

* 자바빈 (JAVA Bean)

- 자바빈이란 JAVA언어의 데이터(변수)와 기능(메서드)으로 이루어진 클래스입니다.
- 자바빈은 데이터를 저장하는 변수, 데이터를 읽어오는 메서드(getter), 데이터를 저장할 때 사용되는 메서드(setter)로 이루어져 있습니다.
- 자바빈은 데이터베이스와의 반복적인 작업을 효율적으로 처리하기 위해 사용합니다.
- JSP에서는 액션태그를 사용하여 자바빈을 사용할 수 있는 방법이 있습니다.
- JSP 액션태그로 자바빈 사용하는 방법

ex) <jsp:useBean id="자바빈 객체 이름" class="자바빈이 위치한 실제 경로" scope="객체를 사용할 범위" />

1. id - JSP 페이지에서 자바빈 객체에 접근할 때 사용할 이름을 지정함.
2. class - 패키지 이름을 포함한 자바빈 클래스의 완전한 경로를 입력함.
3. scope - 자바빈 객체를 저장할 영역을 지정함.

- a. page: 하나의 JSP페이지를 처리할 때 사용되는 영역.
- b. request: 하나의 요청을 처리할 때 사용되는 영역.
- c. session: 하나의 웹 브라우저와 관련된 영역.
- d. application: 하나의 웹 어플리케이션과 관련된 영역.

- JSP 액션태그로 setter와 getter메서드를 사용할 수 있습니다.

1. setter 사용 방법

ex) <jsp:setProperty name="자바빈 id" property="자바빈 클래스의 변수명" value="할당할 값" />

- a. name: 값을 변경할 자바빈 객체의 이름을 지정합니다. useBean태그에서 id속성에 지정한 값을 그대로 사용합니다.
- b. property: 값을 지정할 프로퍼티의 이름을 지정합니다. 자바빈 클래스의 변수명을 적어줍니다.
- c. value: 프로퍼티의 값을 지정합니다. 표현식이나 EL도 사용할 수 있습니다.

2. getter 사용 방법

ex) <jsp:getProperty name="자바빈 id" property="변수명" />

- a. name: useBean태그에서 id속성에 지정한 값을 사용합니다.
- b. property: 출력할 프로퍼티의 이름을 지정합니다. 자바빈 클래스의 변수명을 적습니다.

* Servlet 특징

1. 동적 웹어플리케이션 컴포넌트
2. .java 확장자
3. 클라이언트의 요청에 동적으로 작동하고, 응답은 html을 이용.
4. java thread를 이용하여 동작.
5. MVC패턴에서 Controller로 이용됨.

* URL-Mapping

- URL 매핑을 하지 않으면 URL주소가 너무 길어지고, 경로가 노출되어 보안에 위험이 생기기 때문에 URL 매핑을 사용하여 그 문제들을 해결합니다.

- http://localhost:8181/JSPBasic/servlet/kr.co.koo.HelloWorld

---->> http://localhost:8181/JSPBasic/HelloWorld

- 사용 방법

1. 아노테이션 이용, 클래스 선언부 바로 위에 작성.

ex) @WebServlet("/HelloWorld")

2. web.xml 설정파일 수정.

ex)

```
<servlet>
    <servlet-name>helloworld</servlet-name>
    <servlet-class>kr.co.koo.HelloWorld</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>helloworld</servlet-name>
    <url-pattern>/HWorld</url-pattern>
</servlet-mapping>
```

* Servlet 작동 순서

- 클라이언트에서 요청(request)이 들어오면 서버에서는 servlet 컨테이너를 만들고, 요청이 있을때 thread와 Servlet 객체가 생성됩니다.

* Servlet의 생명주기(LifeCycle)

- Servlet의 장점은 빠른 응답 속도입니다.

- Servlet은 최초 요청시에 객체가 만들어져 메모리에 로딩되고, 이후 추가 요청시에는 기존의 객체를 재활용하게 됩니다. 따라서 동작 속도가 매우 빠릅니다.

1. Servlet 객체를 생성 (최초 한번)

2. Init() 메서드 호출(최초 한번)

3. doGet(), doPost(), service() 호출 (요청시 매번)

4. destroy() 호출 (마지막 한번) - 자원이 해제될 시 호출(Servlet코드를 수정, 서버 재가동할 시)

* 웹 어플리케이션 생명주기 (ServletContextListener)

- 웹 어플리케이션에는 프로그램의 생명주기를 감시하는 리스너가 있습니다.

- 리스너의 해당 메서드가 웹 어플리케이션의 시작과 종료시에 호출됩니다.

1. 시작시에는 contextInitialized()

2. 종료시에는 contextDestroyed()

* 서블릿 초기화 파라미터 (ServletConfig)

- 특정한 서블릿이 생성될 때 초기에 필요한 데이터들이 있습니다.
- 이러한 데이터들을 초기화 파라미터라고 하며, 아노테이션으로 지정하는 방법과, web.xml파일에 기술하는 방법이 있습니다.

* 데이터 공유(ServletContext)

- 여러 서블릿에서 특정 데이터를 공유해야 할 경우 Context Parameter를 이용하여 web.xml파일에 데이터를 기술하고, 여러 서블릿에서 공유하면서 사용할 수 있습니다.