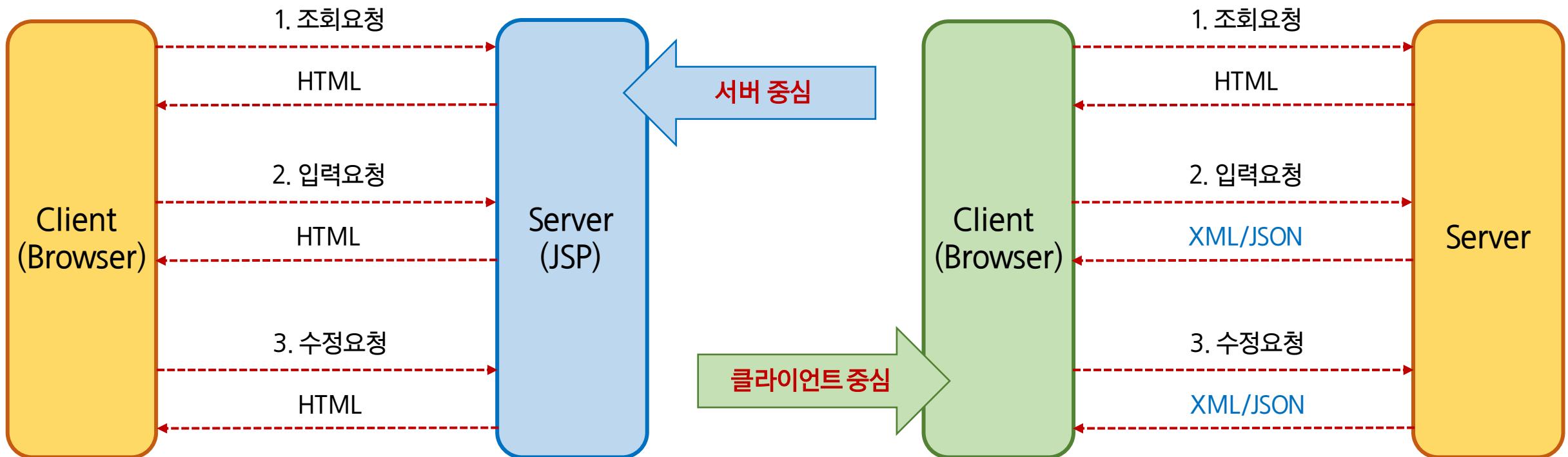
- ✓ Ajax 요청에 대한 응답.
 - data를 입력 후 event 발생.
 - Ajax를 적용하면 event 발생 시 서버에서 요청을 처리한 후 Text, XML 또는 JSON으로 응답.
 - client(Browser)에서는 이 응답 data를 이용하여 화면 전환 없이 현재 페이지에서 동적으로 화면을 재구성.

Ajax (Asynchronous Javascript And XML).

1-2. 서버와 클라이언트의 상호작용.

Confidential

- ✓ 웹 화면을 구성하는 방식은 서버 중심의 상호작용 방식과 클라이언트 중심의 상호작용 방식으로 구분.
- ✓ 서버 중심의 개발방식은 화면 구성이 서버에서 이루어 진다. (프리젠테이션 영역의 JSP나 PHP, ASP 등)
- ✓ 클라이언트 중심의 개발방식은 클라이언트(웹 브라우저)에서 화면을 구성한다. (주로 JavaScript)
- ✓ Ajax는 클라이언트 중심의 개발 방식이며 비동기 요청보다는 동적 화면구성이 관건임.



Javascript AJAX

Ajax (Asynchronous Javascript And XML).

1-3. Javascript AJAX.

Confidential

- ✓ XMLHttpRequest는 자바스크립트가 Ajax 방식으로 통신할 때 사용하는 객체.
- ✓ XMLHttpRequest 객체는 Ajax 통신 시 전송방식, 경로 등 전송정보를 싣는 역할.
- ✓ 실제 서버와의 통신은 브라우저의 Ajax 엔진에서 수행.
- ✓ 직접 자바스크립트로 Ajax를 프로그래밍할 경우 브라우저 별로 통신방식이 달라 코드가 복잡해 진다.

```
<script type="text/javascript">
var httpRequest = null;

function getXMLHttpRequest(){
    if(window.ActiveXObject){ // MS IE
        try{
            return new ActiveXObject("Msxml2.XMLHTTP");
        }catch(e1){
            try{
                return new ActiveXObject("Microsoft.XMLHTTP");
            }catch(e2){
                return null;
            }
        }
    }else if(window.XMLHttpRequest){ // 기타 웹 브라우저
        return new XMLHttpRequest();
    }else{
        return null;
    }
}
```

```
function sendRequest(url, params, callback, method){
    httpRequest = getXMLHttpRequest();

    var httpMethod = method ? method : 'GET';
    if(httpMethod != 'GET' && httpMethod != 'POST'){
        httpMethod = 'GET';
    }

    var httpParams = (params == null || params == '') ? null : params;
    var httpUrl = url;
    if(httpMethod == 'GET' && httpParams != null){
        httpUrl = httpUrl + "?" + httpParams;
    }

    // open() : 요청의 초기화, GET/POST 선택, 접속할 URL 입력
    httpRequest.open(httpMethod, httpUrl, true);
    httpRequest.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded');
    httpRequest.onreadystatechange = callback; // callback 함수 등록
    // send() : web server에 요청 전송.
    httpRequest.send(httpMethod == 'POST' ? httpParams : null);
}

</script>
```

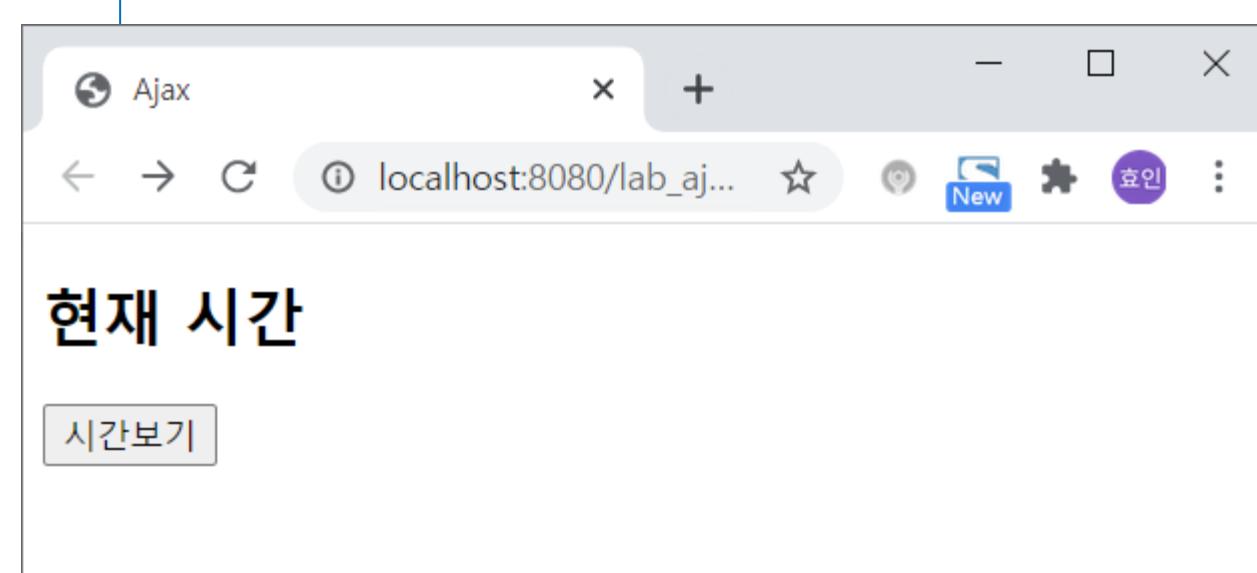
Ajax (Asynchronous Javascript And XML).

1-4. Javascript AJAX 적용 예.

Confidential

- ✓ Server의 시간을 출력하는 예제 (AJAX 미사용).

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Ajax</title>
    <script type="text/javascript">
        // client의 시간이 아닌 server의 시간보기.
        function viewTime() {
            location.href = "1-02currenttime.jsp";
        }
    </script>
</head>
<body>
    <h2>현재 시간</h2>
    <button onclick="javascript:viewTime();">시간보기</button>
</body>
</html>
```



1-02.html

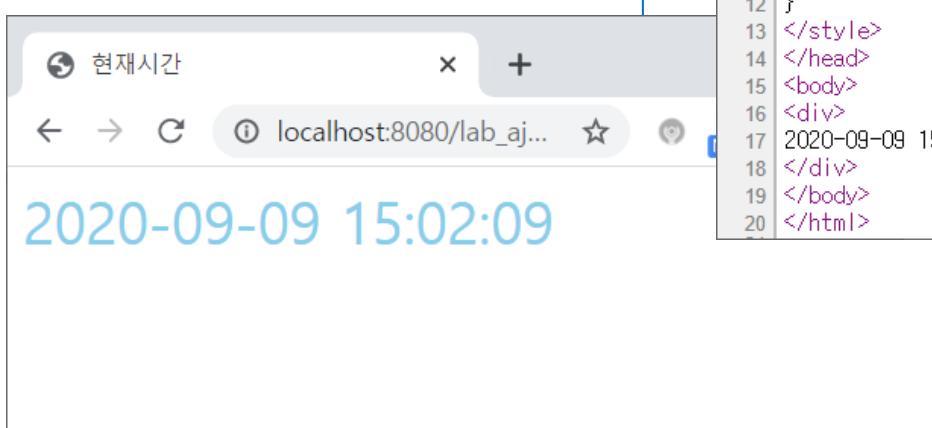
Ajax (Asynchronous Javascript And XML).

1-4. Javascript AJAX 적용 예.

Confidential

- ✓ Server의 시간을 출력하는 예제 (AJAX 미사용).

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" import="java.text.*,java.util.*"%>
<%
DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
String curTime = dateFormat.format(new Date());
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>현재시간</title>
<style type="text/css">
div {
    font-size: 30px;
    color: skyblue;
}
</style>
</head>
<body>
<div>
<%= curTime %>
</div>
</body>
</html>
```



```
1   <!DOCTYPE html>
2   <html>
3   <head>
4   <meta charset="UTF-8">
5   <title>현재시간</title>
6   <style type="text/css">
7   div {
8       font-size: 30px;
9       color: skyblue;
10  }
11  </style>
12  </head>
13  <body>
14  <div>
15  2020-09-09 15:02:09
16  </div>
17  </body>
18  </html>
```

Ajax (Asynchronous Javascript And XML).

1-4. Javascript AJAX 적용 예 : httpRequest의 속성값.

Confidential

✓ readyState.

값	의미	설명
0	Uninitialized	객체만 생성 (open 메소드 호출 전).
1	Loading	open 메소드 호출.
2	Loaded	send 메소드 호출. status의 헤더가 아직 도착되기 전 상태.
3	Interactive	데이터 일부를 받은 상태.
4	Completed	데이터 전부를 받은 상태.

✓ status.

값	텍스트(status Text)	설명
200	OK	요청 성공.
403	Forbidden	접근 거부.
404	Not Found	페이지 없음.
500	Internal Server Error	서버 오류 발생.

Ajax (Asynchronous Javascript And XML).

1-4. Javascript AJAX 적용 예.

Confidential

- ✓ Server의 시간을 출력하는 예제 (AJAX 사용).

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Ajax</title>
    <style type="text/css">
        .viewtime {
            font-size: 30px;
            color: magenta;
        }
    </style>
    <script type="text/javascript" src="../js/httpRequest.js"></script>
    <script type="text/javascript">
        // client의 시간이 아닌 server의 시간보기.
        function getTime() {
            sendRequest("1-03currenttime.jsp", null, viewTime, "GET");
        }
    </script>

```

1-03.html

```
function viewTime() {
    if(httpRequest.readyState == 4) {
        if(httpRequest.status == 200) {
            var time = httpRequest.responseText;
            var div = document.getElementById("curtime");
            div.setAttribute("class", "viewtime");
            div.innerHTML = time;
        }
    }
}

window.onload = function() {
    setInterval("getTime()", 1000);
}
</script>
</head>
<body>
    <h2>현재 시간</h2>
    <button onclick="javascript:getTime();">시간보기</button>
    <div id="curtime"></div>
</body>
</html>
```

Ajax (Asynchronous Javascript And XML).

1-4. Javascript AJAX 적용 예.

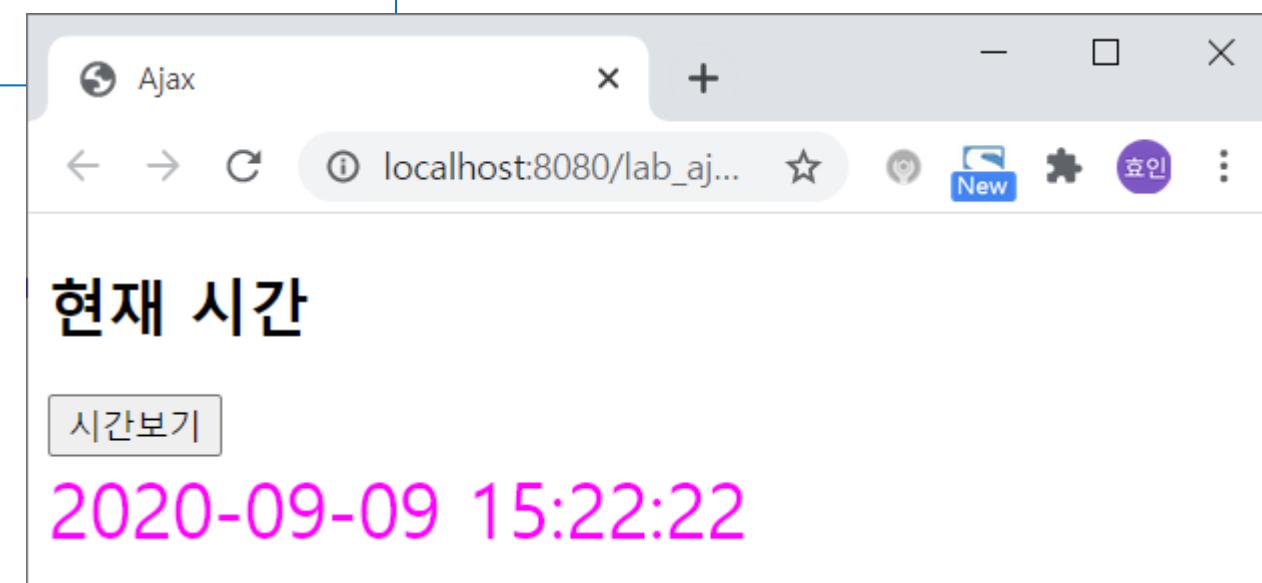
Confidential

- ✓ Server의 시간을 출력하는 예제 (AJAX 사용).

```
<%@ page language="java" contentType="text/plain; charset=UTF-8"
pageEncoding="UTF-8" import="java.text.*,java.util.*"%>
<%
DateFormat dateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
String curTime = dateFormat.format(new Date());
System.out.println(curTime);
%>
<%= curTime %>
```

1-03currenttime.jsp

```
2020-09-09 15:25:34
2020-09-09 15:25:35
2020-09-09 15:25:36
2020-09-09 15:25:37
2020-09-09 15:25:38
2020-09-09 15:25:39
2020-09-09 15:25:40
2020-09-09 15:25:41
2020-09-09 15:25:42
2020-09-09 15:25:43
```



jQuery AJAX

Ajax (Asynchronous Javascript And XML).

1-5. jQuery AJAX 함수 - `$.ajax()` (1/2).

Confidential

- ✓ `$.ajax()` 함수는 jQuery에서 Ajax 기능을 제공하는 가장 기본적인 함수.
- ✓ 다른 함수들에 비하여 옵션을 다양하게 지정할 수 있으며 실무에서 가장 많이 사용.
- ✓ 함수의 옵션은 다양하지만 대부분 자동으로 지정하므로 생략 가능.

`$.ajax(options);`

옵션 속성 이름	설명	자료형
async	동기, 비동기를 지정. (기본 비동기)	Boolean
url	대상 URL 지정.	String
data	요청 매개변수를 지정.	Object, String
type	'GET' 또는 'POST' 등을 지정.	String
success(data, status, xhr)	Ajax 성공 이벤트 리스너를 지정.	Function, Array
complete(xhr, status)	Ajax 완료 이벤트 리스너를 지정.	Function
error(xhr, status, error)	Ajax 실패 이벤트 리스너를 지정.	Function
timeout	만료 시간을 지정.	Number
jsonp	JSONP 매개변수 이름을 지정.	String
jsonpCallback	JSONP 콜백 함수 이름을 지정.	String, Function

Ajax (Asynchronous Javascript And XML).

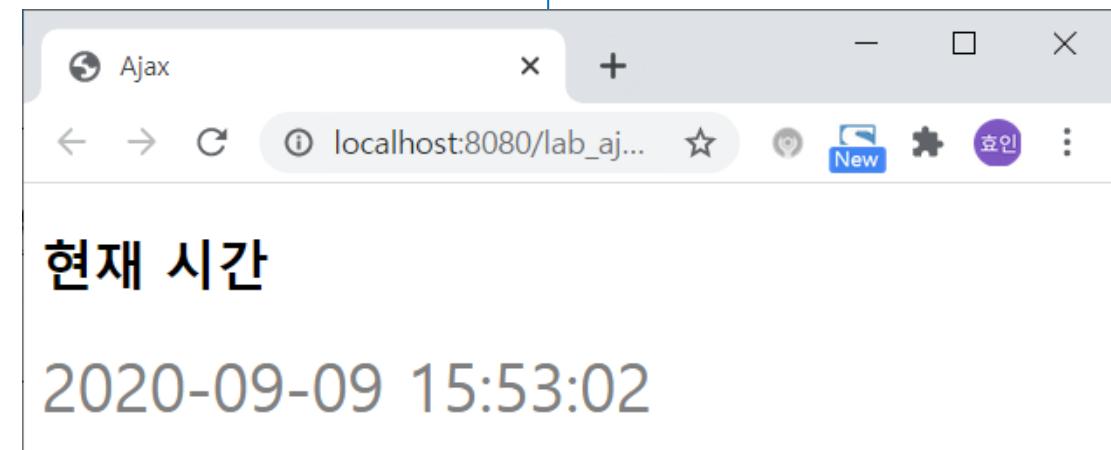
1-5. jQuery AJAX 함수 - `$.ajax()` (2/2).

Confidential

- ✓ Server의 시간을 출력하는 예제 (jQuery AJAX 사용).

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
// client의 시간이 아닌 server의 시간보기.
$(document).ready(function() {
    setInterval("viewTime()", 1000);
});

function viewTime() {
    $.ajax({
        url: "1-03currenttime.jsp",
        type: "GET",
        success: function(response) {
            $("#curtime").addClass("viewtime").empty().append(response);
        },
        error: function(xhr, status, msg){
            console.log("상태값 : " + status + " Http에러메시지 : "+msg);
        }
    });
}
</script>
```



Ajax (Asynchronous Javascript And XML).

1-5. jQuery AJAX 함수 - \$.get(), \$.post().

Confidential

- ✓ \$.get(), \$.post() 함수는 \$.ajax()의 옵션 속성 중 type 옵션이 미리 지정된 함수.
- ✓ 지정된 HTTP Method로 Ajax 통신을 하며 get()은 GET 방식을, post()는 POST 방식을 사용.

```
$.xxx(url, function (result, textStatus, jqXHR){});  
$.xxx(url, data, function (result, textStatus, jqXHR){});
```

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
<script type="text/javascript">  
// client의 시간이 아닌 server의 시간보기.  
$(document).ready(function() {  
    setInterval("viewTime()", 1000);  
});  
  
function viewTime() {  
    $.get("1-03currenttime.jsp", function(response) {  
        $("#curtime").addClass("viewtime").empty().append(response);  
    });  
}  
</script>
```

Ajax (Asynchronous Javascript And XML).

* 참고 : GET방식과 POST방식.

Confidential

✓ GET방식의 특징

- URL에 변수(데이터)를 포함시켜 요청한다.
- 데이터를 Header(헤더)에 포함하여 전송한다.
- URL에 데이터가 노출되어 보안에 취약하다.
- 전송하는 길이에 제한이 있다.
- 캐싱 할 수 있다. 캐싱이란 한번 접근 후, 또 요청할 시 빠르게 접근하기 위해 레지스터에 데이터를 저장시켜 놓는 것

✓ POST방식의 특징

- URL에 변수(데이터)를 노출하지 않고 요청한다.
- 데이터를 Body(바디)에 포함시킨다.
- URL에 데이터가 노출되지 않아서 기본 보안은 되어있다.
- 전송하는 길이에 제한이 없다.
- 캐싱 할 수 없다.

Ajax (Asynchronous Javascript And XML).

1-5. jQuery AJAX 함수 - \$(selector).load().

Confidential

- ✓ `$.load()` 함수는 서버로부터 내용을 조회하여, 선택자를 통해 탐색한 DOM 객체에 동적으로 삽입.
- ✓ 첫 번째 인자는 필수 값으로 HTML을 조회할 서버 URL을 지정.
- ✓ 두 번째 인자는 요청 시 서버에 전달할 데이터를 지정.
- ✓ 세 번째 인자는 서버와 통신을 완료한 후에 수행할 콜백함수를 지정.

```
$(selector).load(url);
$(selector).load(url, data, function (result, textStatus, jqXHR){});
```

```
<script
src="https://ajax.googleapis.com/ajax/Libs/jquery/3.5.1/jquery.min.js"></script>
<script type="text/javascript">
// client의 시간이 아닌 server의 시간보기.
$(document).ready(function() {
    setInterval("viewTime()", 1000);
});

function viewTime() {
    $("#curtime").load("1-03currenttime.jsp").addClass("viewtime");
}
</script>
```

1-6. 데이터 전송 형식 : CSV.

Confidential

- ✓ Server와 Client는 주고 받을 data의 형식을 맞춰야 함.
- ✓ 대표적인 data의 형식은 CSV, XML, JSON 등이 있음.

1. CSV (Comma Separated Values)

- 각 항목을 쉼표(,)로 구분해 데이터를 표현하는 방법.
- 다른 두 형식에 비해 굉장히 짧다. 많은 양의 데이터 전송 시 유리.
- 단, 각각의 데이터가 어떤 내용인지 파악하기 어렵다. (서버와 클라이언트가 완벽히 데이터 구조를 공유할 경우 가능)
- 예.
 - 20201111,김싸피,A,90
 - 20201112,홍싸피,B,92
 - 20201113,박싸피,C,91

Ajax (Asynchronous Javascript And XML).

1-6. 데이터 전송 형식: CSV.

Confidential

1. CSV (html)

1-05.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>CSV</title>
    <style type="text/css">
      table {
        width: 300px;
        height: 100px;
      }
      th, td {
        text-align: center;
      }
    </style>
    <script
      src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("#listBtn").click(function() {
          // server에서 넘어온 data
          $.ajax({
            url:"1-05csv.jsp",
            type:"GET",
            success:function(response) {
              makeList(response);
            },
            error:function(xhr,status,msg){
              console.log("상태값 : " + status + " Http에러메시지 : "+msg);
            }
          });
      });
    </script>
  </head>
  <body>
    <h3>SSAFY 분반</h3>
    <button id="ListBtn">학생정보보기</button>
    <table>
      <tr>
        <th>학번</th>
        <th>이름</th>
        <th>분반</th>
        <th>성적</th>
      </tr>
    </table>
  </body>
</html>
```

```
function makeList(data) {
  var students = data.split("\n");
  for(var i=0;i<students.length;i++) {
    var student = students[i].split(",");
    var tr = $('<tr></tr>');
    for(var j=0;j<student.length;j++) {
      //console.log(student[j]);
      $('<td>' + student[j] + '</td>').appendTo(tr);
    }
    $('#studentinfo').append(tr);
  }
  $("tr:first").css('background', 'darkgray').css('color', 'white');
  $("tr:odd").css('background', 'lightgray');
}

</script>
</head>
<body>
<h3>SSAFY 분반</h3>
<button id="ListBtn">학생정보보기</button>
<table>
<tr>
  <th>학번</th>
  <th>이름</th>
  <th>분반</th>
  <th>성적</th>
</tr>
<tbody id="studentinfo"></tbody>
</table>
</body>
</html>
```

Ajax (Asynchronous Javascript And XML).

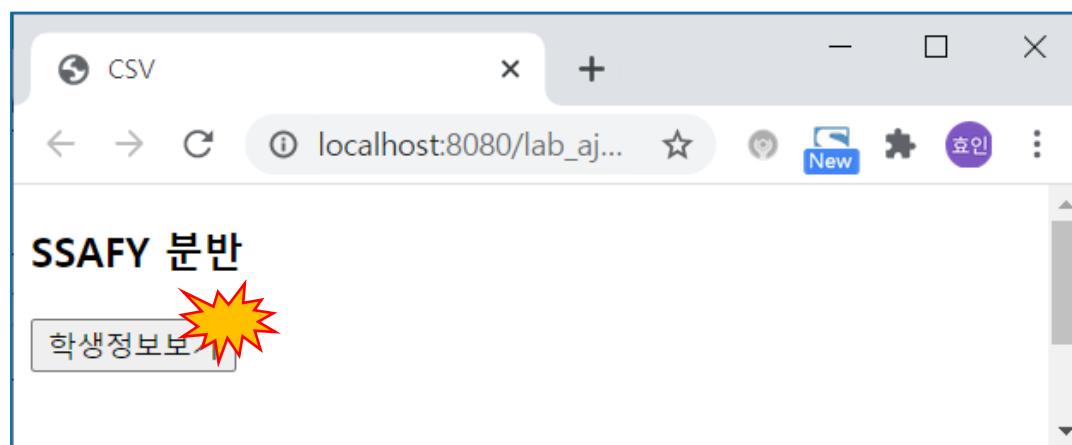
1-6. 데이터 전송 형식 : CSV.

Confidential

1. CSV (jsp, 결과화면)

```
<%@ page language="java" contentType="text/plain; charset=UTF-8"
pageEncoding="UTF-8"%>
20201111, 김싸피, A, 90
20201112, 홍싸피, B, 92
20201113, 박싸피, C, 91
```

1-05csv.jsp

A screenshot of a browser window titled "CSV". The address bar shows "localhost:8080/lab_aj...". The main content area displays the parsed CSV data in a table:
SSAFY 분반
학생정보보기

학번	이름	분반	성적
20201111	김싸피	A	90
20201112	홍싸피	B	92
20201113	박싸피	C	91

Ajax (Asynchronous Javascript And XML).

1-6. 데이터 전송 형식 : XML.

- ✓ server와 client는 주고 받을 data의 형식을 맞춰야 함.
- ✓ 대표적인 data의 형식은 CSV, XML, JSON 등이 있음.

2. XML(eXtensible Markup Language).

- xml은 tag로 data를 표현함.
- tag를 보면 각 data가 무엇을 의미하는지 파악 가능.
- tag에 사용자 정의 속성을 넣을 수 있으므로 복잡한 data 전달 가능.
- 예.

Confidential

```
<?xml version="1.0" encoding="UTF-8"?>

<students>
    <student>
        <id>20201111</id>
        <name>김싸피</name>
        <class>A</class>
        <grade>90</grade>
    </student>
    <student>
        <id>20201112</id>
        <name>홍싸피</name>
        <class>B</class>
        <grade>92</grade>
    </student>
    <student>
        <id>20201113</id>
        <name>박싸피</name>
        <class>C</class>
        <grade>91</grade>
    </student>
</students>
```

Ajax (Asynchronous Javascript And XML).

1-6. 데이터 전송 형식: XML.

Confidential

2. XML(html).

1-06.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>CSV</title>
    <style type="text/css">
      table {
        width: 300px;
        height: 100px;
      }
      th, td {
        text-align: center;
      }
    </style>
    <script
      src="https://ajax.googleapis.com/ajax/Libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("#listBtn").click(function() {
          // server에서 넘어온 data
          $.ajax({
            url:"1-06xml.jsp",
            type:"GET",
            dataType: "xml",
            success:function(response) {
              makeList(response);
            },
            error:function(xhr,status,msg){
              console.log("상태값 : " + status + " Http에러메시지 : "+msg);
            }
          });
      });
    </script>
  </head>
  <body>
    <h3>SSAFY 분반</h3>
    <button id="listBtn">학생정보보기</button>
    <table>
      <thead>
        <tr>
          <th>학번</th>
          <th>이름</th>
          <th>분반</th>
          <th>성적</th>
        </tr>
      </thead>
      <tbody id="studentinfo"></tbody>
    </table>
  </body>
</html>
```

```
function makeList(data) {
  var studentlist = '';
  $(data).find('student').each(function(){
    studentlist += '<tr>';
    studentlist += '<td>' + $(this).find("id").text() + '</td>';
    studentlist += '<td>' + $(this).find("name").text() + '</td>';
    studentlist += '<td>' + $(this).find("class").text() + '</td>';
    studentlist += '<td>' + $(this).find("grade").text() + '</td>';
    studentlist += '</tr>';
    //alert(studentlist);
    $("#studentinfo").append(studentlist);
  });
  $("tr:first").css('background', 'darkgray').css('color', 'white');
  $("tr:even").css('background', 'lightgray');
}

</script>
</head>
<body>
<h3>SSAFY 분반</h3>
<button id="listBtn">학생정보보기</button>
<table>
  <thead>
    <tr>
      <th>학번</th>
      <th>이름</th>
      <th>분반</th>
      <th>성적</th>
    </tr>
  </thead>
  <tbody id="studentinfo"></tbody>
</table>
</body>
</html>
```

Ajax (Asynchronous Javascript And XML).

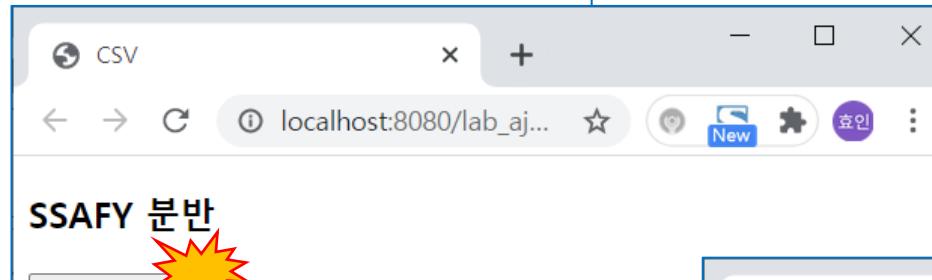
1-6. 데이터 전송 형식 : XML.

Confidential

2. XML(jsp, 결과화면).

1-06xml.jsp

```
<?xml version="1.0" encoding="UTF-8"?>
<%@ page language="java" contentType="text/xml; charset=UTF-8"
   pageEncoding="UTF-8"%>
<students>
  <student>
    <id>20201111</id>
    <name>김싸피</name>
    <class>A</class>
    <grade>90</grade>
  </student>
  <student>
    <id>20201112</id>
    <name>홍싸피</name>
    <class>B</class>
    <grade>92</grade>
  </student>
  <student>
    <id>20201113</id>
    <name>박싸피</name>
    <class>C</class>
    <grade>91</grade>
  </student>
</students>
```



SSAFY 분반			
학생정보보기			
학번	이름	분반	성적
20201111	김싸피	A	90
20201112	홍싸피	B	92
20201113	박싸피	C	91

Ajax (Asynchronous Javascript And XML).

1-6. 데이터 전송 형식 : JSON.

Confidential

- ✓ server와 client는 주고 받을 data의 형식을 맞춰야 함.
- ✓ 대표적인 data의 형식은 CSV, XML, JSON 등이 있음.

3. JSON (JavaScript Object Notation).

- CSV와 XML의 단점을 극복한 형식.
- Javascript에서 사용하는 객체의 형식으로 data를 표현.
- Ajax 사용시 거의 표준으로 사용되는 data 표현 방식.
- 예.

```
[  
 {  
   "id" : "20201111",  
   "name" : "김싸피",  
   "class" : "A",  
   "grade" : "90"  
 },  
 {  
   "id" : "20201112",  
   "name" : "홍싸피",  
   "class" : "B",  
   "grade" : "92"  
 },  
 {  
   "id" : "20201113",  
   "name" : "박싸피",  
   "class" : "C",  
   "grade" : "91"  
 }]
```

1-07.json

Ajax (Asynchronous Javascript And XML).

1-6. 데이터 전송 형식 : JSON.

Confidential

3. JSON (html).

1-07.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>CSV</title>
    <style type="text/css">
      table {
        width: 300px;
        height: 100px;
      }
      th, td {
        text-align: center;
      }
    </style>
    <script
      src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script type="text/javascript">
      $(document).ready(function() {
        $("#listBtn").click(function() {
          $.ajax({
            url:"1-07json.jsp",
            type:"GET",
            contentType:'application/json;charset=utf-8',
            dataType:'json',
            success:function(response){
              makeList(response);
            },
            error:function(xhr,status,msg){
              console.log("상태값 : " + status + " Http에러메시지 : "+msg);
            }
          });
        });
      });
    </script>
  </head>
  <body>
    <h3>SSAFY 분반</h3>
    <button id="listBtn">학생정보보기</button>
    <table>
      <thead>
        <tr>
          <th>학번</th>
          <th>이름</th>
          <th>분반</th>
          <th>성적</th>
        </tr>
      </thead>
      <tbody id="studentinfo"></tbody>
    </table>
  </body>
</html>
```

```
function makeList(data) {
  //alert(data);
  var studentlist = '';
  $.each(data, function(index, item){
    studentlist = '<tr>';
    studentlist += '<td>' + item.id + '</td>';
    studentlist += '<td>' + item.name + '</td>';
    studentlist += '<td>' + item.class + '</td>';
    studentlist += '<td>' + item.grade + '</td>';
    studentlist += '</tr>';
    $("#studentinfo").append(studentlist);
  });
  $("tr:first").css('background', 'black').css('color', 'white');
  $("tr:odd").css('background', 'lightgray');
}
</script>
</head>
<body>
<h3>SSAFY 분반</h3>
<button id="listBtn">학생정보보기</button>
<table>
  <thead>
    <tr>
      <th>학번</th>
      <th>이름</th>
      <th>분반</th>
      <th>성적</th>
    </tr>
  </thead>
  <tbody id="studentinfo"></tbody>
</table>
</body>
</html>
```

Ajax (Asynchronous Javascript And XML).

1-6. 데이터 전송 형식 : JSON.

Confidential

3. JSON (jsp, 결과화면).

```
<%@ page language="java" contentType="application/json; charset=UTF-8"
pageEncoding="UTF-8"%>
```

```
[  
 {  
   "id" : "20201111",  
   "name" : "김싸피",  
   "class" : "A",  
   "grade" : 90  
 },  
 {  
   "id" : "20201112",  
   "name" : "홍싸피",  
   "class" : "B",  
   "grade" : 92  
 },  
 {  
   "id" : "20201113",  
   "name" : "박싸피",  
   "class" : "C",  
   "grade" : 91  
 }]
```

1-07json.jsp

The image displays two browser windows side-by-side, both titled "SSAFY 분반". The left window shows a table with four columns: 학번 (Student ID), 이름 (Name), 분반 (Class), and 성적 (Grade). It contains three rows of data corresponding to the JSON objects in the code above. The right window also shows the same table structure but with only one row of data. A yellow starburst effect is applied to the "학생정보보기" button in the left window, indicating it is the active or target element for the JSON response.

학번	이름	분반	성적
20201111	김싸피	A	90
20201112	홍싸피	B	92
20201113	박싸피	C	91

Ajax (Asynchronous Javascript And XML).

1-7. Event 관리(전역함수).

Confidential

- ✓ Ajax는 서버와 통신하는 과정이 웹 브라우저 내부에서 이루어지므로 사용자가 진행상황을 알기 어렵다.
- ✓ Ajax 전역함수를 사용하여 Ajax 처리 중에 진행 상태를 보여주는 기능을 구현할 수 있다.
- ✓ jQuery는 Ajax 처리가 이루어지는 각 단계 별로 전역함수를 호출.
- ✓ 단, jQuery의 전역함수는 `$.ajaxSetup()` 함수에 global 프로퍼티 설정이 `true`인 경우에만 수행됨. (디폴트 : `true`)

전역함수	호출시점	매개변수
ajaxStart	Ajax함수 실행, XHR 객체 생성 전.	전역 콜백정보 객체.
ajaxSend	XHR 객체 생성 뒤, 서버전송 전.	전역 콜백정보 객체, XHR 인스턴스, <code>\$.ajax()</code> 가 사용하는 프로퍼티.
ajaxSuccess	서버요청 반환, 응답이 성공인 상태.	전역 콜백정보 객체, XHR 인스턴스, <code>\$.ajax()</code> 가 사용하는 프로퍼티.
ajaxError	서버요청 반환, 응답이 실패인 상태.	전역 콜백정보 객체, XHR 인스턴스, <code>\$.ajax()</code> 가 사용하는 프로퍼티. XHR 인스턴스가 반환한 예외객체.
ajaxComplete	서버요청 반환, ajaxSuccess 또는 ajaxError 호출된 후.	전역 콜백정보 객체, XHR 인스턴스, <code>\$.ajax()</code> 가 사용하는 프로퍼티.
ajaxStop	모든 Ajax 진행이 완료, 다른 전역 콜백이 호출된 후.	전역 콜백정보 객체.

전역함수는 `document` 객체에만 사용할 수 있다.

Ajax (Asynchronous Javascript And XML).

1-7. Event 관리(전역함수).

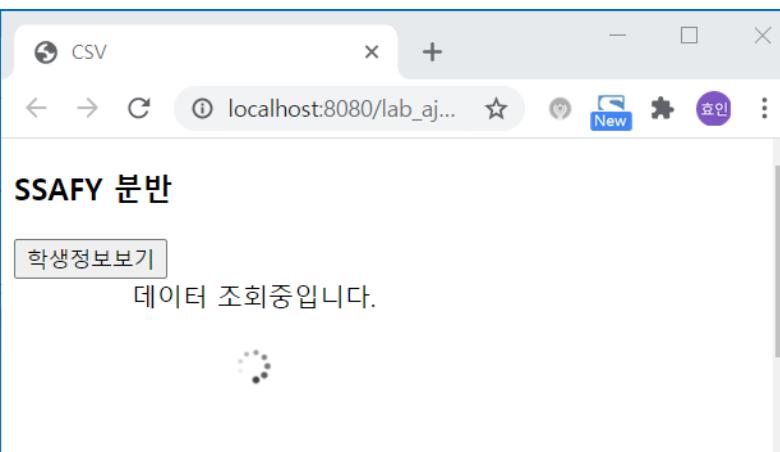
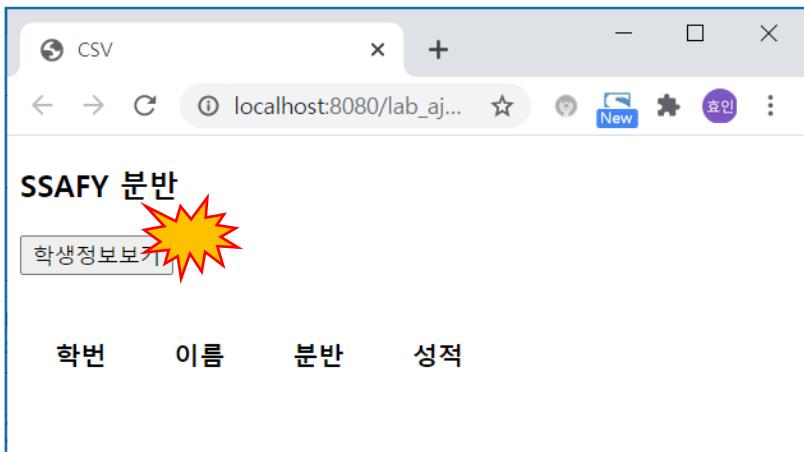
Confidential

- ✓ 로딩화면 구현.

```
<div id="loading" style="display: none;">  
데이터 조회중입니다.<br>   
</div>
```

```
$(document).ajaxStart(function() {  
    $("#loading").fadeIn();  
}).ajaxStop(function() {  
    $("#loading").hide();  
});
```

1-08.jsp



학번	이름	분반	성적
20201111	김싸피	A	90
20201112	홍싸피	B	92
20201113	박싸피	C	91

Bootstrap

반응형 웹

1-1. 반응형 웹의 장단점.

Confidential

- ✓ 모든 스마트 기기에서 접속 가능.
 - 반응형 웹에서 사용하는 기술들은 W3C에서 웹 표준으로 지정한 HTML과 CSS로 이루어져 있음.
 - 스마트 워치 같은 웨어러블 기기 뿐 아니라 스마트 TV나 게임 콘솔 등 웹표준을 지원하는 모든 기기 사용 가능.
- ✓ 가로 모드에 맞추어 레이아웃 변경 가능.
 - 스마트폰이나 태블릿에서 가로 모드로 돌렸을 때 너비 값이 커지면 그에 맞추어 레이아웃을 자동으로 변경.
- ✓ 사이트 유지, 관리 용이.
 - PC용 Mobile용 코드가 따로 있는 것이 아니기 때문에 유지, 관리가 쉽다.
 - 서버 쪽 코드가 아닌 HTML과 CSS로만 되어 있어 복잡하지 않다.

1-2. 뷰포트 (viewport).

Confidential

- ✓ PC화면에 보이는 내용을 모바일 기기에서 그대로 볼 수 없는 이유는 PC화면과 모바일 화면의 픽셀 표현 방법이 다르기 때문.
- ✓ 뷰포트를 지정하면 접속한 기기 화면에 맞추어 확대 또는 축소해 표시할 수 있다.
- ✓ 이때 ‘뷰포트’란 스마트폰 화면에서 실제 내용이 표시되는 영역을 말한다.
- ✓ 웹킷(webkit) 기반인 모바일 브라우저들의 기본 뷰포트 너비는 980px이다. 화면의 크기를 고려하여 320px로 맞추어 웹 사이트를 제작 하더라도 스마트폰의 모바일 브라우저의 기본 뷰포트 너비가 980px이므로 글씨와 그림은 작아진다.
- ✓ 이를 해결하기 위해 뷰포트를 지정.

1. 반응형 웹.

1-2. 뷰포트 (viewport).

Confidential

- ✓ 뷰포트 지정.

```
<meta name="viewport" content="[속성1=값], [속성2=값], …">
```

속성	설명	사용 가능한 값	기본 값
width	뷰포트 너비	device-width 또는 크기	브라우저 기본 값
height	뷰포트 높이	device-height 또는 크기	브라우저 기본 값
user-scalable	확대/축소 가능 여부	yes 또는 no	yes
initial-scale	초기 확대/축소 값	1 ~ 10	1
minimum-scale	최소 확대/축소 값	0 ~ 10	0.25
maximum-scale	최대 확대/축소 값	0 ~ 10	1.6

1. 반응형 웹.

1-2. 뷰포트 (viewport).

Confidential

- ✓ `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- ✓ `<meta name="viewport" content="width=device-width, initial-scale=0.5">`

The image displays two side-by-side browser windows showing a responsive web application for 'SSAFY Cafe'. The left window shows the application with a viewport configuration of `initial-scale=1.0`, resulting in a wider layout where menu items and descriptions are visible. The right window shows the application with a viewport configuration of `initial-scale=0.5`, resulting in a narrower layout where some menu items and descriptions are cut off or require scrolling.

Left Window (initial-scale=1.0):

- Header:** SSAFY Cafe
- Left Sidebar:** [인기메뉴], [신메뉴], [당신의 선택], 최애 메뉴를 골라
- Main Content:** 카페메뉴 (카페라떼, 카푸치노, 에스프레소, 카페모카), 신메뉴 (카페라떼, 카푸치노, 에스프레소, 카페모카)
- Right Sidebar:** [인기메뉴], [최신글]

Right Window (initial-scale=0.5):

- Header:** SSAFY Cafe
- Left Sidebar:** [인기메뉴], [신메뉴], [당신의 선택], 최애 메뉴를 골라
- Main Content:** 카페메뉴 (카페라떼, 카푸치노, 에스프레소, 카페모카), 신메뉴 (카페라떼, 카푸치노, 에스프레소, 카페모카)
- Right Sidebar:** [인기글], [최신글]

DevTools Comparison:

- Left DevTools:** Shows the meta tag `<meta name="viewport" content="width=device-width, initial-scale=1.0">`.
- Right DevTools:** Shows the meta tag `<meta name="viewport" content="width=device-width, initial-scale=0.5">`.

Bootstrap

1-1. Bootstrap?

Confidential

- ✓ Bootstrap은 빠르고 쉬운 웹 개발을 위한 무료 프런트 엔드 프레임워크이다.
- ✓ Bootstrap에는 typography, forms, buttons, tables, navigation, modals, image 및 기타 여러 가지를 위한 HTML 및 CSS 기반 디자인 템플릿과 선택적 JavaScript 플러그인이 포함되어 있다.
- ✓ Bootstrap은 또한 반응형 디자인을 쉽게 만들 수 있는 기능을 제공한다.

1-2. Bootstrap 장점.

Confidential

- ✓ 사용하기 쉬움 : HTML과 CSS에 대한 기본적인 지식만 있으면 누구나 Bootstrap을 사용할 수 있다.
- ✓ 반응형 기능 : Bootstrap의 반응형 CSS는 휴대폰, 태블릿 및 데스크톱에 맞게 조정된다.
- ✓ 모바일 우선 접근 방식 : Bootstrap에서 모바일 우선 스타일은 핵심 프레임워크의 일부이다.
- ✓ 브라우저 호환성 : Bootstrap 4는 모든 최신 브라우저 (Chrome, Firefox, Internet Explorer 10+, Edge, Safari 및 Opera) 와 호환된다.

2. Bootstrap.

1-3. Bootstrap 사용.

Confidential

✓ Bootstrap CDN.

```
<!-- Latest compiled and minified CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<!-- jQuery library -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<!-- Popper JS -->
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"></script>

<!-- Latest compiled JavaScript -->
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
```

✓ Bootstrap Download.

- <https://getbootstrap.com/>에서 다운로드.

✓ Bootstrap 4는 모바일 장치에 반응하도록 설계되었다.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

2. Bootstrap.

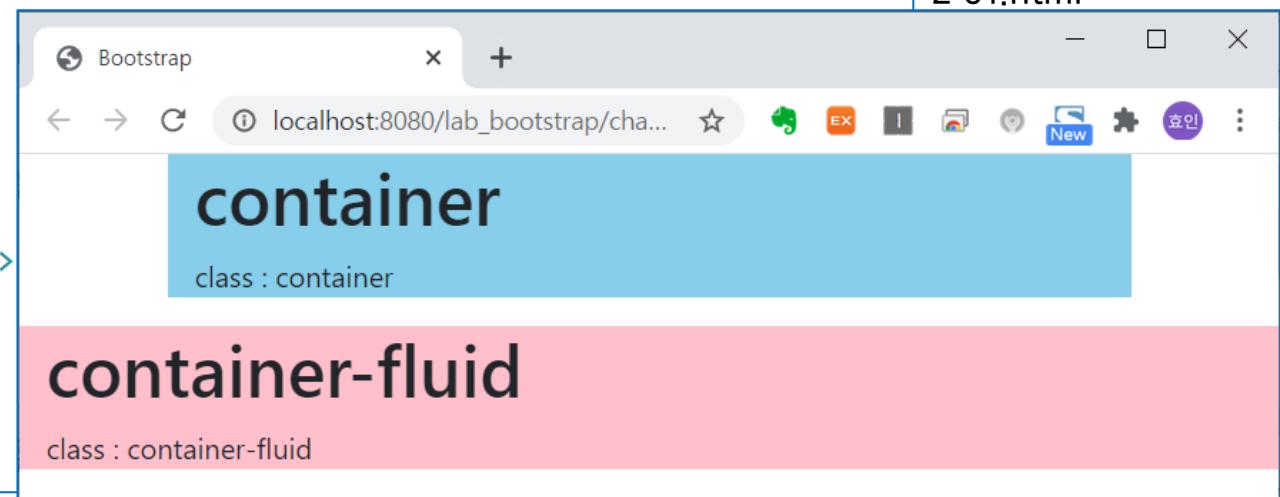
1-3. Container.

Confidential

- ✓ .container 클래스는 반응형 고정 너비 컨테이너를 제공.
- ✓ .container-fluid 클래스는 뷰포트의 전체 너비에 걸쳐 있는 전체 너비 컨테이너를 제공.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Bootstrap</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</head>
<body>
    <div class="container" style="background: skyblue;">
        <h1>container</h1>
        <p>class : container</p>
    </div>
    <div class="container-fluid" style="background: pink;">
        <h1>container-fluid</h1>
        <p>class : container-fluid</p>
    </div>
</body>
</html>
```

2-01.html



2. Bootstrap.

1-5. Grid System.

Confidential

- ✓ 부트 스트랩의 그리드 시스템은 플렉스 박스로 구축되어 페이지에 최대 12 개의 열을 허용한다.
- ✓ 12 개 열을 모두 개별적으로 사용하지 않으려면 열을 함께 그룹화하여 더 넓은 열을 만들 수 있다.

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4				span 4				span 4				
span 4				span 8								
span 6						span 6						
span 12												

https://www.w3schools.com/bootstrap4/bootstrap_grid_basic.asp

1-6. Grid Class.

Confidential

- ✓ 클래스를 결합하여 보다 동적이고 유연한 레이아웃을 만들 수 있다.

class	device	설명 (screen width 대비)
.col-num	extra small	< 576px
.col-sm-num	small	$\geq 576px$
.col-md-num	medium	$\geq 768px$
.col-lg-num	large	$\geq 992px$
.col-xl-num	extra large	$\geq 1200px$

2. Bootstrap.

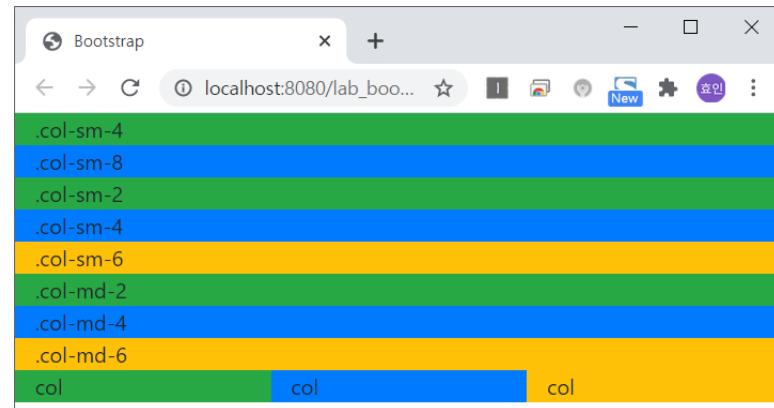
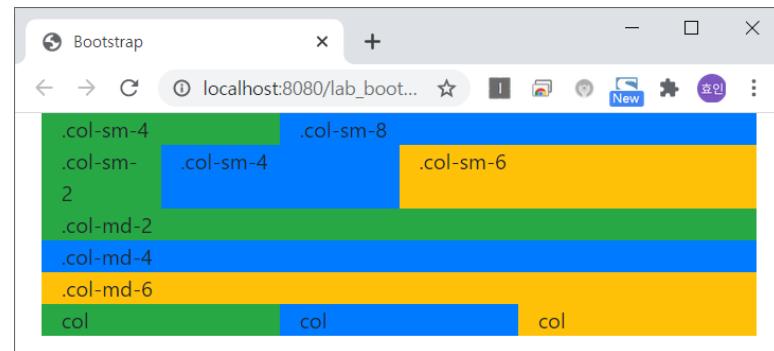
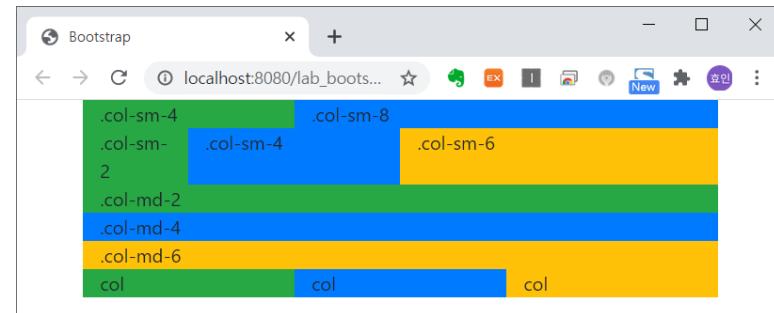
1-6. Grid Class

✓ 적용 예.

```
<div class="container">
  <div class="row">
    <div class="col-sm-4 bg-success">.col-sm-4</div>
    <div class="col-sm-8 bg-primary">.col-sm-8</div>
  </div>
  <div class="row">
    <div class="col-sm-2 bg-success">.col-sm-2</div>
    <div class="col-sm-4 bg-primary">.col-sm-4</div>
    <div class="col-sm-6 bg-warning">.col-sm-6</div>
  </div>
  <div class="row">
    <div class="col-md-2 bg-success">.col-md-2</div>
    <div class="col-md-4 bg-primary">.col-md-4</div>
    <div class="col-md-6 bg-warning">.col-md-6</div>
  </div>

  <div class="row">
    <div class="col bg-success">col</div>
    <div class="col bg-primary">col</div>
    <div class="col bg-warning">col</div>
  </div>
</div>
```

Confidential



창의너비를점점축소

내일
방송에서
만나요!

삼성 청년 SW 아카데미