

## 10장 제이쿼리 비동기 방식 연동

### 10.1 Ajax

#### 10.1.1 Ajax란?

- Ajax(Asynchronous Javascript and XML)란 자바스크립트를 이용해 비동기 방식으로 자료를 전송 요청하는 것을 가리킨다.

##### (1) Ajax 사용 전 방식 (동기방식)

- 방문자가 자료를 요청하면 반드시 서버 컴퓨터를 거쳐야만 자료를 요청할 수 있다. 이 방식으로 자료를 요청하면 잠시 페이지가 서버 스크립트 페이지로 갱신되어 화면이 깜빡거리고, 그동안 사용자는 어떤 작업도 할 수 없게 된다.



##### (2) Ajax 사용 후 방식 (비동기방식)

- Ajax를 사용해 방문자가 서버에 자료를 요청했을 경우, 서버 스크립트 페이지를 거치지 않아도 자료를 받아 올 수 있다. 즉 방문자(client)는 서버에 자료를 요청하는 사이에 다른 작업을 할 수 있다. 블로그의 게시글에 댓글을 달 때 페이지 전환 없이 바로 할 수 있는 것도 Ajax를 사용한 것이다.



#### 10.1.2 웹 서버 설치하기

- Apache Tomcat을 설치한다.

#### 10.1.3 웹 호스팅 등록하여 서버 이용하기

- 닷홈([www.dothome.co.kr](http://www.dothome.co.kr)) 웹호스팅을 등록한다.

### 10.2 Ajax 관련 메서드

#### 10.2.1 Ajax 관련 메서드들



종류	풀이
load()	외부 콘텐츠를 가져올 때 사용한다.
\$.ajax()	데이터를 서버에 HTTP POST, GET 방식으로 전송이 가능하며, (X)HTML, XML, JSON 텍스트 유형에 데이터를 요청할 수 있는 통합적인 메서드이다. 이 표에서 \$post(), \$get(), \$getJSON() 메서드의 기능을 하나로 합쳐 놓은 것이라고 보면 된다.
\$.post()	데이터를 서버에 HTTP POST 방식으로 전송한 후 서버 측의 응답을 받을 때 사용한다.
\$.get()	데이터를 서버에 HTTP GET 방식으로 전송한 후 서버 측에 응답요청 받을 때 사용한다.
\$.getJSON()	데이터를 서버에 HTTP GET 방식으로 전송한 후 서버 측 응답을 JSON 형식으로 받을 때 사용한다.
\$.getScript()	Ajax를 이용하여 외부 자바스크립트를 불러온다.
.ajaxStop(function(){...})	비동기 방식으로 서버에 응답 요청이 완료되었을 때 함수에 실행문이 수행된다.
.ajaxSuccess(function(){...})	ajax 요청이 성공적으로 완료되면 함수에 실행문을 수행한다.

## (1) load() 메서드

- 사용자가 지정한 URL 주소에 데이터를 전송하고 외부 콘텐츠를 요청하여 가져올 때 사용한다. 요청한 콘텐츠를 이용해 선택한 요소에 내용을 바꿀 수 있다.

```
// 기본형
$(요소 선택).load(url, data, 콜백 함수)
```

[실습파일: jquery\_ajax\_1\_test.html, 완성파일: jquery\_ajax\_1.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Ajax</title>
<script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
<script type="text/javascript">
    //
    $(function() {
        $("#d1").load("document_1.txt");
    });
    //]]&gt;
&lt;/script&gt;
&lt;/head&gt;
&lt;body&gt;
    &lt;div id="d1"&gt;내용&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
</div>
<div data-bbox="117 731 281 748" data-label="Section-Header">
<h2>(2) $.ajax() 메서드</h2>
</div>
<div data-bbox="117 762 456 778" data-label="List-Group">
<ul>
<li>■ 다음은 $.ajax() 메서드의 기본형이다.</li>
</ul>
</div>
<div data-bbox="125 805 600 905" data-label="Text">
<pre>// 기본형
$.ajax({
    url: "전송 페이지"(액션 페이지),
    type: "전송 방식"(get, post 방식),
    data: "전송할 데이터",
    dataType: "요청한 데이터 타입"("html", "xml", "json", "text", "jsonp"),
    success: function(result){
        전송 성공하면 실행할 실행문들;</pre>
</div>
```

```
}  
});
```

[실습파일: jquery\_ajax\_2\_test.html, 완성파일: jquery\_ajax\_2.html]

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">  
<head>  
<meta http-equiv="content-type" content="text/html; charset=utf-8" />  
<title>Ajax</title>  
<script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>  
<script type="text/javascript">  
    <![CDATA[  
        $(function() {  
            $("#member").on("submit", function() { // "확인" 버튼을 눌렀을 때...  
                var d = $(this).serialize(); // 폼 안에 파라미터를 한번에 전송한다.  
  
                $.ajax({  
                    url : "welcome.jsp", // 데이터 전송 및 요청할 URL 주소  
                    type : "post", // 데이터 전송 방식  
                    data : d, // 전송할 데이터  
                    success : function(result) { // 정상적으로 되었을 때...  
                        $("#txt1").html(result);  
                    }  
                });  
  
                return false; // action 페이지로 전환되는 것을 차단합니다.  
            });  
        }]);  
    </script>  
</head>  
<body>  
    <form action="member.php" method="post" name="member" id="member">  
        <fieldset>  
            <legend>회원가입</legend>  
            <p>  
                <label for="username">이름</label> <input type="text"  
                    name="username" id="username" />  
            </p>  
            <p>  
                <input type="submit" value="확인" />  
            </p>  
        </fieldset>  
    </form>  
    <h1 id="txt1"></h1>  
</body>  
</html>
```

[welcome.jsp]

```
<%@ page language="java" contentType="text/html; charset=UTF-8"  
    pageEncoding="UTF-8"%>  
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  
<title>Insert title here</title>  
</head>  
<body>  
    환영합니다. ${param.username}님  
</body>  
</html>
```

## [꿀팁] php + apache (server), jsp + (apache) tomcat

- php 언어는 아파치 서버에서만 작동하므로 위 예제는 아파치 서버가 설치된 환경에서 테스트해야 작동된다. 반면 jsp 언어는 tomcat에서 작동된다.

### (3) \$.post() 메서드

[완성파일: example09\_08.html]

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>$.ajax로 jsp 파일 로드하기</title>
<style>
td {
    border: 1px solid gray;
}
</style>
<script src="http://code.jquery.com/jquery-1.11.1.js"
    type="text/javascript"></script>

<script type="text/javascript">
/*
- $.post() 함수는 post방식으로 서버와 통신하는 jQuery 함수이다.
- $.post() 함수는 서버의 welcome.jsp 페이지를 post방식으로 요청한다.
*/

$(function() {
    $("#member").on("submit", function() { //"확인"버튼을 눌렀을 때..
        var username = $("#username").val();
        var passwd = $("#passwd").val();
        var sendData = 'username=' + username + '&passwd=' + passwd;
        alert(sendData);

        $.post( // 문자열 형식의 데이터를 welcome.jsp 파일로 보낸다.
            "welcome.jsp", sendData, function(msg) {
                $('#txt1').html(msg);
            });
        return false;
    });
});
</script>
</head>
<body>

    <form action="member.jsp" method="post" name="member" id="member">
        <fieldset>
            <legend>회원가입</legend>
            <p>
                <label for="username">이름</label> <input type="text"
name="username"
                id="username" />
            </p>
            <p>
                <label for="passwd">비밀번호</label> <input type="password"
name="passwd" id="passwd" />
            </p>
            <p>
                <input type="submit" value="확인" />
            </p>
        </fieldset>
    </form>
    <h1 id="txt1"></h1>
```

```
</body>
</html>
```

#### (4) \$.get() 메서드

[완성파일: example09\_07.html]

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>XML 파일을 GET 방식으로 로드하기</title>
<style>
td { border: 1px solid gray;
}
</style>
<script src="http://code.jquery.com/jquery-1.11.1.js"></script>
<script type="text/javascript">
/*
- $.get() 함수를 이용해서 item.xml 파일을 불러온다.
- $.get() 함수의 첫번째 매개변수에는 서버의 URL 주소를 지정한다.
- $.get() 함수의 두번째 매개변수인 콜백함수를 이용해서 서버에서 보내온 XML 형식의 데이터를 data 로
받는다.
*/

$(function() {
$.get('item.xml', function(data) {
    $("#treeData").append(
        "<tr><td>id</td>" + "<td>name</td>"
        + "<td>price</td>" + "<td>description</td>" + "</tr>");
    $(data).find('item').each(function() {
        // 서버에서 얻어온
        // 데이터에서 선택터로 item태그를 찾는다.
        var $item = $(this);
        // 6개의 item태그중 현재 처리중인 item태그를 this로 접근한후에 이를
        $("#treeData").append("<tr>" + "<td>" // this로 접근한 후에 이를 $(this)를 사용하여
        jQuery객체를
        + $item.attr("id") + "</td>" + "<td>" //생성함
        + $item.attr("name") + "</td>" + "<td align='right'>"
        + $item.find("price").text() + "</td>" + "<td>"
        + $item.find("description").text() + "</td>" + "</tr>");
    });
});
});
</script>
</head>
<body>
<table id="treeData"></table>
</body>
</html>
```

[item.xml]

```
<?xml version="1.0" encoding="UTF-8"?>
<items>
<item id="1" name="레몬">
    <price>3000</price>
    <description> 레몬에 포함되어 있는 쿠엔산은 피로회복에 좋다. 비타민C도 풍부하다. </description>
</item>

<item id="2" name="키위">
    <price>2000</price>
    <description> 비타민C가 매우 풍부하다. 다이어트와 미용에도 매우 좋다. </description>
</item>
```

```

<item id="3" name="블루베리">
  <price>5000</price>
  <description> 블루베리에 포함된 anthocyanin(안토시아닌)은 눈피로에 효과가 있다. </description>
</item>

<item id="4" name="체리">
  <price>5000</price>
  <description> 체리는 맛이 단 성분이 많고 피로회복에 잘 듣는다. </description>
</item>

<item id="5" name="메론">
  <price>5000</price>
  <description> 메론에는 비타민A와 칼륨이 많이 포함되어 있다. </description>
</item>

<item id="6" name="수박">
  <price>15000</price>
  <description> 수분이 풍부한 과일이다.</description>
</item>
</items>

```

## (5) \$.getJSON() 메서드

[완성파일: example09\_05.html]

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSON 이용하기</title>
<style>
td {
  border: 1px solid gray;
}
</style>
<script src="http://code.jquery.com/jquery-1.11.1.js" type="text/javascript"></script>
<script type="text/javascript">
/*
  - JSON은 JavaScript Object Notation 의 약어로 XML 데이터를 대신하기 위해서 사용된다.
  - JSON은 키와 값을 쌍으로 가지는 구조이다.
  - 배열을 사용할 때는 대괄호([ ])안에 중괄호({ })를 사용하여 조합한다.
  - jQuery에서는 JSON으로 표현한 데이터를 파일에 저장해 두었다가 필요할 때 이를 로드할 수 있는
  $.getJSON()함수를 제공한다.

  - 서버로 부터 get방식의 요청을 하고, 응답을 JSON 형식으로 받기 위해서 $.getJSON()함수를 사용한다.
  - $.getJSON()함수의 첫번째 매개변수는 서버의 URL주소를 지정한다. ('item.json')
  - 서버로 요청이 성공적으로 완료되면, 두번째 매개변수로 기술한 콜백함수가 호출된다. (function(data,
  textStatus))
  콜백함수의 첫번째 매개변수인 data는 서버에서 돌려받은 JSON 객체 형식의 데이터이고, 두번째
  매개변수인 textStatus는 성공일때
  "success" 라는 문자열이 전달된다.

*/

$(function() {
  $.getJSON('item.json', function(data, textStatus) {
    $("#treeData").append("<tr><td>id</td>" + "<td>name</td>"
      + "<td>price</td>" + "<td>description</td>" + "</tr>");
    $.each(data, function() {
      $("#treeData").append("<tr>" + "<td>"
        + this.id + "</td>" + "<td>"
        + this.name + "</td>" + "<td align='right>"
        + this.price + "</td>" + "<td>"
        + this.description + "</td>" + "</tr>");
    });
  });
}

```

```

    });
});

</script>
</head>
<body>
    <table id="treeData"></table>
</body>
</html>

```

[item.json]

```

[
  {
    "id": "1",
    "name": "레몬",
    "price": "3000",
    "description": "레몬에 포함되어 있는 쿠엔산은 피로회복에 좋다. 비타민C도 풍부하다."
  },
  {
    "id": "2",
    "name": "키위",
    "price": "2000",
    "description": "비타민C가 매우 풍부하다. 다이어트와 미용에도 매우 좋다."
  },
  {
    "id": "3",
    "name": "블루베리",
    "price": "5000",
    "description": "블루베리에 포함된 anthocyanin(안토시아닌)은 눈피로에 효과가 있다."
  },
  {
    "id": "4",
    "name": "체리",
    "price": "5000",
    "description": "체리는 맛이 단 성분이 많고 피로회복에 잘 듣는다."
  },
  {
    "id": "5",
    "name": "매론",
    "price": "5000",
    "description": "매론에는 비타민A와 칼륨이 많이 포함되어 있다."
  },
  {
    "id": "6",
    "name": "수박",
    "price": "15000",
    "description": "수분이 풍부한 과일이다."
  }
]

```

[완성파일: example09\_05\_02.html]

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JSON 이용하기</title>
<style>
td {
    border: 1px solid gray;
}
</style>
<script src="http://code.jquery.com/jquery-1.11.1.js" type="text/javascript"></script>
<script type="text/javascript">
/*

```





```

</head>
<body>
    <h1>인기 검색어</h1>
    <div id="wrap">
        <ol></ol>
    </div>
</body>
</html>

```

```

[rank.xml]

<?xml version="1.0" encoding="UTF-8"?>
<result>
    <item>
        <rank>
            <k>김연아</k>
        </rank>
        <rank>
            <k>이상화</k>
        </rank>
        <rank>
            <k>모태범</k>
        </rank>
    </item>
</result>

```

### 10.2.3 JSON 데이터 요청하기

- JSON은 JavaScript Object Notation의 약자로 자바스크립트 객체 표기법을 말한다. JSON으로 데이터를 표기할 때는 중괄호 내에 key와 value 값을 쌍으로 표기한다.

```

// 기본형
{key:value1, key2:value2, key3:value3, ...}

```

[실습파일: jquery\_ajax\_4\_test.html, 완성파일: jquery\_ajax\_4.html]

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>Ajax</title>
<script type="text/javascript" src="js/jquery-1.10.2.min.js"></script>
<script type="text/javascript">
    <![CDATA[
        $(function() {
            $.ajax({
                url : "rank.json", //데이터를 요청할 URL 주소
                dataType : "json", //데이터 타입 설정
                success : function(result) { //데이터 성공적으로 요청되었을 때...
                    $.each(result.rank, function(i, d) {
                        $("#wrap ol").append("<li>" + d["k"] + "</li>");
                    });
                }
            });
        });
    ]>
</script>
</head>
<body>
    <h1>인기 검색어</h1>

```

```

        <div id="wrap">
            <ol></ol>
        </div>
    </body>
</html>

```

[rank.json]

```

{
  "rank": [
    { "k": "김연아" },
    { "k": "이상화" },
    { "k": "모태범" }
  ]
}

```

## [실습] Ajax로 RSS 연동하기

- RSS(Really Simple Syndication)란, 사전적으로 '초간편 배급(배포)'이라는 의미를 가지고 있다.

[완성파일: jquery\_ajax\_rss.html]

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko" xml:lang="ko">
<head>
<title> RSS 연동하기 </title>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>
<script type="text/javascript">
$(function(){
//변수 g에 크로스 도메인의 데이터를 가져올 수 있도록 구글 주소를 저장하였습니다.
//var g="https://ajax.googleapis.com/ajax/services/feed/load?v=2.0&num=10&callback=?&q=";
var g = "https://api.rss2json.com/v1/api.json?rss_url=";

$("#news_wrap h2 a").on("click",
function(e){
    e.preventDefault(); //<a> 태그에 링크를 차단하였습니다.
    $("#news_wrap h2 a").removeClass("on"); //<a>에 "on" class를 삭제합니다.
    $(this).addClass("on"); //클릭한 <a>에 class="on"을 생성합니다.
    $.ajax({
        url:g+$(this).attr("href"), //요청할 데이터 경로.
        dataType: "json", //요청할 데이터에 타입.
        success: function (data) { //성공적으로 요청했었을 때...
            $("#news_list").empty(); //<ul>에 하위 요소를 모두 제거합니다.
            //console.log(data.responseData.feed.entries);

            //$.each(data.responseData.feed.entries,function(i, d){
            $.each(data.items,function(i, d){
                if(i==5) return false;
                var title=d["title"]; //title 키의 값을 가져옵니다.
                //var date=new Date(d["publishedDate"]); //뉴스 작성 날짜 객체 생성
                var date=new Date(d["pubDate"]); //뉴스 작성 날짜 객체 생성
                //작성한 년-월-일 형식으로 변수 m에 저장합니다.
                var m=date.getFullYear()+"-"+(date.getMonth()+1)+"-"+date.getDate();
                var lk=d["link"];
                //앞서 구해온 데이터를 이용해 정보 목록 태그를 생성합니다.
                $("#news_list").append('<li><a href="'+lk+'" target="_blank">'+
                    title+'</a> <span>'+m+'</span></li>');
            }); // " $.each " 메서드에 종료
        } // " success: function " 에 종료
    }); // $.ajax() 메서드에 종료
}); // 이벤트 on 메서드에 종료

```

```

$("#news_wrap h2 a").eq(0).click(); //최초 첫 번째 탭 버튼을 강제로 클릭시킵니다.
});
</script>
<style type="text/css">
/*기본 세팅*/
*{margin:0;padding:0;}
li{list-style:none;}
body{font:12px Margun Gothic, "맑은고딕",gulim,"굴림",sans-serif;
color:#333;padding-top:20px;}
h1,h2,h3,h4,h5,h6{font-size:100%;font-weight:normal;}
a{color:#333;text-decoration:none;outline:none;}
h1{font-size:2em;text-align:center;margin-bottom:10px;color:#c00;}

/*탭 메뉴 디자인*/
#news_wrap{width:380px;margin:0 auto;
background:#FFF url("images/rss_bg.gif") 0 0 repeat-x;}

#news_wrap li{height:25px;line-height:25px;position:relative;}
#news_wrap span{position:absolute;top:0;right:3px;}
#news_wrap li a{display:block;width:210px;
white-space:nowrap;overflow:hidden;
text-overflow:ellipsis;padding-left:15px;
background:url("images/bullet.gif") 5px 10px no-repeat;
}
#news_wrap h2{float:left;padding-right:3px;
margin-bottom:10px;}
#news_list{clear:both;}

/*비활성화 탭 버튼 모양*/
#news_wrap h2 a{
display:inline-block;padding:8px 10px;
border:1px solid #aaa;
border-bottom:none;
border-radius:5px 5px 0 0;background:#ccc;
}
/*비활성화 탭 버튼 모양*/
#news_wrap h2 a.on{background-color:#fff;font-weight:bold;}
</style>
</head>
<body>
<h1>조선닷컴</h1>
<div id="news_wrap">
<h2> <!-- 초기에 첫 번째 버튼에만 class="on"이 포함되어 있습니다. -->
<a href="http://myhome.chosun.com/rss/www_section_rss.xml" class="on">
오늘의 주요뉴스</a>
</h2>
<h2>
<a href="http://www.chosun.com/site/data/rss/ent.xml">엔터테인먼트</a>
</h2>
<h2>
<a href="http://newsplus.chosun.com/hitdata/xml/se/sports/index.xml">스포츠</a>
</h2>
<ul id="news_list">
</ul>
</div>
</body>
</html>

```