

# JWP Basic for Web UI - 1주차

—



## 학습 목표

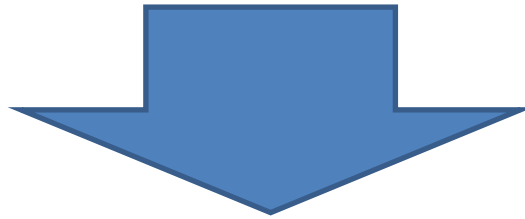
- Servlet/JSP의 역할, 동작 방식에 대한 이해
- Servlet Lifecycle, Servlet과 Servlet Container의 관계
- Eclipse에서 Servlet/JSP를 개발하는 환경에 친숙해 지기
- Servlet/JSP 간에 페이지 이동
- URL과 Servlet 매핑 - web.xml 또는 Annotation 기반

## 미션 1

- 아무 것도 없는 상태에서 자바 웹 애플리케이션을 개발할 수 있는 프로젝트를 생성한다.
  - 힌트: eclipse에서 Java Project로 생성한다.
- 웹 애플리케이션 서버는 embedded tomcat을 사용하도록 설정한다.
  - 힌트: <http://youtu.be/lqczJuKzmLU> 동영상 참고
- index.jsp를 추가한 후 브라우저에서 <http://localhost:8080> 으로 접속했을 때 “Hello World”를 볼 수 있어야 한다.

## 미션 2

- 사용자는 회원 가입을 할 수 있어야 한다.
  - 회원 가입시 사용자 정보는 아이디, 비밀번호, 이메일, 이름 정보를 입력할 수 있다.
- 회원 가입한 사용자 정보는 메모리에 저장한다.

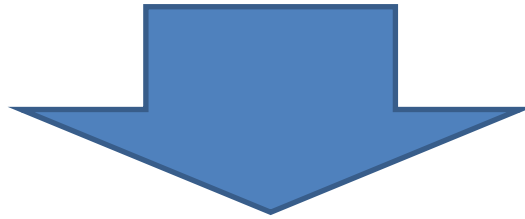


## 힌트

- JSP로 회원 가입 페이지를 구현한다.
- 사용자가 회원 가입 완료 후 form submit을 하면 입력한 값을 처리할 수 있는 Servlet을 구현한다.
- form submit URL과 Servlet을 mapping한다.
- Servlet에서 사용자가 입력한 값을 추출해 Map<String, User> Collection에 저장한다.

## 추가 미션(미션 2까지 끝내고 시간이 남는 경우)

- 회원 가입한 사용자의 정보를 수정할 수 있다.



## 힌트

- 이미 회원 가입한 사용자의 회원 아이디를 URL 인자로 전달하면 개인정보 수정 페이지로 이동한다.
  - `http://localhost:8080/user/updateForm.jsp?userId=pobi`
- 개인 정보 수정 후 form submit을 하면 수정한 값을 처리할 수 있는 Servlet을 구현한다.
- form submit URL과 Servlet을 mapping한다.
- ArrayList Collection에 저장되어 사용자를 찾아 사용자 정보를 수정한다.

## 마감시한 및 제출 방법

- 마감시한 : 3월 10일 10시 00분
- 제출방법 : 소스 코드는 github을 통해 제출하고, github url은 이 강의 문서가 올라가 있는 게시글의 댓글을 통해 공유한다.

# 과제 및 사전 학습

—



## 과제

- 미션 1,2를 완료하지 못했을 경우 완료한다.
- 추가 미션을 완료한다.



## 3월 11일 수업 전까지 사전 학습

기본 설명	동영상	내용
개발자에게 테스트는 귀찮으면서도 정말 중요한 작업이죠. 테스트를 너무 멀리하지 말고 내 가까이 둔다면 개발이 더 재미있을 수 있다. 내가 개발한 기능을 다른 누군가가 테스트하는 것이 아니라 내가 테스트해야 된다는 생각으로 접근하면 좋겠다. 또한 테스트를 통해 개발 속도를 높일 수 있다는 인식의 전환도 있었으면 한다.	<a href="http://youtu.be/P3cYfVMdFHY">http://youtu.be/P3cYfVMdFHY</a>	<ul style="list-style-type: none"><li>• 단위 테스트를 활용하지 않을 때의 개발/테스트 과정 설명</li><li>• test source folder 추가</li><li>• Database와 User 객체에 대한 단위 테스트 코드 추가</li><li>• 로그인 기능에 대한 단위 테스트 추가</li></ul>
HTTP는 무상태 프로토콜이기 때문에 로그인 기능과 같이 상태를 유지하는 기능을 구현하기 힘들다. 무상태 프로토콜에서 상태를 유지하기 위해 세션 또는 쿠키를 사용한다. 이 동영상은 로그인 기능에 세션 사용 방법을 다룬다. 세션이 어떻게 활용되는지 이해하고 책을 통해 세션과 쿠키에 대해 추가 학습하면 좋겠다.	<a href="http://youtu.be/hQulJST8C2c">http://youtu.be/hQulJST8C2c</a>	<ul style="list-style-type: none"><li>• 정상적인 로그인 기능 구현</li><li>• 로그인 성공 후 session에 정보 저장</li><li>• 로그 아웃 기능 구현</li></ul>
JSP에서 페이지 이동의 한 방법으로 forward 방식이 있다. forward 방식은 redirect 방식의 한계점을 극복하기 위한 방법으로 무상태 프로토콜에서 반드시 이해하고 사용해야 할 방법이다.	<a href="http://youtu.be/ANhfLsFAmas">http://youtu.be/ANhfLsFAmas</a>	<ul style="list-style-type: none"><li>• jsp에 예외 처리하는 흐름 이해</li><li>• RequestDispatcher.forward()를 활용한 jsp 이동</li></ul>

## 3월 11일 수업 전까지 사전 학습

기본 설명	동영상	내용
JSP만으로 구현할 경우 코드의 복잡도가 증가해 유지보수하는데 많은 어려움이 있다. 이에 대한 대안으로 JSP와 Servlet을 용도에 맞게 사용할 필요가 있다. JSP와 Servlet을 각각의 역할에 맞도록 사용하면서 MVC 패턴도 등장하면서 웹 애플리케이션 개발은 발전해 왔다. JSP와 Servlet을 사용해야 하는 시점을 이해하고 적절히 사용하는 연습을 하자.	<a href="http://youtu.be/EfsVt7-ol_I">http://youtu.be/EfsVt7-ol_I</a>	<ul style="list-style-type: none"><li>• JSP만으로 구현했을 때의 이슈 공유</li><li>• 이 문제를 해결하기 위한 역할 분리로 MVC 등장</li><li>• JSP와 Servlet을 병행해서 구현</li></ul>
Servlet과 JSP 페이지를 이동하기 위해 redirect와 forward의 차이점을 이해하는 것은 중요하다. 이 개념은 자바 뿐만 아니라 모든 웹 프레임워크가 가지고 있는 개념이기 때문에 반드시 이해하고 넘어가면 좋겠다.	<a href="http://youtu.be/PxvrpcZJwGs">http://youtu.be/PxvrpcZJwGs</a>	<ul style="list-style-type: none"><li>• redirect와 forward 방식의 차이</li><li>• 개발할 때 고려할 사항</li></ul>
JSP에서 html 코드와 자바 코드를 같이 사용하는 것은 가독성 측면이나 유지보수 좋지 않다. 이 같은 문제점을 해결하기 위해 JSP에서는 jstl과 expression language(이하 el)를 지원하고 있다. jstl과 el을 활용해 자바 코드를 제거해 보자. 앞으로 JSP에서는 자바 코드를 절대 사용하지 않겠다고 선언을 하고 구현하는 것도 좋은 방법이다.	<a href="http://youtu.be/Sk9-mfKpvtQ">http://youtu.be/Sk9-mfKpvtQ</a>	<ul style="list-style-type: none"><li>• jstl 라이브러리 다운로드 및 프로젝트에 추가</li><li>• jstl 활용해 if/else 구현</li><li>• expression language 활용해 스크립틀릿 코드 제거</li></ul>