



# JSP 프로그래밍

16차시

- MVC 패턴의 이해
- MVC 패턴의 활용



- 모델-뷰-컨트롤러(Model-View-Controller)
- SW공학에서 사용되는 디자인 패턴
- 어플리케이션의 역할을 모델, 뷰, 컨트롤러로 구분
- 사용자 인터페이스로부터 비즈니스 로직을 분리하여 개발하여 서로 영향 없이 쉽게 수정이 가능한 어플리케이션을 만들 수 있다.



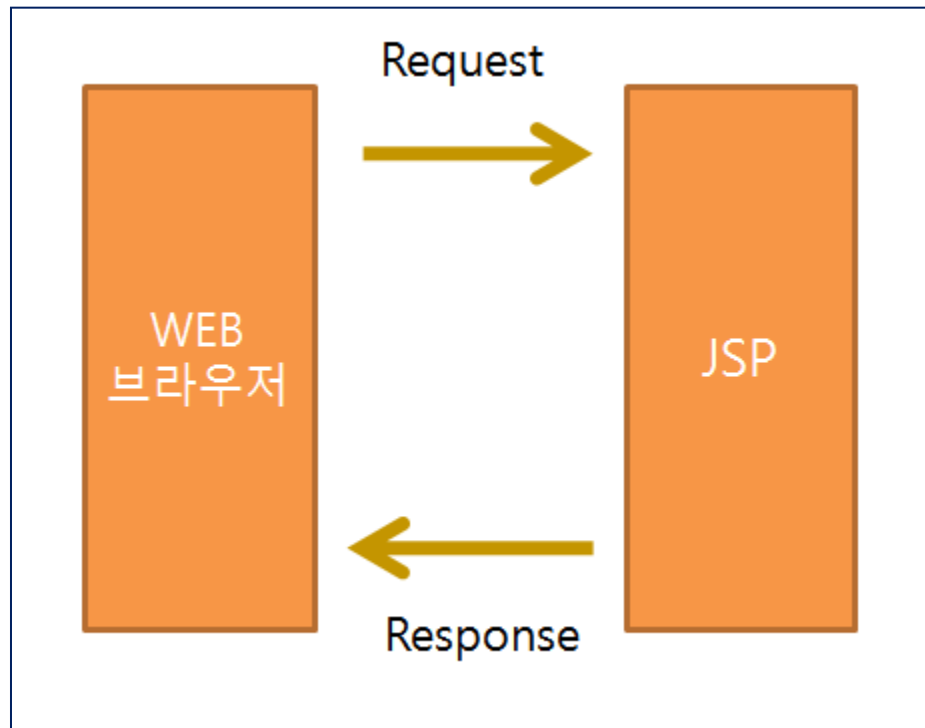
- 모델: 정보(데이터)
- 뷰: 텍스트, 체크박스 등과 같은 사용자 인터페이스
- 컨트롤러: 데이터와 비즈니스 로직 사이의 상호동작



- 모델1(Model1) VS 모델2(Model2)
- 여기서 말하는 모델1, 모델2는 MVC에서 나오는 Model과는 다른 의미라고 생각하는게 편하다.



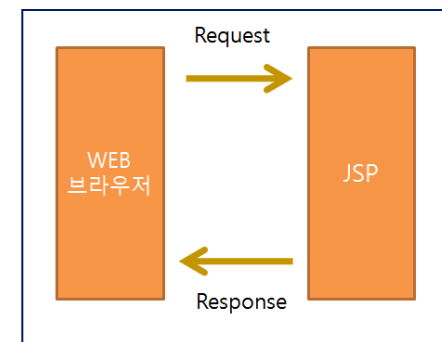
## □ 모델 1 (Model1)





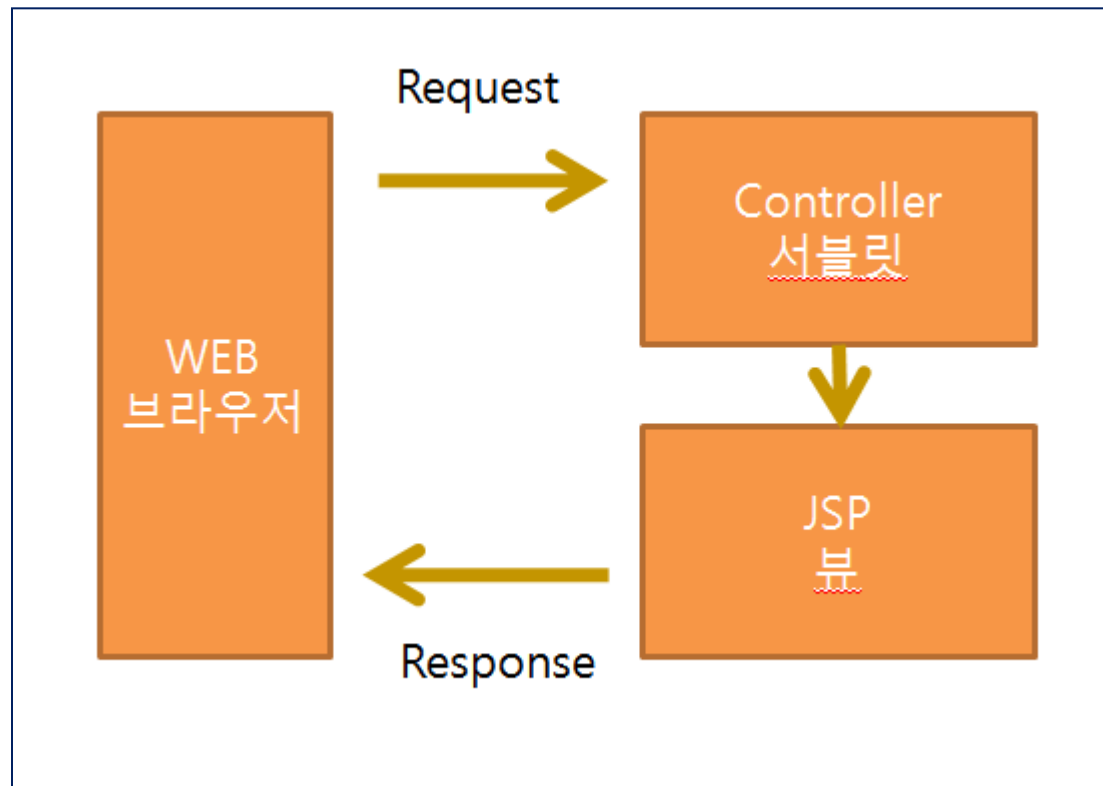
## □ 모델1(Model1)

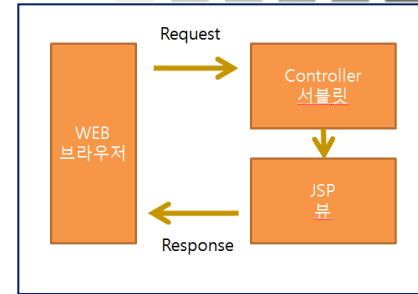
- JSP에서 비즈니스 로직, View 모두 처리
- 구조가 단순하여 익히기 쉽다.
- 구현이 익숙하고, 작은 단위의 어플리케이션에 적합
- 출력을 위한 뷰와 로직 처리를 위한 자바 코드가 섞여있기  
때문에 JSP 코드의 복잡성이 증가
- 업무 분업이 어렵다.





## □ 모델2(Model2)





## □ 모델2(Model2)

○ JSP에서 모든 것을 담당하던 모델1 방식과 달리

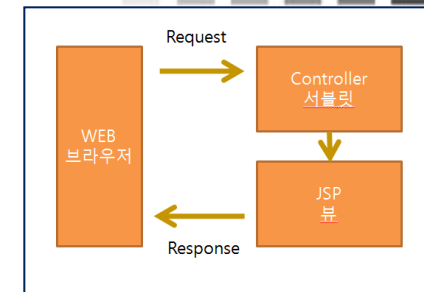
JSP페이지, 서블릿, 로직을 위한 클래스로 나누어서 브라우저  
의 요청을 처리함

○ 요청이 들어오면 로직 처리는 모델이 담당하고, 요청 결과를  
보여주는 뷰는 JSP가 담당, 이를 위한 흐름 제어를 컨트롤  
러인 서블릿이 담당한다.





## □ 모델2(Model2)



- 출력을 위한 뷰와 로직을 위한 자바 코드가 분리되어 JSP 코드가 복잡하지 않음
- 뷰, 로직처리에 대한 분업이 용이, 유지보수성 좋음
- 모델1에 비하여 설계가 어려움
- 모델1에 비하여 많은 경험치와 이해가 필요



- 자바 퍼시스턴스 프레임워크로 XML서술자나 어노테이션을 이용하여 저장 프로시저나 SQL문을 객체와 연결한다.
- JDBC로 처리하는 상당부분의 코드와 파라미터 설정 및 결과 매핑을 대신 수행해준다.
- 생산성의 향상, 관점의 분리, 작업의 분리가 가능해짐



□ 주의사항!!

□ 모델1보다 모델2가 무조건 좋은가요?

상황에 따라서 다릅니다.  
개발 범위, 자원, 인력 구성 등  
프로젝트에 따라서 장점, 단점이 있습니다.



하지만 현실은  
스프링입니다.



# 추가적으로 공부 꼭!! 하세요.



- SW공학, 전산학 기초
- 프로젝트 관리 – Maven, Gradle
- 형상관리 – SVN, Git
- CI/CD – 젠킨스
- Javascript, JQuery

이제는  
필수가 아닌  
기본입니다.



고생하셨습니다.

어제보다 오늘을 행복한  
하루로 만들어 나가길 기원드립니다.

