**Web개발\_Day46\_노트정리**

**내장객체**

ㆍJSP는 추후에 Servlet 파일로 변환되기 때문에 application, session, out. Request 등이 내장객체로써 제공된다.

ㆍ내장객체들은 주로 ‘데이터를 주고 받을 때’ 사용된다.

* 요쳥/ request와 응답/response에 데이터를 전달하는 용도

**Reqeust**

ㆍ값이 하나일 땐 request.getParameter(“value값”);

ㆍ값이 여러 개 넘어오는 checkbox등의 경우엔 request.getParameterValues(“value값”);

* 이 경우엔 표현식 <%= %> 로 출력하기 전에 <% %> 로 배열등에 담아주는 것이 먼저이다.

ㆍ<% %> 에서 나온 결과값을 출력하고 싶다면, 스크립트릿이 닫힌 시점에서 …

1. <%= %>으로 출력하는 것이 가능하며
2. <%= %>와 동일한 내장객체인 “out”을 이용하여 out.println(값)을 사용하는 것도 가능하다.

ㆍrequest로 전달되는 값이 ‘한글’인 경우 “전달 행위” 자체는 문제가 되지 않지만, 전달된 값을 출력하는 과정에서 인코딩이 깨질 수 있다.

* 최상단에 <% request.setCharacterEncoding(“UTF-8”); %> 이 필요하다.
* 최상단인 이유는, 해당 값을 다루는 페이지 같은 경우 제일 먼저 해줘야 하는 행동이 인코딩이기 때문이다.

**Response**

ㆍ reponse.sendRedirect(“주소”); : GET요청처럼 해당 주소로 이동

* 제어권을 ‘해당주소’로 옮기되, 요청헤더(=데이터)(=request/response 한쌍)을 교체해버린다.
* A페이지에서 B페이지로 이동할땐 request/response 한쌍이 이루어지는데, B페이지에서 C페이지로 이동시엔 새로운 request/response한쌍이 생성된다.
* Scope범위가 A페이지에서 B페이지까지 밖에 되지 않기 때문이다.

ㆍA페이지에서 값을 B페이지를 통과하여 C페이지로 넘기려 한다면…

* A페이지에서 B페이지로 이동하고, B페이지에서 forward를 사용하여 제어권을 C페이지로 향하게 한다.

**Session**

ㆍ request/response가 하나의 페이지에서 다음 페이지까지만 값을 저장한다면, session은 해당 창에서 값을 쭉 유지한다.

ㆍsession 사용

* Session.isNew() : 새로운 세션인지를 확인한다
* Session.setAttribute(“id값”, “value값”); : Session안에 값을 지정한다.
* Session.getAttribute(“id값”); : session에 넣어둔 id값을 통해 value값을 불러온다.
* Session.setMaxInactiveInterval(초단위); : session의 유효범위인 창이 꺼질 때, session을 유지할 시간을 초딘위로 설정한다.

ㆍsession의 중요성:

* 해당 창에 있다가 다른 창으로 이동 했을 경우(예: 쇼핑몰 상품목록 🡪 장바구니), 정보가 손실되면 안되므로, 해당 창에서 정보를 유지하는 것을 위해 사용한다.

**application**

ㆍ 서버 scope를 갖는다.

* 위의 session이 브라우저 단위로 scope가 유지된다면, 이건 server단위로 scope가 유지된다.
* 창이 꺼져도 정보가 저장되어야 할 경우 사용된다.
* 서버가 꺼지기 전까지, application의 데이터가 저장된다.

ㆍapplication 사용 : session과 동일한 set/getAttribute를 사용한다.

**정보 유지 범위, Scope**

ㆍ 위의 Response 파트에서 말한 것처럼 정보의 유지범위, scope를 확인하여 정확하게 세팅하는 것은 중요!

ㆍscope범위의 크기 차이

* Request : 현재 페이지 🡪 다음 페이지
* Session : 해당 브라우저(창)
* Application : 연결된 서버

ㆍ마냥 넓은 scope를 갖도록 모든 데이터를 큰 scope (예: application)에 저장한다면, 브라우저가 무거워지므로 데이터의 scope를 적재적소로 사용하는 것이 좋다.

* 서버에 부하를 줄 수 있고 연결속도에 저하가 생길 수 있다.