

JSP

[07] JSP쿠키

※ 웹브라우저에서 서버로 어떤 데이터를 요청 하면, 서버측에서는 알맞은 로직을 수행한 후 데이터를 웹브라우저에 html형태로 응답한다. 그리고, **서버는 웹브라우저와의 관계를 종료한다(무상태서버 or 무상태 로직)**. 이렇게, 웹브라우저에 응답 후 관계를 끊는 것은 http프로토콜의 특징.

****여러 브라우저에서 요청이 들어오게되면 스레드를 생성해 각각의 브라우저에게 뿌림 (동시에 진행) -> 응답이 끝난 애의 관계는 중단**

※ 연결이 끊겼을 때 어떤 정보를 지속적으로 유지하기 위한 수단으로 쿠키라는 방식을 사용한다.

※ 쿠키는 서버에서 생성하여, **서버가 아닌 클라이언트측에 특정 정보를 저장**(txt 파일 형태로 저장)한다. 그리고 서버에 요청 할 때 마다 쿠키의 속성값을 참조 또는 변경 할 수 있다.

※ 쿠키는 4kb로 용량이 제한적이며, 300개까지 데이터 정보를 가질 수 있다.

쿠키는 서버가 만들.

쿠키를 만드는 문법 보고 , 쿠키의 속성 생성(쿠키의 이름 내용, 쿠키는 유효기간이 있음 (쿠키마다 다름)), response에 쿠키 탑재, 쿠키를 확인 하는 것 , 쿠키를 삭제하는 것

```
//쿠키 생성 - 쿠키는 객체형태. 매개변수에는 (쿠키이름, 쿠키값)
Cookie cookie = new Cookie("cookieName", "cookieValue");

//쿠키의 유효 시간 setMaxAge(1시간 유효한 쿠키 : 60*60)
cookie.setMaxAge(60*60);

//response에 탑재
response.addCookie(cookie);
```

쿠키를 생성하면 쿠키가 객체형태로 생성이 되어야 함. 매개변수에는 쿠키이름과 쿠키값, 나중에 로그인로직 만들 때 쿠키 이름에 id 넣으면 됨

쿠키의 유효시간 등 속성까지 생성한 다음에 response에 탑재시켜 브라우저에 날려줘야함

```

<h2>지금 막 생성된 쿠키 이름 : <%=cookie.getName() %></h2>
<h2>지금 막 생성된 쿠키의 값 : <%=cookie.getValue() %></h2>
<hr>
<a href="ex1_cookieConstruct.jsp">쿠키생성(cookieName)</a><br>
<a href="ex2_cookieList.jsp">쿠키들(쿠키이름-쿠키값) 리스트 확인</a><br>
<a href="ex3_thatCookie.jsp">특정 쿠키(이름이 cookieName) 찾기</a><br>
<a href="ex4_cookieDel.jsp">쿠키 삭제</a>

```

쿠키이름과 쿠키 값을 확인할 수 있는 로직 보기

버튼을 누를때마다 쿠키를 생성하고, 쿠키리스트확인, 특정쿠키 찾기, 쿠키삭제 로직을 jsp에 넣어주기

쿠키 리스트 확인

```

<!-- 쿠키리스트확인 -->
<%
    //쿠키는 배열로 받아야됨
    Cookie[] cookies = request.getCookies();
    for(int idx=0; idx<cookies.length; idx++){
        String name      = cookies[idx].getName(); //idx번째 쿠키의 이름
        String value     = cookies[idx].getValue(); //idx번째 쿠키의 값
        out.println("<h2>" + idx + "번째 쿠키 이름 : " + name + ", 값 : " + value + "</h2>");
    }
%>

```

쿠키는 배열로 받아야 함

쿠키 배열 변수 안에 쿠키를 request로 받아와주고 for문 돌려주기

for문안에 name과 value값을 받은후 출력

```
<%
Cookie[] cookies = request.getCookies();
if(cookies!=null){
    int idx=0; //idx를 밖에 놓은 이유 : for 밑에서 if(idx==cookies.length)이냐 라는걸 물어보기위해
    for(idx=0; idx<cookies.length; idx++){
        //idx번째 cookies이름이 내가 찾고자 하는 쿠키의 이름과 같은지 만약 같으면 break;
        String name = cookies[idx].getName(); //idx번째 쿠키 이름
        if(name.equals("cookieName")){
            //name이 동일하면 value값을 받아서 뿌리기
            String value = cookies[idx].getValue(); //idx번째 쿠키의 값
            out.println("<h2>찾고자 하는 쿠키 이름 : " + name + ", 쿠키값: " + value + "</h2>");
            break;
        }
    }
}

if(idx==cookies.length){
    //내가 찾고자 하는 특정쿠키가 없다는 뜻
    // 왜 ? cookies.length가 3이라는건 쿠키갯수가 3이라는 뜻 하지만 idx는 0부터 돌기 때문에
    // idx==cookies.length가 같다는건 위에서 찾는 쿠키가 없어서 방을 다 돌고 3이 되었을때 떨어졌다는 뜻임
    out.println("찾고자 하는 쿠키가 없습니다");
}

}else {
    out.println("생성된 쿠키가 없습니다");
}
%>
```

[쿠키 삭제하기](#)

```
<%
//쿠키는 모든 쿠키를 삭제하려면 for문을 돌려야 함
//쿠키 삭제는 보통 특정 쿠키를 삭제한다는 뜻
//쿠키 삭제는 보통 로그아웃할때 활용
Cookie[] cookies = request.getCookies();
if(cookies!=null){
    for(Cookie c : cookies){
        String name = c.getName();
        if(name.equals("cookieName")){
            // 쿠키 삭제 : 유효기간이 0초짜리 같은 이름의 쿠키를 만들어 덮어 씌
            Cookie cookie = new Cookie("cookieName", "xxxxxx");//0초후에 사라질 아이
            cookie.setMaxAge(0); //타재하자마자 파괴될 아이
            response.addCookie(cookie);
            out.println("<h3>쿠키 삭제 성공</h3>");
        }
    }
}
%>
```