

삼성 청년 SW 아카데미

Web-BackEnd

<알림>

본 강의는 삼성 청년 SW아카데미의 콘텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

Back-End (Servlet & JSP)

목차

1. [WebArchitecture](#)
2. [Servlet](#)
3. [Servlet Life-Cycle](#)
4. [Servlet Parameter 처리](#)
5. [JSP \(Java Server Page\)](#)
6. [JSP Scriptlet](#)
7. [JSP Directive](#)
8. [JSP 기본객체](#)

- 핵심내용 -

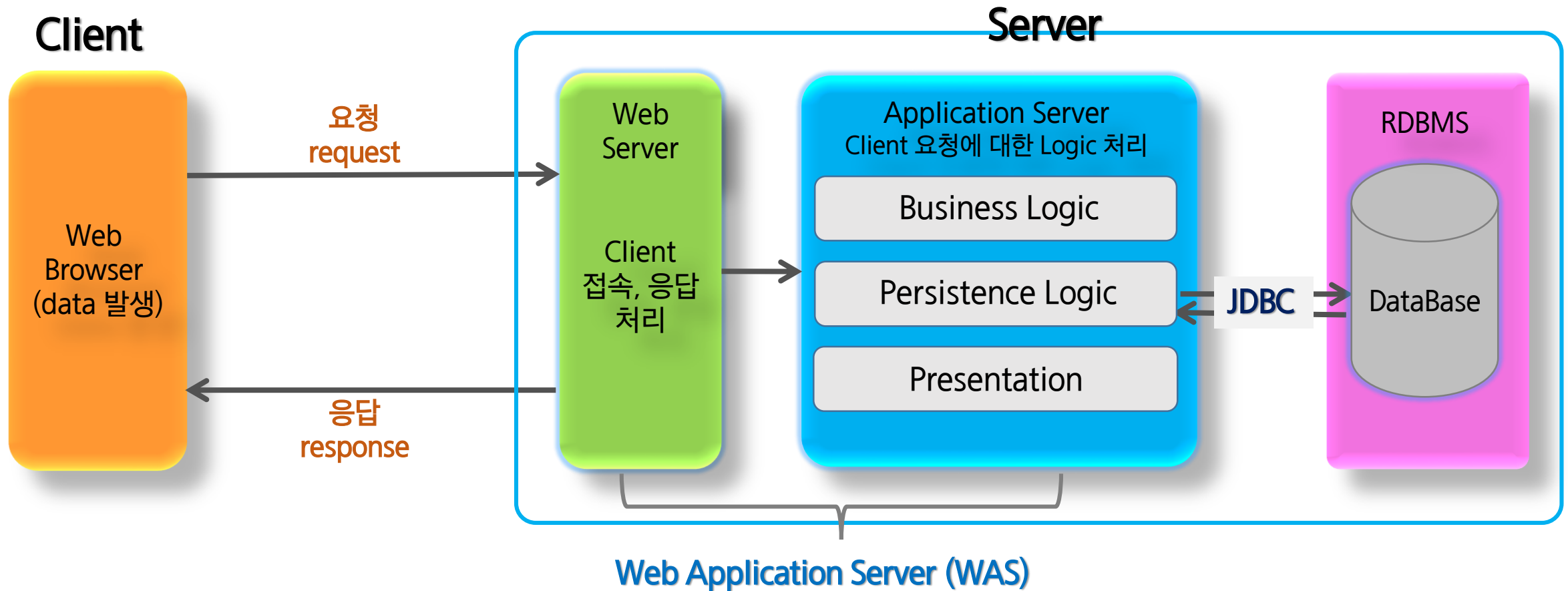
- Web Architecture
- Servlet?
- Servlet Life-Cycle
- Servlet에서의 parameter 처리

Web Architecture

삼성 청년 SW 아카데미

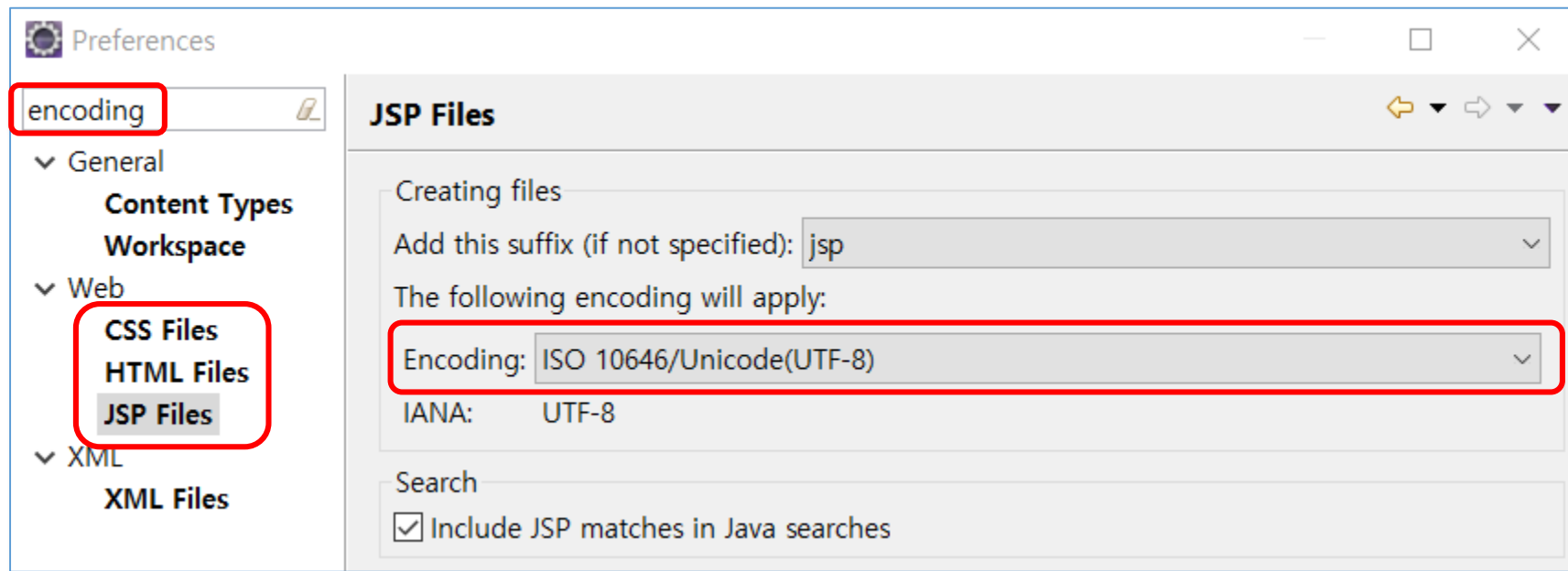
Web Architecture.

✓ Web Architecture

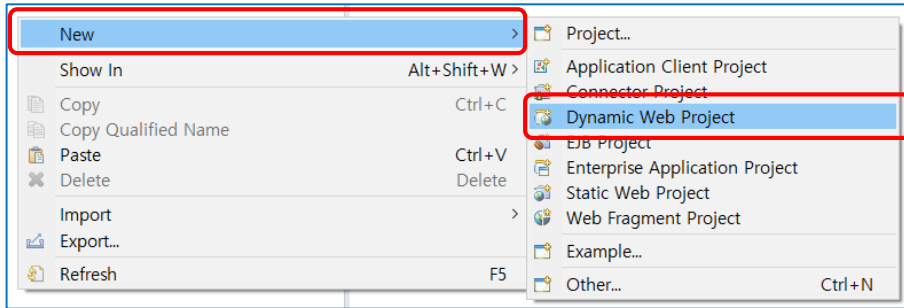


✓ eclipse encoding 설정

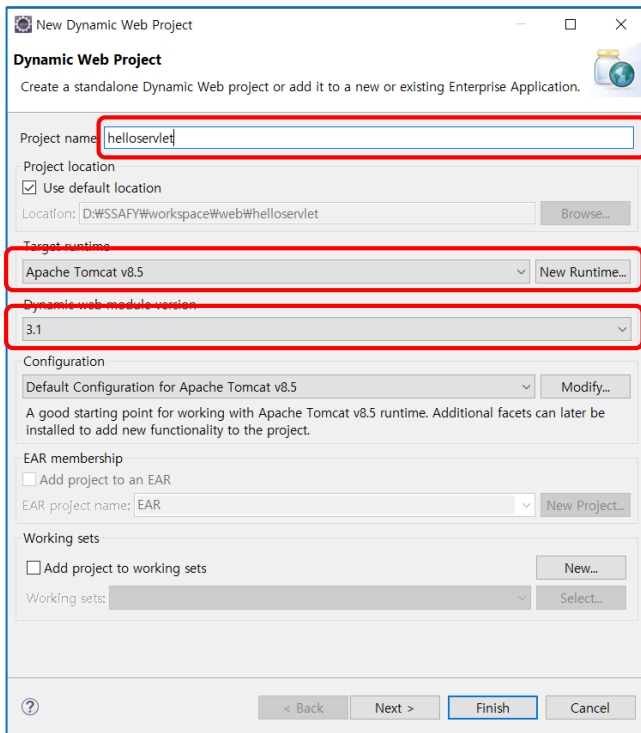
window메뉴의 preferences에서 encoding 검색 후 아래 그림과 같이
CSS Files, HTML Files, JSP Files의 Encoding을 UTF-8로 설정한다.



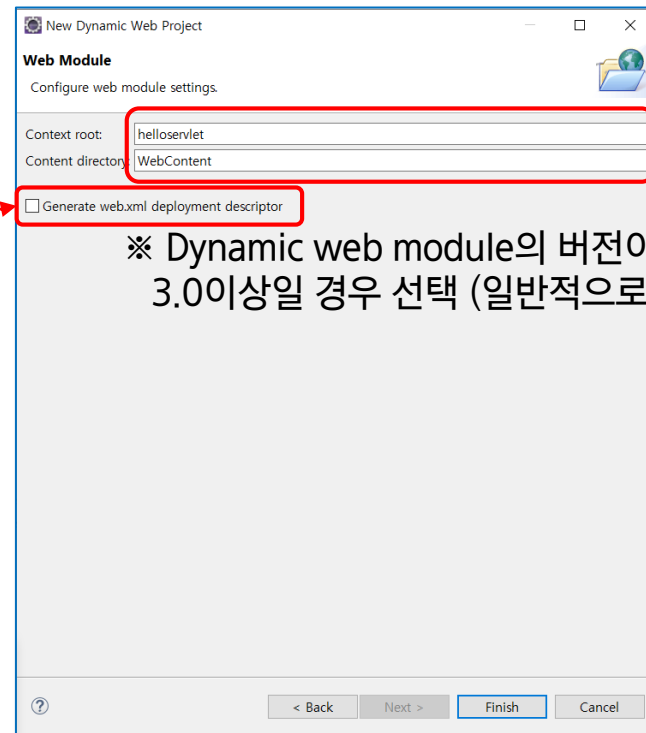
✓ helloworld project 생성



① New >> Dynamic Web Project 선택



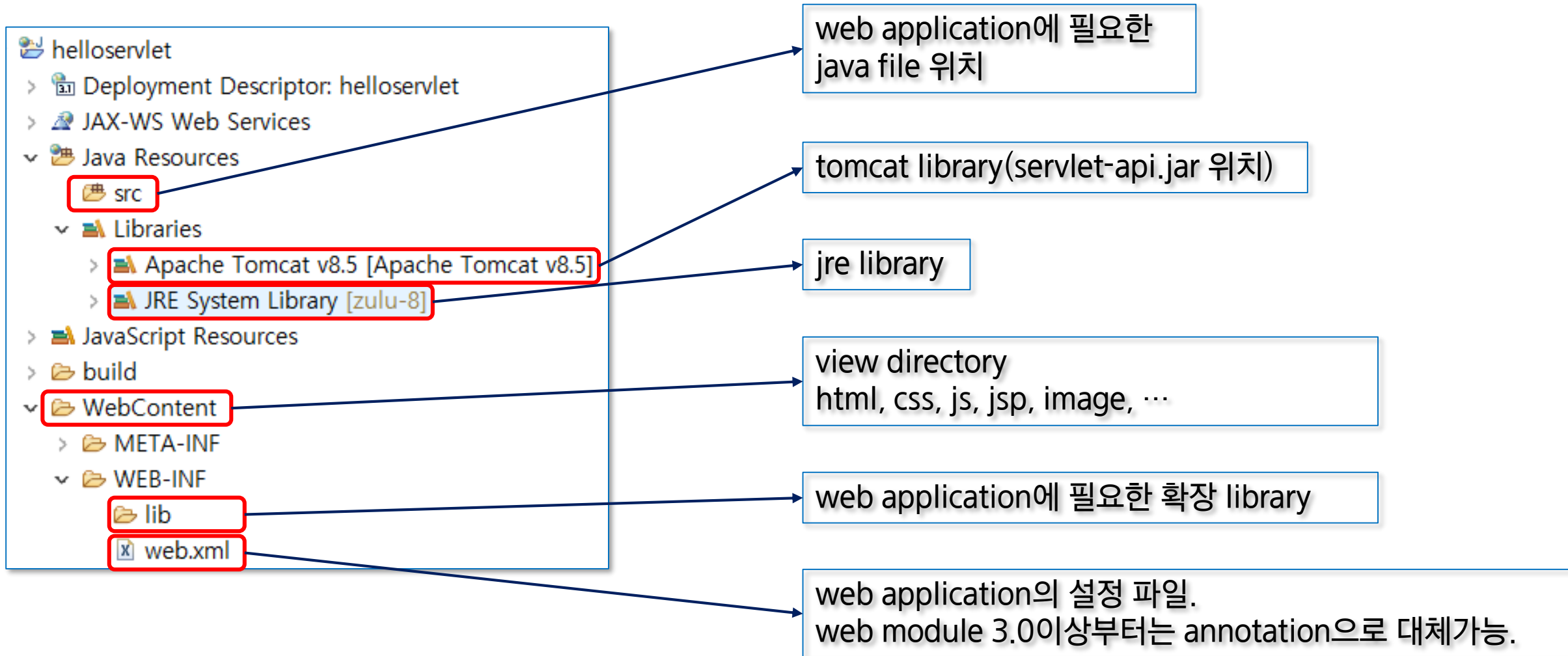
② project name 설정
target runtime 선택



③ context root와
content directory 확인
(변경 가능)

※ Dynamic web module의 버전이 3.0미만일 경우 반드시 체크
3.0이상일 경우 선택 (일반적으로 체크X, annotation사용)

✓ project 구조



HelloSsafety class

✓ HelloSsafety.java servlet class 생성.

Java Resources
src
com.ssafy.hello
HelloSsafety.java

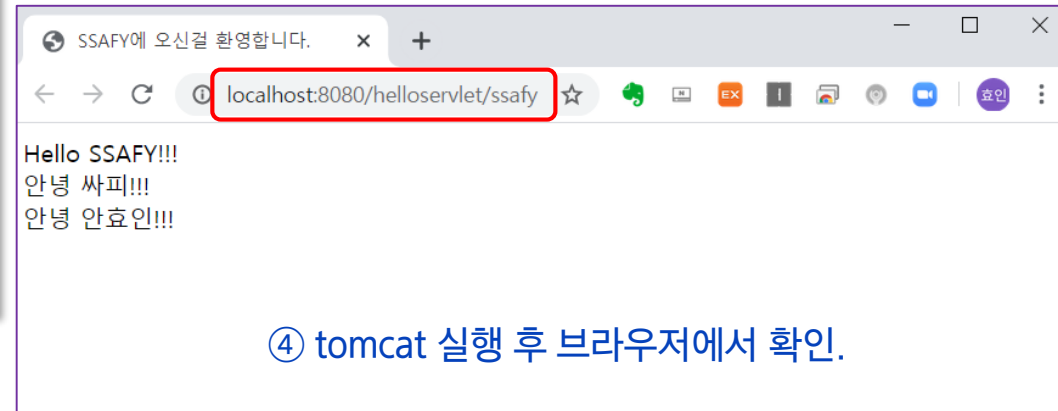
① servlet 파일 만들기.

```
String name = "안효인";  
//response.setContentType("text/html");  
//response.setCharacterEncoding("UTF-8");  
response.setContentType("text/html;charset=UTF-8");  
PrintWriter out = response.getWriter();  
out.println("<html>");  
out.println("<head>");  
out.println("    <title>SSAFY에 오신걸 환영합니다.</title>");  
out.println("</head>");  
out.println("<body>");  
out.println("Hello SSAFY!!!<br>");  
out.println("안녕 싸피!!!<br>");  
out.println("안녕 " + name + "!!!");  
out.println("</body>");  
out.println("</html>");
```

③ service method 구현.

```
package com.ssafy.hello;  
  
import java.io.IOException;  
import javax.servlet.ServletException;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
  
public class HelloSsafety extends HttpServlet {  
    private static final long serialVersionUID = 1L;  
  
    protected void doGet(HttpServletRequest request, HttpServletResponse response)  
        throws ServletException, IOException {  
  
    }  
}
```

② file 정리.



④ tomcat 실행 후 브라우저에서 확인.

이어서..

삼성 청년 SW 아카데미

Servlet?

삼성 청년 SW 아카데미

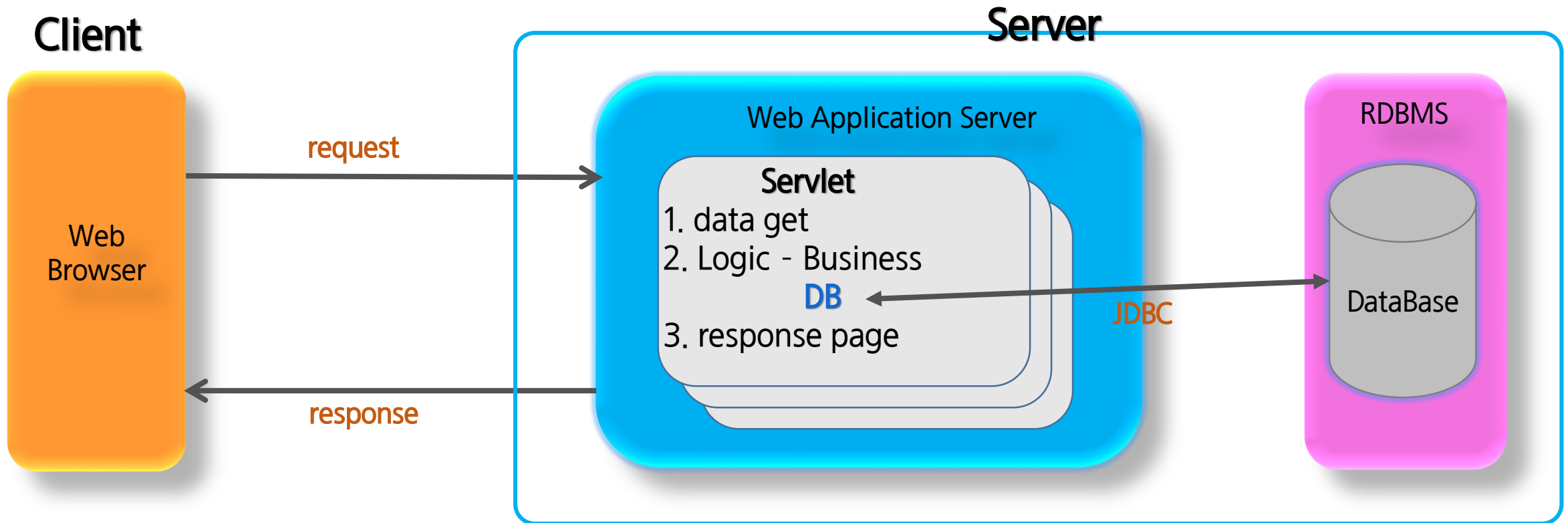
✓ Servlet?

자바 서블릿(Java Servlet)은 [자바](#)를 사용하여 웹페이지를 동적으로 생성하는 서버측 프로그램 혹은 그 사양을 말하며, 흔히 "서블릿"이라 불린다. 자바 서블릿은 웹 서버의 성능을 향상하기 위해 사용되는 자바 클래스의 일종이다. 서블릿은 JSP와 비슷한 점이 있지만, [JSP](#)가 HTML 문서 안에 Java 코드를 포함하고 있는 반면, 서블릿은 자바 코드 안에 HTML을 포함하고 있다는 차이점이 있다.

[출처 : 위키백과]

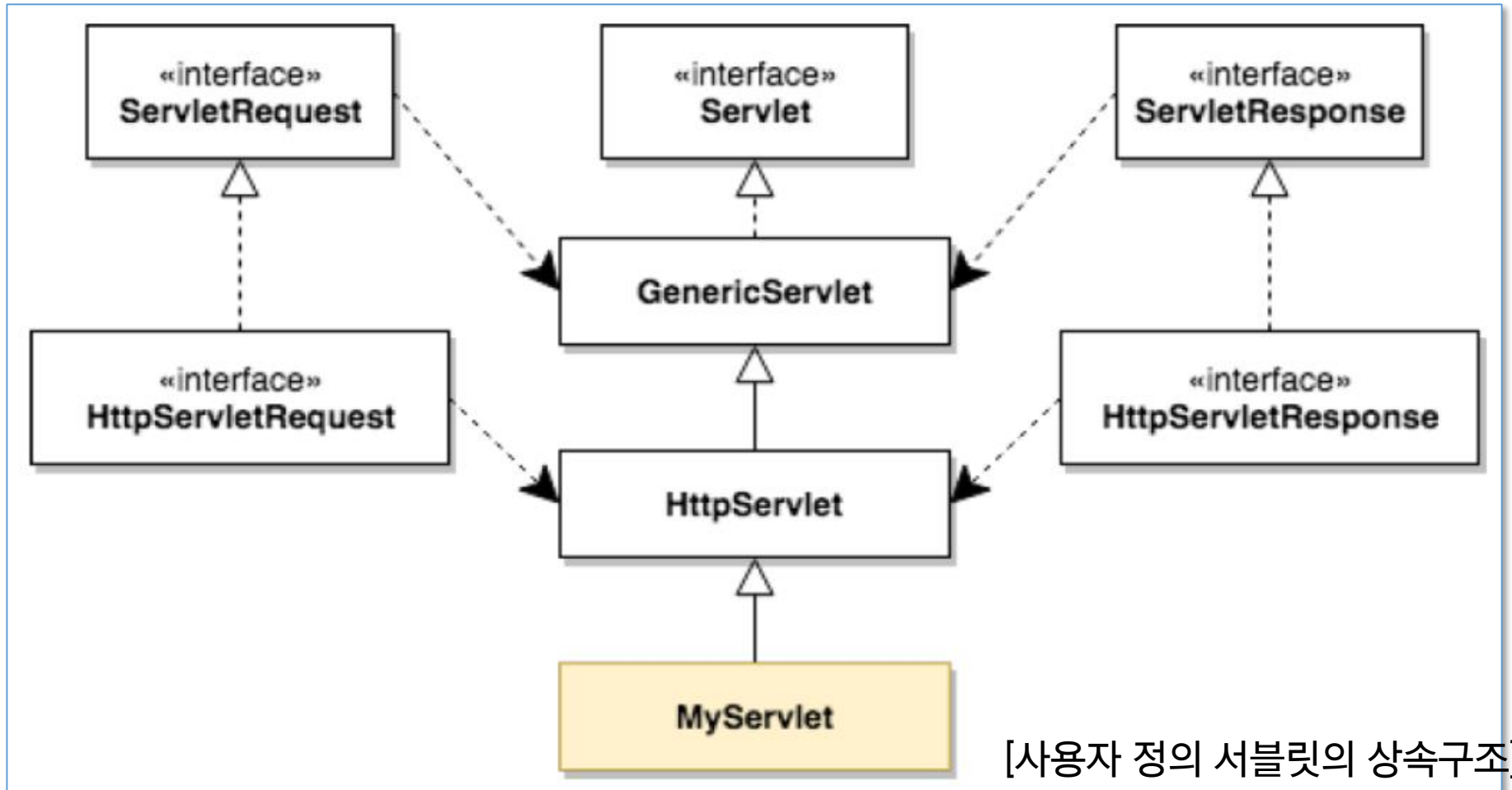
Servlet Process

✓ Servlet 동작 흐름



✓ Servlet API

[Servlet API Document](#)



이어서..

삼성 청년 SW 아카데미

Servlet Life-cycle

삼성 청년 SW 아카데미

Servlet's Life-Cycle

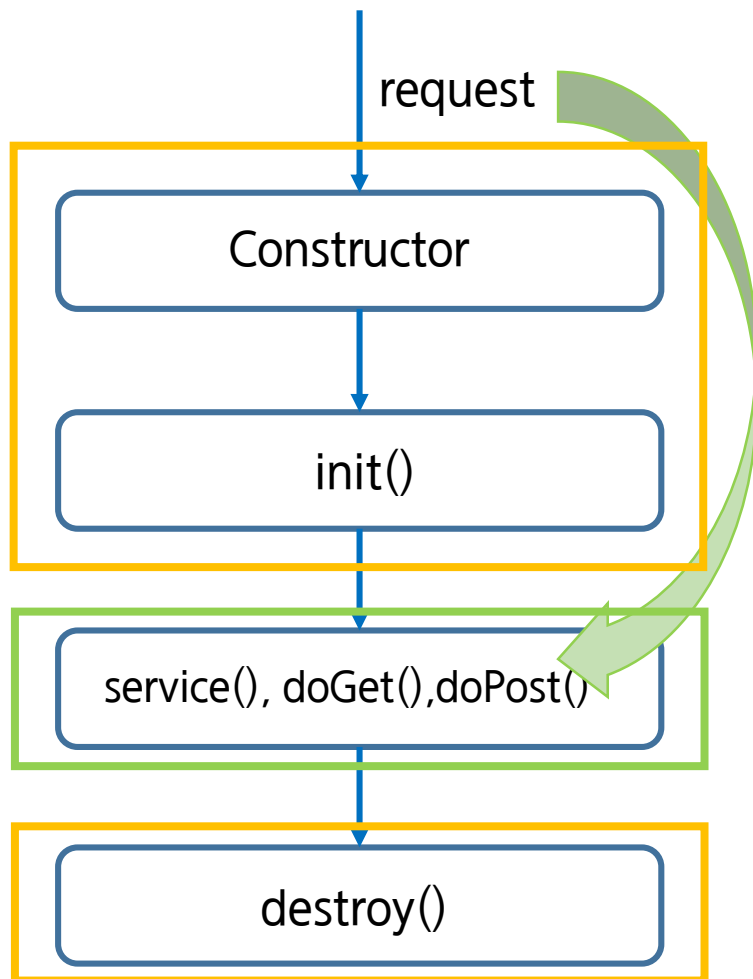
✓ Servlet Life-Cycle

Servlet class는 javaSE에서의 class와는 다르게 main method가 없다. 즉 객체의 생성부터 사용(method call)의 주체가 사용자가 아닌 Servlet Container에게 있다.

Client가 요청(request)을 하게 되면 Servlet Container는 Servlet객체를 생성(한번만)하고, 초기화(한번만) 하며 요청에 대한 처리(요청시마다 반복)를 하게 된다. 또한 Servlet객체가 필요 없게 되면 제거하는 일까지 Container가 담당하게 된다.

Servlet's Life-Cycle

✓ Servlet Life-Cycle의 주요 method



method	description
init()	서블릿이 메모리에 로드 될 때 한번 호출. 코드 수정으로 인해 다시 로드 되면 다시 호출.
doGet()	GET방식으로 data전송 시 호출.
doPost()	POST방식으로 data전송 시 호출.
service()	모든 요청은 service()를 통해서 doXXX()메소드로 이동.
destroy()	서블릿이 메모리에서 해제되면 호출. 코드가 수정되면 호출.

— 최초 요청 시 한번만 실행

— 요청 시 마다 반복 실행

이어서..

삼성 청년 SW 아카데미

Servlet Parameter 처리

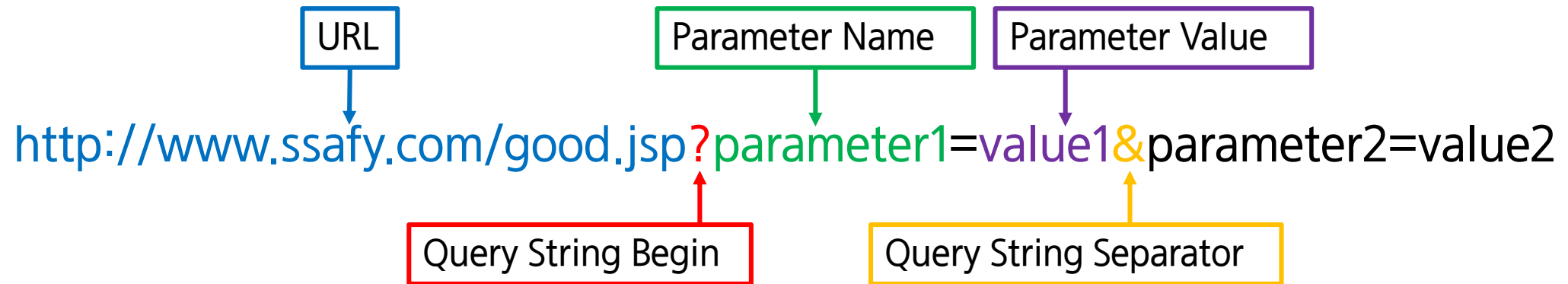
삼성 청년 SW 아카데미

✓ Parameter 전송 방식

	GET	POST
특징	전송되는 데이터가 URL뒤에 Query String으로 전달. 입력 값이 적은 경우나 데이터가 노출이 되도 문제가 없을 경우 사용.	URL과 별도로 전송. HTTP header 뒤 body에 입력 스트림 데이터로 전달.
장점	간단한 데이터를 빠르게 전송. form tag뿐만 아니라 직접 URL에 입력하여 전송 가능.	데이터의 제한이 없다. 최소한의 보안 유지 효과를 볼 수 있다.
단점	데이터 양에 제한이 있다. (location bar(URL + parameters)를 통해 전송할 수 있는 데이터의 사이즈는 2kb(2048byte)로 제한된다.	전달 데이터의 양이 같을 경우 GET방식보다 느리다. (전송 패킷을 body에 데이터를 구성해야 하므로)

[GET과 POST방식 비교]

✓ URL? QueryString? Parameter?



이어서..

삼성 청년 SW 아카데미

JSP

(Java Server Page)

삼성 청년 SW 아카데미

- 핵심내용 -

- WEB + JDBC
- JSP(Java Server Page)?
- Scripting Element
- directive
- JSP 기본객체

✓ JSP (Java Server Page)

자바 서버 페이지([영어](#): JavaServer Pages^[1], JSP)는 [HTML](#) 내에 [자바](#) 코드를 삽입하여 [웹 서버](#)에서 동적으로 웹 페이지를 생성하여 [웹 브라우저](#)에 돌려주는 언어이다. [Java EE](#) 스펙 중 일부로 [웹 애플리케이션 서버](#)에서 동작한다.

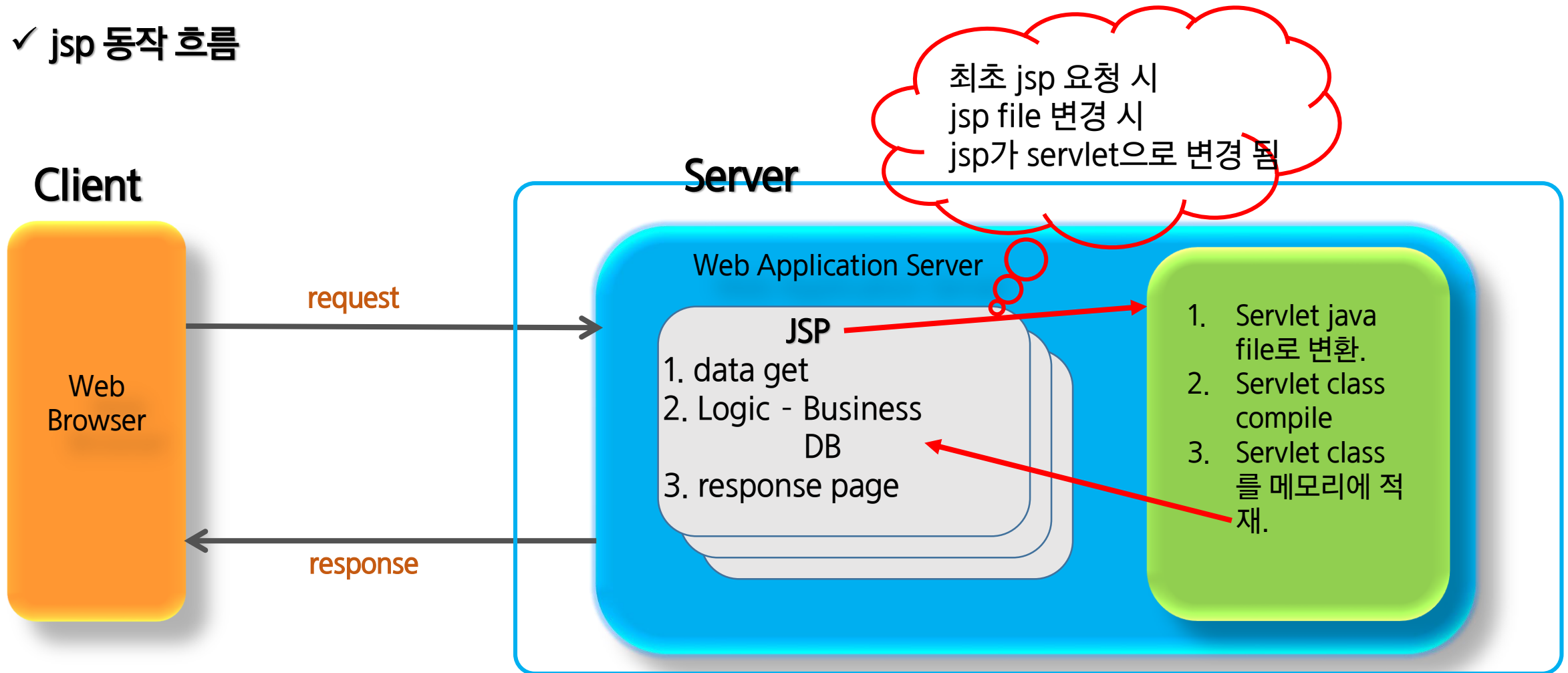
자바 서버 페이지는 실행시에는 [자바 서블릿](#)으로 변환된 후 실행되므로 서블릿과 거의 유사하다고 볼 수 있다. 하지만, 서블릿과는 달리 HTML 표준에 따라 작성되므로 웹 디자인하기에 편리하다. [1999년 썬 마이크로시스템즈](#)에 의해 배포되었으며 이와 비슷한 구조로 [PHP](#), [ASP](#), [ASP.NET](#) 등이 있다.

[아파치 스트럿츠](#)나 [자카르타 프로젝트](#)의 [JSTL](#) 등의 JSP 태그 라이브러리를 사용하는 경우에는 자바 코딩없이 태그만으로 간략히 기술이 가능하므로 생산성을 높일 수 있다.

[출처 : 위키백과]

JSP Process

✓ jsp 동작 흐름



servlet 변환 파일 확인 :

%workspace%\W.metadata\W.plugins\Worg.eclipse.wst.server.core\Wtmp0\work\WCatalina\Wlocalhost\W[project]\Worg\Wapache\Wjsp

이어서..

삼성 청년 SW 아카데미

JSP Scriptlet

삼성 청년 SW 아카데미

Declaration

✓ JSP 스크립팅 요소(Scripting Element) - 선언

1. 선언 (Declaration)

멤버변수 선언이나 메소드를 선언 하는 영역.

형식)

<%! 멤버변수와 method작성 %>

예)

```
<%!  
String name;  
  
public void init() {  
    name = "안효인";  
}  
%>
```


✓ JSP 스크립팅 요소(Scripting Element) - 스크립트릿

2. 스크립트릿 (Scriptlet)

Client 요청 시 매번 호출 영역으로, Servlet으로 변환 시 service() method에 해당되는 영역.
request, response에 관련된 코드 구현.

형식)

`<% java code %>`

예)

```
<%  
for(int dan=2; dan<10; dan++) {  
    out.println("<tr>");  
    String classname = dan % 2 == 0 ? "color1" : "color2";  
    for(int i=1; i<10; i++) {  
        out.println("<td class=\"\" + classname + \"\">\" + dan +  
            \" * \" + i + \" = \" + dan * i + "</td>");  
    }  
    out.println("</tr>");  
}  
%>
```

✓ JSP 스크립팅 요소(Scripting Element) - 표현식

3. 표현식 (Expression)

데이터를 브라우저에 출력할 때 사용.

형식)

`<%= 문자열 %>`

예)

```
안녕 <%= name %>!!!
```

주의) `<%= 문자열; %>` : 문자열 뒤 세미콜론(;)은 작성X.

`<%= 문자열 %>` == `<% out.print(문자열); %>` : 같은 표현.

✓ JSP 스크립팅 요소(Scripting Element) - 주석

4. 주석 (Comment)

코드 상에서 부가 설명을 작성.

형식)

`<%-- 주석 할 code --%>`

예)

```
<!-- HTML 주석 -->  
<%-- JSP 주석 --%>  
<%  
/*  
JAVA 주석  
*/  
%>
```

* html 주석과 jsp 주석은 차이점?

이어서..

삼성 청년 SW 아카데미

JSP Directive

삼성 청년 SW 아카데미

✓ JSP 지시자 (Directive)

1. page Directive

컨테이너에게 현재 JSP페이지를 어떻게 처리할 것인가에 대한 정보를 제공한다.

형식) `<%@ page attr1="val1" attr2="val2" ... %>`

2. include Directive

특정 jsp file을 페이지에 포함.

여러 jsp페이지에서 반복적으로 사용되는 부분을 jsp file로 만든 후 반복 영역에 include 시켜 반복되는 코드를 줄일 수 있다.

형식) `<%@ include file="/template/header.jsp" %>`

3. taglib Directive

JSTL 또는 사용자에게 의해서 만든 커스텀 태그(custom tag)를 이용할 때 사용되며

JSP 페이지 내에 불필요한 자바 코드를 줄일 수 있다.

형식) `<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>`

✓ JSP 지시자 (Directive) - page

속성		기본값	설명
language	java		스크립트에서 사용 할 언어 지정
info			현재 JSP 페이지에 대한 설명
contentType	text/html; charset=ISO-8859-1		브라우저로 보내는 내용의 MIME 형식 지정 및 문자 집합 지정
pageEncoding	ISO-8859-1		현재 JSP 페이지 문자집합 지정
import			현재 JSP 페이지에서 사용할 Java 패키지나 클래스를 지정
session	true		세션의 사용 유무 설정

✓ JSP 지시자 (Directive) - page

속성	기본값	설명
errorPage		에러가 발생할 때에 대신 처리될 JSP 페이지 지정
isErrorPage	false	현재 JSP 페이지가 에러 핸들링 하는 페이지인지 지정하는 요소.
buffer	8KB	버퍼의 크기
autoflush	true	버퍼의 내용을 자동으로 브라우저로 보낼 지에 대한 설정
isThreadsafe	true	현재 JSP 페이지가 멀티 쓰레드로 동작해도 안전한지 여부를 설정하는 것으로 false인 경우 JSP페이지는 SingleThread로 서비스 된다.
extends	javax.servlet.jsp.HttpJspPage	현재 JSP 페이지를 기본적인 클래스가 아닌 다른 클래스로부터 상속하도록 변경.

이어서..

삼성 청년 SW 아카데미

JSP 기본객체

삼성 청년 SW 아카데미

✓ JSP 기본객체 - 1

기본 객체명	Type	설명
request	javax.servlet.http.HttpServletRequest	HTML 폼 요소의 선택 값 등 사용자 입력 정보를 읽어올 때 사용
response	javax.servlet.http.HttpServletResponse	사용자 요청에 대한 응답을 처리하기 위해 사용
pageContext	javax.servlet.jsp.PageContext	각종 기본 객체를 얻거나 forward 및 include 기능을 활용할 때 사용
session	javax.servlet.http.HttpSession	클라이언트에 대한 세션 정보를 처리하기 위해 사용 page directive의 session 속성을 false로 하면 내장 객체는 생성이 안된다.
application	javax.servlet.ServletContext	웹 서버의 애플리케이션 처리와 관련된 정보를 레퍼런스하기 위해 사용

✓ JSP 기본객체 - 2

기본 객체명	Type	설명
out	javax.servlet.jsp.JspWriter	사용자에게 전달하기 위한 output 스트림을 처리할 때 사용
config	javax.servlet.ServletConfig	현재 JSP에 대한 초기화 환경을 처리하기 위해 사용
page	java.lang.Object	현재 JSP페이지에 대한 참조 변수에 해당됨
exception	java.lang.Exception	Error를 처리하는 JSP에서 isErrorPage속성을 true로 설정하면 exception내장 객체를 사용할 수 있고 기본은 false로 설정되어 있다. 전달된 오류 정보를 담고 있는 내장 객체

✓ JSP 기본객체의 영역(scope)

기본객체	설명
pageContext	하나의 JSP페이지를 처리할 때 사용되는 영역. 한번의 클라이언트 요청에 대하여 하나의 JSP 페이지가 호출되며, 이때 단 한 개의 page객체만 대응이 된다. 페이지 영역에 저장한 값은 페이지를 벗어나면 사라진다. ※ 커스텀 태그에서 새로운 변수를 추가할 때 사용한다.
request	하나의 HTTP 요청을 처리할 때 사용되는 영역. 웹 브라우저가 요청을 할 때마다 새로운 request객체가 생성됨. request영역에 저장한 속성은 그 요청에 대한 응답이 완료되면 사라진다.
session	하나의 웹 브라우저와 관련된 영역. 같은 웹브라우저 내에서 요청되는 페이지들은 같은 session들을 공유하게 됨. ※ 로그인 정보 등을 저장한다.
application	하나의 웹 어플리케이션과 관련된 영역. 웹 어플리케이션당 1개의 application 객체가 생성됨. 같은 웹 어플리케이션에서 요청되는 페이지들은 같은 application객체를 공유함.

✓ JSP 기본객체의 영역(scope) - 공통 method

servlet과 jsp페이지 간에 특정 정보를 주고 받거나 공유 하기 위한 메소드를 지원.

method	설명
void setAttribute (String name , Object value)	문자열 name 이름으로 Object형 데이터를 저장한다. Object형이므로 어떠한 Java 객체도 저장이 가능하다.
Object getAttribute(String name)	문자열 name에 해당하는 속성 값이 있다면 Object 형태로 가져오고 없으면 null 을 리턴 한다. 따라서 리턴 값에 대한 적절한 형 변환이 필요하다.
Enumeration getAttributeNames()	현재 객체에 저장된 속성들의 이름들을 Enumeration 형태로 가져온다.
void removeAttribute(String name)	문자열 name에 해당하는 속성을 삭제한다.

✓ WEB Page 이동

	forward(request, response)	sendRedirect(location)
사용 방법	<pre>RequestDispatcher dispatcher = request.getRequestDispatcher(path); dispatcher.forward(request, response);</pre>	<pre>response.sendRedirect(location);</pre>
이동 범위	동일 서버(project)내 경로	동일 서버 포함 타 URL 가능.
location bar	기존 URL 유지 (실제 이동되는 주소 확인 불가)	이동하는 page로 변경
객체	기존의 request와 response가 그대로 전달.	기존의 request와 response는 소멸되고, 새로운 request와 response가 생성.
속도	비교적 빠름	forward()에 비해 느림
데이터 유지	request의 setAttribute(name, value)를 통해 전달.	request로는 data 저장 불가능. session이나 cookie를 이용.

내일 방송에서 만나요!

삼성 청년 SW 아카데미