

□ jQuery : Javascript를 이용하여 HTML 태그로 구성된 웹 페이지의 구성 요소들을 동적으로 변화 시킬 수 있게 도움을 줄 수 있도록 구현 된 Framework 이다.

□ jQuery를 이용하기 위해서는 <http://code.jquery.com> 에서 제공하는 jquery.js 파일을 HTML 페이지에서 호출한다.

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>
```

※ <!DOCTYPE html> 은 HTML 5 기준으로 작성되었음을 브라우저에게 알리는 역할을 한다.

□ jQuery 는 HTML 태그의 요소들을 동적으로 변화시키기 위해 변화 시킬 태그에 접근 할 수 있도록 Selector 라는 개념을 이용한다.

○ Selector 는 HTML 태그의 고정된 명칭이나(<p>, , <div>, 등) HTML 태그의 ID 또는 CLASS 값으로 설정된 변수명을 통해 해당하는 태그를 선택할 수 있도록 돕는다.

○ Selector를 이용하기 위해서는 \$ 문자를 사용하며 선택하려는 태그를 괄호안에 넣는데, 단순히 태그만으로 지정된 경우 괄호 안에 태그의 명칭을 넣어서 사용하며, ID 값이 설정된 경우 “#d명”, CLASS 로 설정된 경우 “.class명”을 괄호안에 넣는다.

□ Selector를 이용하면 해당하는 태그에 접근이 가능하고, 해당 태그에 포함된 문자열은 \$(selector).text()와 같이 읽어올 수 있다.

※ alert 함수는 Javascript 함수로 브라우저에 경고창을 띄워주며, 괄호안에 포함된 문자열을 경고창에 출력해준다.

< ex1-1 - 태그로만 이뤄진 경우 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<p>안녕하세요</p>

<script>
    alert($(".p").text());
</script>
```

< ex1-2 -ID 값이 설정된 경우 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<p id="select">안녕하세요</p>

<script>
  alert($("#select").text());
</script>
```

< ex1-3 -CLASS 값이 설정된 경우 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<p class="select">안녕하세요</p>

<script>
  alert($(".select").text());
</script>
```

- 주의 할 점은 특수기호로 ID 또는 CLASS 명이 설정되어 있다면 Escape 처리를 해줘야 한다는 것이다.

< Escape 처리가 필요한 특수기호 >

```
! " # $ % & ' ( ) * + , . / : ; < = > ? @ [ \ ] ^ ` { | } ~
```

< ex1-4 -특수기호 Escape 처리 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<p class="!select">안녕하세요</p>

<script>
  alert($(".!select").text());
</script>
```

- 태그에 포함된 문자열을 읽어오는 방식은 태그에 따라 다른데, VALU

E 값이 포함되는 INPUT 태그의 경우 \$("selector").val()을 사용해야 한다.

< ex1-5 -val 함수를 이용한 문자열 출력 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<input type="text" name="select" value="test">

<script>
  alert($("#input").val());
</script>
```

○ INPUT 태그의 경우 ID, CLASS 명을 설정하지 않더라도 NAME 에 설정된 변수명을 이용하여 접근 할 수 있다.

< ex1-6 -NAME 에 설정된 변수명 이용 접근 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<input type="text" name="select" value="test">

<script>
  alert($("#input[name='select']").val());
</script>
```

□ 태그에서 값을 읽어오는 방식과 같이 값을 설정하는 것도 동일한 함수를 이용한다.

< ex1-7 -ID 값 설정된 변수명 이용 값 설정 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<p id="select">안녕하세요</p>

<script>
  $("#select").text("값 변경하기");
</script>
```

< ex1-8 -CLASS 값 설정된 변수명 이용 값 설정 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<p class="select">안녕하세요</p>

<script>
    $(".select").text("값 변경하기");
</script>
```

< ex1-9 -NAME 값 설정된 변수명 이용 값 설정 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<input type="text" name="select" value="test">

<script>
    $("input[name='select']").val("값 변경하기");
</script>
```

○ HTML 태그로 값을 설정하기 위해서는 .text() 대신 .html()을 이용한다.

< ex1-10 -HTML 태그 추가하기 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<p id="select">안녕하세요</p>

<script>
    $("#select").html("<b>값 변경하기</b>");
</script>
```

○ .text() 또는 .html() 을 이용하면 함수안에 포함된 문자열로 문자열을 대체하는데, .append() 함수를 사용하면 기존의 문자열과 결합할 수 있다.

< ex1-11 -내용 결합하여 출력하기 >

```
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<p id="select">안녕하세요</p>

<script>
    $("#select").append(" 반갑습니다.");
</script>
```

□ jQuery를 이용한 이벤트 처리 방법

- 이벤트란 특정한 상황이 발생한 경우를 말하는데 키보드 입력, 마우스 클릭과 같은 상황을 말한다.
- 이벤트를 처리하기 위해서는 Handler 가 필요한데, Handler 는 데이터 처리에 대한 순서와 방법을 정해놓은 것으로서 “함수”를 Handler 로 등록하여 사용한다.
- 함수의 형태는 아래와 같다.

< ex2-1 -함수 정의하고 호출하기 >

```
<!DOCTYPE html>
<script>
    function func(){ // 함수 정의
        alert("함수 호출");
    };

    func(); // 함수 호출
</script>
```

- 키보드가 입력된 경우 위에서 작성한 함수를 호출하는 이벤트처리 Handler 를 등록하는 방법은 아래와 같다.

< ex2-2 -키보드 DOWN 이벤트 처리 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<input type="text" name="input">

<script>
    function func(){
        alert("함수 호출");
    };

    $("input[name='input']").keydown(func);
//Handler 등록 시에는 괄호를 제외 한다.
</script>
```

< ex2-3 -키보드 UP 이벤트 처리 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<input type="text" name="input">

<script>
    function func(){
        alert("함수 호출");
    };

    $("input[name='input']").keyup(func);
//Handler 등록 시에는 괄호를 제외 한다.
</script>
```

○ 마우스 클릭 이벤트를 처리하는 방법은 아래와 같다.

< ex2-4 -마우스 버튼 DOWN 이벤트 처리 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div id="mouse_event">Click ME !!!</div>

<script>
  function func(){
    alert("Mouse Down");
  };

  $("#mouse_event").mousedown(func);
</script>
```

< ex2-5 -마우스 버튼 UP 이벤트 처리 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div id="mouse_event">Click ME !!!</div>

<script>
  function func(){
    alert("Mouse UP");
  };

  $("#mouse_event").mouseup(func);
</script>
```

- Mouse DOWN / Mouse UP 이벤트의 경우 해당 위치에서 이벤트가 발생한 경우에 함수가 호출되는데, CLICK 이벤트의 경우 해당 위치에서 MOUSE DOWN 과 UP 이벤트가 동시에 해당 위치에서 발생해야 함수가 호출된다.

< ex2-6 -마우스 버튼 CLICK 이벤트 처리 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div id="mouse_event">Click ME !!!</div>

<script>
  function func(){
    alert("Mouse CLICK");
  };

  $("#mouse_event").click(func);
</script>
```

< ex2-7 -마우스 OVER 이벤트 처리 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div id="mouse_event">Mouse OVER</div>

<script>
  function func(){
    alert("Mouse OVER");
  };

  $("#mouse_event").mouseover(func);
</script>
```


< ex2-8 -마우스 OUT 이벤트 처리 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div id="mouse_event">Mouse OUT</div>

<script>
    function func(){
        alert("Mouse OUT");
    };

    $("#mouse_event").mouseout(func);
</script>
```

□ jQuery 의 이벤트 처리 방법을 이용하면, 체크박스 전체 체크하기, 사용자 선택에 따른 내용 추가 등 사용자의 행동에 따른 동적인 웹 페이지를 만드는데 활용이 가능하다.

○ 체크박스는 체크가 된 경우 태그 속성 값으로 “checked” 라는 값이 추가 되는데 아래의 예제를 통해 확인해 볼 수 있다.

< ex3-1 -체크된 체크박스 만들기 >

```
<!DOCTYPE html>

전체 선택 <input type="checkbox" name="checkALL"><br>
1번 <input type="checkbox" name="check[]" checked><br>
2번 <input type="checkbox" name="check[]"><br>
3번 <input type="checkbox" name="check[]"><br>
```

○ 특정 태그에 속성 값을 추가하기 위해서는 .prop() 함수를 사용한다.

< ex3-2 -전체선택 체크박스에 체크한 경우 전체선택 하기 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

전체 선택 <input type="checkbox" name="checkALL"><br>
1번 <input type="checkbox" name="check[]"><br>
2번 <input type="checkbox" name="check[]"><br>
3번 <input type="checkbox" name="check[]"><br>

<script>
    function func(){
        $("input[name='check[]']").prop("checked", true);
    };

    $("input[name='checkALL']").click(func);
</script>
```

- 위 코드의 경우 전체선택 박스를 클릭한 경우 모든 체크박스에 체크하도록 되어 있어 체크 해제되는 되지 않는 상태이다. 이를 해결하기 위해서 IF(조건) 문을 추가한다.

< ex3-3 -전체선택 체크박스 구현 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

전체 선택 <input type="checkbox" name="checkALL"><br>
1번 <input type="checkbox" name="check[]"><br>
2번 <input type="checkbox" name="check[]"><br>
3번 <input type="checkbox" name="check[]"><br>

<script>
    function func(){
        if($("#input[name='checkALL']").prop("checked"))
        {
            $("#input[name='check[]']").prop("checked", true);
        }
        else
        {
            $("#input[name='check[]']").prop("checked", false);
        }
    };

    $("#input[name='checkALL']").click(func);
</script>
```

- jQuery 를 이용하여 SELECT 박스에서 사용자가 선택한 값을 확인하는 방법에 대한 예제는 아래와 같다.

○ SELECT 박스는 선택된 경우 속성 값으로 SELECTED 값이 추가 된다.

< ex3-4 -SELECT 박스 사용자 선택 값 출력하기 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div id="date">
  <select name="YEAR">
    <option value="2017">1. 2017</option>
    <option value="2018">2. 2018</option>
    <option value="2019" selected>3. 2019</option>
  </select>
</div>

<script>
  alert($("#select[name='YEAR'] option:selected").val());
  alert($("#select[name='YEAR'] option:selected").text());
</script>
```

○ 사용자가 SELECT 박스에서 다른 값을 선택한 경우의 이벤트를 처리하기 위해서는 아래와 같이 .change() 함수를 이용한다.

< ex3-5 -SELECT 박스 사용자 선택 값 출력하기 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div id="date">
  <select name="YEAR">
    <option value="2017">1. 2017</option>
    <option value="2018">2. 2018</option>
    <option value="2019" selected>3. 2019</option>
  </select>
</div>

<script>
  function func(){
    alert($("#select[name='YEAR'] option:selected").val());
  }

  $("#select[name='YEAR']").change(func);
</script>
```


- SELECT 박스를 이용하여 년, 월, 일을 선택할 수 있는 SELECT 박스를 구현할 수 있는데, 이를 모두 HTML 태그를 이용하여 구현하려면 코드가 길어지고 복잡해지는데 jQuery 와 Javascript의 FOR(반복문)을 이용하면 간단히 해결이 가능하다.

< ex3-6 -SELECT 박스에 년/월/일 표시하기 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div id="date">
  <select name="YEAR">
  </select>
  <select name="MONTH">
  </select>
  <select name="DAY">
  </select>
</div>

<script>
  for( i = 2017; i <= 2030; i++ )
  {
    $("select[name='YEAR']").append("<option value='" + i + "'>" + i + "</option>");
  }
  for( i = 1; i <= 12; i++ )
  {
    $("select[name='MONTH']").append("<option value='" + i + "'>" + i + "</option>");
  }
  for( i = 1; i <= 31; i++ )
  {
    $("select[name='DAY']").append("<option value='" + i + "'>" + i + "</option>");
  }
</script>
```

- 위에서 만든 코드는 모든 월의 일수가 31일로 고정되어 정확한 월의 일수가 표현되지 않고 있으므로 사용자가 월을 변경할 때마다 해당하는 월의 일수에 맞춰 표시되도록 변경해야 한다.

< ex3-7 -사용자 선택에 따른 월의 일수 표시 >

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div id="date">
  <select name="YEAR">
  </select>
  <select name="MONTH">
  </select>
  <select name="DAY">
  </select>
</div>

<script>
  for( i = 2017; i <= 2030; i++ )
  {
    $("select[name='YEAR']").append("<option value='" + i + "'>" + i + "</option>");
  }
  for( i = 1; i <= 12; i++ )
  {
    $("select[name='MONTH']").append("<option value='" + i + "'>" + i + "</option>");
  }

  $("select[name='MONTH']").change(function(){
    $("select[name='DAY']").text("");

    year = $("select[name='MONTH'] option:selected").val();
    month = $("select[name='MONTH'] option:selected").val();
    date = new Date(year, month, 0);
    days = date.getDate();

    for( i = 1; i <= days; i++ )
    {
      $("select[name='DAY']").append("<option value='" + i + "'>" + i + "</option>");
    }
  });
</script>
```

□ jQuery 에 포함된 기능 중 하나는 Ajax 이다.

Ajax는 페이지 전환 없이 사용자의 입력을 서버에서 처리할 수 있도록 돕는 프레임워크 중 하나로서 동적인 웹 페이지를 구성하는데 중요한 요소로 사용된다.

○ Ajax 는 아래와 같은 형태로 구성되어 있다.

< jQuery Ajax 의 구성 >

```
function func(data){
    alert(data); // 서버로부터 전달받은 데이터를 출력
};

$.ajax(
    type: "GET" 또는 "POST",
    url: "데이터를 전송할 URL",
    data: {"변수명": "값", "변수명": "값"},
    success: func // URL 접근 성공 및 응답 받은 경우
);
```

○ Ajax 는 먼저 사용자의 입력을 서버에 전달하는 과정이 필요하기 때문에, 서버로의 데이터 전달을 위해 GET/POST 방식을 이용한다.

○ 먼저 GET 방식은 요청하는 URL 뒤에 사용자의 데이터를 추가하여 전송하는 방식으로 아래와 같이 사용한다.

- 예제로 사용된 코드는 ajax를 통해 서버의 “jquery_test.php” 페이지로 데이터를 전달하고, “jquery_test.php” 페이지에서는 넘겨받은 데이터를 출력 해주는 단순한 형태이다.
- ajax 는 “jquery_test.php” 페이지에서 출력된 문자열을 넘겨받아 출력한다.

< ex4-1 - Ajax 이용하기 GET 방식 >

ajax-get.html
<pre><!DOCTYPE html> <script src="http://code.jquery.com/jquery-3.2.1.js"></script> <script> function func(data){ alert(data); // 서버로부터 전달받은 데이터를 출력 }; \$.ajax({ type: "GET", url: "/jquery_test.php", data: {"name":"tester"}, success: func // URL 접근 성공 및 응답 받은 경우 }); </script></pre>
jquery_test.php
<pre><?php \$name = \$_GET['name']; echo \$name; ?></pre>

< ex4-2 - Ajax 이용하기 GET 방식 >

ajax-get.html
<pre><!DOCTYPE html> <script src="http://code.jquery.com/jquery-3.2.1.js"></script> <script> function func(data){ alert(data); // 서버로부터 전달받은 데이터를 출력 }; \$.ajax({ url: "/jquery_test.php?name=tester" success: func // URL 접근 성공 및 응답 받은 경우 }); </script></pre>
jquery_test.php
<pre><?php \$name = \$_GET['name']; echo \$name; ?></pre>

○ POST 방식의 경우 아래와 같은 형태로 사용이 가능하다.

< ex4-3 - Ajax 이용하기 POST 방식 >

ajax-post.html
<pre><!DOCTYPE html> <script src="http://code.jquery.com/jquery-3.2.1.js"></script> <script> function func(data){ alert(data); // 서버로부터 전달받은 데이터를 출력 }; \$.ajax({ type: "POST", url: "/jquery_test.php", data: {"name":"tester"}, success: func // URL 접근 성공 및 응답 받은 경우 }); </script></pre>
jquery_test.php
<pre><?php \$name = \$_POST['name']; echo \$name; ?></pre>

○ 위의 코드를 응용하면, HTML 코드에 데이터를 추가하여 표시되도록 할 수 있다.

- 아래 예제에서는 POST 방식으로 데이터를 전달하고 그 응답 값을 받아 HTML 페이지의 <div> 태그 위치에 응답 값을 출력하도록 하는 코드 이다.

< ex4-4 - Ajax 사용하기 POST 방식 >

ajax-post.html

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div id="result"></div>

<script>
    function func(data){
        $("#result").text(data);
    };

    $.ajax({
        type: "POST",
        data: {name:"tester"},
        url: "/jquery_test.php",
        success: func
    });
</script>
```

jquery_test.php

```
<?php
    $name = $_POST['name'];
    echo $name;
?>
```

- 위와 같이 서버에 데이터를 전송하고, 그 결과를 화면에 출력하는 형태의 코드는 회원가입 시 “ID 중복체크” 기능에서 주로 사용된다.
- 아래 예제는 “ID 중복체크” 버튼을 누르면 해당하는 ID가 데이터베이스에 존재하는지 확인 후 그 결과를 알려주는 코드이다.

< ex4-5 - Ajax 활용 ID 중복체크 기능 구현 >

id-check.html

```
<!DOCTYPE html>
<script src="http://code.jquery.com/jquery-3.2.1.js"></script>

<div>ID : <input type="text" name="id"><input type="button" value="ID체크" onclick="id_check();"></div>
<div id="result"></div>

<script>
function func(data){
    if(data == 1){
        result = "사용중인 ID 입니다.";
        color = "red";
    }
    else if(data == -1){
        result = "ID가 입력되지 않았습니다.";
        color = "red";
    }
    else{
        result = "사용할 수 있는 ID 입니다.";
        color = "blue";
    }
    $("#result").text(result).css({"color":color, "font-weight":"bold"});
};

function id_check(){
    id = $("input[name='id']").val();
    $.ajax({
        type: "POST",
        data: {id:id},
        url: "/id-check.php",
        success: func
    });
}
</script>
```

< ex4-5 - Ajax 활용 ID 중복체크 기능 구현 >

id-check.php

```
<?php
    $host = "127.0.0.1";
    $id = "root";
    $pass = "1234";
    $db = "board";

    $conn = mysqli_connect($host, $id, $pass, $db);

    $id = mysqli_real_escape_string($conn, $_POST['id']);

    if($id == '')
    {
        echo -1;
        exit;
    }

    $sql = "select id from user where id = '{$id}'";
    $result = mysqli_query($conn, $sql);
    $rows = mysqli_num_rows($result);

    echo $rows;

    mysqli_free_result($result);
    mysqli_close($conn);
?>
```