

## 9. Ajax

1. Ajax의 개요 및 구조
2. Ajax 라이브러리를 사용한 웹 페이지 작성

# 1. Ajax의 개요 및 구조

## □ Ajax의 개요

- Ajax라는 용어는 2005년 2월 18일 제임스 제시 개럿(Jesse James Garrett)이 자신의 블로그에 "Ajax : a new approach to web application"라는 제목의 글을 싣게 되면서 붙여짐
- Ajax: Asynchronous JavaScript+XML의 약어
- 오래된 새로운 기술: 이미 있었던 기술을 조합하여 새롭게 발전시킨 기술

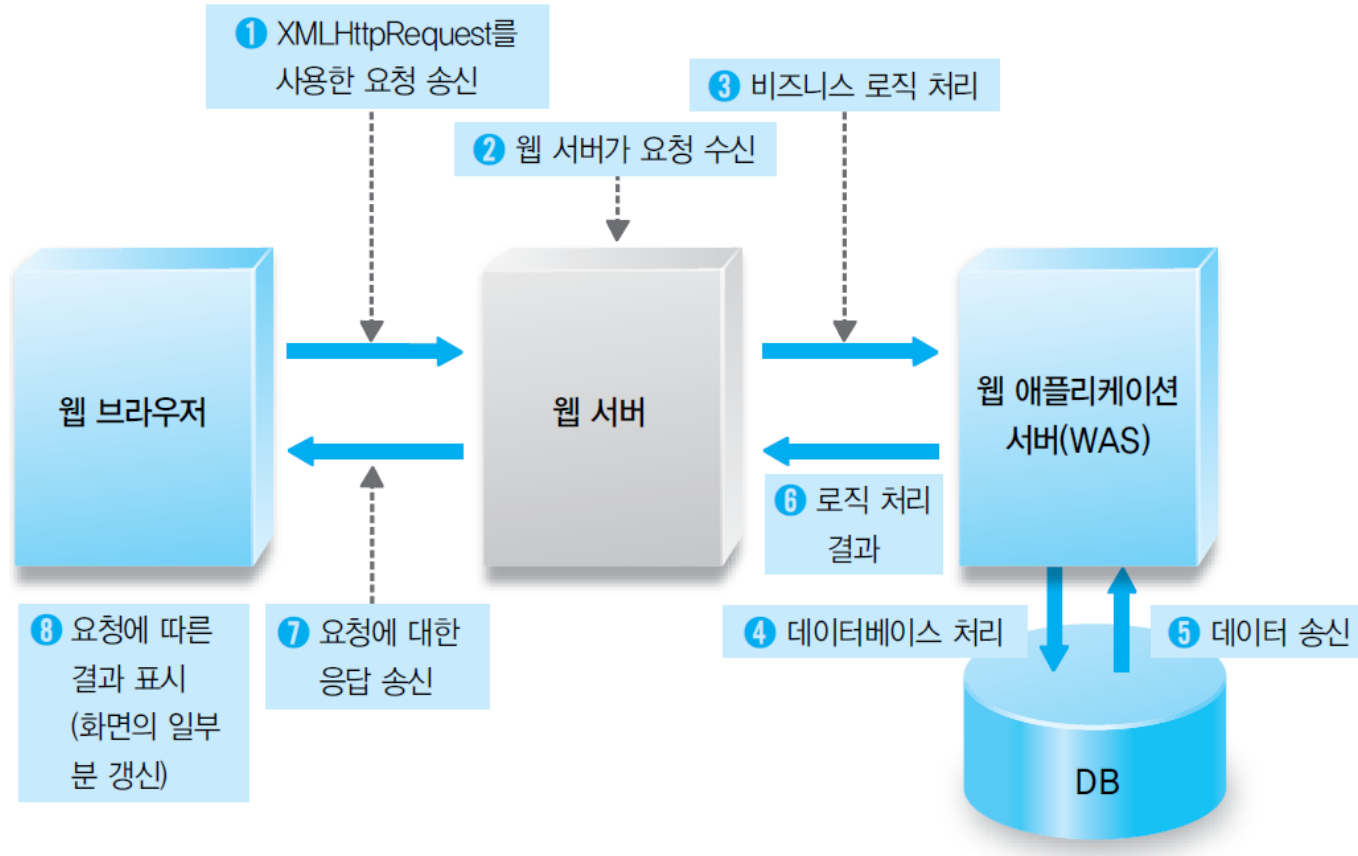
# 1. Ajax의 개요 및 구조

## □ Ajax를 이루는 구성 요소

- ▣ XHTML(Extensible HyperText Markup Language, 현재는 HTML5)과 CSS(Cascading Style Sheets)를 사용한 표준 기술 기반의 웹 페이지
- ▣ DOM(Document Object Model)을 사용한 동적인 화면 표시와 상호 작용
- ▣ XML과 XSLT 등을 사용한 데이터의 변경과 조작
- ▣ XMLHttpRequest를 사용한 비동기적인 데이터 전송
- ▣ 이것들을 결합해서 사용하는 자바스크립트

# 1. Ajax의 개요 및 구조

## □ Ajax의 비동기적 통신의 구조(데이터의 흐름)



### ▲ Ajax의 동작 방식

# 1. Ajax의 개요 및 구조

## □ Ajax의 비동기적 통신의 구조(데이터의 흐름)

### ▣ 특징

- 웹 브라우저는 요청을 송신하면 응답을 기다리지 않음
- 서버는 필요한 데이터만을 응답

## 2. Ajax 라이브러리를 사용한 웹 페이지 작성

### □ Ajax 라이브러리의 개요

- ▣ Ajax를 사용한 웹 애플리케이션을 직접 작성하는 것은 상당히 힘든 작업
- ▣ 라이브러리를 사용하면 이런 힘든 부분들을 쉽게 구현
  - jQuery 라이브러리가 현재 가장 많이 사용됨

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery 라이브러리

- ▣ 웹 애플리케이션의 작성이 쉬어지며, 자바스크립트나 Ajax 및 DOM 관련 작업을 간단히 처리
- ▣ 기능
  - HTML/DOM 작업
  - CSS 작업
  - HTML 이벤트 처리
  - 각종 효과 및 애니메이션
  - Ajax
  - 각종 유틸리티 등

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery 사용 방법

#### ▣ 방법 1

- <http://jquery.com/> 사이트에서 최신 라이브러리를 다운로드
- [프로젝트]-[Webcontent]-[js]에 라이브러리 배치
- 웹 페이지에 <script> 코드 추가
  - `<script src="../js/jquery-3.11.0.min.js"></script>`

#### ▣ 방법 2

- <http://jquery.com/> 사이트에서 최신 CDN 경로 복사
- 웹 페이지에 <script> 코드 추가
  - `<script src="http://code.jquery.com/jquery-1.11.0.min.js"></script>`



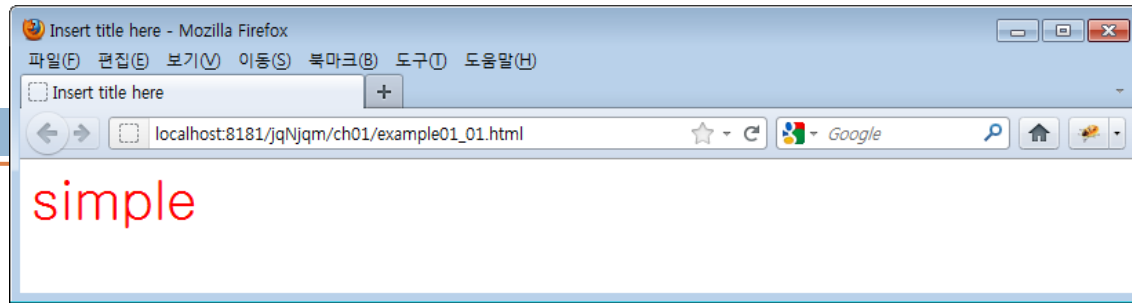
## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ 실습 1:

- ▣ <http://jquery.com/download> 로 이동
- ▣ Download the compressed, production jQuery 3.11.0 다운로드
- ▣ 프로젝트의 [WebContent]에 [js] 폴더 작성
- ▣ 다운로드한 jquery-3.11.0.min.js를 [js]폴더에 복사
  
- ▣ jQuery가 필요할 때 HTML 문서에 파일을 포함시킴
- ▣ 스타일 시트를 읽어 들인 다음 자바스크립트를 읽어 들임.
- ▣ 다른 스크립트들을 로드하기 전 jQuery(jquery.js)를 로드해야 함

# 실습문제

```
<title>Insert title here</title>
<style type="text/css">
  span { font-size: 30pt }
  .redtext { color : #f00 }
  .italictext { font-style: italic; }
</style>
<script src="../../js/jquery-3.1.1.min.js" type="text/javascript"> </script>
<script type="text/javascript">
  $('span').addClass('italictext');
</script>
</head>
<body>
<span> simple </span>
<script type="text/javascript">
  $('span').addClass('redtext');
</script>
</body>
```



이곳의 내용이 변경

표시



표시

```
...  
<title>Insert title here</title>  
<style type="text/css">  
    div#displayArea{  
        width:200px; height:200px; border:5px double #6699ff;  
    }  
</style>  
<script src="../js/jquery-1.11.3.min.js"> </script>  
<script type="text/javascript">  
    $(document).ready(function(){  
        $("button").click(function(){  
            $("#displayArea").html("<img src='../images/ansi_main2s.png' border='0'>")  
        });  
    });  
</script>  
</head>  
<body>  
    <div id="displayArea"> 이곳의 내용이 변경 </div>  
    <button> 표시 </button>  
</body> </html>
```

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery 기본 사용법

#### ▣ 기본 문법

- 선택한 엘리먼트에 어떤 동작을 수행
- `$(selector).action()`
  - `$` : *jQuery*에서 정의 및 접근에 사용
  - `(selector)` : *HTML* 엘리먼트
  - `action()` : 해당 엘리먼트에서 수행할 동작

예:

`$(this).hide()` // 현재 엘리먼트를 숨김

`$("p").hide()` // <p> 엘리먼트를 숨김

`$(".test").hide()` // class 속성의 값이 test인 엘리먼트를 숨김

`$("#test").hide()` // id 속성의 값이 test인 엘리먼트를 숨김

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery 기본 사용법

- ▣ 실렉터 - HTML 엘리먼트에 접근
- ▣ 엘리먼트를 선택해 작업을 처리하기 위해 사용
- ▣ \$로 시작해 \$()와 같은 형태로 사용
  - ( ) 안에는 엘리먼트명, 엘리먼트의 id속성 값, class속성의 값 등이 올 수 있음

실렉터	표시형태	사용예
엘리먼트명	<code>\$("엘리먼트명")</code>	- html 태그 <code>&lt;p&gt;연습&lt;/p&gt;</code> - JS에서 접근 : <code>\$("p")</code>
엘리먼트의 id 속성값	<code>\$("#id 속성 값")</code>	- HTML 태그 <code>&lt;p id="test"&gt;연습&lt;/p&gt;</code> - JS에서 접근: <code>\$("#test")</code>
엘리먼트의 class 속성 값	<code>\$(".class 속성 값")</code>	- HTML 태그 <code>&lt;p class="t1"&gt;연습&lt;/p&gt;</code> - JS에서 접근: <code>\$(".t1")</code>

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery 기본 사용법

#### ▣ HTML 엘리먼트의 내용에 접근-get/set

##### ■ 엘리먼트의 내용에 접근하려면 메소드를 사용

- `text()` : 선택한 엘리먼트의 내용을 텍스트 형태로 지정하거나 얻어냄. (예)  
`$("#p").text();` , `$("#p").text( "작업" );`
- `html()` : 선택한 엘리먼트의 내용을 HTML 태그를 포함하여 지정하거나 얻어냄
- `val()` : 폼 필드의 값을 지정하거나 얻어냄

HTML 태그

<P> 연습 </P>

-JS에서 접근

`$("#P").text()` //text() 메소드를 사용하여 <p> 태그의 내용을 얻어옴

HTML 태그

<P> 연습 </P>

-JS에서 접근

`$("#P").text("작업")` //text() 메소드를 사용하여 <p> 태그의 내용을 변경

## 엘리먼트 내용 변경

```
...
<script src="../js/jquery-1.11.3.min.js"> </script>
<script type="text/javascript">
    $(document).ready(function(){
        $("#b1").click(function(){
            $("#b2").text($("#p").text());
        });
        $("#b2").click(function(){
            $("#display").html("<img src='../images/myFace.png' border='0'/>");
        });
    });

</script>
</head>
<body>
<p> 이미지 표시 </p>
<button id="b1">버튼 레이블 변경</button>
<div id="display"> </div>
<button id="b2">버튼</button>
</body>
</html>
```

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery 기본 사용법

#### ▣ 이벤트

- 웹 페이지에서 발생한 이벤트를 제어하는 방법 제공
- 마우스 이벤트 종류
  - *click* : 엘리먼트를 클릭 시 발생
  - *dblclick* : 엘리먼트를 더블 클릭 시 발생
  - *mouseenter* : 엘리먼트에 마우스 포인터가 위치
  - *mouseleave* : 엘리먼트에 위치된 마우스 포인터가 나가면 발생

예 :

```
$("#p").click(function(){  
    alert("test");  
});
```

```
$("#s1").mouseenter(function(){  
    alert("마우스포인터가 id가 s1에 위치됨");  
});
```



## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery 기본 사용법

#### ▣ 이벤트

- 웹 페이지에서 발생한 이벤트를 제어하는 방법 제공
- 키보드 이벤트 종류
  - *keypress* : 키보드를 눌렀다 놓으면 발생
  - *keydown* : 키보드를 누르면 발생
  - *keyup* : 눌린 키보드를 놓으면 발생

예:

```
$("#input").keypress(function(){  
    $(this).css("font-color", "#cccccc");  
});
```

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery 기본 사용법

#### ▣ 이벤트

- 웹 페이지에서 발생한 이벤트를 제어하는 방법 제공
- 폼 이벤트 종류
  - *submit* : 폼을 전송 시 발생
  - *change* : 폼 엘리먼트가 변경되면 발생
  - *focus* : 폼 엘리먼트의 포커스가 위치되면 발생
  - *blur* : 폼 엘리먼트가 포커스를 잃으면 발생

```
예:  
$("input").focus (function(){  
    $(this).css("background-color", "#cccccc");  
});
```

## □ jQuery 기본 사용법

### ▣ 이벤트

- 웹 페이지에서 발생한 이벤트를 제어하는 방법 제공
- 도큐먼트/윈도우 이벤트 종류
  - *ready* : 페이지의 로드가 완료되면 발생
  - *load* : 페이지가 로드되면 발생
  - *resize* : 웹 브라우저의 창의 크기를 변경하면 발생
  - *scroll* : 웹 브라우저 창의 스크롤을 이동하면 발생
  - *unload* : 페이지가 언로드되면 발생

예:

```
$(document).ready(function(){  
    $('#result').load("test.txt");  
});
```

## 엘리먼트 이벤트 처리 예

...

```
<title>jQuery 이벤트처리</title>
```

```
<script src="../js/jquery-1.11.0.min.js"></script>
```

```
<script>
```

```
$(document).ready(function(){
```

```
    $("p").mouseenter(function(){//<p>엘리먼트에 마우스포인터를 위치시키면 자동실행  
        $(this).text("왔구려, 마우스포인터!!!");  
    });
```

```
    $("p").mouseleave(function(){//<p>엘리먼트에서 마우스포인터가 나가면 자동실행  
        $(this).text("돌아와 마우스포인터!!!");  
    });
```

```
    $("button").dblclick(function(){//<button>엘리먼트를 더블클릭하면 자동실행  
        $(this).css("background-color","#cccccc");  
    });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
    <p>마우스 포인터를 여기에!!!</p>
```

```
    <button>더블클릭하시구려.</button>
```

```
</body>
```

```
</html>
```

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery Ajax

- ▣ Ajax 기능을 구현하는 메소드들을 제공
  - 서버로부터 text, HTML, XML 또는 JSON 형태의 파일을 요청하고 응답 받을 수 있는 기능을 제공
- ▣ 간단한 코드만으로 Ajax 기능을 구현 가능

메소드명	설명
\$.get()	서버로 HTTP get 방식의 요청을 함
\$.getJSON()	HTTP get방식을 사용하여 JSON 데이터 요청함
\$.post()	서버로 HTTP post 방식의 요청을 함
.load()	서버로 데이터를 요청하고 HTML 엘리먼트에 응답받은 결과를 로드함
\$.ajax()	비동기 Ajax 요청을 수행함. post 방식을 지정해서 사용

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery Ajax

#### ▣ 서버 요청과 관련된 메소드

- `.load()` : 서버로 데이터를 요청하고 HTML 엘리먼트에 응답받은 결과를 로드
- 응답 받은 결과를 화면에 표시해야 하는 로그인폼, 회원가입 폼, 글 목록 등을 실행할 때 주로 사용
- 사용 예
  - `$( "#result" ).load( "ajax/test.html" );`

// load() 메소드 예:

...

<title>jQuery Ajax메소드 -load() </title>

<script src="../js/jquery-1.11.0.min.js"> </script>

<script>

\$(document).ready(function(){

    //[결과]버튼을 클릭하면 xhrTest1.jsp페이지가 실행된다.

    \$("#b1").click(function(){

        \$("#result").load("../images/xhrTest1.jsp");

    });

});

</script>

</head>

<body>

    <button id="b1">결과</button>

    <div id="result"> </div>

</body>

</html>

```
<title>jQuery Ajax메소드 -load() 응답처리 </title>
<style type="text/css">
  #result{ width : 200px; height : 200px; border : 5px double #6699FF; }
</style>
<script src="../js/jquery-1.11.0.min.js"> </script>
<script>
  $(document).ready(function(){
    //[결과]버튼을 클릭하면 xhrTest3.txt가 <div id="result">엘리먼트에 로드
    $("#b1").click(function(){
      $("#result").load("xhrTest3.txt", function(response,stu,xhr){
        if(stu=="success")//서버요청이 성공시
          alert("로드 성공!"); //메시지상자 표시
        if(stu=="error")//서버요청 실패시
          alert("에러: "+xhr.status+": "+xhr.stu);
      });
    });
  });
</script>
</head>
<body>
  <button id="b1">결과</button>
  <div id="result"> </div>
</body>
</html>
```



## □ jQuery Ajax

### ▣ 서버 요청과 관련된 메소드

- \$.get() : 서버로 HTTP get 방식의 요청
- .load() 메소드를 사용하는 것과 결과가 거의 같음
- 사용 예
  - `$.get( "ajax/test.html", function(){});`
  - `$.get( "ajax/test.html", function(data,status){});`

```
<title>jQuery Ajax메소드 - $.get()</title>
<style type="text/css">
  #result{ width : 200px; height : 200px; border : 5px double #6699FF; }
</style>
<script src="../js/jquery-1.11.0.min.js"> </script>
<script>
  $(document).ready(function(){
    $("#b1").click(function(){//[결과]버튼을 클릭하면 자동실행
      //xhrTest3.txt를 get방식으로 요청
      $.get("xhrTest3.txt", function(data,status){
        //data: xhrTest3.txt의 내용 + status: 처리상태=> resultStr에 저장
        var resultStr = "데이터: " + data + "\n처리상태: " + status;
        //resultStr내용을 <div id="result">에 표시
        $("#result").text(resultStr);
      });
    });
  });
</script>
</head>
<body>
  <button id="b1">결과</button>
  <div id="result"> </div>
</body>
</html>
```

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery Ajax

#### ▣ 서버 요청과 관련된 메소드

- \$.post() : 서버로 HTTP post 방식의 요청을 함
- 사용 예

```
$.post( "ajax/test.jsp",  
    { //전송할 파라미터  
        name:"kingdora",  
        status:"homebody "  
    },  
    function(data, status){  
    }  
);
```

```
<title>jQuery Ajax메소드 - $.post()</title>
<style type="text/css">
  #result{ width : 200px; height : 200px; border : 5px double #6699FF; }
</style>
<script src="../js/jquery-1.11.0.min.js"></script>
<script>
  $(document).ready(function(){
    $("#b1").click(function){//[결과]버튼을 클릭하면 자동실행
      //process.jsp페이지에 요청데이터를 보낸 후 결과를 반환받음
      $.post("process.jsp", //요청페이지
        { //요청페이지에 실어서 보낼 데이터
          name:"kingdora",
          stus:"homebody"
        },
        function(data,status){//응답내용 처리
          if(status == "success")//요청이 제대로 처리되었으면
            $("#result").html(data);
        });
    });
  });
</script>
</head>
<body>
  <button id="b1">결과</button>
  <div id="result"></div>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<meta name="viewport" content="width=device-width,initial-scale=1.0"/>

<% request.setCharacterEncoding("utf-8");%>

<%
    String resultStr = "처리결과: <br>";
    String name = request.getParameter("name");
    String stus = request.getParameter("stus");
    resultStr += "이름은 " + name + "이고,<br>";
    resultStr += "현재상태는 " + stus ;
    out.println(resultStr);
%>
```

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery Ajax

#### ▣ 서버 요청과 관련된 메소드

- \$.ajax() : 비동기 Ajax 요청을 수행함. get, post방식을 지정해서 사용
- 주로 로그인 처리, 회원가입처리, 글쓰기처리 등과 같이 DB와 연동 후 처리 결과만을 반환하는 경우에 사용

## 2. Ajax 라이브러리를 사용한 웹페이지 작성

### □ jQuery Ajax

#### ▣ 서버 요청과 관련된 메소드

##### ■ 사용 예

```
$.ajax({  
    type: "POST",  
    url: "loginPro.jsp",  
    data: query,  
    success: function(data){}  
});
```

```
<title>jQuery Ajax메소드 - $.ajax()</title>
<style type="text/css">
    #result{ width : 200px; height : 200px; border : 5px double #6699FF; }
</style>
<script src="../js/jquery-1.11.0.min.js"> </script>
<script>
    $(document).ready(function(){
        $("#b1").click(function(){ //[결과]버튼을 클릭하면 자동실행
            var query = {name : "kingdora", stus : "homebody"}; //요청페이지에 전송할 데이터
            //process.jsp페이지에 요청데이터를 보낸 후 결과를 반환받음
            $.ajax({
                type: "POST", //전송방식
                url: "process.jsp", //요청페이지
                data: query, //전송데이터
                success: function(data){ //요청페이지를 실행한 결과
                    $("#result").html(data);
                }
            });
        });
    });
</script>
</head>
<body>
    <button id="b1">결과</button>
    <div id="result"> </div>
</body>
</html>
```