



Day13; 20220921

날짜	@2022년 9월 21일
유형	@2022년 9월 21일
태그	

spring/src at main · u8yes/spring

Contribute to u8yes/spring development by creating an account on GitHub.

u8yes/spring



<https://github.com/u8yes/spring/tree/main/src>

Contributor 0 Issues 1 Star 0 Forks

BoardDAO.java

```
BoardDAO.java x root-context.xml getBoardList.jsp index.jsp board.sql sqlMapConfig.xml boardMapping.xml
1 package com.springproj.biz.dao;
2
3 import java.util.List;
4
11
12 @Repository("boardDAO")
13 public class BoardDAO extends SqlSessionDaoSupport { // DAO(Data Access Object) 마이바티스에게 전달해줌.
14
15     @Autowired
16     public void setSqlSessionFactory(SqlSessionFactory sqlSessionFactory) { // 부모에게 객체를 전달받음
17         super.setSqlSessionFactory(sqlSessionFactory); // 다시 부모에게 변수를 보냄.
18     }
19
20
21     // CRUD 기능의 메서드 구현
22     // 글 등록(insert문)
23     public void insertBoard(BoardVO vo) { // DTO(DO), 매개변수를 파라미터라고도 한다. 전달되어지는 타입 (BoardVO vo)
24
25         getSqlSession().insert("Board.insertBoard", vo); // get메서드로 요청하면 마이바티스가 반환해준다.
26     }
27
28     // 목록보기(select문)
```

sqlMapConfig.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE configuration
3 PUBLIC "-//mybatis.org/DTD Config 3.0//EN"
4 "https://mybatis.org/dtd/mybatis-3-config.dtd">
5
6 <configuration>
7
8     <!-- Alias(별명, 가명) 설정 -->
9     <typeAliases>
10         <typeAlias alias="board" type="com.springproj.biz.domain.BoardVO"/> <!-- 만들 때 나타나는 속성(type)은 무조건 써줘야 하는 속성 -->
11     </typeAliases>
12
13     <!-- SQL Mapper 설정 -->
14     <mappers>
15         <mapper resource="mappings/boardMapping.xml"/>
16     </mappers>
17
18 </configuration>

```

boardMapping.xml

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3 PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
4 "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
5 <!-- 쿼리가 담기는 설정 파일이다 -->
6
7 <mapper namespace="Board">
8     <insert id="insertBoard" parameterType="board"> <!-- DAO에서 BoardVO vo(파라미터)에 담아져있기에 parameterType="board"생략가능함 -->
9         <![CDATA[
10             insert into board(seq, title, writer, content)
11             values ((select nvl(max(seq),0)+1 from board), #{title}, #{writer}, #{content})
12         ]]>
13     </insert>
14     <!-- '<', '>'는 비교 연산자로 인식하지 않고 여기에서는 객체로 본다. 설정을 <![CDATA[]>로 --> <!-- // SQL문 - nvl: Null값이면 0으로 실행해라
15     <!-- ${}는 변수를 바로 가져온다. #{}은 실행되어지는 시점에 변수를 가져온다. 둘이 같다고 생각해도 상관없다. -->
16
17     <select id="getBoardList_T" resultType="board"> <!-- resultType="board" 반환받는 타입을 명시하고 있어야 한다. 짧게 별명alias로 board로 -->
18     <![CDATA[
19         select * from BOARD where title like '%||#{searchKeyword}||%' order by seq desc
20     ]]>
21 </select>
22
23
24 <select id="getBoardList_C" resultType="board">

```

새로운 Mybatis로 설정하면 지워도 됨.

```

<!-- Spring과 Mybatis 연동 설정 방법 (1) -->
<bean id="sqlSession" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource"></property> <!-- name="dataSource" Set호출하면서 ref="dataSource"자료를 전달한다 -->
    <property name="configLocation" value="classpath:sqlMapConfig.xml" />
    <!-- Mybatis설정에 대한 위치정보를 value="classpath:sqlMapConfig.xml"라고 알려주는 것 -->
</bean>

<!-- Spring과 Mybatis 연동 설정 방법 (2) -->
<bean class="org.mybatis.spring.SqlSessionTemplate">
    <constructor-arg ref="sqlSession"/> <!-- constructor-arg = 생성자 호출// 객체 생성은 ref="" -->
</bean>

```

```

// 목록보기(select문)
public BoardVO getBoard(int seq) {
    return getSession().selectOne("Board.getBoard", seq);
}

public List<BoardVO> getBoardList(BoardVO vo) {
    System.out.println("DAO getBoardList");

    if(vo.getSearchCondition().equals("TITLE")) {
        return getSession().selectList("Board.getBoardList_T", vo); |
        // TITLE이 맞으면 실행. 요구한 것을 vo에 담아서 SQL로 넘겨줌. selectList로 전부 리턴해줌.
    }else if(vo.getSearchCondition().equals("CONTENT")) {
        return getSession().selectList("Board.getBoardList_C", vo);
    }else {
        return getSession().selectList("Board.getBoardList_T", vo);
    }
}
}

```

```

public List<BoardVO> getBoardList(BoardVO vo) {

    /*
    * if(vo.getSearchCondition().equals("TITLE")) { return
    * mybatis.selectList("Board.getBoardList_T", vo); // TITLE이 맞으면 실행. 요구한 것을 vo에
    * 담아서 SQL로 넘겨줌. selectList로 전부 리턴해줌. }else
    * if(vo.getSearchCondition().equals("CONTENT")) { return
    * mybatis.selectList("Board.getBoardList_C", vo); }else { return
    * mybatis.selectList("Board.getBoardList_T", vo); }
    */

    return mybatis.selectList("Board.getBoardList", vo);
}

```

BoardDAOMybatis.java

```

package com.springproj.biz.dao;

import java.util.List;

import org.mybatis.spring.SqlSessionTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

import com.springproj.biz.domain.BoardVO;

@Repository
public class BoardDAOMybatis {

    @Autowired
    SqlSessionTemplate mybatis;

    // CRUD 기능의 메서드 구현
    // 글 등록(insert문)
    public void insertBoard(BoardVO vo) { // DTO(DO), 매개변수를 파라미터라고도 한다. 전달되어지는 타입 (BoardVO vo)

        mybatis.insert("Board.insertBoard", vo); // get메서드로 요청하면 마이바티스가 반환해준다.
    }

    // 목록보기(select문)
    public BoardVO getBoard(int seq) {
        return mybatis.selectOne("Board.getBoard", seq);
    }

    public List<BoardVO> getBoardList(BoardVO vo) {

        /*
        * if(vo.getSearchCondition().equals("TITLE")) { return
        * mybatis.selectList("Board.getBoardList_T", vo); // TITLE이 맞으면 실행. 요구한 것을 vo에
        * 담아서 SQL로 넘겨줌. selectList로 전부 리턴해줌. }else
        * if(vo.getSearchCondition().equals("CONTENT")) { return

```

```

        * mybatis.selectList("Board.getBoardList_C", vo); }else { return
        * mybatis.selectList("Board.getBoardList_T", vo); }
        */

return mybatis.selectList("Board.getBoardList", vo);
}

// 글 수정(update문)
public void updateBoard(BoardVO vo) { // DTO(DO)
    mybatis.update("Board.updateBoard", vo);

}

// 글 삭제(delete문)
public void deleteBoard(int seq) {
    //System.out.println("BoardDAOMybatis.deleteBoard 실행" + seq);

    mybatis.delete("Board.deleteBoard", seq);

}
}
}

```

boardMapping.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<!-- 쿼리가 담기는 설정 파일이다 -->

<mapper namespace="Board">
    <insert id="insertBoard" parameterType="board"> <!-- DAO에서 BoardVO vo(파라미터)에 담겨있기에 parameterType="board"생략가능함 -->
    <![CDATA[
        insert into board(seq, title, writer, content)
        values ((select nvl(max(seq),0)+1 from board), #{title}, #{writer}, #{content})
    ]]>
    </insert>
    <!-- '<', '>'는 비교 연산자로 인식하지 않고 여기에서는 꺾쇠로 본다. 설정을 <![CDATA[]]>로 --> <!-- // SQL문 - nvl: Null값이면 0으로 실행해라 -->
    <!-- ${}는 변수를 바로 가져온다. #{}은 실행되어지는 시점에 변수를 가져온다. 둘이 같다고 생각해도 상관없다. -->

    <!-- <select id="getBoardList_T" resultType="board"> resultType="board" 반환받는 타입을 명시하고 있어야 한다. 짧게 별명alias로 board로
    <![CDATA[
        select * from BOARD where title like '%'||#{searchKeyword}||'%' order by seq desc
    ]]>
    </select>

    <select id="getBoardList_C" resultType="board">
    <![CDATA[
        select * from BOARD where content like '%'||#{searchKeyword}||'%' order by seq desc
    ]]>
    </select> -->

    <!-- Mybatis가 제공하는 동적(Dynamic) SQL 쿼리문을 추가적으로 제공해줌. Spring(X) -->
    <!-- resultType="board" 반환받는 타입을 명시하고 있어야 한다. 짧게 별명alias로 board로 -->
    <select id="getBoardList" resultType="board">
    select *
    from BOARD
    where 1 = 1
    <if test="searchCondition == 'TITLE'">
        and title like '%'||#{searchKeyword}||'%'
    </if>
    <if test="searchCondition == 'CONTENT'">
        and content like '%'||#{searchKeyword}||'%'
    </if>
    order by seq desc
    </select>

    <!-- List 보여주기 -->
    <select id="getBoard" resultType="board">
    <![CDATA[
        select * from BOARD where seq = #{seq}
    ]]>
    </select>

    <update id="updateBoard">
    <![CDATA[

```

```

        update board set title = #{title}, content = #{content} where seq = #{seq}
    ]]>
</update>

<delete id="deleteBoard">
<![CDATA[
    delete from board where seq = #{seq}
]]>
</delete>

</mapper>

```

sqlMapConfig.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE configuration
PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">

<configuration>

    <!-- Alias(별명, 가명) 설정 -->
    <typeAliases>
        <typeAlias alias="board" type="com.springproj.biz.domain.BoardVO"/>    <!-- 만들 때 나타나는 속성(type)은 무조건 써줘야 하는 속성 -->
    </typeAliases>

    <!-- SQL Mapper 설정 -->
    <mapper>
        <mapper resource="mappings/boardMapping.xml"/>
    </mapper>

</configuration>

```

두 개의 쿼리를 하나로 모아줌

```

<!-- Mybatis가 제공하는 동적(Dynamic) SQL 쿼리문을 추가적으로 제공해줌. Spring(X) -->
<select id="getBoardList" resultType="board"> resultType="board" 반환받는 타입을 명시하고 있어야 한다. 짧게 별명alias로 board로

    select *
    from BOARD
    where 1 = 1
    <if test="searchCondition == TITLE">
        and title like '%'||#{searchKeyword}||'%'
    </if>
    <if test="searchCondition == CONTENT">
        and content like '%'||#{searchKeyword}||'%'
    </if>
    order by seq desc

</select>

```

오늘의 실수(BoardServiceImpl에서 BoardDAOMybatis로 보내준 vo와 seq를 작성 안해주고 있었음. 그걸 Mybatis로 보내줘야 하는데 중간에 안 써주면 당연히 DAO에서 Mybatis로 보낼 자료가 없을수밖에....

```

2
3
4 // 글 수정(update문)
5 public void updateBoard(BoardVO vo) { // DTO(DO)
6     mybatis.update("Board.updateBoard", vo);
7
8
9 }
0
1 // 글 삭제(delete문)
2 public void deleteBoard(int seq) {
3     //System.out.println("BoardDAOmybatis.deleteBoard 실행" + seq);
4
5     mybatis.delete("Board.deleteBoard", seq);
6
7 }
8 }
9

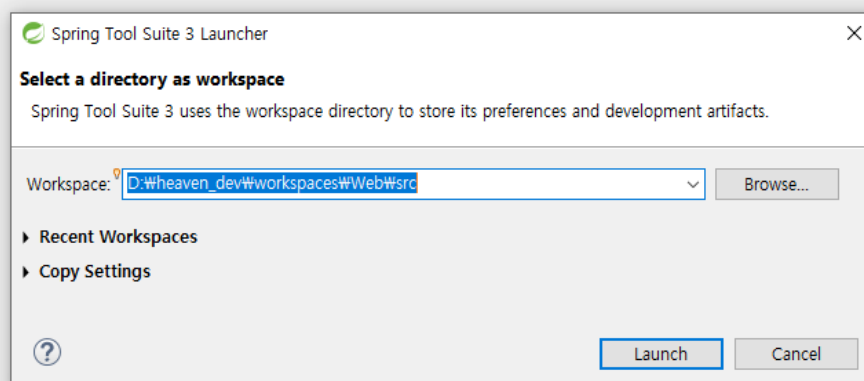
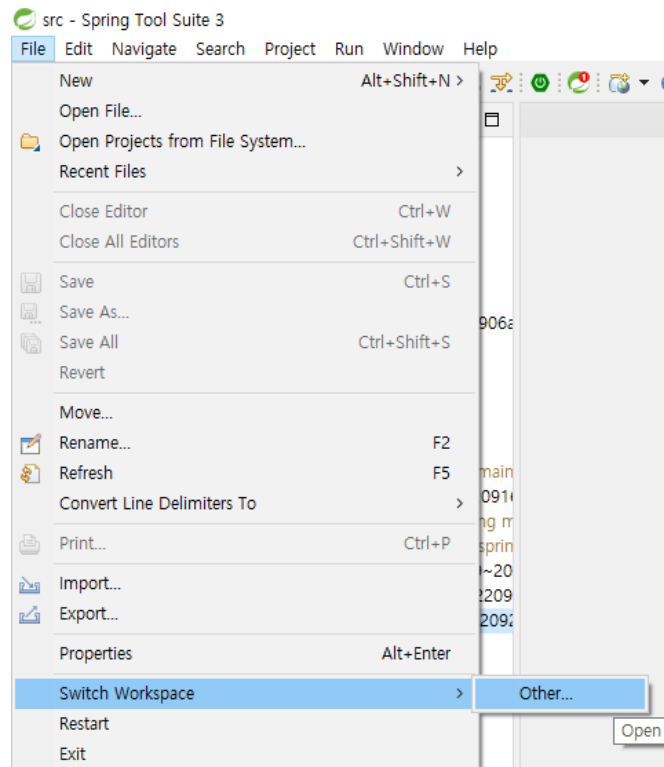
```

FLASH는 이제 안 받아줌

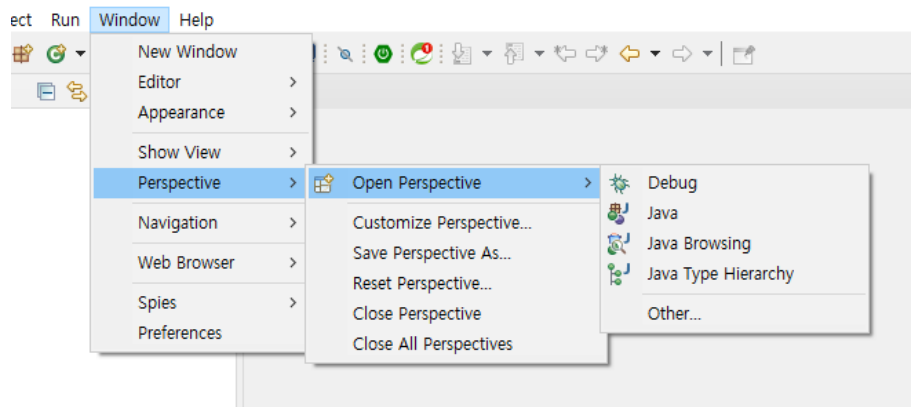
아이작스 기술을 가지고 네이버 로그인 부분만 실시간 업데이트 가능하게 해줌.

로그인 할 때 전체를 업데이트하면 느려짐.

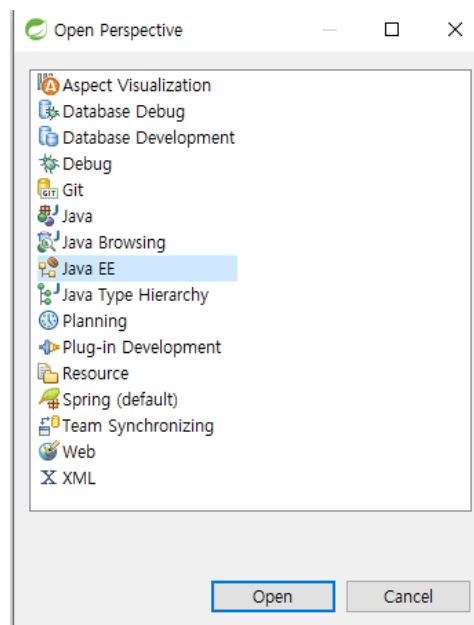
JQuery.



첫번째 방법 Java EE 클릭하기.

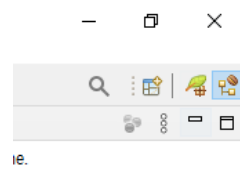


두번째 방법 Java EE 클릭

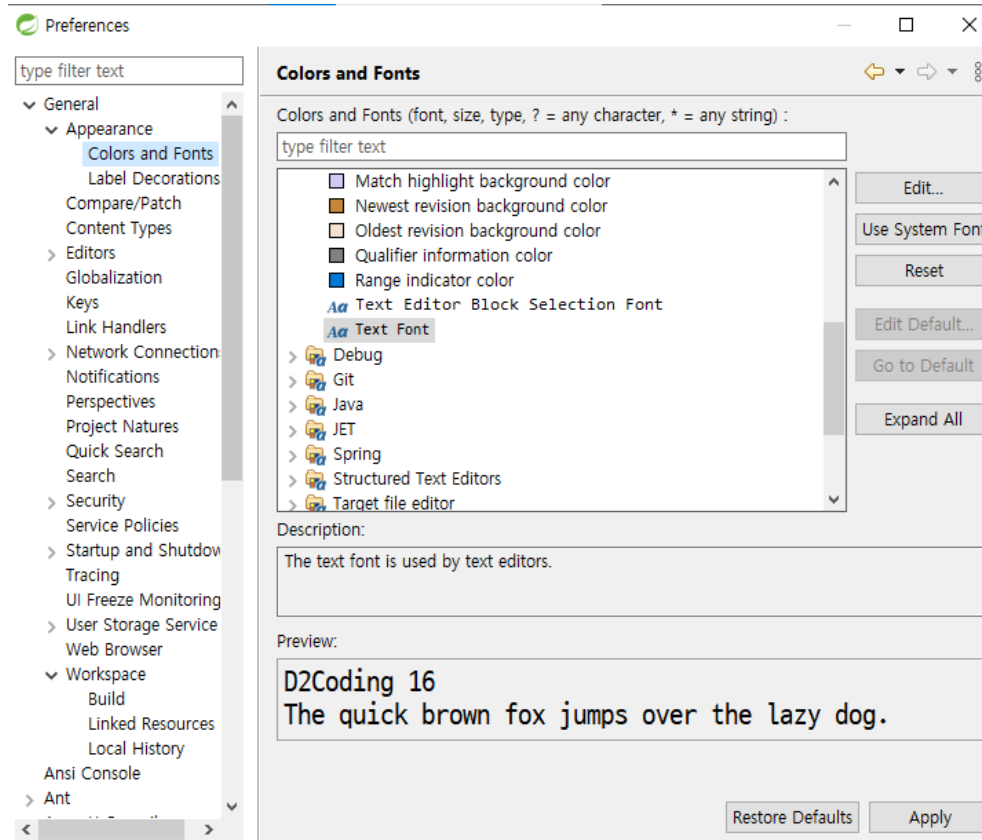


최적의 환경을 선택할 수 있음

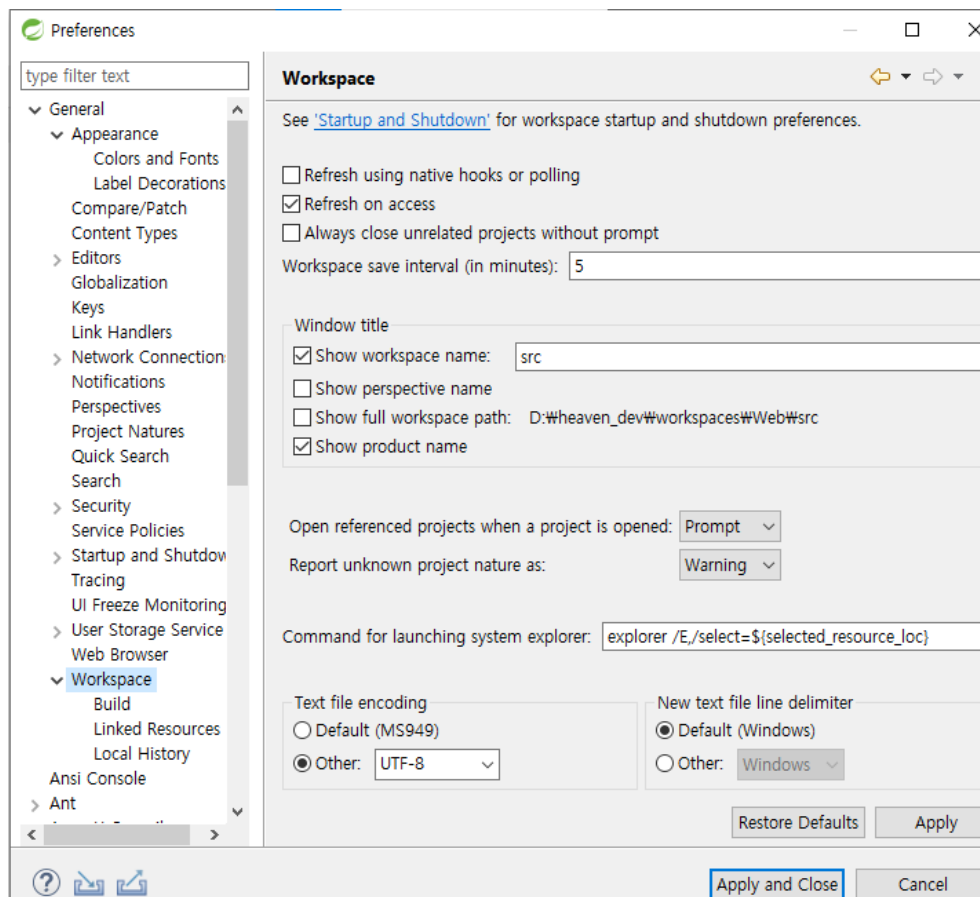
Spring or Java EE



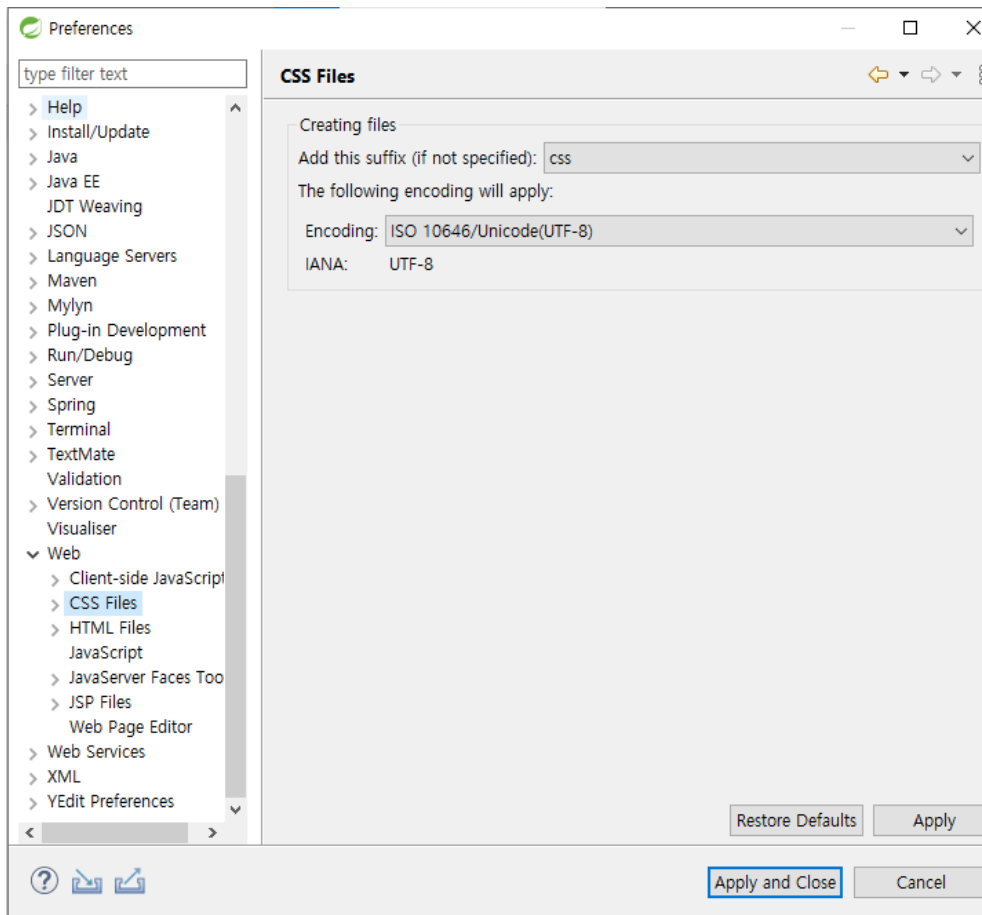
설정 시작



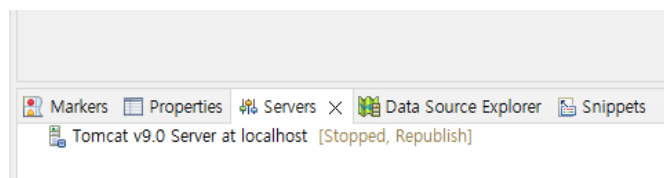
Workspace에서 한글로 바뀜(UTF-8)



CSS, HTML, JSP 설정을 한글로 UTF-8



tomcat9 설정해줌.



포트번호를 8080을 오라클이 쓰고 있기 때문에 8090으로 변경

Tomcat v9.0 Server at localhost X

Overview

General Information

Specify the host name and other common settings.

Server name:

Host name:

Runtime Environment:

Configuration path:

[Open launch configuration](#)

Server Locations

Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.

☒ Use workspace metadata (does not modify Tomcat installation)

☐ Use Tomcat installation (takes control of Tomcat installation)

☐ Use custom location (does not modify Tomcat installation)

Server path:

[Set deploy path to the default value \(currently set\)](#)

Deploy path:

Server Options

Enter settings for the server.

☒ Serve modules without publishing

☐ Publish module contexts to separate XML files

☒ Modules auto reload by default

☐ Enable security

☐ Enable Tomcat debug logging (not supported by this Tomcat version)

Publishing

Timeouts

Ports

Modify the server ports.

Port Name	Port Number
Tomcat admin port	8005
HTTP/1.1	8090

MIME Mappings

절대 경로로 잡아줌.

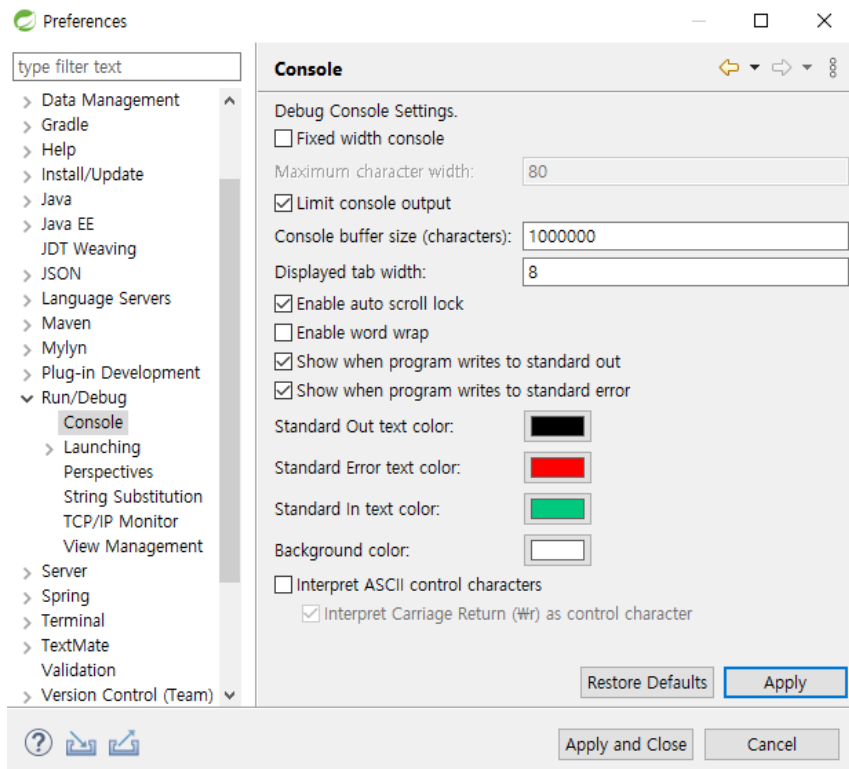
Server Options

Enter settings for the server.

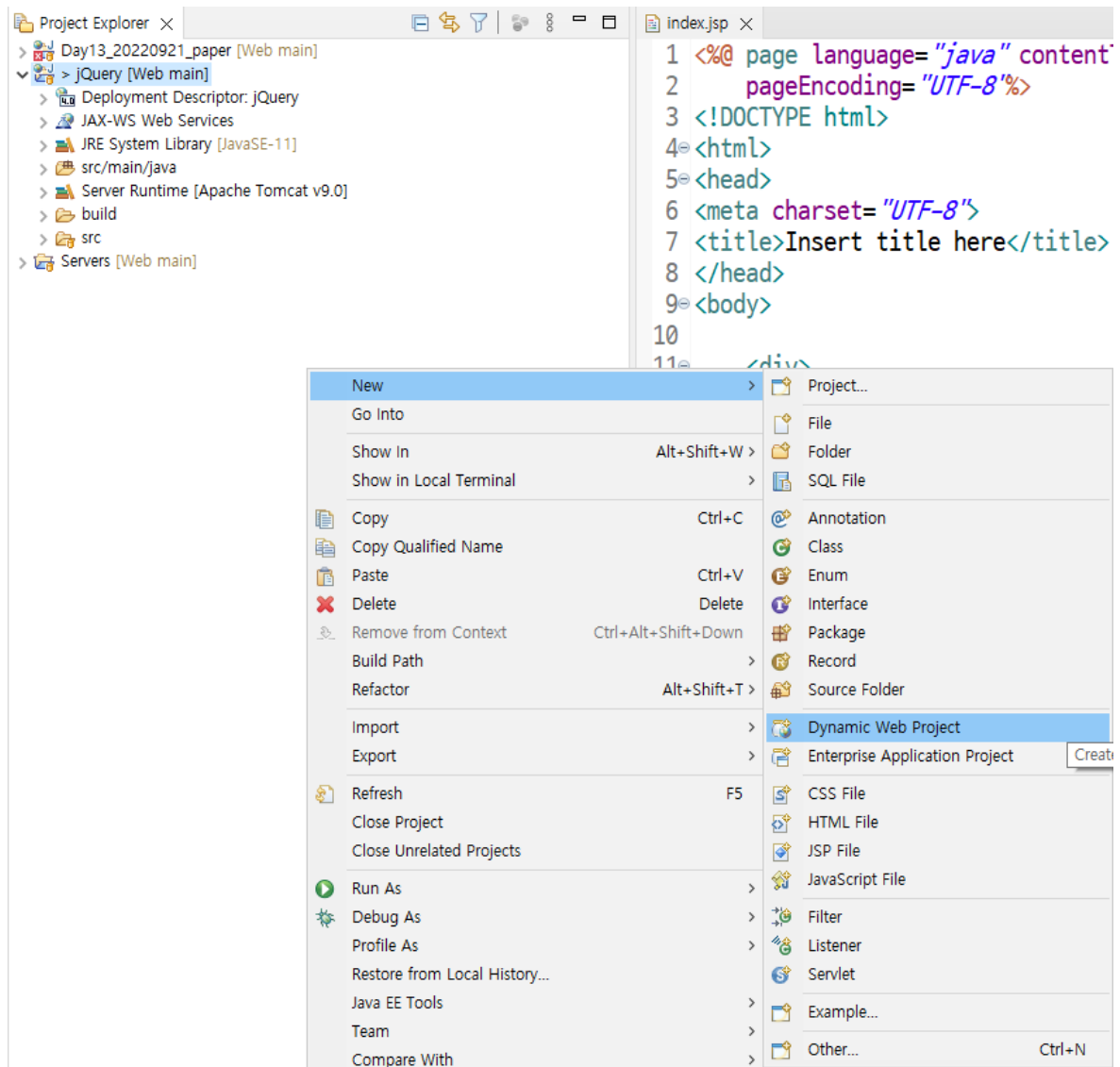
☒ Serve modules without publishing

☐ Publish module contexts to separate XML files

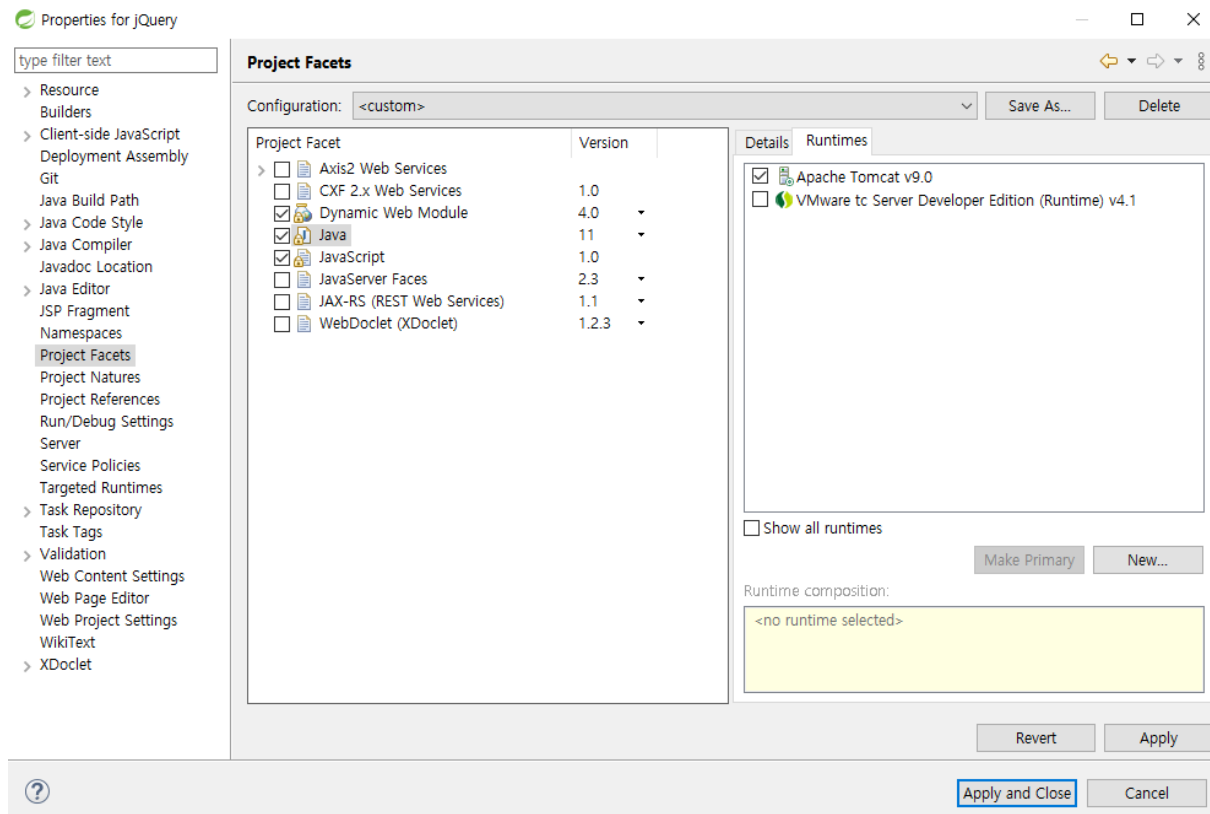
tomcat 실행할 때 자꾸 창이 뜨니깐 없어지게 하려고 Preference 들어가서 1000000으로 셋팅해줌.



새 프로젝트 만들자.



JAVA 버전을 11로



https://s3-us-west-2.amazonaws.com/secure.notion-static.com/2961444c-37b3-4f4c-997a-083bdf0103c2/01_jQuery.Basic.html

```

<!-- jQuery 기본
1. jQuery 개요
- 모든 브라우저에서 동작하는 클라이언트 자바스크립트 라이브러리.
- 다음의 기능을 위해 제작
  . DOM(Document Object Model)과 관련된 처리를 쉽게 구현
  . 일관된 이벤트 연결을 쉽게 구현
  . 시각적 효과를 쉽게 구현
  . ajax 애플리케이션을 쉽게 구현

- https://jquery.com
- jQuery 사용 방법
  1) 직접 다운로드 사용하는 방법.
  2) CDN 호스트 사용하는 방법.
    . Content Delivery Network
    . 사용자에게 간편하게 콘텐츠를 제공하는 방식을 의미.
    . 구글, 마이크로소프트, jQuery측에서 사용자가 jQuery를 사용하기 편하게 콘텐츠를 제공.

-->

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>jQuery 태그 구조</title>
</head>
<body>

<!-- 3-1/3-2. 전체/태그 선택자
<h1>Lorem ipsum</h1>
<p>Lorem ipsum dolor sit amet.</p>
<h1>Lorem ipsum</h1>
<p>Lorem ipsum dolor sit amet.</p>
<hr>

```

```

-->

<!-- 3-3. 아이디 선택자
<h1>Header-0</h1>
  <h1 id="target">Header-1</h1>
<h1>Header-2</h1>
<hr>
-->

<!-- 3-4. 클래스 선택자
<h1 class="item">Header-0</h1>
<h1 class="item select">Header-1</h1>
<h1 class="item">Header-2</h1>
<hr>
-->

<!-- 4. 자식(손) 선택자와 후손 선택자
<div>
  <ul>
    <li>Apple</li>
    <li>Bag</li>
    <li>Cat</li>
    <li>Dog</li>
  </ul>
</div>
<hr>
-->

<!-- 5. 속성 선택자
<input type="text"/></br>
<input type="password"/></br>
<input type="radio"/></br>
<input type="checkbox"/></br>
<input type="file"/></br>
<hr>
-->

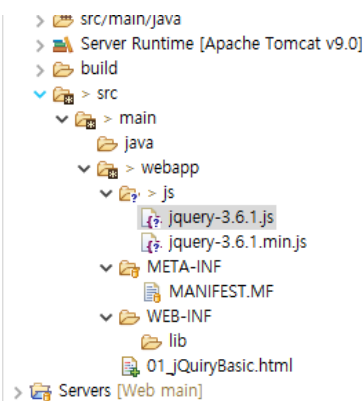
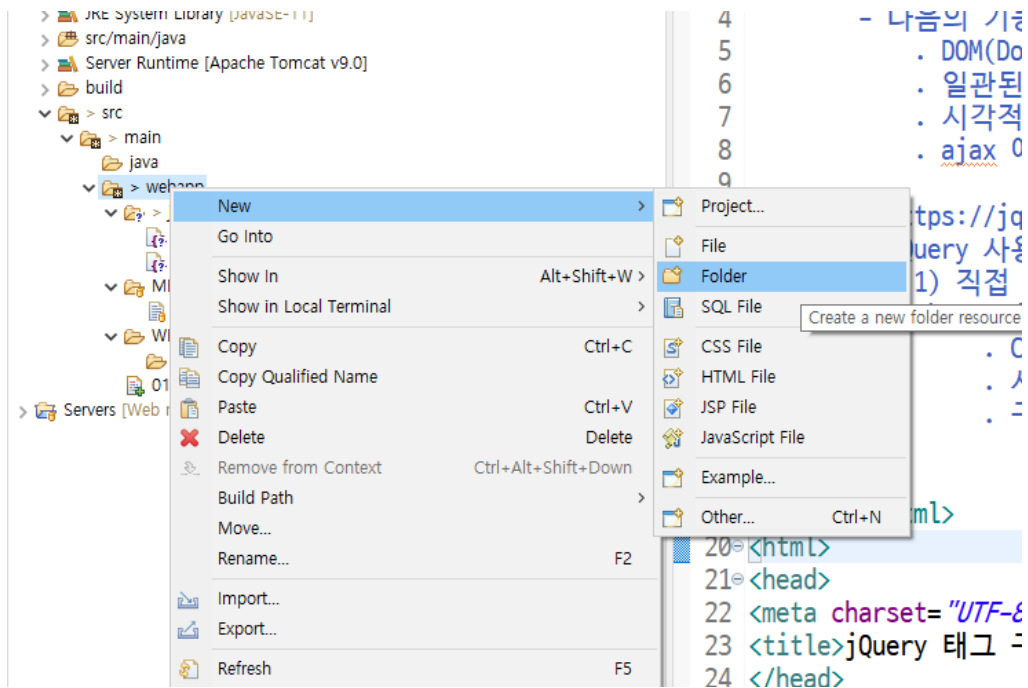
<!-- 6-1. 입력 양식 필터 선택자
<select>
  <option>Apple</option>
  <option>Bag</option>
  <option>Cat</option>
  <option>Dog</option>
</select>
<hr>
-->

<!-- 6-2/6-3. 필터선택자
<table border="1">
  <tr><th>이름</th><th>혈액형</th><th>지역</th></tr>
  <tr><td>홍길동</td><td>AB형</td><td>서울시 강남구</td></tr>
  <tr><td>홍길서</td><td>A형</td><td>서울시 마포구</td></tr>
  <tr><td>홍길남</td><td>B형</td><td>서울시 서초구</td></tr>
  <tr><td>홍길북</td><td>O형</td><td>서울시 송파구</td></tr>
  <tr><td>김길동</td><td>O형</td><td>서울시 영등포구</td></tr>
  <tr><td>이길동</td><td>A형</td><td>서울시 강서구</td></tr>
  <tr><td>박길동</td><td>B형</td><td>서울시 용산구</td></tr>
</table>
<hr>
-->

<!-- 8. jQuery를 사용한 배열 관리(2)
<h1>item - 0</h1>
<h1>item - 1</h1>
<h1>item - 2</h1>
<h1>item - 3</h1>
<h1>item - 4</h1>
-->

</body>
</html>

```

js에 아래 파일 압축폴고 난 후 붙여넣기

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0abbf312-c909-4dfe-b2f2-5db7c5dd72f1/jquery-3.6.1.zip>

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/3220d933-7555-45c6-9074-f67474b43507/jquery-3.6.1.min.zip>

01_jQueryBasic.html

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1162a3ce-d44b-438b-96d3-09b8e9ac6e7f/01_jQuery.Basic.html

```
<!-- jQuery 기본
1. jQuery 개요
- 모든 브라우저에서 동작하는 클라이언트 자바스크립트 라이브러리.
- 다음의 기능을 위해 제작
  . DOM(Document Object Model)과 관련된 처리를 쉽게 구현
  . 일관된 이벤트 연결을 쉽게 구현
  . 시각적 효과를 쉽게 구현
  . ajax 애플리케이션을 쉽게 구현

- https://jquery.com
- jQuery 사용 방법
  1) 직접 다운로드 사용하는 방법.
  2) CDN 호스트 사용하는 방법.
    . Content Delivery Network
    . 사용자에게 간편하게 콘텐츠를 제공하는 방식을 의미.
    . 구글, 마이크로소프트, jQuery측에서 사용자가 jQuery를 사용하기 편하게 콘텐츠를 제공.
-->

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>jQuery 태그 구조</title>

<style type="text/css">

</style>

<!-- 방법1. 직접 다운로드, src=""는 외부로 import해준다.
기능을 사용하려면 반드시 js 밑으로 실행할 함수, 예제들을 넣어야 한다. -->
<script src="./js/jquery-3.6.1.min.js"></script>

<!-- 방법2. CDN 방식, 네트워크로 직접 접근해서 다운로드 받음.(인터넷 연결 필요)
-->
<script src="https://code.jquery.com/jquery-3.6.1.min.js"></script>
-->
<script type="text/javascript">
/*
jQuery(document).ready(function(){
  alert("First Ready");
});
*/
$(function(){
/* 1. 라이브러리 포함 방법 : 2가지 방식
- 방법1, 방법2 참조
*/

/* 2. jQuery를 사용한 웹 페이지 형식 ($가 jQuery이다, document는 자료형
, ready는 </body>를 다 읽고 끝난 시점에 function(){css문법}실행하게끔 대기)

- $(document).ready(function(){});
  . 문서가 준비 완료되면 매개변수로 전달된 함수를 실행하라는 의미.
  . $(function(){}); // 간단한 형식 정의.(위낙 많이 사용해서)
  alert("First Ready");
*/

/* 3. jQuery 기본 선택자
var headers = document.getElementsByTagName("h1"); // var은 배열로 담아준다.

headers[0].innerHTML = '더조은';
headers[1].innerHTML = '컴퓨터 학원';

1) 전체 선택자(*)
$('*').css('color','red'); // key, value값

2) tag 선택자
- 한 개 태그 선택
$('p').css('color','blue');
- 두 개 태그 선택
$('h1, p').css('color','blue');

3) id 선택자 //
$('#target').css('color','blue');
```

```

4) class 선택자 //
$('.item').css('color','green');
$('h1.select').css('color','red');
$('.item.select').css('background','blue');
*/

/* 4. jQuery 자식(손) 선택자와 후손 선택자
- 자식(손) 선택자
*/

});

</script>
</head>
<body>

<!-- 3-1/3-2. 전체/태그 선택자
<h1>Lorem ipsum</h1>
<p>Lorem ipsum dolor sit amet.</p>
<h1>Lorem ipsum</h1>
<p>Lorem ipsum dolor sit amet.</p>
<hr>
-->

<!-- 3-3. 아이디 선택자
<h1>Header-0</h1>
<h1 id="target">Header-1</h1>
<h1>Header-2</h1>
<hr>
-->

<!-- 3-4. 클래스 선택자
<h1 class="item">Header-0</h1>
<h1 class="item select">Header-1</h1>
<h1 class="">Header-2</h1>
<h2 class="select">Header-2</h1>
<hr>
-->

<!-- 4. 자식(손) 선택자와 후손 선택자 -->
<div>
<ul>
<li>Apple</li>
<li>Bag</li>
<li>Cat</li>
<li>Dog</li>
</ul>
</div>
<hr>

<!-- 5. 속성 선택자
<input type="text"/><br>
<input type="password"/><br>
<input type="radio"/><br>
<input type="checkbox"/><br>
<input type="file"/><br>
<hr>
-->

<!-- 6-1. 입력 양식 필터 선택자
<select>
<option>Apple</option>
<option>Bag</option>
<option>Cat</option>
<option>Dog</option>
</select>
<hr>
-->

<!-- 6-2/6-3. 필터선택자
<table border="1">
<tr><th>이름</th><th>혈액형</th><th>지역</th></tr>
<tr><td>홍길동</td><td>AB형</td><td>서울시 강남구</td></tr>
<tr><td>홍길서</td><td>A형</td><td>서울시 마포구</td></tr>
<tr><td>홍길남</td><td>B형</td><td>서울시 서초구</td></tr>
<tr><td>홍길북</td><td>O형</td><td>서울시 송파구</td></tr>
<tr><td>김길동</td><td>O형</td><td>서울시 영등포구</td></tr>
<tr><td>이길동</td><td>A형</td><td>서울시 강서구</td></tr>
<tr><td>박길동</td><td>B형</td><td>서울시 용산구</td></tr>
</table>

```

```
<hr>
-->

<!-- 8. jQuery를 사용한 배열 관리(2)
  <h1>item - 0</h1>
  <h1>item - 1</h1>
  <h1>item - 2</h1>
  <h1>item - 3</h1>
  <h1>item - 4</h1>
-->

</body>
</html>
```
