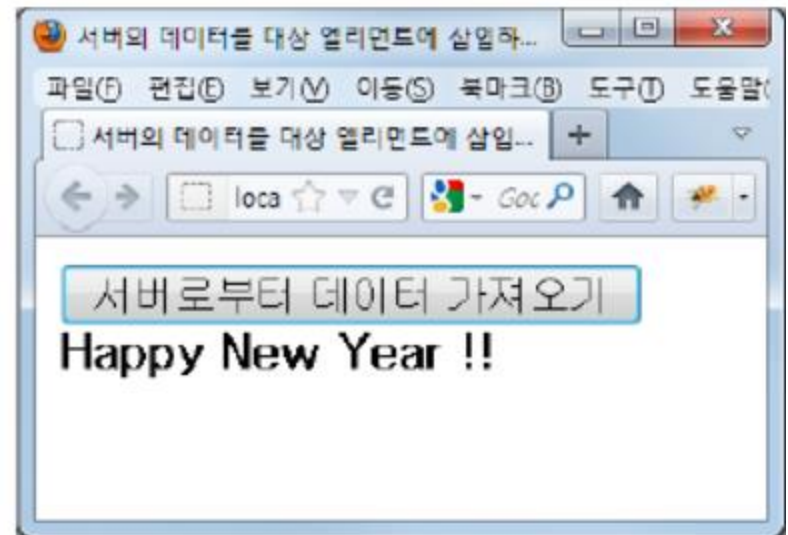
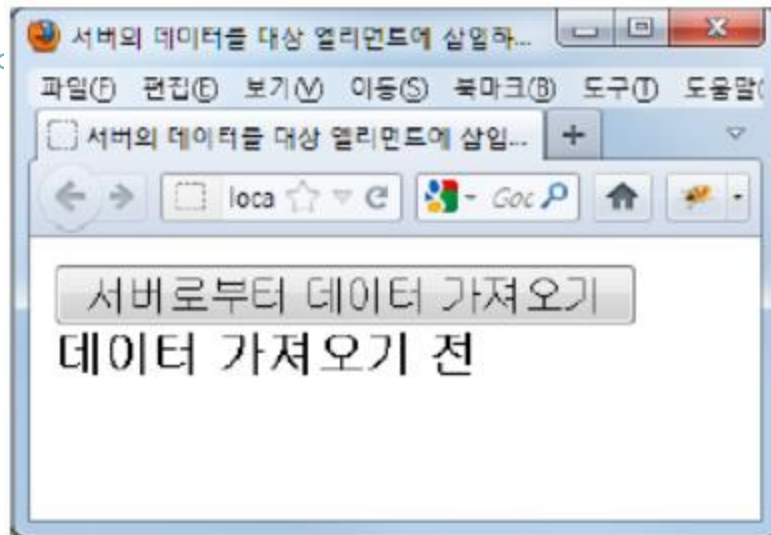


19장. Ajax2(jQuery)

서버의 데이터를 대상 엘리먼트에 삽입하기

페이지를 동적으로 로드하기 위해서는 load() 메소드를 호출한다



`load(url[, data][, success])`

1. url은 요청할 URL 주소이고, data는 요청할 때 서버에 보낼 자바 스크립트 객체 맵이나 문자열 형식의 데이터이다.
2. success는 요청이 성공했을 때 호출할 콜백 함수이다.
3. load() 메소드는 완료된 응답을 일치하는 집합에 있는 엘리먼트의 콘텐츠로 삽입한다.

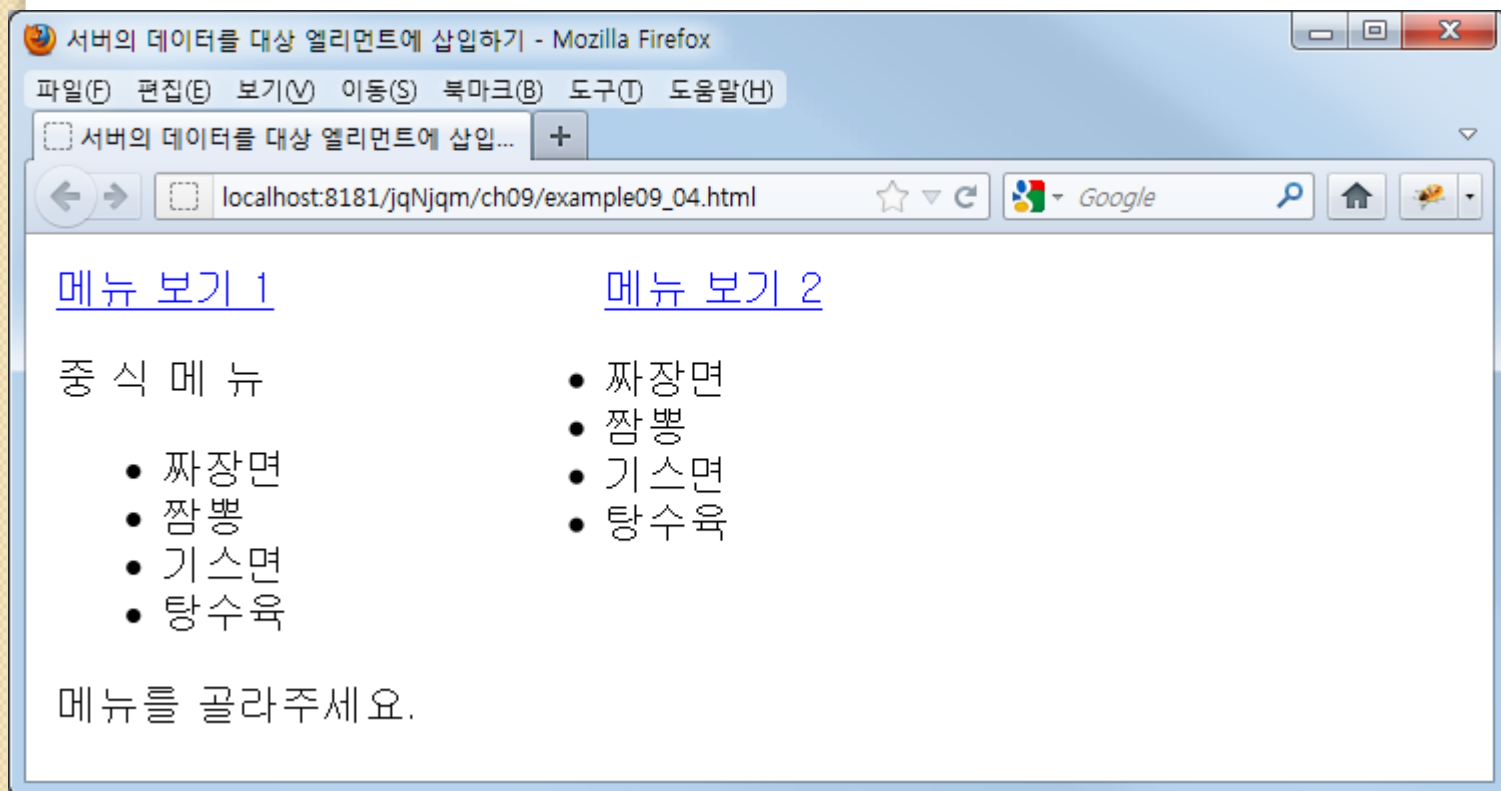
`$("#container").load("resource");`

Ajax 요청 직접 제어하기

- jQuery에서는 \$.ajax()라는 유틸리티 함수로 Ajax 요청을 설정 제어한다.

\$.ajax(options)

- 요청의 생성 방법과 통보 받을 콜백을 제어하고자 전달된 options 를 사용하여 Ajax 요청을 전송한다. 반환값은 XHR 인스턴스이다.



Ajax 요청 직접 제어하기

1. url (String)
 - 요청 URL
2. type (String)
 - 사용할 HTTP 메서드, 일반적으로 POST나 GET을 사용한다. 생략하면 기본값으로 GET을 사용한다.
3. data (Object)
 - 요청에 전달되는 프로퍼티를 가진 객체. GET 요청이면 데이터는 쿼리 문자열로 제공된다. POST 요청이면 데이터는 요청의 본문으로 제공된다. 두 경우 모두 \$.ajax() 유틸리티 함수가 값의 인코딩을 처리한다

Ajax 요청 직접 제어하기

4. dataType (String)

- 응답의 결과로 반환되는 데이터의 종류를 식별하는 키워드

1] xml - 응답 텍스트는 XML 문서로 파싱되며, 콜백에 결과로 생성된 XML DOM을 전달

2] html - 응답 텍스트는 처리 과정 없이 콜백 함수로 전달된다.

반환된 HTML 코드에 있는 모든 `<script>` 블록이 평가된다.

3] json - 응답 텍스트는 JSON 문자열로 평가되며, 생성된 객체는 콜백에 전달된다.

4] jsonp - 원격 스크립트를 허용한다는 점을 제외하고는 json 과 유사하다.

원격 서버가 이와 같은 방식을 지원한다고 가정한다.

5] script - 응답 텍스트는 콜백에 전달된다. 응답은 모든 콜백의 호출보다 먼저 자바스크립트 구문으로 처리된다.

6] text - 응답 텍스트는 일반 텍스트다.

5. timeout (Number)

- Ajax 요청의 제한 시간을 밀리초 단위로 설정한다. 제한 시간 안에 요청이 완료되지 않으면 요청을 취소하거나, error 콜백이 정의되어 있다면 호출

6. global (Boolean)

- true나 false에 따라 전역 함수(global function)를 활성화하거나 비활성화한다.

전역 함수는 엘리먼트에 덧붙일 수 있으며 Ajax 호출 동안 다양한 위치나 조건에서 실행된다. 값이 생략되면 기본 값으로 전역 함수를 활성화한다.

7. contentTypeString

- 요청에 명시되는 콘텐츠 타입. 생략하면 'application/x-www-form-urlencoded'가 기본으로 설정되며, 이는 폼 전송이 기본으로 사용하는 타입과 동일하다.

Ajax 요청 직접 제어하기

8. success (Function)

- 응답이 성공 상태 코드를 반환하면 호출되는 함수. 응답 본문은 이 함수의 첫 번째 매개 변수로 전달되며, dataType 프로퍼티에 명시한 형태로 구성된다. 두 번째 매개 변수는 상태 값을 나타내는 문자열이며, 이번 경우에는 항상 'success' 다.

9. error (Function)

- 응답이 에러 상태 코드를 반환하면 호출되는 함수. 매개 변수가 세 개 전달되는데, 각각 XHR 인스턴스, 상태 값이 항상 'error' 인 메시지 문자열, 선택사항으로 XHR 인스턴스가 반환한 예외 객체

10. complete (Function)

- 요청이 완료되면 호출되는 함수, 매개 변수 두 개가 전달되는데, 각각 XHR 인스턴스와 'success' 혹은 'error'를 나타내는 상태 메시지 문자열이다. success 혹은 'error'를 나타내는 상태 메시지 문자열이다. success 나 error 콜백을 명시했다면, 이 함수는 해당 콜백이 호출된 후에 실행된다.

11. beforeSend (Function)

- 요청이 전송되기 전에 먼저 호출되는 함수. 이 함수는 XHR 인스턴스를 전달 받으며, 사용자 정의 헤더를 설정하거나 요청 전에 필요한 연산을 수행하는 데 사용할 수 있다.

Ajax 요청 직접 제어하기

12. async (Boolean)

- false 이면 요청이 동기 호출로 전송된다. 기본은 비동기 요청이다.

13. processDataBoolean

- false로 설정되면, URL 인코딩된 형태로 처리되어 전달된 데이터를 금지한다.
기본 값은 데이터가 'application/x-www-form-urlencoded' 타입의 요청에 사용하는 형태의 URL로 인코딩된다.

14. ifModified Boolean

- true 일 때 Last-Modified 헤더를 확인하여 마지막 요청 이후에 응답 콘텐츠가 변경되지 않았다면 요청이 성공한다. 만일 생략하면 헤더를 확인하지 않는다.

JSON 이용하기

1. JavaScript Object Notion의 약어
2. XML 데이터를 대신하기 위해서 사용한다.
3. 키와 값을 쌍으로 가지는 구조
4. 배열을 사용할 경우에는 대괄호 안에 중괄호를 사용하여 조합

5. JSON에 저장되는 정보의 형태

- 배열
 - 대괄호 안에 값을 콤마로 구분하여 나열하며, 대괄호 안에 나오는 순서대로 배열 요소의 순서가 매겨진다.
 - [1, 2, 3]
- 객체
 - 중괄호 안에 있는 이름: 값의 형태로 멤버 하나를 표현
 - 각 멤버는 콤마로 구분
 - 순서가 아닌 이름으로 읽기 때문에 멤버의 순서는 의미가 없다.
 - {"name": "레몬" }

```
[
  {
    "id": "1",
    "name": "레몬",
    "price": " 3000",
    "description": "레몬에 포함되어 있는 쿠엔산은 피로회복에 좋다. "
  },
  {
    "id": "2",
    "name": "키위",
    "price": " 2000",
    "description": "비타민C가 매우 풍부하다."
  }
]
```


XML 파일을 GET 방식으로 로드하기-\$.get()

\$.get(URL, fn)

- GET 방식으로 서버와 통신하는 jQuery Ajax 함수
- URL로 지정한 파일을 로드해서 그 데이터를 텍스트 형식으로 콜백 함수(fn)에게 넘겨주는 기능을 한다

```
$.get('item.xml', function(data) {  
    $(data).find('item').each(function() {  
        var $item = $(this);  
        $("#treeData").append("<tr>" + "<td>"  
            + $item.attr("id") + "</td>" + "<td>"  
            + $item.attr("name") + "</td>" + "<td align='right'>"  
            + $item.find("price").text() + "</td>" + "<td>"  
            + $item.find("description").text() + "</td>" + "</tr>");  
    });  
});
```

POST 방식으로 서버와 통신하기-\$.post()

\$.post(URL[, data][,fn][,dataType])

- POST 방식으로 서버와 통신하는 jQuery Ajax 함수이다. \$.post()는 URL로 지정한 파일을 로드해서 그 데이터를 텍스트 형식으로 콜백 함수(fn)에게 넘겨주는 기능

```
var username = $('#username').val();
var sendData = 'username=' + username;

$.post("welcome.jsp",
    sendData,
    function (data) {
        $('#message').html(data);
    });
```

Ajax에 대한 global 옵션 설정-\$.ajaxSetup()

\$.ajaxSetup()

- jQuery Ajax에 대한 공통적인 기본 설정들을 지정하기 위해서 사용.

```
$.ajaxSetup({  
    type:"POST",  
    url:"logincheck.jsp",  
    dataType : "text",  
    success: function (msg) {  
        $('#message').html(msg);  
    }  
});  
$.ajax({  
    data: sendData  
});
```

JSON-simple **사용법**

-라이브러리 추가

다운로드

<https://code.google.com/archive/p/json-simple/downloads>

1. json-simple-1.1.1.jar 다운로드
2. 다운로드 받은 파일을 아파치 톰캣의 lib 폴더에 넣는다.
3. 추가할 프로젝트에서 오른쪽버튼을 클릭 후
Build Path -> Configure Build Path... 를 누른다.
4. Java Build Path -> Libraries -> Add Library...
5. 다운받은 json-simple-1.1.1.jar 를 추가한다.
6. 라이브러리 추가 완료 !

- 사용법

1. 기본 객체

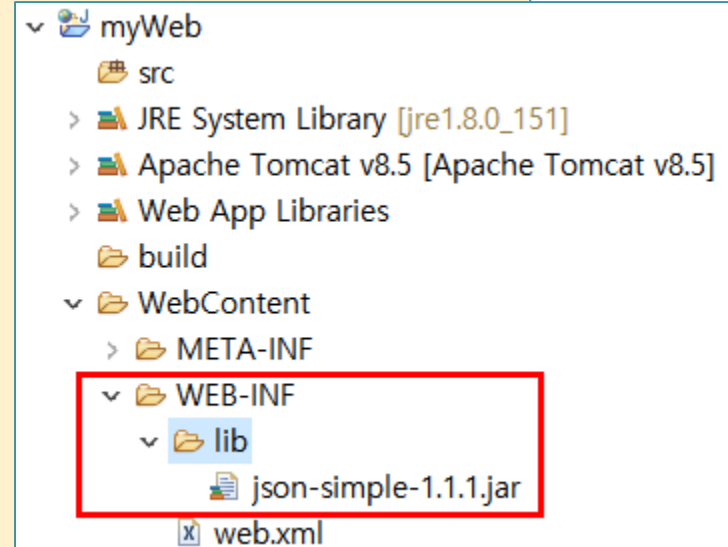
1) JSON 객체를 생성

```
JSONObject jsonObj = new JSONObject();
```

2) JSON 객체에 데이터 저장(복수저장 가능)

```
jsonObj.put("이름1", "내용1");
```

```
jsonObj.put("이름2", "내용2");
```



JSON-simple **사용법**

- 사용법

3) 결과

=> {"이름2":"내용2","이름1":"내용1"}

2. 배열 객체

1) JSON 배열 객체 생성

```
JSONArray jArr = new JSONArray();
```

2) 여러개의 JSON 객체 생성

```
JSONObject obj1 = new JSONObject();
```

```
obj1.put("name","사과");
```

```
obj1.put("age","21");
```

```
JSONObject obj2 = new JSONObject();
```

```
obj2.put("name","바나나");
```

```
obj2.put("age","27");
```