



Today's lecture

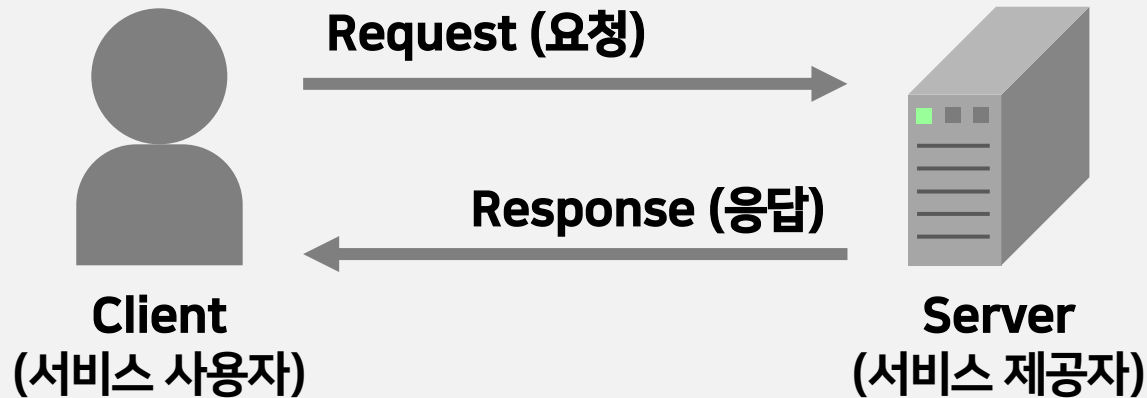


```
System.out.print('Servlet/JSP 개요');
```

네트워크 통신 개요

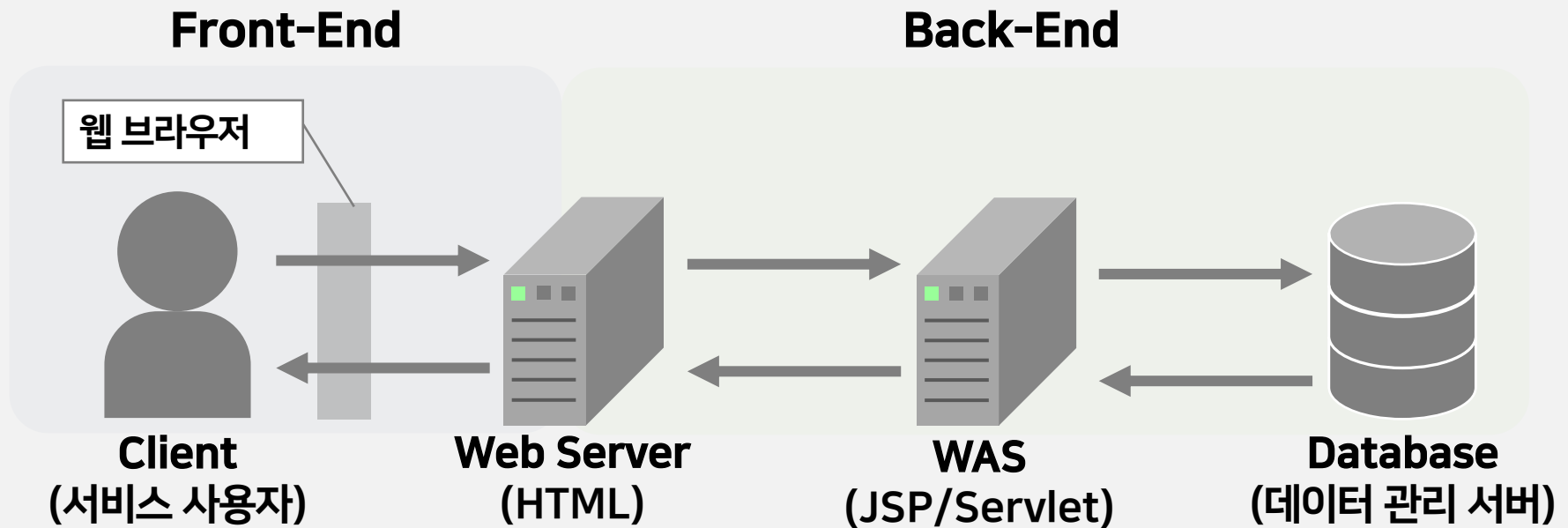
▶ Server-client Model

서버는 특정한 서비스를 제공하는 컴퓨터를,
클라이언트는 이러한 서비스를 이용하는 사용자를 말함



웹 통신 개요

▶ Web통신 구조



▶ Web Server

사용자에게 HTML페이지나 jpg, png같은 이미지를 HTTP프로토콜을 통해 웹 브라우저에 제공하는 서버로 내부의 내용이 이미 만들어져 있는 정적인 요소들을 화면에 보여주는 역할을 함

✓ Web Server의 종류

- Apache : Apache Software Foundation에서 만든 서버로
HTTP통신에 대한 여러 라이브러리 제공
- Windows IIS : Window OS에서 제공하는 웹 서버로
높은 수준의 보안성과 성능 제공
- NGINX : 무료 오픈 소스 서버로 사용자 요청을 스레드가 아닌
확장성이 있는 이벤트 기반 설계로 리소스만 할당해 사용

Web Application Server의 약자로 사용자가 요청한 서비스의 결과를 스크립트 언어 등으로 가공하여 생성한 동적인 페이지를 사용자에게 보여주는 역할

✓ WAS의 종류

- tomcat : Apache Software Foundation에서 Servlet과 JSP를 통한 동적인 웹 문서를 처리하기 위해 만든 웹 애플리케이션 서버
- wildfly : Jboss라고도 불리며 톰캣이 제공하는 Servlet Container 뿐만 아니라 EJB Container를 별도로 제공하여 폭 넓은 서비스 구현
- jeus : 국산 WAS, 대용량 데이터 트랜잭션을 고성능으로 처리하며 개발 및 운영에 관한 기술 지원이 뛰어남

▶ 서블릿 컨테이너와 JSP 컨테이너

✓ 서블릿 컨테이너(Servlet-Container)

서블릿의 생명 주기 관리(생성, 초기화, 소멸), HttpServletRequest/
HttpServletResponse 객체 생성, 요청에 따라 멀티 스레딩 구성,
전송 방식에 따라 동적으로 페이지 구성하는 작업 진행

✓ JSP 컨테이너(JSP-Container)

JSP파일을 다시 java코드로 변경해주고 class파일로 전환하여
메모리 공간에 로드한 뒤 실행 가능하게 만드는 작업 진행(Servlet화),
처리 결과를 HTML파일로 만들어주는 작업 진행

▶ Web Server VS WAS

구분	장점	단점
Web Server	<ul style="list-style-type: none"> - 빠른 처리 속도 요청에 대한 결과 페이지만 전송 - 구현이 쉬움 HTML같은 단순한 문서만으로 구성 	<ul style="list-style-type: none"> - 한정적 서비스 만들어진 정보만 보여주기 때문에 서비스가 한정적 - 글의 추가, 수정, 삭제가 어려움 문서의 내용이 변경될 경우 직접 수정
WAS	<ul style="list-style-type: none"> - 서비스의 다양성 여러 데이터를 활용할 수 있음 - 글의 추가, 수정, 삭제가 쉬움 문서의 내용이 변경될 경우 직접 수정하지 않음 	<ul style="list-style-type: none"> - 느린 처리 속도 데이터를 처리하여 결과를 전송함 - 구현이 어려움 서비스에 해당하는 소스를 직접 작성