Ajax (Asynchronous Javascript And XML)

CONTENTS

- aJax API

  https://api.jquery.com/category/ajax/

- $.aJax()

  https://api.jquery.com/jQuery.ajax/

- 추가적인 aJax 메서드

  https://api.jquery.com/category/ajax/shorthand-methods/

- 폼 관련 보조 메서드

  https://api.jquery.com/category/ajax/helper-functions/

Ajax (Asynchronous Javascript And XML)

- **Ajax (Asynchronous Javascript And XML)**
  - 클라이언트와 서버측의 **데이터 전송 및 처리**를 **비동기적(Asynchronous)**으로 처리
  - Ajax를 사용함으로써, 서버측에 데이터를 요청한 후, 그 데이터의 수신이 완료될 때까지 기다리지 않고, 다른 작업을 바로 진행할 수 있고, 좀더 웹페이지를 자유롭게 구성할 수 있게 되었고, 불필요한 페이지 변경 또는 로딩을 줄일 수 있다.

$ajax()

- 기본
  - jQuery를 활용한 Ajax 메서드

```
$.ajax(options);

$.ajax(url, option);
```

$ajax()

- 기본
  - $ajax() 메서드(2)

```
<script>
        $(document).ready(function () {
            $.ajax('data.html', {
                success: function (data) {
                    $('body').append(data);
                }
            });
        });
</script>
```

$ajax()

- 기본
  - $ajax() 메서드(2)

```
<script>
        $(document).ready(function () {
            $.ajax({
                url: 'data.html',
                success: function (data) {
                    $('body').append(data);
                }
            });
        });
</script>
```

$ajax()

- 기본
  - $ajax() 메서드(2)

```
<input type="text" id="txt" value="akwldrkwpxm">

<script>
        $(document).ready(function () {
            $.ajax({
                url: 'parameter.jsp',
                type: 'GET',
                data: {
                    name: $('#txt').val(),
                    price: 'test'
                },
                success: function (data) {
                    $('body').append(data);
                }
            });
        });
</script>
```

$ajax()

```
$.ajax({
        url : requestUrl,
        type : 'DELETE',
        async : true,
        data : JSON.stringify(requestParam),
        dataType : "json",
        timeout : 10000,
        contentType : "application/json",
        beforeSend : function(){
            $('.wrap-loading').removeClass('display-none');
        },
        complete : function(){
            $('.wrap-loading').addClass('display-none');
        },
        success : function(data){
            alert(data);
        },
        error : function(request,status,error){
            alert("code:"+request.status +"\n"+"message:"+request.responseText+"\n"+"error:"+error);
            $('.wrap-loading').addClass('display-none');
        }
});
```

$ajax()

- **$ajax() Option**

옵션 속성 이름	설명	자료형
async	비동기 통신 플래그. true (비동기 통신)에서 요청 이후 응답입 없어도 비동기 처리를 계속한다. false로 설정 (동기 통신)하면 통신에 응답이 있을 때까지 브라우저는 잠겨 이벤트 실행이 대기한다.	**true** , false
complete	AJAX 통신 완료될 때 호출되는 함수 success이나 error가 호출된 후에 호출되는 Ajax Event	Function
data	요청 매개변수를 지정	Object, String
dataType	서버에서 반환되는 데이터 형식을 지정 생략했을 경우는, jQuery이 MIME 타입 등을 보면서 자동으로 결정 "json" : JSON 형식 데이터로 평가하고 JavaScript의 개체로 변환합니다.	"xml", "html", "script", "jsonp", "json", "text"
error	통신에 실패했을 때 호출되는 Ajax Event	Function
contentType	서버에 데이터를 보낼 때 사용 content - type 헤더의 값 "application / x - www - form - urlencoded"	
timeOut	제한 시간 (밀리초)을 설정합니다.	숫자
type	HTTP 통신의 종류를 설정합니다.	"**GET**", "POST", "PUT", "DELETE"
success	AJAX 통신이 성공하면 호출되는 Ajax Event 입니다. 돌아온 데이터와 dataType 지정한 값 2 개의 인수를받습니다.	
URL	대상 URL을 지정	

■ 추가적인 ajax 메서드

- **추가적인 jQuery Ajax 메서드**

메서드 이름	설명
jQuery.get()	Get방식으로 수행 jQuery.get(url [, data] [, success] [, dataType]) jQuery.get([settings])
jQuery.post()	Post방식으로 수행 jQuery.post(url [, data] [, success] [, dataType]) jQuery.post([settings])
jQuery.getJSON()	Get 방식으로 ajax를 수 행해 Json 데이터로 반환 jQuery.getJSON(url [, data] [, success])
jQuery.getScript()	Get 방식으로 ajax를 수 행해 script 데이터로 반환 jQuery.getScript(url [, success])
.load()	Ajax를 수행하고 선택자로 선택한 문서 객체 안에 데이터 삽입 .load(url [, data] [, complete])

추가적인 ajax 메서드

- **jQuery.get()**

```
<script>
        $(document).ready(function () {
            $.get('data.html', function (data) {
                $('body').html(data);
            });
        });
</script>
```

■ 추가적인 ajax 메서드

- **jQuery.post()**

```
<script>
        $(document).ready(function () {
            $.post('data.html', function (data) {
                $('body').html(data);
            });
        });
</script>
```

■ 추가적인 ajax 메서드

- **jQuery.getJSON()**

```
<script>
        $(document).ready( function() {
                $.getJSON('data.json', function(data) {
                        $.each(data, function(key, value) {
$('body').append('<h1>'+value.name+':'+value.price+'</h1>');
                        });
                });
        });
</script>
```

■ 추가적인 ajax 메서드

- **.load()**

```
<script>
        $(document).ready(function () {
            $('body').load('data.html');
        });
</script>
```

■ 추가적인 ajax 메서드

- **XML 조작**

```
<script>
    $(document).ready(function () {
        $.ajax({
            url: 'data.xml',
            success: function (data) {
                $(data).find('product').each(function () {
                    // 변수를 선언합니다.
                    var name = $(this).find('name').text();
                    var price = $(this).find('price').text();

                    // 출력합니다.
                    $('<h1></h1>').text(name + ':' + price).appendTo('body');
                });
            }
        });
    });
</script>
```

■ 추가적인 ajax 메서드

- 추가적인 jQuery Ajax 메서드

메서드 이름	설명
jQuery.param()	입력 양식의 내용을 요청 매개변수 문자열로 변환
.serialize()	입력 양식의 내용을 객체로 변환
.serializeArray()	객체의 내용을 요청 매개변수 문자열로 변환

■ 추가적인 ajax 메서드

- 추가적인 jQuery Ajax 메서드

```
<form id="my-form">
    <table>
        <tr>
                <td><label for="name">Name</label></td>
                <td><input id="name" name="name" type="text" /></td>
        </tr>
        <tr>
                <td><label for="price">Region</label></td>
                <td><input id="price" name="price" type="text" /></td>
        </tr>
    </table>
    <input type="submit" value="Get Ajax String" />
</form>
<div id="wrap"></div>
```

■ 추가적인 ajax 메서드

- **query String**

```
<script>
        $(document).ready(function () {
            $('#my-form').submit(function (event) {
                // 입력 양식의 value 속성을 가져옵니다.
                var name = $('#name').val();
                var region = $('#price').val();

                // Ajax 요청을 수행합니다.
              var url = '/parameter.jsp?name='+name+'&price='+region;
                $('#wrap').load(url);

                // 기본 이벤트를 제거합니다.
                event.preventDefault();
            });
        });
</script>
```

■ 추가적인 ajax 메서드

- **Javascrip Object**

```
<script>
        $(document).ready(function () {
            $('#my-form').submit(function (event) {
                // 입력 양식의 value 속성을 가져옵니다.
                var name = $('#name').val();
                var region = $('#price').val();

                // Ajax 요청을 수행합니다.
                var url = '/parameter.jsp';
                var data = { name: name, region: region };
                $('#wrap').load(url, data);

                // 기본 이벤트를 제거합니다.
                event.preventDefault();
            });
        });
</script>
```

■ 추가적인 ajax 메서드

- **jquery.param()**

```
<script>
        $(document).ready(function () {
            $('#my-form').submit(function (event) {
                // 입력 양식의 value 속성을 가져옵니다.
                var name = $('#name').val();
                var region = $('#price').val();

                // Ajax 요청을 수행합니다.
                var url = '/parameter.jsp';
                var data = { name: name, region: region };
                var params = $.param(data);
                $('#wrap').load(url, params);

                // 기본 이벤트를 제거합니다.
                event.preventDefault();
            });
        });
</script>
```

■ 추가적인 ajax 메서드

- **Serialize()**

```
<script>
        $(document).ready(function () {
            $('#my-form').submit(function (event) {
                // Ajax 요청을 수행합니다.
                $('#wrap').load('/parameter.jsp', $(this).serialize());

                // 기본 이벤트를 제거합니다.
                event.preventDefault();
            });
        });
</script>
```

■ 추가적인 ajax 메서드

- **serializeArray()**

```
<script>
        $(document).ready(function () {
            $('#my-form').submit(function (event) {
                // Ajax 요청을 수행합니다.
                $('#wrap').load('parameter.jsp', $(this).serialize());

                // 기본 이벤트를 제거합니다.
                event.preventDefault();
            });
        });
</script>
```