



jQuery를 이용한 ajax 통신

안 화 수

❖ Ajax (Asynchronous JavaScript + XML)

- JavaScript와 XML을 이용한 비동기 통신처리를 구현하는 기술
- JavaScript로 웹 페이지 전체를 다시 읽어 들이지 않고, 일부 정보를 서버와 데이터를 주고 받는 경우에 사용한다.
- JavaScript를 이용해서 서버에서 데이터를 가져와 페이지 전체의 갱신(refresh)없이 특정 부분만을 변경하는 것이 가능하다.
ex. ID중복 검사, 댓글작성

❖ 동기 통신과 비동기 통신

❖ 동기통신

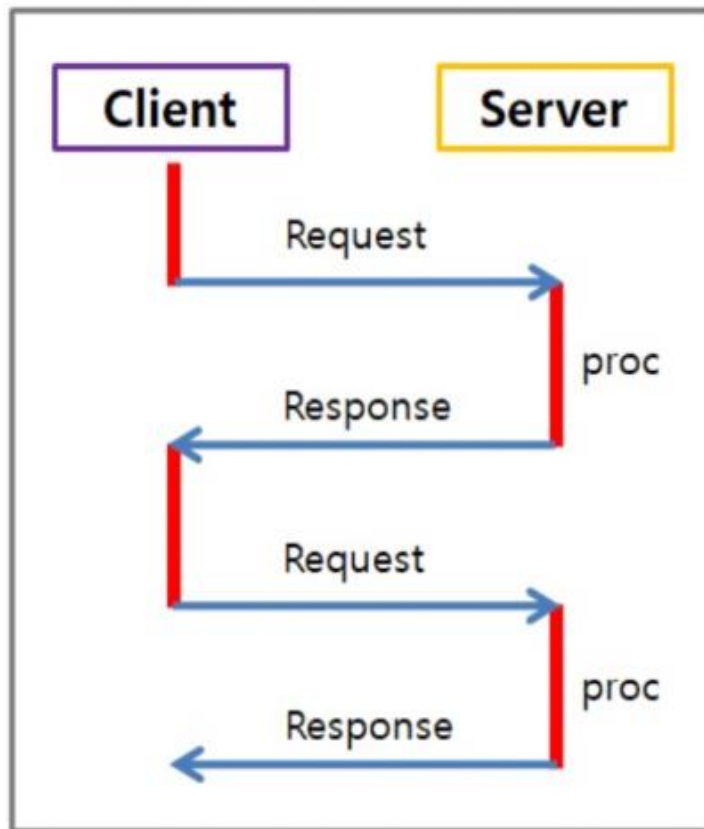
1. 웹 브라우저에서 서버에 요청을 보내고 응답을 받는다.
2. 서버에 요청을 하고, 서버로 부터 응답이 끝날 때까지 다른 작업을 할 수 없게 된다.
3. 요청 페이지와 응답 받는 페이지가 바뀐다.

❖ 비동기 통신

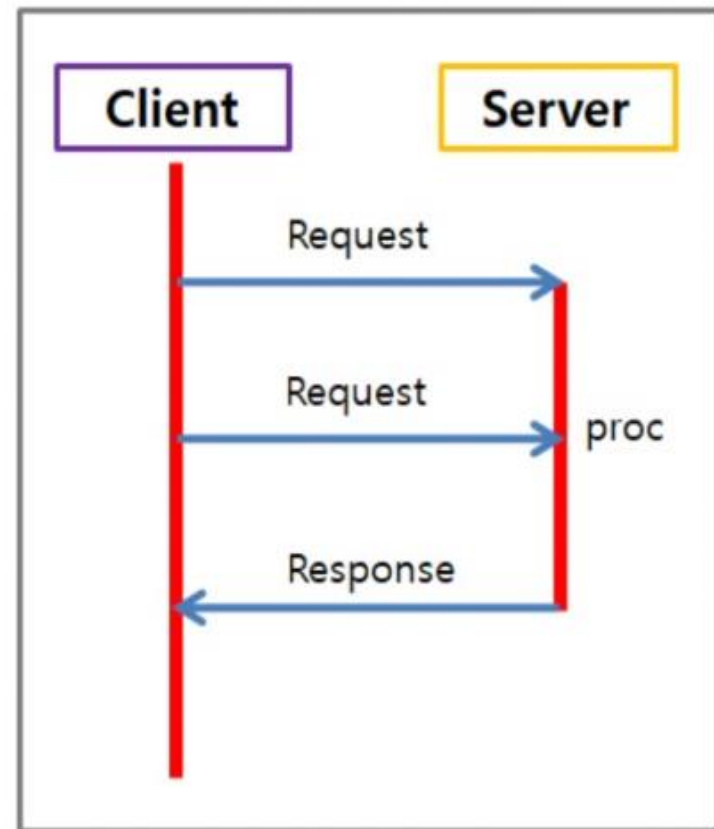
1. 웹 브라우저로부터 일부의 정보를 요청할 수 있다.
2. 서버에 요청을 하고, 서버로 부터 응답이 끝나지 않더라도 다른 작업을 할 수 있다.
3. 요청 페이지와 응답 받는 페이지가 바뀌지 않는다.

ajax

❖ 동기 통신과 비동기 통신



<동기식>



<비동기식>

❖ Ajax의 장단점

❖ 장점

- 웹페이지 속도가 향상된다.(필요한 데이터만 리로드하면 되니까!)
- 서버의 처리가 완료될 때 까지 기다리지 않고 처리 가능하다.
- 서버에서 Data만 전송하면 되므로 코딩의 양이 줄어든다.

❖ 단점

- 히스토리 관리가 안된다.
- 페이지 이동없는 통신으로 인한 보안상 문제가 있을 수 있다.
- 연속으로 데이터를 요청하면 서버 부하가 생길 수 있다.

❖ jQuery Ajax 메소드

\$.ajax() : get, post 방식으로 Ajax를 수행한다.

\$.get() : get 방식으로 Ajax를 수행한다.

\$.post() : post 방식으로 Ajax를 수행한다.

\$.getJSON() : get 방식으로 Ajax를 수행해 JSON 데이터를 가져온다.

\$.getScript() : get 방식으로 Ajax를 수행해 Script 데이터를 가져온다.

\$(selector).load() : Ajax를 수행한 후에 선택한 문서 객체안에 응답받을 파일(문자열)을 불러온다.

load 함수

❖ load 함수 : load01.html

```
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
    $(function(){
        $('button').click(function(){
            // load함수로 sample1.txt파일을 불러와서 p태그 사이에 출력
            $('p').load('sample1.txt');
        });
    });
</script>
</head>
<body>
    <button>변경</button>
    <p>변경전</p>
</body>
```

load 함수

❖ 요청 파일 : **sample1.txt**

sample1.txt 의 텍스트 입니다.

load 함수

❖ load 함수 : load02.html

```
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
    $(function() {
        $('button').click(function() {
            // load함수로 resource.html파일을 불러와서 div태그 사이에 출력
            $('#container').load('resource.html');
        });
    });
</script>
</head>
<body>
    <button>서버로부터 데이터 가져오기</button>
    <div id="container">데이터 가져오기 전</div>
</body>
```

load 함수

❖ 요청 파일 : resource.html

```
<!DOCTYPE html>  
<html>  
<head>  
<meta charset="UTF-8">  
<title>요청 파일 : resource.html</title>  
</head>  
<body>  
  
<b>Happy New Year !!</b>  
  
</body>  
</html>
```

load 함수

❖ load 함수 : load03.html

```
<style>
  div { width: 180px; height: 80px; margin: 3px; float: left; }
</style>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
  $(function() {
    $('#menu1').click(function () {
      // load() 함수를 이용해서 menu.html 문서전체를 읽어와서
      // id가 message1인 태그 사이에 추가한다.
      $('#message1').load('menu.html');
    });
    $('#menu2').click(function () {
      // load() 함수를 이용해서 menu.html 문서의 내용중 li 태그만 읽어와서
      // id가 message2인 태그 사이에 추가한다.
      $('#message2').load('menu.html li');
    });
  });
</script>
</head>
<body>
<div>
  <a href="#" id="menu1">메뉴 보기 1</a><p>
  <span id="message1"></span>
</div>
<div>
  <a href="#" id="menu2">메뉴 보기 2</a><p>
  <span id="message2"></span>
</div>
</body>
```

load 함수

❖ 요청 파일 : menu.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>요청 파일 : menu.html</title>
</head>
<body>
<p> 중 식 메 뉴 </p>
<ul>
    <li> 짜장면 </li>
    <li> 짬뽕 </li>
    <li> 기스면 </li>
    <li> 탕수육 </li>
</ul>
<p> 메뉴를 골라주세요.</p>
</body>
</html>
```

ajax 함수

❖ \$.ajax()

- 서버에 get, post 방식 모두 요청할 수 있는 함수이다.
- \$.ajax({
 url : "전송 페이지",
 type : "전송 방식(get, post방식),
 data : "전송할 데이터",
 dataType : "요청한 데이터입("html","xml","json","text"),
 success : function(result){
 콜백함수로 돌려 받은 후 실행될 문장;
 }
});
- 웹 브라우저에 출력되는 결과가 콜백함수로 리턴 된다.

ajax 함수

❖ ajax 함수 : ajax01.html(1/2)

```
<style>
    div { width: 180px; height: 80px; margin: 3px; float: left; }
</style>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
    $(function() {
        $('#menu1').click(function () {
            $.ajax({
                url: 'menu.html',      // 서버에 요청할 파일명
                dataType: "html",
                // 서버로부터 응답받은 데이터는 콜백함수의 매개변수로 전달된다.
                success: function(data) {
                    // 콜백함수로 전달된 data를 id가 message1인 태그에 설정한다.
                    $('#message1').html(data);
                }
            }); // $.ajax() end
        });
        $('#menu2').click(function () {
            $.ajax({
                url: 'menu.html',
                dataType: "html",
                // 서버로부터 응답받은 데이터는 콜백함수의 매개변수로 전달된다.
                success: function(data) {
                    // menu.html문서의 내용중 li엘리먼트만 읽어서 id가 message2인 태그에 설정한다.
                    $('#message2').html($(data).find('li'));
                }
            }); // $.ajax() end
        });
    });
</script>
</head>
```

ajax 함수

❖ ajax 함수 : ajax01.html(2/2)

```
<body>
<div>
    <a href="#" id="menu1">메뉴 보기 1</a><p>
    <span id="message1"></span>
</div>
<div>
    <a href="#" id="menu2">메뉴 보기 2</a><p>
    <span id="message2"></span>
</div>
</body>
</html>
```

post 함수

❖ \$.post()

- 서버에 post 방식으로 요청할 때 사용하는 함수이다.
- \$.post("요청할 파일명",
 { "key" : value } ,
 function(msg){
 콜백함수로 돌려 받은 후 실행될 문장;
 }
);
- 웹 브라우저에 출력되는 결과가 콜백함수로 리턴 된다.

post 함수

❖ post함수로 welcome.jsp 파일 요청 결과

이름 입력 :

환영합니다. 홍길동님

post 함수

❖ post 함수 : post01.html

```
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
$(function() {
    $('#b1').click(function() {
        $.post('welcome.jsp',
            {'username' : $('#username').val()},
            function(msg) {
                alert(msg);
                $('#message').html(msg);
            });
    });
});
</script>
</head>
<body>
<form>
    이름 입력 :
    <input type="text" id="username"><br>
    <input type="button" id="b1" value="전송">
</form>
<div id="message"></div>
</body>
```

post 함수

❖ welcome.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>서버에 요청파일 : welcome.jsp</title>
</head>
<body>
    <!-- 웹브라우저에 출력되는 결과가 콜백함수로 리턴된다. -->
    환영합니다. <%=request.getParameter("username") %>님 <br>
</body>
</html>
```

post and ajax 비교

❖ ajax 함수 : ajax02.html

```
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
$(function() {
    $('#b1').click(function() {
        $.ajax({
            url : 'welcome.jsp',
            type : 'post',
            data : { 'username' : $('#username').val() },
            success : function(msg) {
                alert(msg);
                $('#message').html(msg);
            }
        }); // $.ajax() end
    });
});
</script>
</head>
<body>
<form>
    이름 입력 :
    <input type="text" id="username"><br>
    <input type="button" id="b1" value="전송">
</form>
<div id="message"></div>
</body>
```

get 함수

❖ \$.get()

- 서버에 get 방식으로 요청할 때 사용하는 함수이다.
- \$.get("요청할 파일명",
function(data){
 콜백함수로 돌려 받은 후 실행될 문장;
}
);
- 웹 브라우저에 출력되는 결과가 콜백함수로 리턴 된다.

get 함수

❖ get함수로 item.xml 파일 요청 결과

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

get 함수

❖ get 함수 : get01.html

```
<style>
    td { border: 1px solid gray;}
</style>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
$(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() {
            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>");
        }); // each() end
    });
});
</script>
</head>
<body>
    <table id="treeData"></table>
</body>
```

xml

❖ item.xml

```
<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>
```


JSON

❖ JSON (JavaScript Object Notation)

- 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.

JSON

❖ JSON (JavaScript Object Notation)

```
[
  {
    "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."
  }
]
```

getJSON 함수

❖ \$.getJSON()

- 서버에 json 데이터를 요청할 때 사용하는 함수이다.
- \$.getJSON("요청할 json파일명",
function(data, textStatus){
콜백함수로 돌려 받은 후 실행될 문장;
}
);
- data : 콜백함수로 서버에서 돌려받는 데이터
- textStatus : 콜백함수로 잘 돌려 받으면 'SUCCESS' 리턴

getJSON 함수

❖ getJSON 함수 : getJSON01.html

```
<style>
    td { border: 1px solid gray;}
</style>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
$(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>");
        }); // $.each() end
    });
});
</script>
</head>
<body>
    <table id="treeData"></table>
</body>
```

getJSON and ajax 비교

❖ ajax 함수 : ajax03.html

```
<style>
    td { border: 1px solid gray;}
</style>
<script src="http://code.jquery.com/jquery-latest.js"></script>
<script>
$(function() {
    $.ajax({
        url : "item.json",
        dataType : "json",
        success : function(data) {
            $("#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>");
            }); // $.each() end
        }
    }); // $.ajax() end
});
</script>
</head>
<body>
    <table id="treeData"></table>
</body>
```