

* JSP란??

- JSP(Java Server Page) : JAVA코드가 들어가 있는 HTML 코드
- Java의 웹 서버 프로그램 스펙(서블릿)으로 변환되어 서비스 됨.

* Servlet과 JSP의 차이점

- Servlet

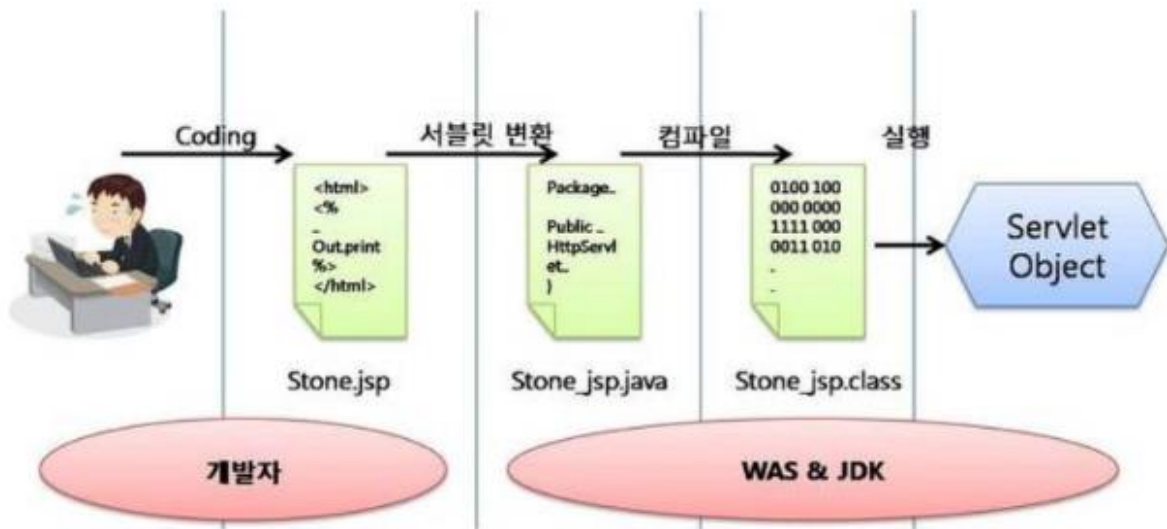
- > "웹 서비스 기능을 해주는 자바 클래스"를 말하는 것으로
자바 소스코드 속에 HTML 코드가 들어가는 형태
-> HTML 문서를 작성하는데 복잡하고 번거롭다는 단점이 있음.

- JSP

- > 복잡한 Servlet을 좀 더 간단히 사용할 수 있음.
- > Servlet과 반대로 HTML소스코드 속에
자바 소스코드(<% %> 또는 <%= %>)가 들어가는 형태.

> 컴파일을 통해 클래스 파일로 변환되어 웹 서버(WAS)에서 실행됨.

* JSP 동작 구조



`jspFile.jsp ==> _jspFile.java ==> _jspFile.class`

* Servlet/JSP 동작 순서

- 1) 웹 서버가 사용자로부터 서블릿에 대한 요청을 받으면 서블릿 컨테이너에 그 요청을 넘김
- 2) 요청을 받은 컨테이너는 `HttpRequest`와 `HttpResponse` 객체를 만들어 이를 서블릿 `doPost()`나 `doGet()` 메소드 중 하나를 호출
- 3) 클라이언트가 요청에 대한 응답 웹 페이지를 보여줄 때

기존 Servlet에서 출력 객체를 이용하여

HTML 문서 작성하던 부분(프레젠테이션 로직)을 JSP로 분리함.

-> 데이터의 입력, 수정 등의 대한 제어를 JSP로 넘겨 가독성을 높임.

- 4) JSP에게 넘겨진 프레젠테이션 로직을 수행한 후 컨테이너에게 Response 전달
- 5) 전달받은 JSP 프레젠테이션 로직 수행 결과와 Servlet이 결합되어 `.class` 파일이 만들어짐

* JSP 장점

- 서블릿보다 쉽고 작성하기 빠름
- 디자인 부분(html)과 로직 부분(java)으로 이루어져 있음.
정보, 디자인 부분(html)과 로직 부분(java)을 분리시킬 수도 있음.
- 프로그래머가 직접 코딩한 서블릿 보다 최적화 된 서블릿으로 생성시켜주므로
효율적인 코드가 만들어짐
- 웹 애플리케이션 상에서 변수의 사용 가능한 범위(scope)설정이 쉬움

* JSP를 이루는 구성인자(element)

1. 지시어(directive) :

- 1) **page** : 현재의 JSP페이지를 컨테이너에서 처리하는데 필요한 각종 속성을

기술하는 부분으로 보통 소스의 맨 앞에 위치

```
<%@ page 속성1="속성값1" 속성2="속성값2" ... %>
```

page 지시어 속성 종류

- > **contentType** : MIME 형식 지정 및 캐릭터셋 설정(text.html; charset=ISO-8891-1)

```
<%@ page contentType="text/html" %>
```

- > **import** : 자바 클래스 사용, JSP내에서 사용할

외부 자바 패키지나 클래스의 import 지정 = java import

```
<%page import="java.sql.*, java.util.*" %>
```

- > **errorPage** : 오류 페이지 관리, 현재 페이지에서 오류 발생 시 호출될 페이지 지정,

에러 발생시 포워딩(페이지 주소는 그대로고 화면은 이동) 됨

```
<%@ page errorPage="error.jsp" %>
```

> **isErrorPage** : 오류 페이지 관리, `errorPage`속성에 설정된 오류 처리 파일로

다른 용도 사용 못하고 오직 오류만을 처리하는 페이지로 지정(true)

true로 두면 `exception`객체를 쓸 수 있게 된다.

```
<%@ page isErrorPage="true" %>
```

> **language** : 사용할 스크립트 언어 유형 지정(java)

```
<%@ page language = "java" %>
```

> **session** : 세션의 사용유무 (true), 웹 브라우저와 웹 서버가 지속적으로 상대를 인식하기 위해 필요한 정보를 임시로 저장해두는 방법

```
<%@ page session="true" %>
```

> **buffer** : 버퍼의 크기 (8KB), jsp페이지 내용을 출력하려면 `JspWriter`객체인 `out`변수 사용,

이때 `out`객체로 사용할 버퍼의 크기 지정 가능

버퍼는 데이터를 좀 더 효율적이고 안전하게 전송하기 위한 프로그래밍 기법으로

전송할 데이터를 미리 확보한 후 조금씩 전송하는 방법

```
<%@ page buffer="8KB" %>
```

> **autoFlush** : 버퍼의 내용 자동 비움 (true)

```
<%@ page autoFlush="true" %>
```

> isThreadSafe : 단일스레드 모델을 사용함으로써 true 동시성 제어 여부 지정

서블릿의 장점은 각 사용자 접속에 대한 요청을 프로세스가 아닌 스레드로 처리

스레드를 이용하면 좀 더 효율적으로 시스템을 운영할 수 있으나

각 스레드가 자원을 공유하게 때문에 이들 데이터에 대한 안정성을 보장할 수 없게 됨

이 옵션은 스레드 상태에서 데이터의 안정성을 보장하는 옵션

```
<% page isThreadSafe="true" %>
```

> info : JSP 페이지에 대한 설명부분, 일종의 주석

```
<% page info="JSP Example" %>
```

> **pageEncoding** : JSP파일(페이지)에 기록된 소스코드 자체의 인코딩 방식

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
```

(pageEncoding 은 ContentType의 charset 과 동일한 기능을 함)

2) include

jsp 파일이 자바파일로 바뀔 때 자바문서에 삽입되어 컴파일 되는 파일 추가

```
<%@include file="today.jsp" %>
```

3) taglib

JSP 기능을 좀더 확장할 수 있는 인자

정의된 사용자 정의 태그 가져와 사용할 수 있는 방법 제공

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

2. 스크립팅 원소(**scripting element**) : JSP페이지에서 자바코드를 직접 기술할 수 있게 하는 기능

1) 선언문(declaration)

<%! 자바코드 %>

2) 스트립틀릿(**scriptlet**)

<% 자바코드 %>

3) 출력식, 표현식(**expression**)

<%= 자바코드 %>

3. JSP액션(표준 액션, 커스텀 액션)

* JSP페이지에서 사용할 수 있는 내장객체변수의 종류

- 일반적인 JAVA프로그램에서는 변수를 사용하기 전에 반드시 선언을 해야하지만

JSP페이지에서는 선언을 사용하지 않고도 사용 할 수 있는 변수가 있음

-> 이러한 변수를 JSP페이지의 "내장객체변수(implicit variable)"라 함

왜냐하면 웹컨테이너(=WAS, 톰캣서버)가 JSP페이지를 Servlet클래스로 변환해줄 때 자동적으로 변수 선언을 해주기 때문에 변수 선언 없이 그냥 사용 가능

- request

- > doGet, doPost메소드의 첫번째 파라미터와 동일한 역할
- > 웹클라이언트에서 보내온 데이터 값을 받아서 처리해주는 변수
- > 데이터 저장소 역할을 할 수 있음

- response

- > doGet, doPost메소드의 두번째 파라미터와 동일한 역할
- > 웹클라이언트의 요청에 대해 응답결과 처리를 위한 변수

- out

- > 웹브라우저로 HTML코드를 출력해주는 기능
- > Servlet클래스 안에서 response.getWriter()메소드를 호출해서 얻은 PrintWriter객체와 동일한 기능을 함

- session

- > 세션 : 여러 웹 컴포넌트들이 협력작업을 시작해서 끝내기까지의 기간 의미
- > session 내장 객체 속에 세션을 시작하고 끝나는 기능과 그 기간동안 웹 컴포넌트(웹서비스를 하고 있는 어디서나)들이 데이터를 주고 받을 수 있는 기능이 들어있음

- application

> JSP페이지가 속하는 웹애플리케이션에 관련된 기능을 함

- config

> JSP페이지가 속하는 웹애플리케이션의 초기상태와 관련된 기능을 함