

# JSP/Servlet

## <비동기 함수>

비동기 함수(Asynchronous function)은 동기적으로 실행되지 않고, 백그라운드에서 병렬적으로 실행되는 함수입니다. 일반적으로 비동기 함수는 장기 실행 작업(네트워크 요청, 파일 읽기/쓰기, 데이터베이스 쿼리 등)을 처리하는 데 사용됩니다. 비동기 함수는 작업이 완료될 때까지 기다리지 않고 다른 작업을 수행할 수 있으므로, 애플리케이션의 반응성을 향상시킬 수 있습니다.

비동기 함수는 다양한 방식으로 구현될 수 있습니다. 여기에는 콜백(callback) 함수, 프로미스(Promise), async/await, RxJS 등이 포함될 수 있습니다. 각 방식은 다른 문법과 패턴을 가지고 있지만, 모두 비동기 작업을 처리하기 위한 목적으로 사용됩니다.

비동기 함수는 JSON을 사용하여 처리를 한다.

## <Ajax 활용>

(exam02.jsp)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>JQuery Test02</title>
<script src="https://code.jquery.com/jquery-3.7.0.js"></script>
<script>
    // getBtn 클릭시 아래쪽 띄우기(비동기 함수 사용)
    $(document).ready(function() {
        $("#getBtn").click(function() {
            $.get("process.jsp", {
                id : $("#id").val(),
                pwd : $("#pwd").val(),
                method : "get"
            }, function(resp) { // Callback 함수
                // alert(resp)
                $("#result").html(resp); // Set과 같은 역할
            })// get

        }) // getBtn

        $("#postBtn").click(function() {
            $.post("process.jsp", {
                id : $("#id").val(),
                pwd : $("#pwd").val(),
                method : "post"
            }, function(resp) {
                $("#result").html(resp);
            })// post

        })// postBtn
    })
}
```

```

$("#ajaxBtn").click(function() {
    $.ajax({
        type : "get",
        url : "process.jsp",
        data : {
            id : $("#id").val(),
            pwd : $("#pwd").val(),
            method : "ajax"
        },
        success : function(resp) {
            $("#result").html(resp);
        },
        error : function(e) {
            alert(e + "error")
        }
    })//ajax

```

```

})//ajaxBtn

```

```

$("#ajaxBtn_on").on("click", function() {
    $.ajax({
        type : "get",
        url : "process.jsp",
        data : {
            id : $("#id").val(),
            pwd : $("#pwd").val(),
            method : "ajax_on"
        },
        success : function(resp) {
            $("#result").html(resp);
        },
        error : function(e) {
            alert(e + "error")
        }
    })// ajax

```

```

})//ajaxBtn_on

```

```

}) // document

```

```

</script>

```

```

</head>

```

```

<body>

```

```

    id :

```

```

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

```

```

    <br /> pwd :

```

```

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

```

```

    <br />

```

```

<button type="button" id="getBtn">get전송</button>
<button type="button" id="postBtn">post전송</button>
<button type="button" id="ajaxBtn">ajax전송</button>
<button type="button" id="ajaxBtn_on">ajax_on전송</button>
<hr />

```

```

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

```

```

</body>
</html>

```

(process.jsp 생성)

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
request.setCharacterEncoding("utf-8");
String id = request.getParameter("id");
String pwd = request.getParameter("pwd");
String method = request.getParameter("method");
String str = "[처리결과]<br/>";
str += "id : " + id + "<br/>";
str += "pwd : " + pwd + "<br/>";
str += "method : " + method + "<br/>";
out.println(str);
%>

```

val() : form tag의 값을 가져올 때 input type  
html() : tag 처리  
text() : 일반적인 문자열 처리

<JSON의 형태>

```

{ { "memberNumber" : "m001", "irum" : "goroman" "picture" : "goroman.png" }, {
"memberNumber" : "m002", "irum" : "goroman2" "picture" : "goroman2.png" }, {
"memberNumber" : "m003", "irum" : "goroman3" "picture" : "goroman3.png" }, {
"memberNumber" : "m004", "irum" : "goroman4" "picture" : "goroman4.png" } }

```

Parsing을 하기 위해서는 Parser가 필요함

자세한 것은 JSP프로젝트 파일 참고