jQuery Ajax & JSON

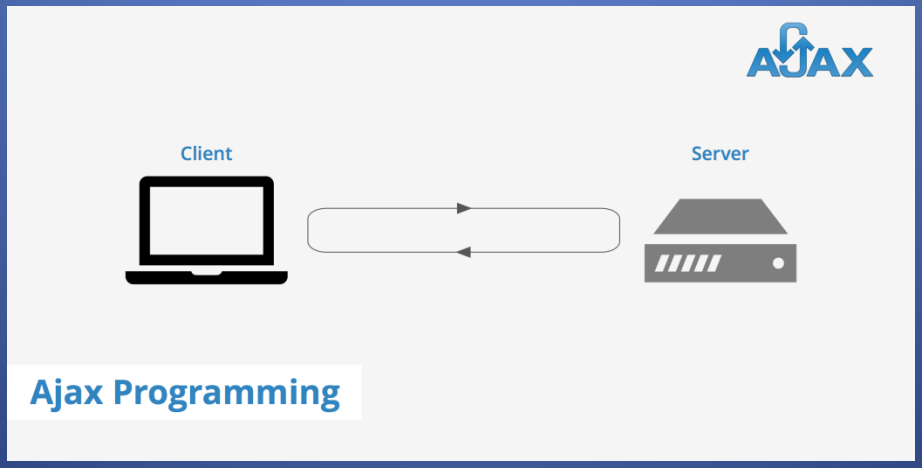
jQuery Ajax

- Ajax(Asynchronous JavaScript and XML, 에이잭스)란 비동기 방식의 Javascript와 XML을 가리킴

동기방식과 비동기 방식의 차이점

- 동기방식 : 서버에 신호를 보냈을 때, 응답이 돌아와야 다음 동작을 수행하는 방식
- 비동기 방식 : 동기방식과는 반대로 신호를 보냈을 때, 응답 상태와 상관없이 다음 동작을 수행

- Ajax 이용 목적
  - ✓ 화면 전환 없이 클라이언트 측과 서버 측간에 XML, JSON(Javascript Object Notation), 텍스트, HTML 등의 정보를 교환하기 위함
  - ✓ 자료 요청시 시간이 소요되지만, Ajax 이용시 방문객이 컴퓨터 서버에 자료 요청시 화면 전환 없이 내용만 교체되어 볼 수 있다는 장점을 활용



jQuery Ajax

동기방식과 비동기 방식의 차이점

- 동기방식 : 서버에 신호를 보냈을 때, 응답이 돌아와야 다음 동작을 수행하는 방식
- 비동기 방식 : 동기방식과는 반대로 신호를 보냈을 때, 응답 상태와 상관없이 다음 동작을 수행

- 동기 방식은 요청이 들어오면 그 뒤에 들어온 요청은 이전에 들어온 요청의 해결이 다될 때까지 대기 (진행 속도는 느리지만, 안정적 접근 가능)
- 비동기 방식은 첫번째 요청이 들어와 그것을 해결하는 도중 두번째 요청이 들어오면 두번째 요청과 첫번째 요청을 동시에 처리하는 방식 (속도는 빠르지만 불안정적 접근)

jQuery Ajax

- 웹 서버 설치하기

<웹 서버>란

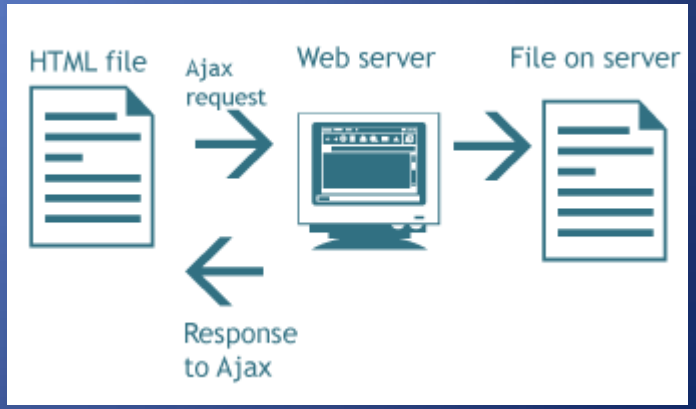
- 웹 서비스를 제공하는 컴퓨터를 말함.
- 완성된 웹 문서를 컴퓨터에 저장만 해 둔다면 외부의 방문자가 웹 문서를 볼 수 없기 때문에
- 웹 서버를 구축해 웹 문서를 인터넷 환경 하에 누구나, 어디에서나 접근이 가능하도록 구성

- 웹 서버 설치 조건

✓ Ajax를 이용하여 방문객이 폼에 입력 요소를 작성하여 데이터를 전송하고 답을 요청하는 테스트 페이지 만들기 : 웹 서버 + 액션 페이지(서버 측 스크립트 페이지, 즉 PHP/ASP/JSP)

✓ 위의 서버 측 스크립트 페이지는 전송된 데이터를 이용하여 정상적인 데이터인지를 검사한 후, 응답 값을 되돌려 전송하게 됨

✓ 서버 측 스크립트 페이지는 서버컴퓨터에 저장



jQuery Ajax

- 웹 서비스를 제공하는 방법
  - ✓ APM(Apache PHP MySQL)을 이용하여
    - ① 서버를 현재 컴퓨터에 설치하는 방법
    - ② 외부에 구축된 서버인 웹 호스팅을 임대하여 이용하는 방법

웹 서버와 웹 호스팅의 차이점

- 공통점 : 방문객인 클라이언트에세 웹 서비스를 제공
- 웹 서버 : 사용자가 직접 서버를 구축하여 웹서비스를 제공 (웹 서버에 대한 전문 지식 필요. 트래픽, 네트워크, 웹 프로그래밍, 방화벽, OS별 접근방식 등)
- 웹 호스팅 : 이미 서버가 구축된 외부 컴퓨터에 다수 이용자가 소량의 저장 공간을 나누어 임대 사용(웹 서버에 대한 간단한 지식만으로 가능)

jQuery Ajax

- Ajax 관련 메서드

종류	설명
load( )	외부 콘텐츠를 가져올 때 사용
$.ajax( )	데이터 서버에 HTTP POST, GET 방식으로 전송이 가능. (X)HTML, XML, JSON, 텍스트 유형에 데이터를 요청할 수 있는 통합적인 메서드. 이 표 상에서 $.post( ), $.get( ), $.getJSON( ) 메서드의 기능을 하나로 합쳐 놓은 것이라고 보면 됨.
$.post( )	데이터를 서버에 HTTP POST 방식으로 전송한 후 서버 측의 응답을 받을 때 사용
$.get( )	데이터를 서버에 HTTP GET 방식으로 전송한 후 서버 측의 응답요청을 받을 때 사용
$.getJSON( )	데이터를 서버에 HTTP GET 방식으로 전송한 후, 서버 측 응답을 JSON 형식으로 받을 때 사용
$.getScript( )	Ajax를 이용하여 외부 자바스크립트를 불러옴 $("button").click(function(){ $.getScript("demo_ajax_script.js"); });

jQuery Ajax

- Ajax 관련 메서드

종류	설명
.ajaxStop(function(){...})	비동기 방식으로 서버에 응답 요청이 완료 되었을 때 함수에 실행문이 수행
.ajaxSuccess(function(){...})	Ajax 요청이 성공적으로 완료되면 함수에 실행문을 수행
serialize( )	방문자가 입력 요소에 값을 입력하고 전송하였을 때, 데이터 전송 방식인 'name1=value1 & name2=value2, ...' 쿼리 스트링 형식의 데이터로 변환하여 액션 페이지에 전송
serializeArray( )	방문자가 입력 요소에 값을 입력하고 전송하였을 때, 전송 방식인 key1:value1, key2:value2, ... JSON 데이터로 변환하여 액션 페이지에 전송
.ajaxComplete(function(){...})	Ajax 통신이 완료되면 함수문에 실행문을 실행

jQuery Ajax

- load( ) 메서드
  - ✓ 사용자가 지정한 URL 주소에 데이터를 전송하고 외부 콘텐츠를 요청하여 가져올 때 사용
  - ✓ 요청한 콘텐츠를 이용해 선택한 요소에 내용을 바꿀 수 있음
  - ✓ 기본형 $("요소선택").load(URL, data, 콜백함수);
  - ✓ URL 주소 : 외부 콘텐츠를 요청할 외부주소 입력
  - ✓ Data : 전송할 데이터 입력
  - ✓ 콜백함수 : 전송이 완료되면 콜백함수에 저장된 실행문 실행

[HTML]

```
<script type="text/javascript">
$(function(){
   $("#d1").load("document_1.txt");
});
</script>
<body>
   <div id="d1">내용</div>
</body>
```

[document_1.txt]

```
<h1>삽입 제목</h1>
<p>삽입 내용</p>
```

jQuery Ajax

- load( ) 메서드 – 실습 과제
  - ✓ 상단의 헤더와 하단의 푸터를 외부 파일로 만들고 각각의 위치로 불러오세요.

[index.html]

```
<style>
   #cont_01{height:200px; background:#ffaaff;}
   #cont_02{height:300px; background:#aaaaff;}
</style>
<body>
   <div id=“header"></div>
   <section id="cont_01">내용</div>
   <section id="cont_02">내용</div>
   <div id=“footer"></div>
</body>
```

[header.html]

```
<div style=“height:100px; background:#aaffaa;”>
   <div style=“float:left”>LOGO</div>
   <nav style=“float:right”>MENUS</nav>
</div>
```

[footer.html]

```
<div style=“height:80px; background:#555; color:#fff;”>
   <p>CopyRights &copy; James 2018. All rights reserved.</>
</div>
```

jQuery Ajax

- $.ajax( ) 메서드
  - ✓ 사용자가 지정한 URL 경로에 파일의 데이터를 전송하고 입력한 URL 경로 파일로부터 요청한 데이터를 불러옴
  - ✓ 이때 불러올 수 있는 외부데이터로는 텍스트, HTML, XML, JSON 유형 등이 있음
  - ✓ 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;
    }
});
```

  - ✓ url에는 데이터 전송 및 요청할 외부 주소를 입력
  - ✓ type에는 전송방식을 입력
  - ✓ data에는 전송할 데이터를 입력
  - ✓ dataType은 서버로부터 받아올 데이터 형식을 지정
  - ✓ success은 데이터 전송 및 요청이 정상적으로 이뤄지면 함수 내의 실행문이 실행

jQuery Ajax

- $.ajax( ) 메서드 옵션 종류

종류	설명
async	통신을 동기 또는 비동기 방식으로 설정하는 옵션. 기본값은 비동기 방식(true). 비동기 방식으로 설정되어 있다면, 방문객이 컴퓨터에서 서버로 데이터를 전송하고 요청하는 동안에서도 다른 작업 수행이 가능
beforeSend	http 요청하기 전에 함수를 실행하는 이벤트 핸들러
cache	요청한 페이지를 인터넷에 캐시(저장)할지 여부를 설정. 기본 값은 true.
complete	Ajax가 완료되었을 때 함수를 실행하는 이벤트 핸들러 또는 http 요청 완료시 이벤트 핸들러
contentType	서버로 전송시킬 데이터의 content-type을 설정. 기본 값은 application/x-www-form-urlenecoded.
data	서버로 전송할 데이터를 지정. http 요청 후 return 하는 값
dataType	서버에서 요청 받아올 데이터의 형식을 지정. (xml, html, json, jsonp, script, text) 생략하면 요청한 자료에 맞게 자동으로 형식이 설정
error	통신에 문제가 발생했을 때 함수를 실행. http 요청 실패시 이벤트 핸들러

jQuery Ajax

- $.ajax( ) 메서드 옵션 종류

종류	설명
success	Ajax로 통신이 정상적으로 이뤄지면 함수를 발생
timeout	통신 시간을 제한. 시간단위를 밀리초
type	데이터를 전송할 때 방식(get/post)을 설정
url	데이터를 전송할 페이지를 설정. (문자열) 기본 값은 현재 페이지.
username	HTTP 액서스를 할 때 인증이 필요할 경우 사용자 이름을 지정

jQuery Ajax

- $.ajax( ) 메서드

```
<script type="text/javascript">
$(function(){
   $("#member").on("submit",function(){ //"확인"버튼을 눌렀을 때...
      var d=$(this).serialize(); //폼 요소에 전송할 데이터를 재가공
      $.ajax({
          url:"member.php", //데이터 전송 및 요청할 URL 주소
          type:"post",  //데이터 전송 방식
          data:d,   //전송할 데이터
          success:function(result){ //전송 및 요청이 정상적으로 되었을 때...
            $("#txt1").text(result);
          }
      });
      return false; //action 페이지로 전환되는 것을 차단
   });
});
</script>
<body>
<form action="member.php" method="post" name="member"  id="member">
  <fieldset>
    <legend>회원가입</legend>
    <p>
      <label for="user_name">이름</label>
      <input type="text" name="user_name" id="user_name" />
    </p>
    <p><input type="submit" value="확인" /></p>
  </fieldset>
</form>
<h1 id="txt1"></h1>
</body>
```

jQuery Ajax

- $.ajax( ) 메서드

[member.php]

```
<?php
  echo $_POST["user_name"]."님 반갑습니다";
?>
```

jQuery Ajax

- XML 데이터 요청하기
  - ✓ XML(Extensible Markup Language)은 확장 가능 언어.
  - ✓ 태그명을 사용자가 임의로 작성 가능
  - ✓ 주로 데이터 배포 목적으로 작성(예로 네이버 등의 포털 사이트에서 메인 페이지에서 뉴스 / 정보 기사 등이 해당)
  - ✓ 기본형

```
<?xml version="1.0" encoding="UTF-8"?>
<tag1>
   <tag2>내용</tag2>
</tag2>
```

jQuery Ajax

- XML 데이터 요청하기

```
<script type="text/javascript">
$(function(){
   $.ajax({ //외부에 데이터 전송 또는 요청합니다.
      url:"rank.xml", //데이터를 요청할 URL 주소
      dataType:"xml", //요청할 데이터의 타입
      success:function(result){ //성공적으로 데이터가 요청되었을 때...
         if($(result).find("rank").length>0){ //요청한 xml 데이터에서 <rank> 태그의 개수가 0보다 클 영우 블록 내의 실행문을 실행
            $(result).find("rank").each(function(){ //<rank> 태그의 총 개수만큼 function(){...}에 실행문을 실행
               var name=$(this).find("k").text(); //<rank 태그만큼 함수 내의 실행문이 실행될 때마다 $(this)에는 <rank> 태그가 순차적으로 선택. 그리고, <k> 태그에 텍스트를 변수 name에 할당
               var result="<li>"+name+"</li>"; //<li> 태그의 변수 name에 할당된 값을 텍스트로 결합시켜 변수 result에 할당
               $("#wrap ol").append(result); //<ol> 태그의 하위 맨 마지막에 result 값을 삽입
            });
         }
      }
   });
});
</script>
<body>
   <h1>인기 검색어</h1>
   <div id="wrap">
      <ol></ol>
   </div>
</body>
```

jQuery Ajax

- XML 데이터 요청하기

[rank.xml]

```
<?xml version="1.0" encoding="UTF-8"?>
<result>
   <item>
      <rank>
         <k>방탄 소년단</k>
      </rank>
      <rank>
         <k>여자친구</k>
      </rank>
      <rank>
         <k>트와이스</k>
      </rank>
   </item>
</result>
```

jQuery Ajax

- JSON 데이터 요청하기
  - ✓ JSON(Javascript Object Notation)은 자바스크립트 객체 표기법.
  - ✓ key와 value 값이 쌍으로 구성된 형태의 객체 표기법
  - ✓ 클라이언트와 서버 사이에 정보를 교환하기 위해 사용
  - ✓ 형식은 xml보다 간단
  - ✓ 기본형 `{key1:value1, key2:value2, key3:value3, ... }`

jQuery Ajax

- JSON 데이터 요청하기

```
<script type="text/javascript">
$(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){  //JSON 데이터의 rank에 저장된 값 만큼 함수 내의 실행문을 반복하여 실행. 이때 매개변수 d에는 rank의 값으로 저장된 객체가 순서대로 할당
            $("#wrap ol").append("<li>"+d["k"]+"</li>");
         });
      }
   });
});
</script>
<body>
   <h1>인기 검색어</h1>
   <div id="wrap">
      <ol></ol>
   </div>
</body>
```

[rank.json]

```
{"rank":[
  {"k": " 방탄소년단 " },
  {"k": " 여자친구 " },
  {"k": " 트와이스 " }]
}
```

jQuery Ajax

- 실습 예제 – 메뉴를 구성한 후, 각 메뉴 클릭시 해당하는 페이지 ajax로 불러오기(계속)

```
<script type="text/javascript">
$(document).ready(function(){
  var $menu = [
    {$name : "menu_01", $url : "/sub-menu/menu-detail_01.html"},
    {$name : "menu_02", $url : "/sub-menu/menu-detail_02.html"},
    {$name : "menu_03", $url : "/sub-menu/menu-detail_03.html"},
    {$name : "menu_04", $url : "/sub-menu/menu-detail_04.html"}
  ];
  for(i=0; i<$menu.length; i++){
    $(".menu ul").append("<li><a href='#0"+i+"'>"+$menu[i].$name+"</a></li>");
  };
  $(".cont").load($menu[0].$url);
  $(".menu ul li").each(function(index){
    $(this).click(function(){
      $.ajax({
            type : 'post',
            url : $menu[index].$url,
            dataType : 'html' ,
            success: function(data) {
                  $(".cont").html(data);
            }
      });
    });
  });
});
</script>
```

jQuery Ajax

- 실습 예제 – 메뉴를 구성한 후, 각 메뉴 클릭시 해당하는 페이지 ajax로 불러오기(계속)

```
<header>
		<div class="logo">LOGO</div>
		<nav class="menu">
			<ul></ul>
		</nav>
</header>
<section>
		<div class="cont"></div>
</section>
```

[menu-detail_01.html]

```
<div>
		<p>menu-cont-01-01</p>
		<p>menu-cont-01-02</p>
		<p>menu-cont-01-03</p>
		<p>menu-cont-01-04</p>
</div>
```

[menu-detail_02.html]

```
<div>
		<p>menu-cont-02-01</p>
		<p>menu-cont-02-02</p>
		<p>menu-cont-02-03</p>
		<p>menu-cont-02-04</p>
</div>
```