application 객체

웹 애플리케이션 당 하나만 생성

모든 JSP 페이지에서 접근할수 있음

서버의 정보, 서버의 물리적 경로 등

web.xml에 설정한 context 초기화 매개변수 읽어올 수 있음

.getInitParameter() : 설정된 내용 읽어옴

.getRealPath() : 서버의 물리적 경로 얻어옴

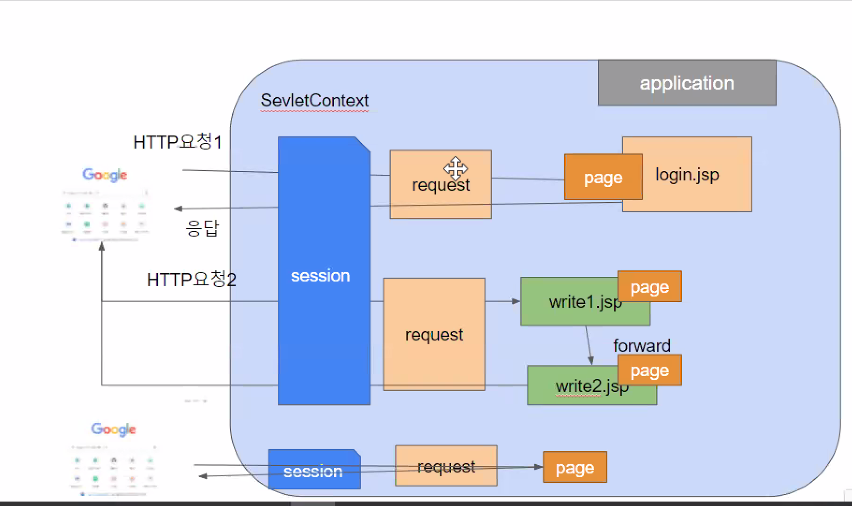
.useImplicitObject() : this

.setAttribute(name,value) : 어플리케이션에 값을 저장

.getAttribute(name) : 애플리케이션에 저장된 데이터 불러옴

.removeAttribute(name) : 지움

String.format () =sysout.printf



JSP의 영역(scope)

page 영역

페이지 영역은 한 번의 클라이언트의 요청에 하나의 JSP 페이지를 범위로 갖는다.

요청을 처리하는 JSP는 요청에 대한 Page를 가짐

그에 해당하는 pageContext라는 객체를 할당받는다.

=> (pageContext객체가 메모리에 로딩된다는 의미)

pageContext객체에 정보를 저장하면 해당 페이지 내에서만 사용할 수 있음

pageContext.setAttribute, getAttribute 이용

getAttribute 할 시 object 자료형이므로 자료형 일치시켜줘야 함

request영역

브라우저에서 오는 요청 자체가 request영역(링크 클릭, url 입력시 요청 발생)

요청 시 영역에 request 내장객체 저장

Forward된 모든 JSP와 영역 공유

브라우저가 결과(response)를 받으면 request 내장객체는 사라짐

request 객체는 getParameter, getAttribute를

통해 이전 페이지의 입력값들을 가져올 수 있음

Session 영역

브라우저를 닫기 전까지 페이지를 이동하더라도 정보를 잃지 않고 서버에 보관할 수 있는 객체

1개에 브라우저엔 1개의 Session만 생성 가능

application영역

하나의 웹 어플리케이션과 관련된 전체 영역

영역 내 모든 JSP는 application 내장객체를 공유

application 종료 시 반환

request 영역 forward

다른 서블릿이나 JSP와 연동하는 방법

request에 대한 추가 작업을 다른 서블릿에서 수행하게 함

forward된 JSP는 request 영역을 공유함

HttpServletResponse 객체의 sendRedirect()

서블릿의 요청이 웹브라우저를 거쳐 새롭게 요청할 곳을 알려줌

=> url 주소가 바뀜

JSP주소? name=”lee” 방식으로 getParameter 할 수 있음

HttpServletResponse 객체의 addHeader()이용

마찬가지로 웹브라우저에 재요청

response.addHeader("Refresh","2;url=second2");

location 객체의 href 속성 이용

자바 스크립트에서 재요청함

location.href=’주소’

RequestDispatcher의 .forward()이용

클라이언트의 브라우저를 거치지 않고 서블릿 이동

sendRedirect(), addHeader(), location.href()

: 서블릿=>클라이언트=>서블릿 요청, 응답객체 새로 생성

forward()

한 웹컨테이너 안에서 페이지 이동=> 요청,응답객체 공유