5. Session 관리

5.1 Session

5.2 Cookie

5.3 URL 재작성

5.1 Session

5.2 Cookie

5.3 URL 재작성

❖ Stateless Protocol 의 문제점

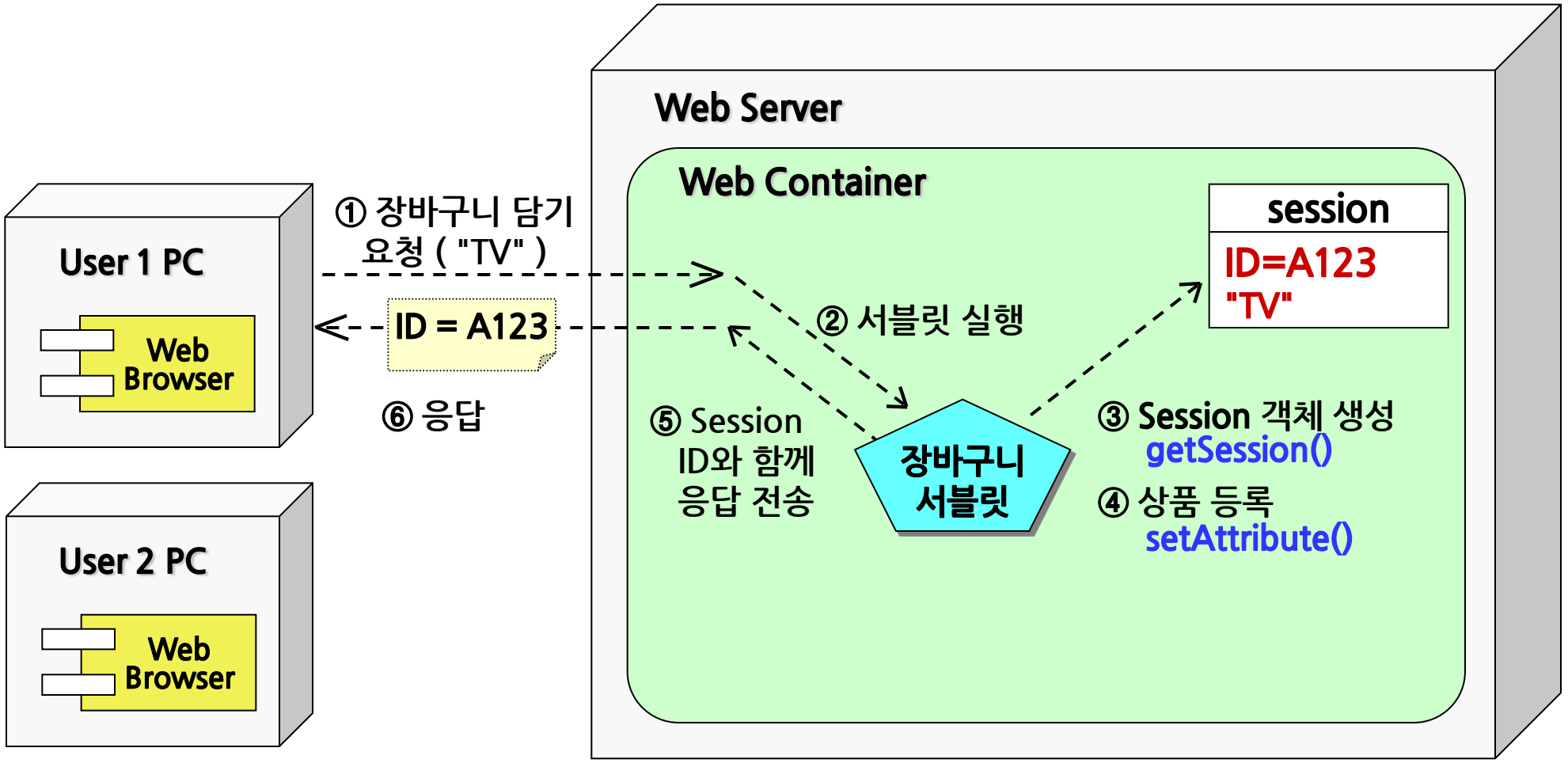
- 기본적으로 HTTP는 연결 당 하나의 요청만 처리하는 무상태(stateless) 프로토콜이다.
- 웹 서버는 하나의 요청에 대해 응답한 후, 연결을 끊는다. 따라서 동일한 사용자가 다시 요청을 했을 때 웹 서버는 동일한 사용자라는 것을 인식하지 못한다.

❖ 상태 관리 매커니즘

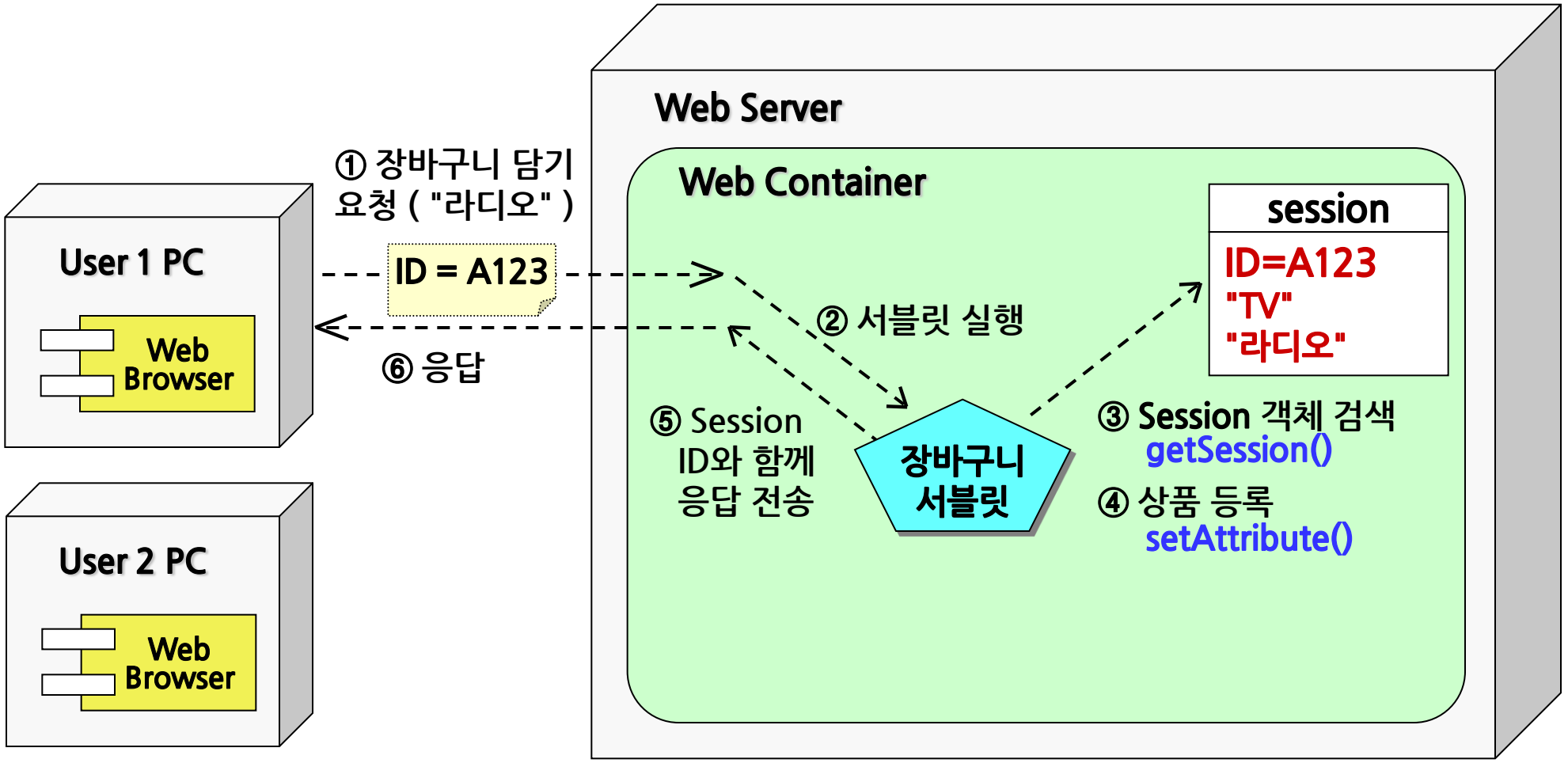
- HTTP 프로토콜의 문제점을 극복하기 위해서 사용자의 상태 정보를 관리하는 매커니즘이 필요하다.
- 이러한 매커니즘에는 세션, 쿠키, URL 재작성이 있다.

Session 관리 프로세스

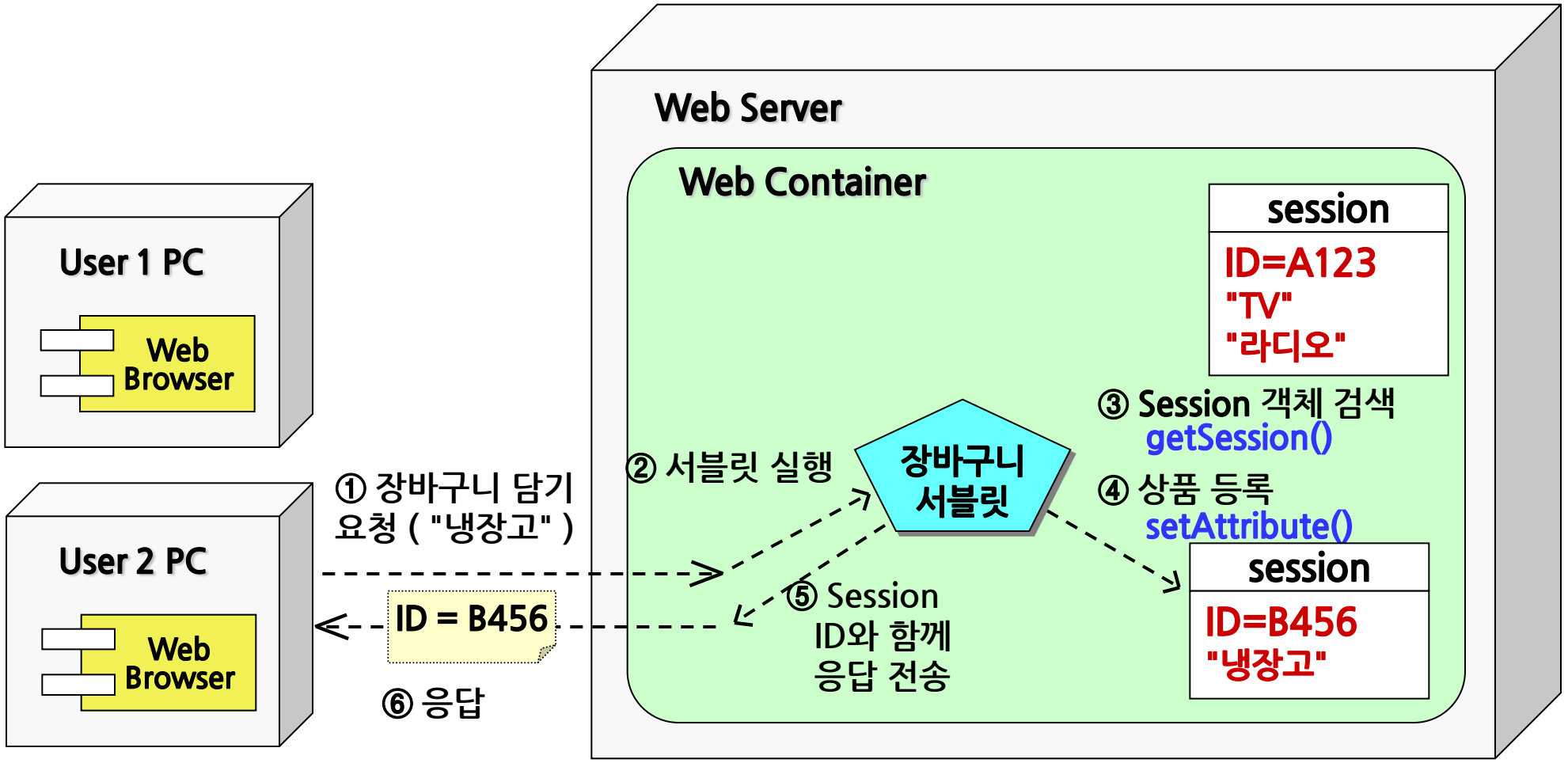
1. User1이 장바구니에 처음으로 상품을 담는다



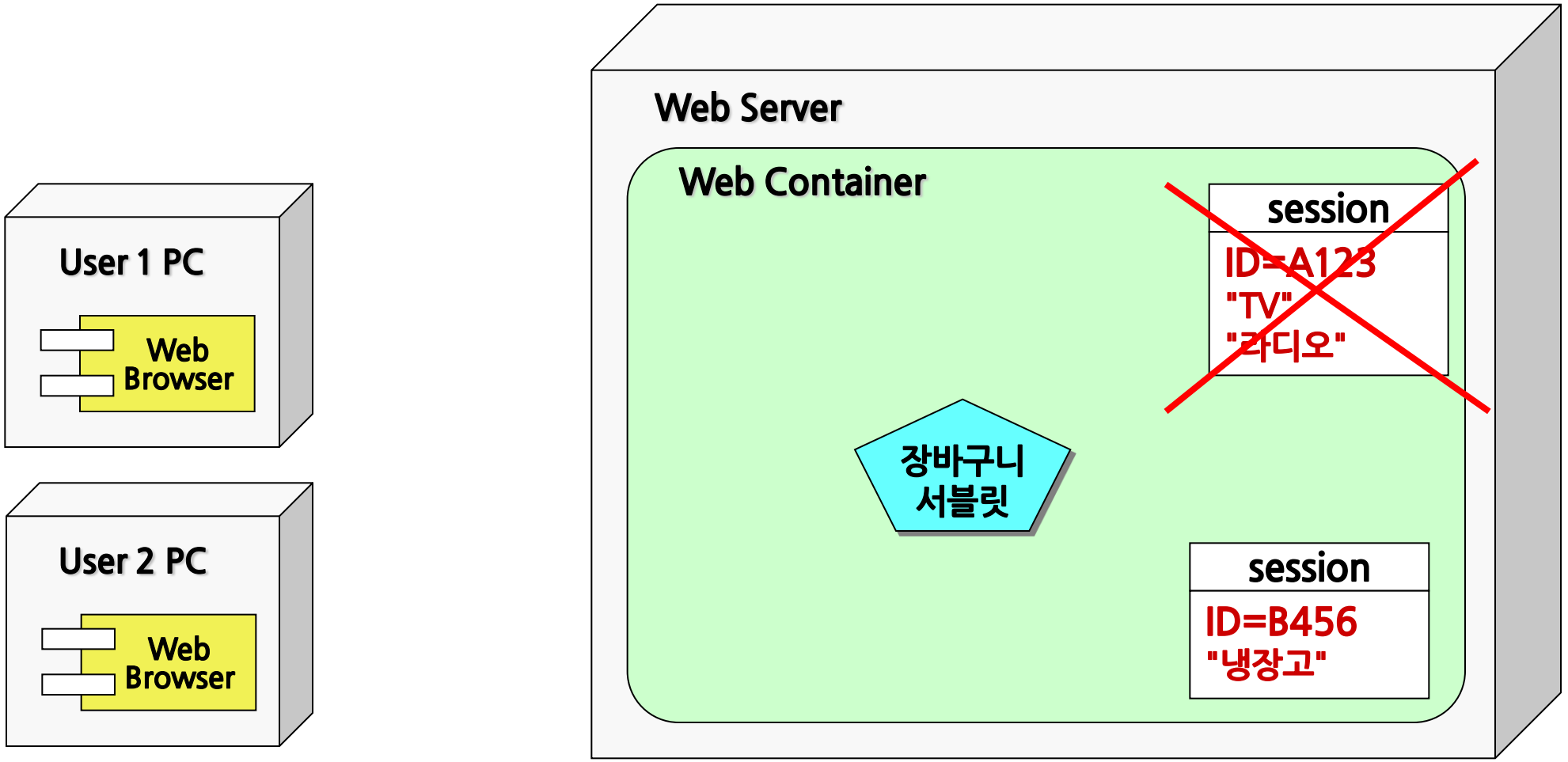
2. User1이 장바구니에 다른 상품을 담는다



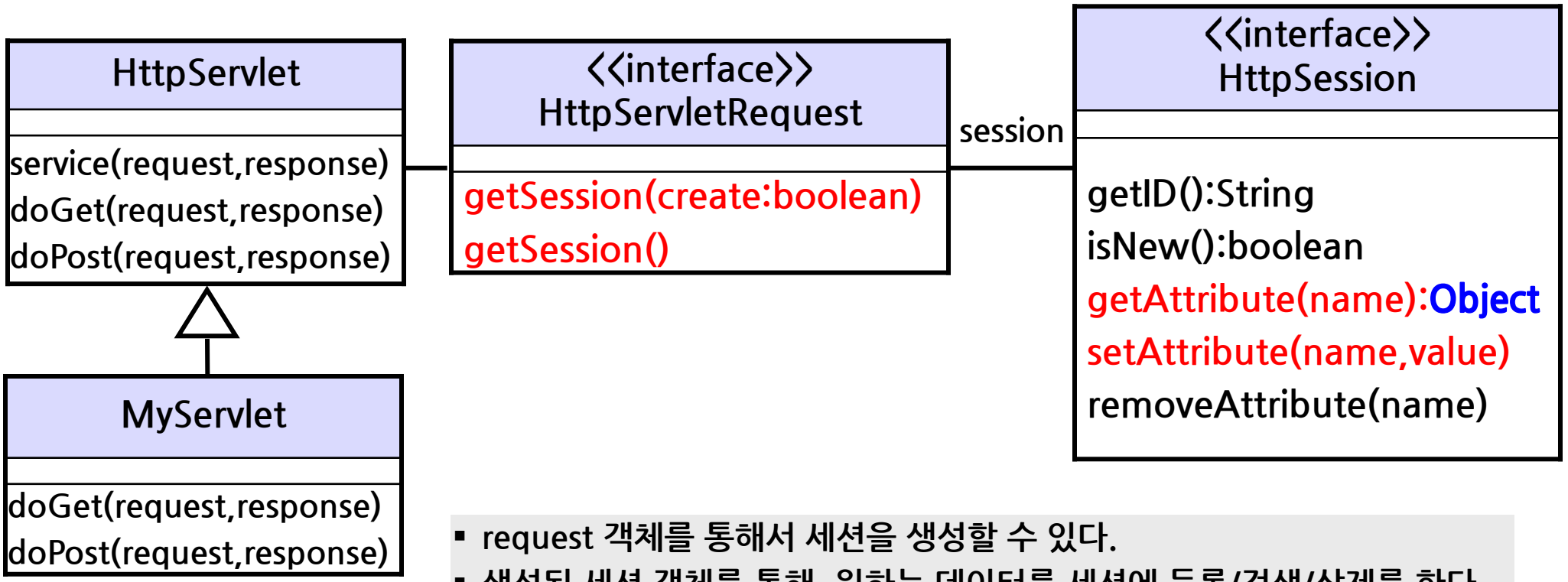
3. User2가 장바구니에 처음으로 상품을 담는다



4. User1이 세션 제거(logout)를 요청하거나, 정해진 시간 안에 다시 요청을 보내지 않는 경우



❖ 세션 관리를 위한 메소드들이 선언된 interface



- request 객체를 통해서 세션을 생성할 수 있다.
- 생성된 세션 객체를 통해, 원하는 데이터를 세션에 등록/검색/삭제를 한다.

- **request.getSession(), request.getSession(true)**
  : 기존 세션이 있다면 그 세션객체를, 없다면 새로운 세션 객체를 생성하여 return한다.
- **request.getSession(false) :**
  기존 세션이 있다면 그 세션 객체를, 없다면 null을 return한다.

public String getId()

- 세션 객체의 session id를 return한다.

public Object getAttribute(String name)

- 세션에서 파라미터 name으로 등록된 정보를 return한다.

public void setAttribute(String name, Object value)

- 세션 객체에 키/값 쌍으로 정보를 저장한다.

public void removeAttribute(String name)

- 세션에서 파라미터 name으로 등록된 정보를 제거한다.

public void invalidate()

- 세션 객체를 무효화한다.

product.html

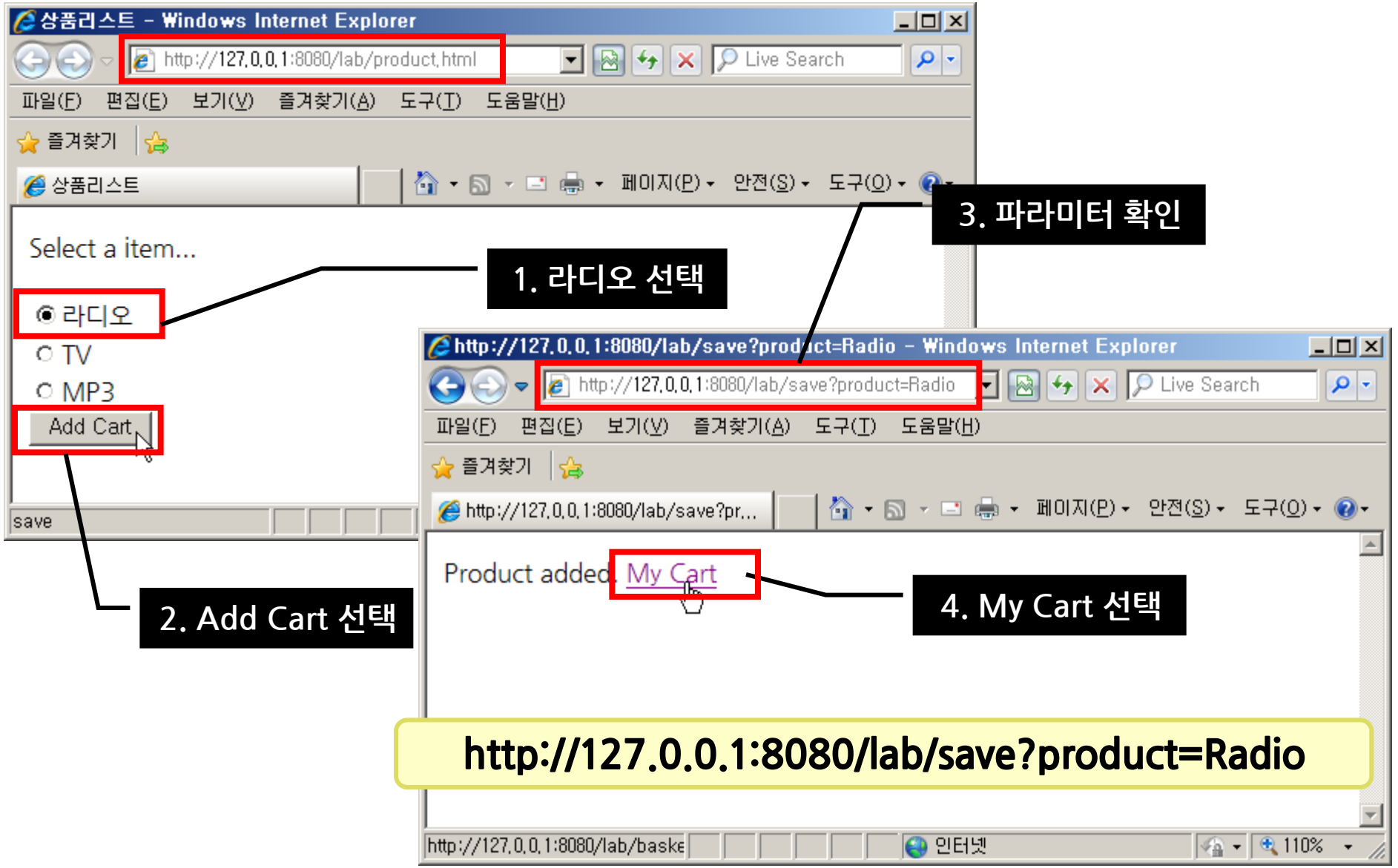
```
<html>
<head>
 <title>상품리스트</title>
</head>
<body>
Select a item...
<form action = "save" method="get">
   <input type="radio" name="product" value="Radio">라디오<br>
   <input type="radio" name="product" value="TV">TV<br>
   <input type="radio" name="product" value="MP3">MP3<br>
   <input type="submit" value="Add Cart">
</form>
</body>
</html>
```

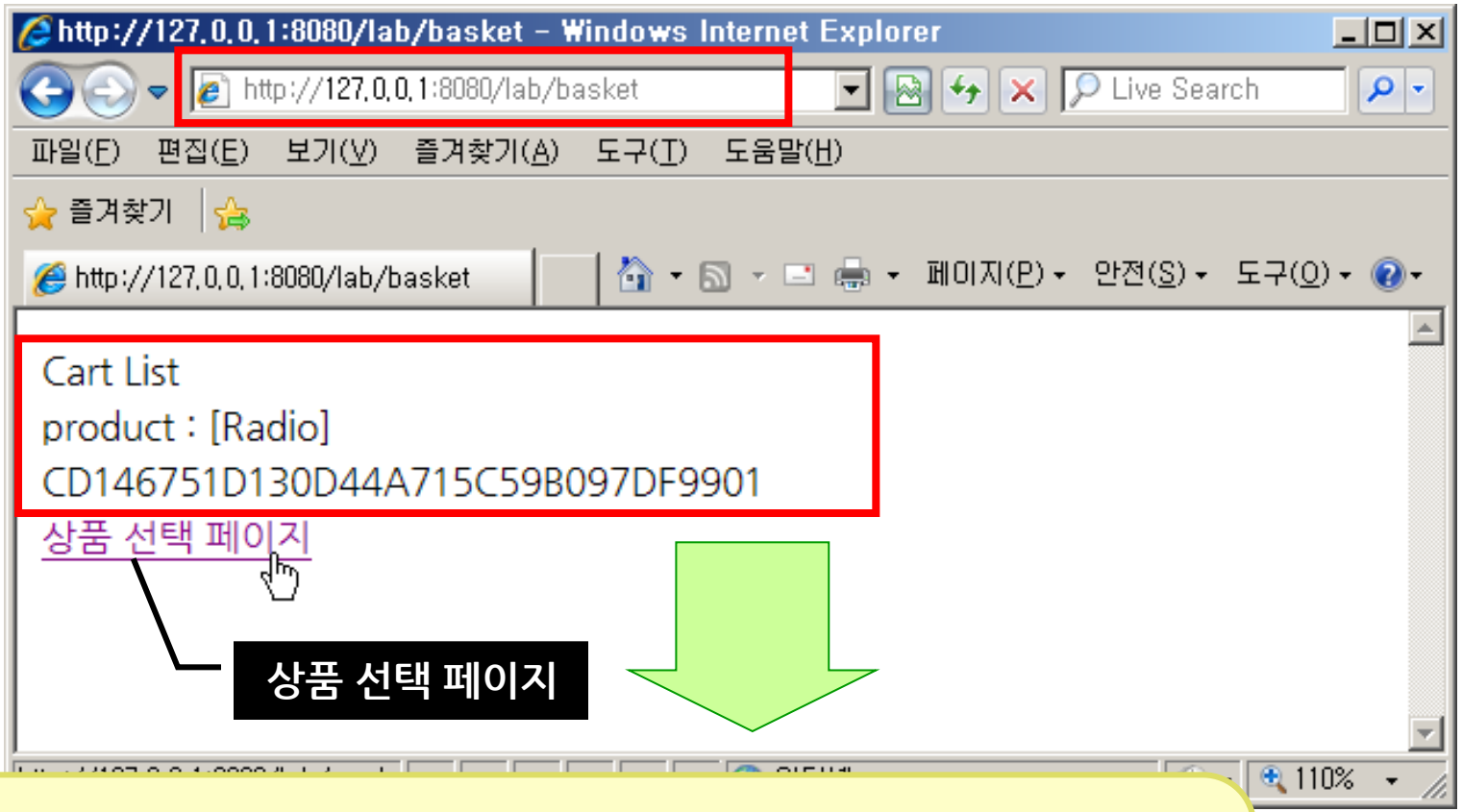
SaveServlet.java, url-pattern : /save

```
public void doGet( HttpServletRequest request, HttpServletResponse response )
                    throws ServletException, IOException {

    response.setContentType("text/html;charset=euc-kr");

    String p = request.getParameter("product");
    HttpSession session = request.getSession();
    ArrayList<String> list = (ArrayList<String>) session.getAttribute("product");
    if (list == null) {
        list = new ArrayList<String>();
        list.add(p);
        session.setAttribute("product", list);
    } else {
        list.add(p);
    }

    PrintWriter out = response.getWriter();
    out.println("<html><body>");
    out.println("Product added.");
    out.println("<a href=basket>My Cart</a>");
    out.println("</body></html>");
    out.close();
}
```

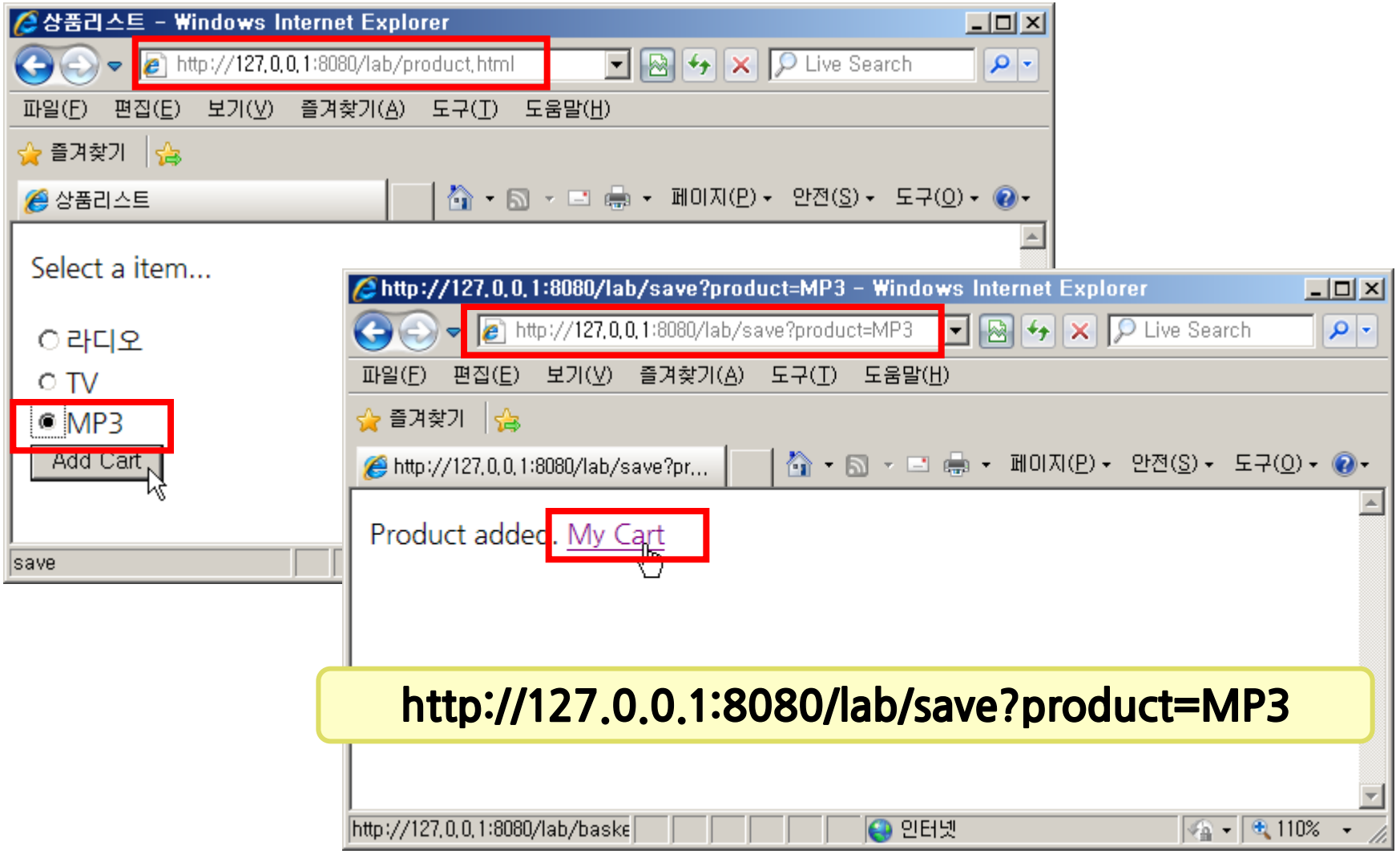
BasketServlet.java, url-pattern : /basket

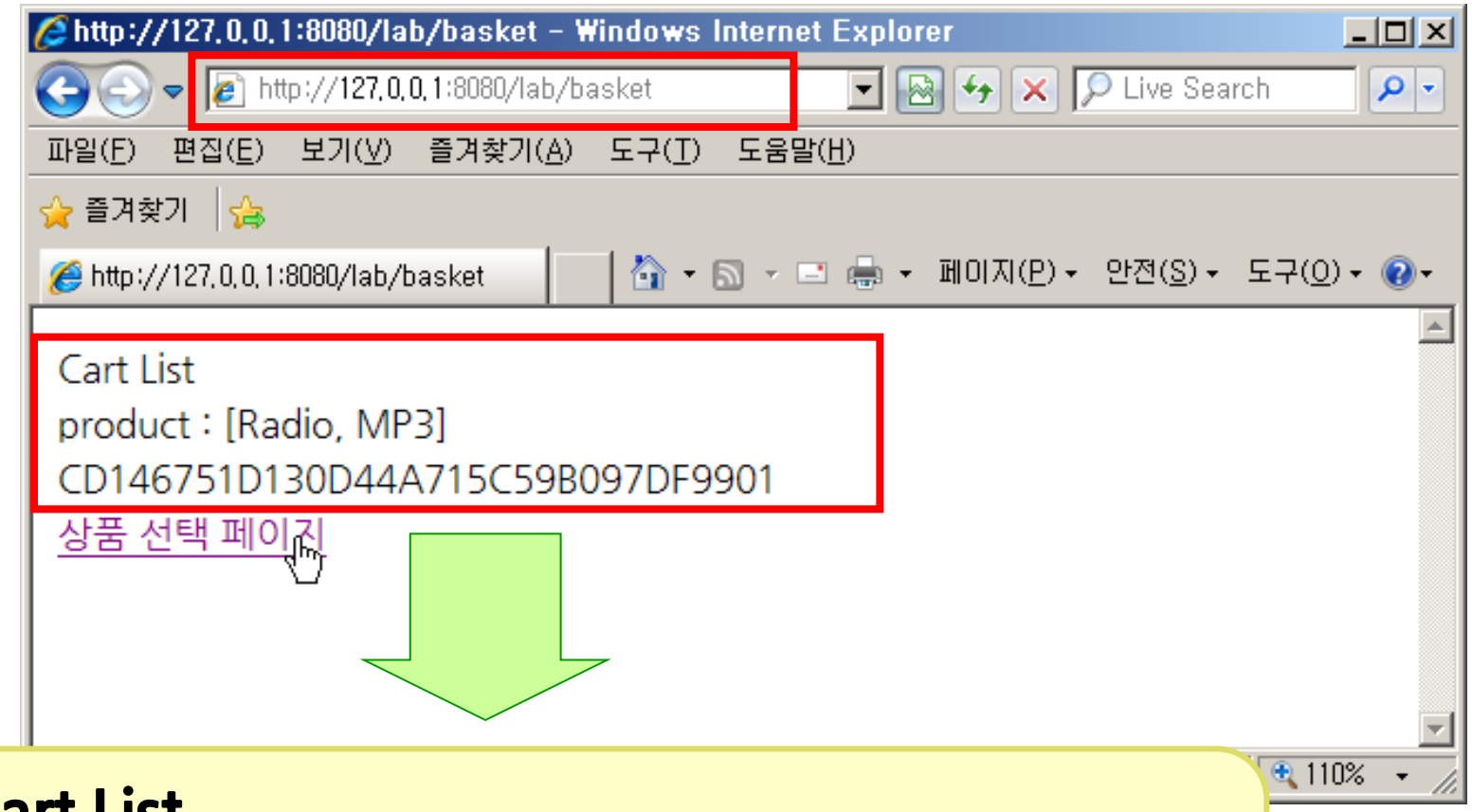
```
public void doGet( HttpServletRequest request, HttpServletResponse response )
            throws ServletException, IOException {
    response.setContentType( "text/html;charset=euc-kr" );

    PrintWriter out = response.getWriter();
    out.println( "<html><body>" );
    out.println( "Cart List<br>" );

    HttpSession session = request.getSession( false );

    if ( session != null ) {
        ArrayList<String> list = (ArrayList<String>)session.getAttribute( "product" );
        out.println( "product : " + list + "<br>" );
        out.println( session.getId() );
    } else {
        out.println( "No Session" );
    }
    out.println( "<br/><a href='product.html'>상품 선택 페이지</a>" );
    out.println( "</body></html>" );
    out.close();
}
```


상품리스트 - Windows Internet Explorer
http://127.0.0.1:8080/lab/product.html
Live Search
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
즐겨찾기
상품리스트
페이지(P) 안전(S) 도구(O)
Select a item...
라디오
TV
MP3
Add Cart
save
1. 라디오 선택
2. Add Cart 선택
3. 파라미터 확인
http://127.0.0.1:8080/lab/save?product=Radio - Windows Internet Explorer
http://127.0.0.1:8080/lab/save?product=Radio
Live Search
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
즐겨찾기
http://127.0.0.1:8080/lab/save?pr...
페이지(P) 안전(S) 도구(O)
Product added. My Cart
http://127.0.0.1:8080/lab/baske
인터넷
110%
4. My Cart 선택
http://127.0.0.1:8080/lab/save?product=Radio



Cart List
product : [Radio]
CD146751D130D44A715C59B097DF9901

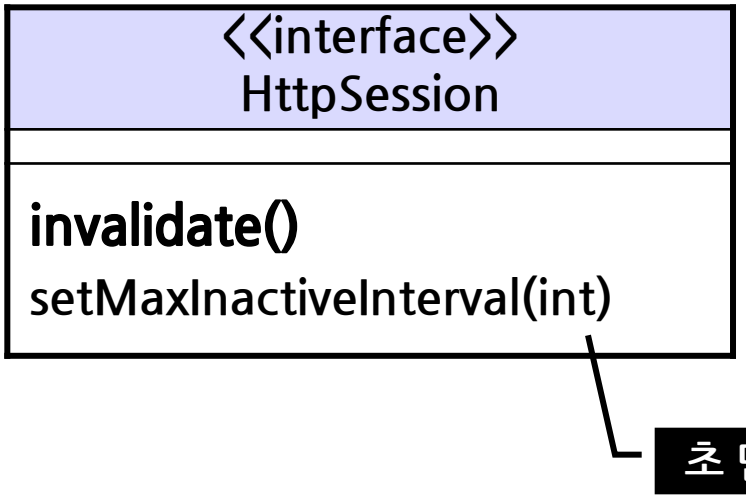

상품리스트 - Windows Internet Explorer
http://127.0.0.1:8080/lab/product.html
Live Search
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
즐겨찾기
상품리스트
페이지(P) 안전(S) 도구(O)
Select a item...
라디오
TV
MP3
Add Cart
save
http://127.0.0.1:8080/lab/save?product=MP3 - Windows Internet Explorer
http://127.0.0.1:8080/lab/save?product=MP3
Live Search
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
즐겨찾기
http://127.0.0.1:8080/lab/save?pr...
페이지(P) 안전(S) 도구(O)
Product added. My Cart
http://127.0.0.1:8080/lab/baske
인터넷
110%
http://127.0.0.1:8080/lab/save?product=MP3



Cart List
product : [Radio, MP3]
CD146751D130D44A715C59B097DF9901

Session 무효화

❖ Servlet에서 세션 무효화




```
HttpSession session = request.getSession(false);

if ( session != null ) {
    session.invalidate();
}
```

❖ DD(web.xml)에서 설정

```
<session-config>
   <session-timeout>10</session-timeout>
</session-config>
```

분 단위

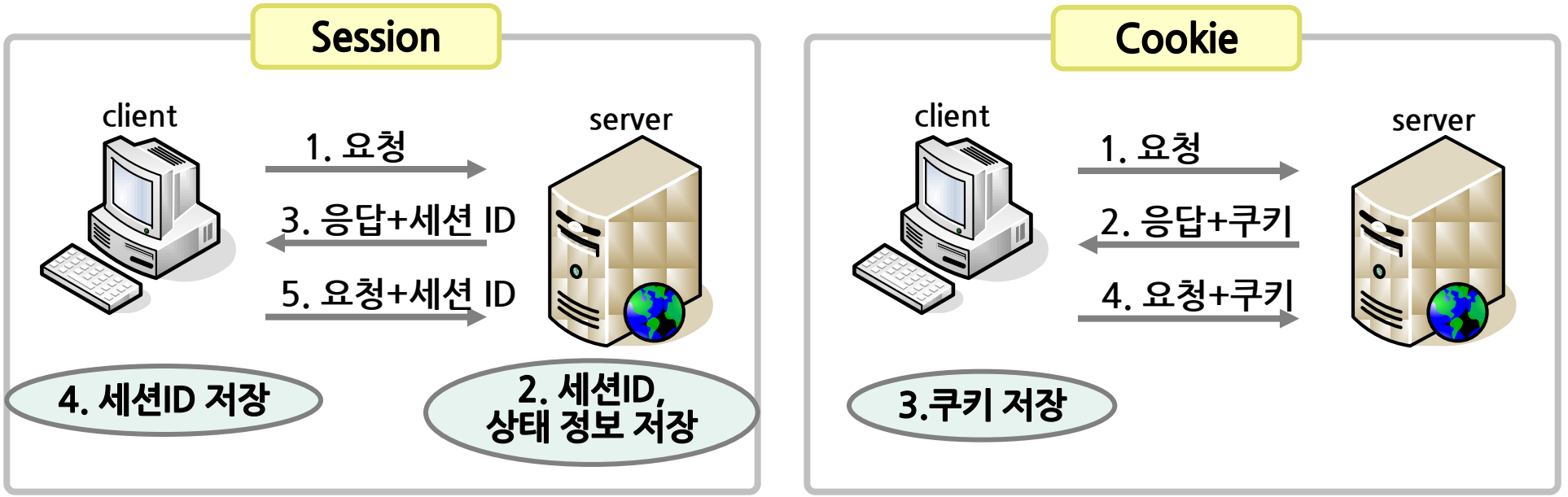
5.1 Session

5.2 Cookie

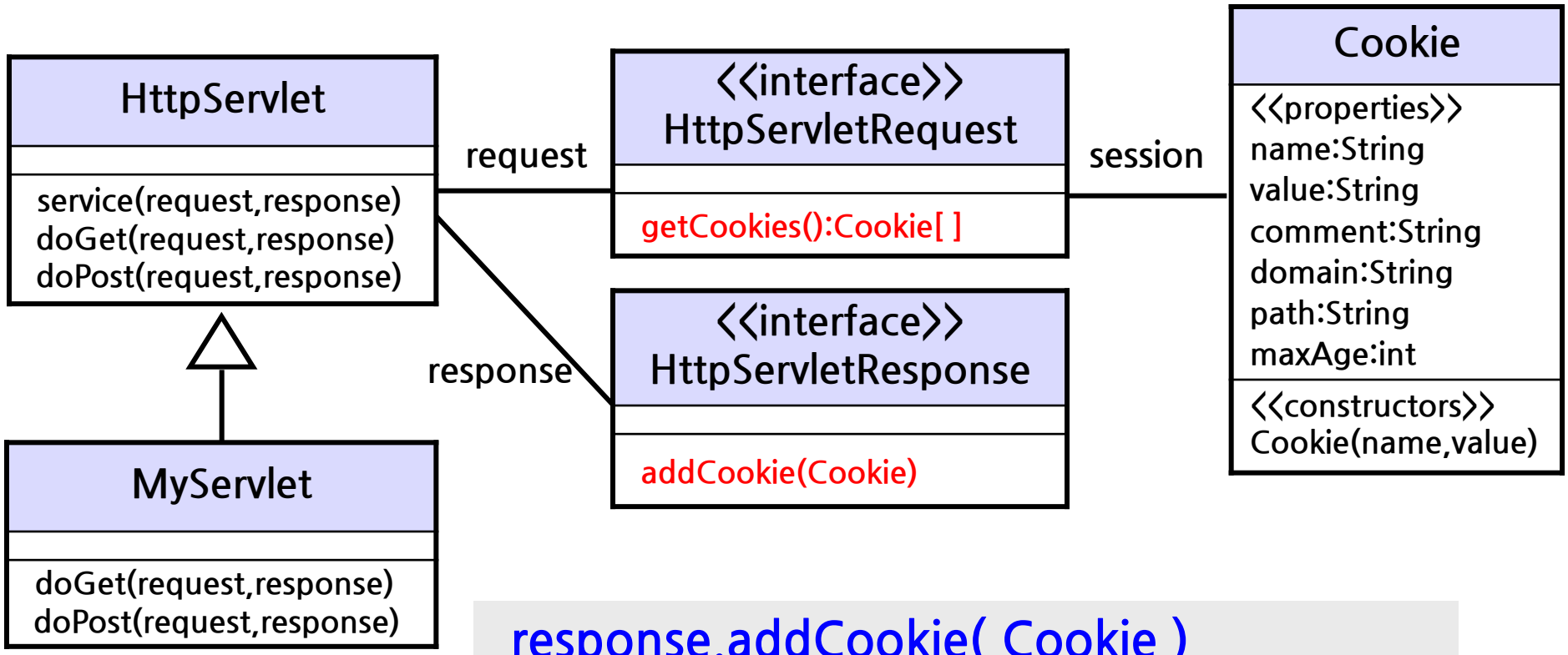
5.3 URL 재작성

❖ Session vs. Cookie

- 세션은 상태 정보를 서버에서 관리하고 세션 ID만 클라이언트에서 관리한다.
- 반면 **쿠키는 모든 상태 정보를 클라이언트에서 관리한다.**
- 서버가 응답 시 보낸 상태 정보를 클라이언트의 브라우저는 텍스트 형태로 사용자 PC에 저장하고, 다음 요청 시 그 내용을 서버에 보낸다.
- Cookie는 지정한 수명에 다다르면, 브라우저에 의해 삭제된다.



❖ Cookie 관리를 위한 메소드들이 선언된 interface



response.addCookie(Cookie)

생성한 Cookie를 응답에 실어 보내는 메소드

request.getCookies()

클라이언트로 부터 얻은 쿠키 정보를 읽어오는 메소드

public Cookie (String name, String value)

- name, value 쌍으로 쿠키를 생성한다.

public String getValue ()

- 현재 쿠키가 갖고 있는 값을 return한다.

public void setMaxAge (int expiry)

- 쿠키를 언제 브라우저가 삭제할지를 초 단위로 지정한다.

productCookie.html

```
<html>
<head>
 <title>상품리스트</title>
</head>
<body>
Select a item...
<form action = "saveCookie" method="get">
   <input type="radio" name="product" value="Radio">라디오<br>
   <input type="radio" name="product" value="TV">TV<br>
   <input type="radio" name="product" value="MP3">MP3<br>
   <input type="submit" value="Add Cart">
</form>
</body>
</html>
```

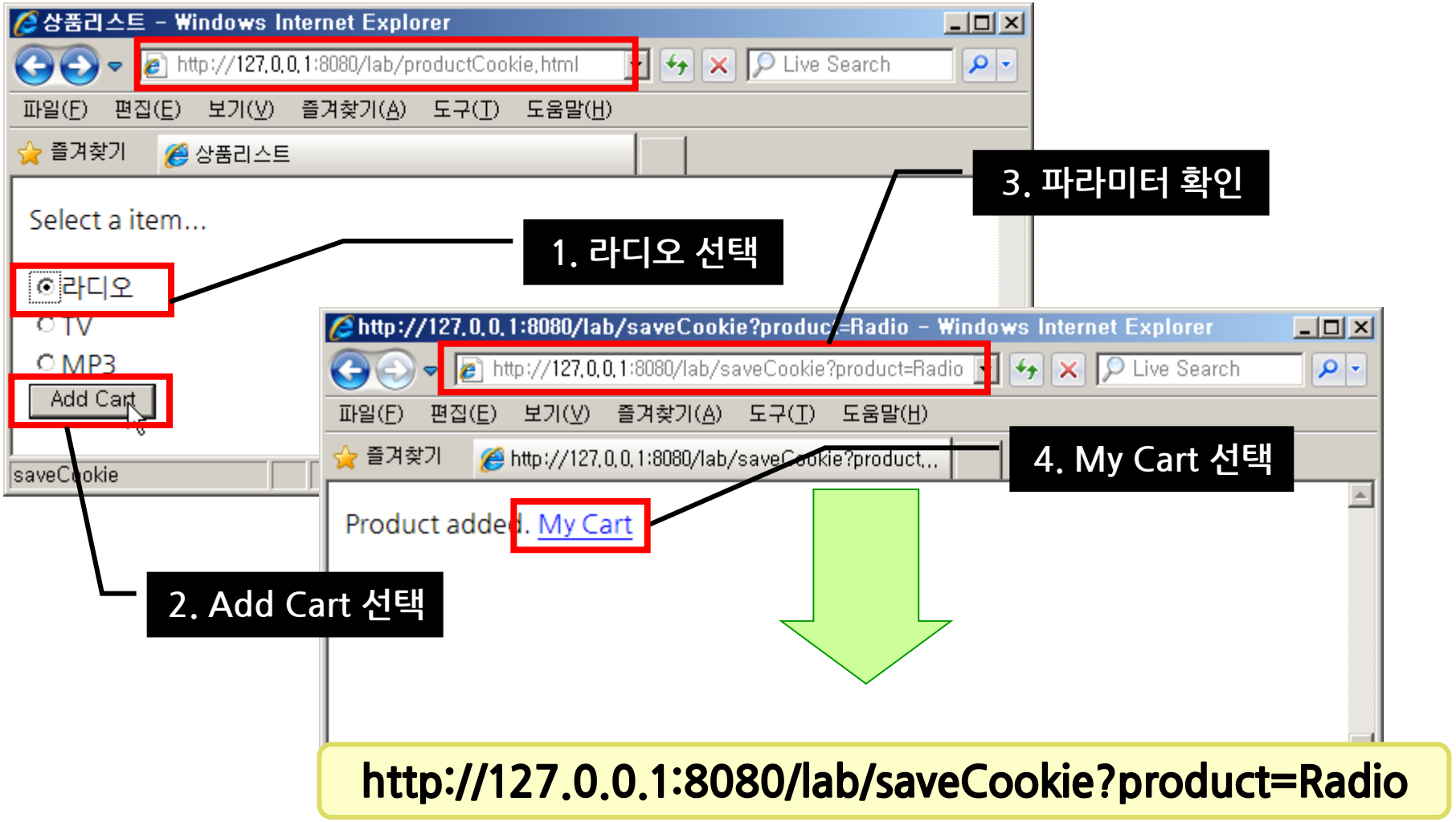
- chap05.cookie.SaveCookieServlet.java
- url-pattern : /saveCookie

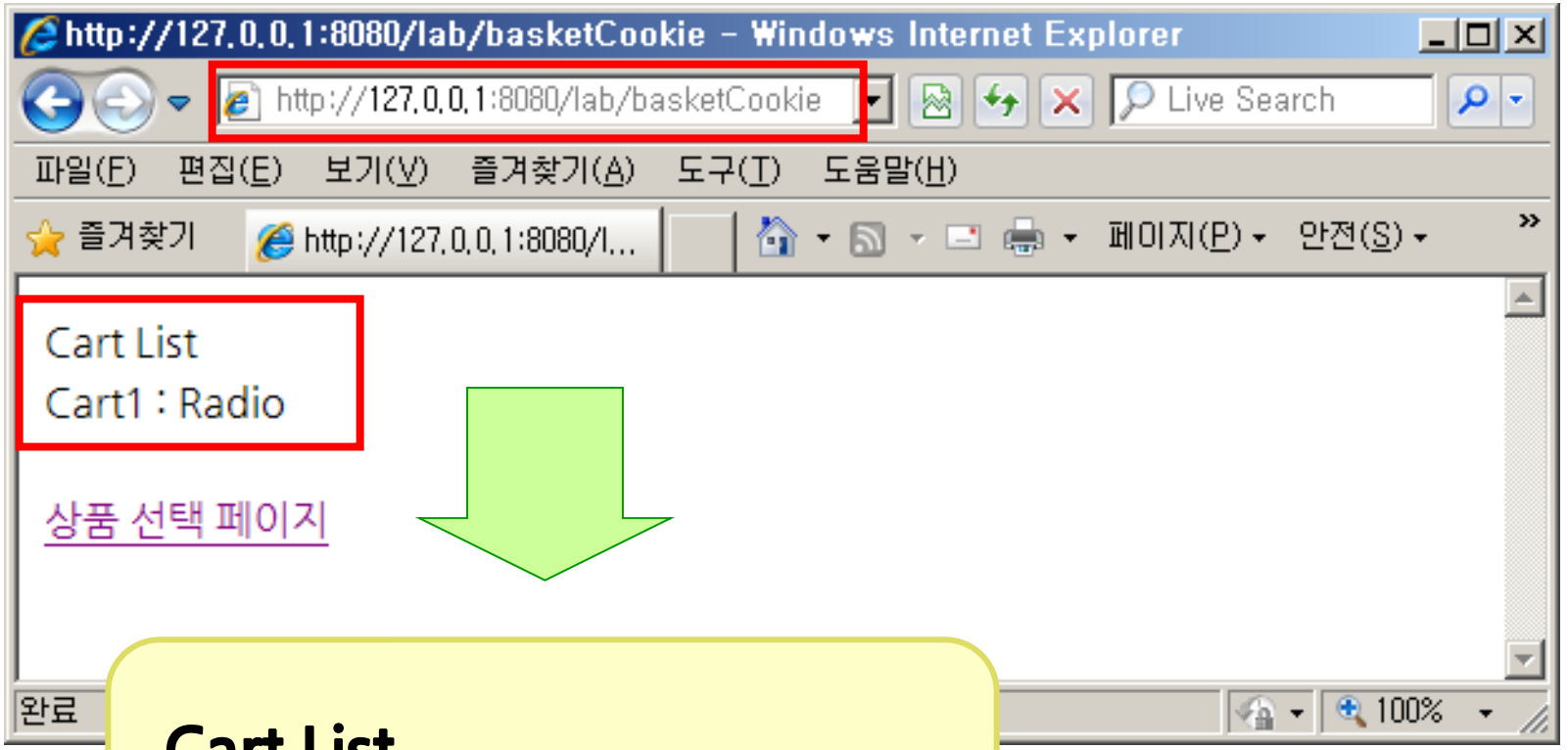
```
public void doGet( HttpServletRequest request, HttpServletResponse response )
                    throws ServletException, IOException {

        response.setContentType( "text/html;charset=euc-kr" );
        String p = request.getParameter( "product" );
        System.out.println( "product : " + p  );
        // 기존 쿠키를 가져온다.
        Cookie[] cookies = request.getCookies();
        Cookie c = null;
        // 쿠키에 unique한 이름을 주고 쿠키를 생성한다.
        if ( cookies == null || cookies.length == 0 ) {
            c = new Cookie( "Cart1", p );
        } else {
            c = new Cookie( "Cart" + (cookies.length + 1), p );
        }
        // 쿠키는 3600초 후에 브라우저에 의해 삭제된다.
        c.setMaxAge(60*60);
        // 응답에 쿠키를 저장한다.
        response.addCookie( c );

        PrintWriter out = response.getWriter();
        out.println( "<html><body>" );
        out.println( "Product added." );
        out.println( "<a href=basketCookie>My Cart</a>" );
        out.println( "</body></html>" );
        out.close();
    }
```

- chap05.cookie.BasketCookieServlet.java
- url-pattern : /basketCookie

```
public void doGet( HttpServletRequest request, HttpServletResponse response )
        throws ServletException, IOException {

    response.setContentType( "text/html;charset=euc-kr" );

    PrintWriter out = response.getWriter();
    out.println( "<html><body>" );
    out.println( "Cart List<br>" );

    // 쿠키를 요청정보에서 가져온다.
    Cookie cookies[] = request.getCookies();
    // 쿠키가 존재하면, 이름/값을 출력한다.
    if ( cookies != null ) {
       for ( int i = 0 ; i < cookies.length ; i++ ) {
          Cookie cookie = cookies[i];
          out.println( cookie.getName() + " : "
                + cookie.getValue()  + "<br>" );
       }
    } else {
       out.println( "물품이 없습니다." );
    }
    out.println( "<br/><a href='productCookie.html'>상품 선택 페이지</a>" );
    out.println( "</body></html>" );
    out.close();
}
```

[실습] Cookie

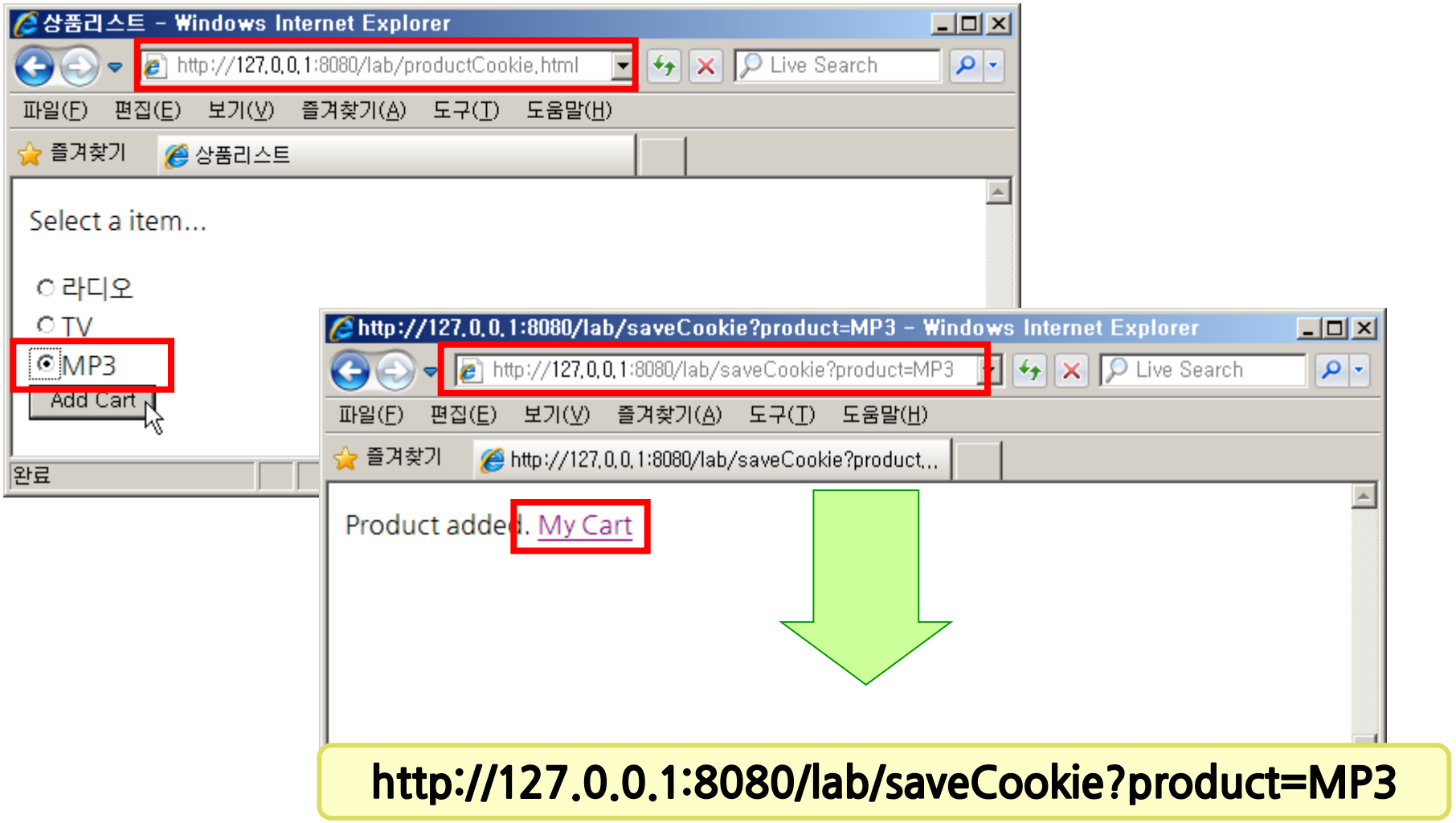

상품리스트 - Windows Internet Explorer
http://127.0.0.1:8080/lab/productCookie.html
Live Search
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
즐겨찾기
상품리스트
Select a item...
라디오
TV
MP3
Add Cart
saveCookie
1. 라디오 선택
2. Add Cart 선택
3. 파라미터 확인
http://127.0.0.1:8080/lab/saveCookie?product=Radio - Windows Internet Explorer
http://127.0.0.1:8080/lab/saveCookie?product=Radio
Live Search
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
즐겨찾기
http://127.0.0.1:8080/lab/saveCookie?product...
Product added. My Cart
4. My Cart 선택
http://127.0.0.1:8080/lab/saveCookie?product=Radio

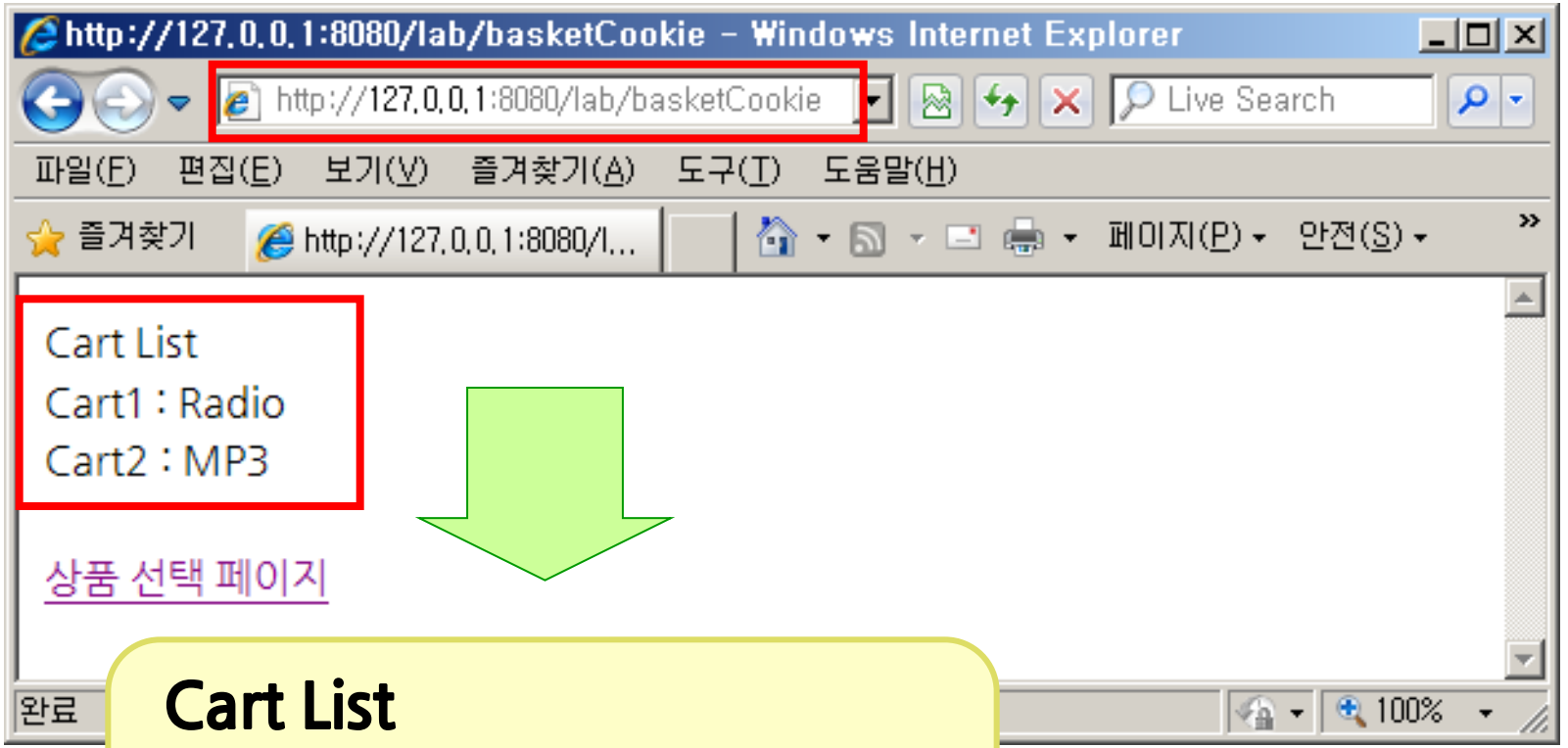


Cart List

Cart1 : Radio

[실습] Cookie

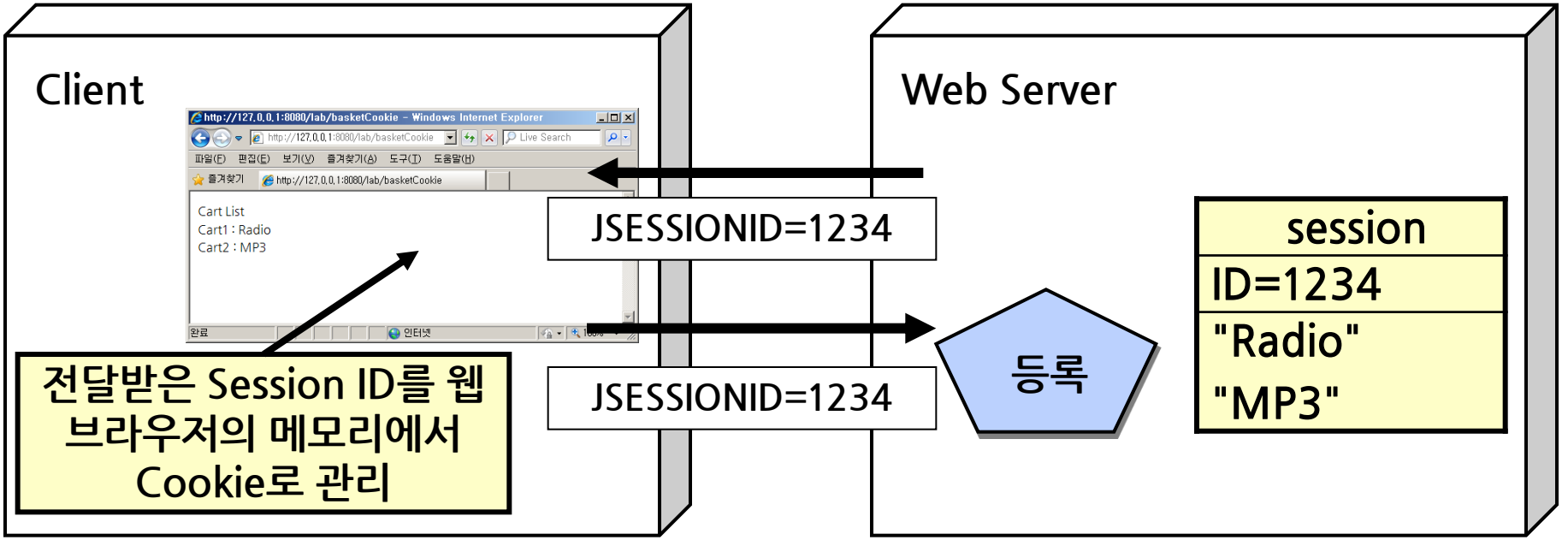

상품리스트 - Windows Internet Explorer
http://127.0.0.1:8080/lab/productCookie.html
Live Search
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
즐겨찾기
상품리스트
Select a item...
라디오
TV
MP3
Add Cart
완료
http://127.0.0.1:8080/lab/saveCookie?product=MP3 - Windows Internet Explorer
http://127.0.0.1:8080/lab/saveCookie?product=MP3
Live Search
파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)
즐겨찾기
http://127.0.0.1:8080/lab/saveCookie?product...
Product added. My Cart
http://127.0.0.1:8080/lab/saveCookie?product=MP3



Cart List
Cart1 : Radio
Cart2 : MP3

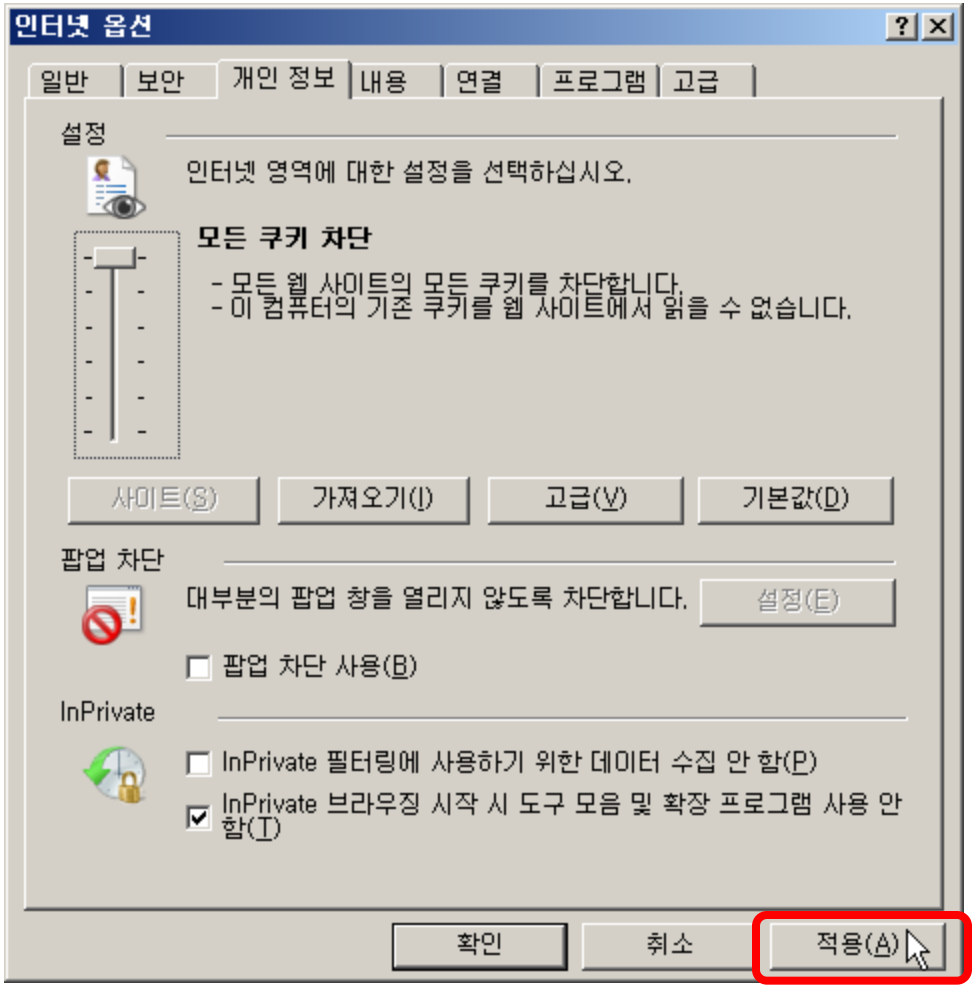
Session 관리를 위한 Cookie 사용

- Session ID 는 쿠키를 통해 관리된다.
- Session이 활성화되어 있으면 Client의 모든 HTTP 요청은 Client에 저장된 Session ID 쿠키를 포함한다.



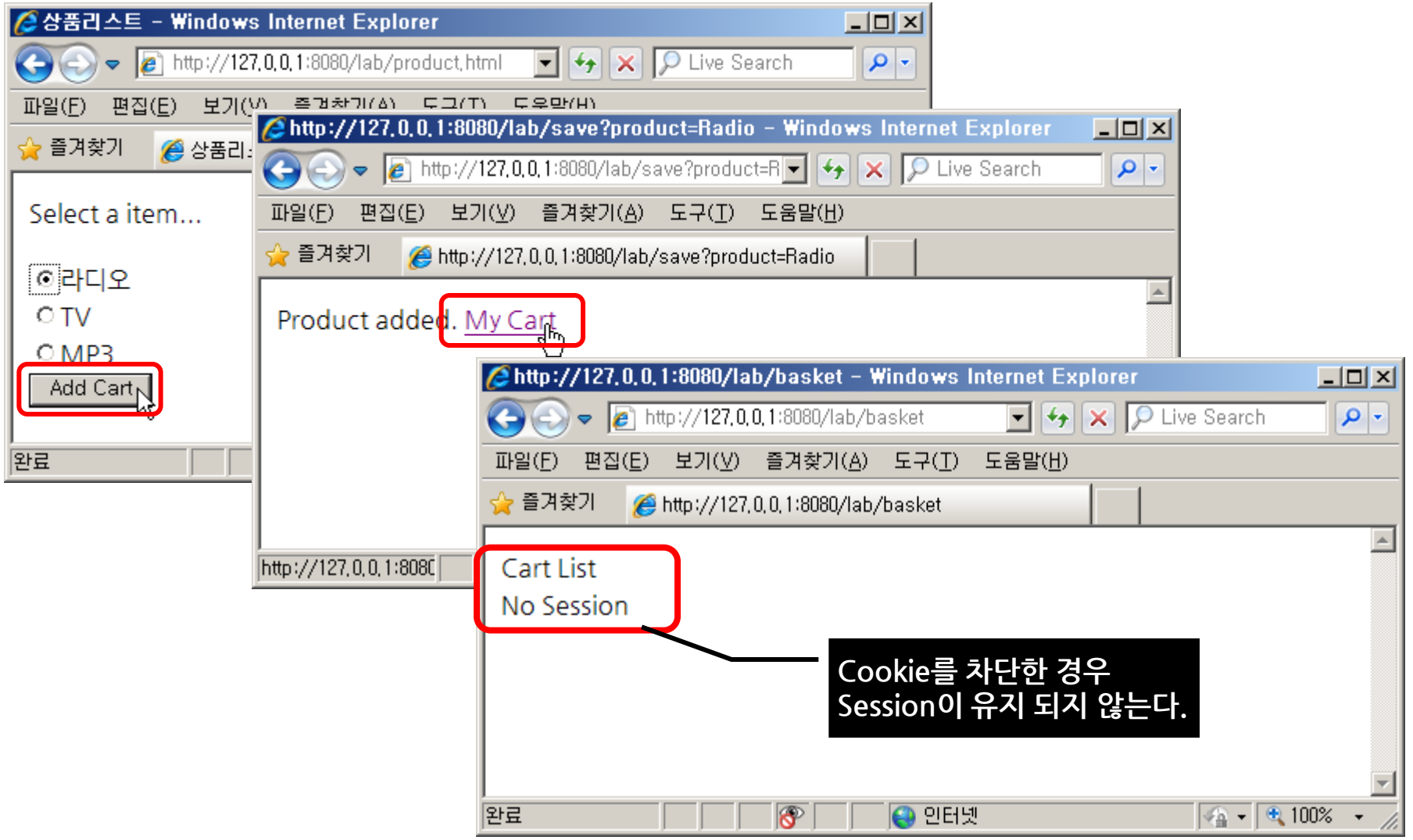
[실습] 모든 쿠키 차단 실행

❖ 모든 쿠키 차단 (도구 → 인터넷 옵션 → 개인 정보)



[실습] 모든 Cookie 차단 후 실행

❑ http://127.0.0.1:8080/lab/product.html 로 접속한 후 테스트한다.



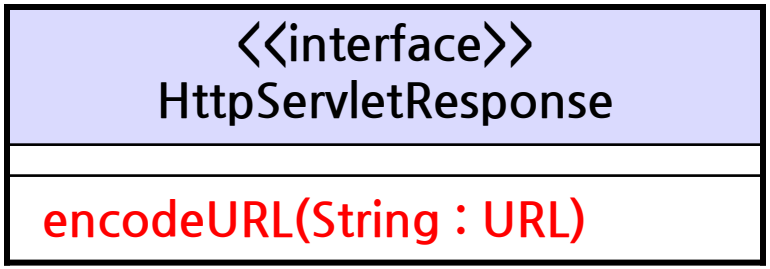
5.1 Session

5.2 Cookie

5.3 URL 재작성

❖ URL 재작성

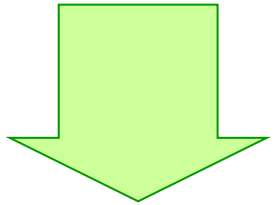
- HttpSession 객체의 session id를 URL에 직접 담아서 보내는 방법으로 Cookie를 사용 못하는 브라우저에서 주로 사용한다.
- HttpServletResponse 객체의 encodeURL() 를 이용한다. → response.encodeURL();
- 즉, Session ID를 Cookie로 하는 것이 아니라, URL에 추가하는 것이다.



❑ SaveServlet.java 수정

SaveServlet.java 수정 전

```
out.println( "<a href=basket>My Cart</a>" );
```

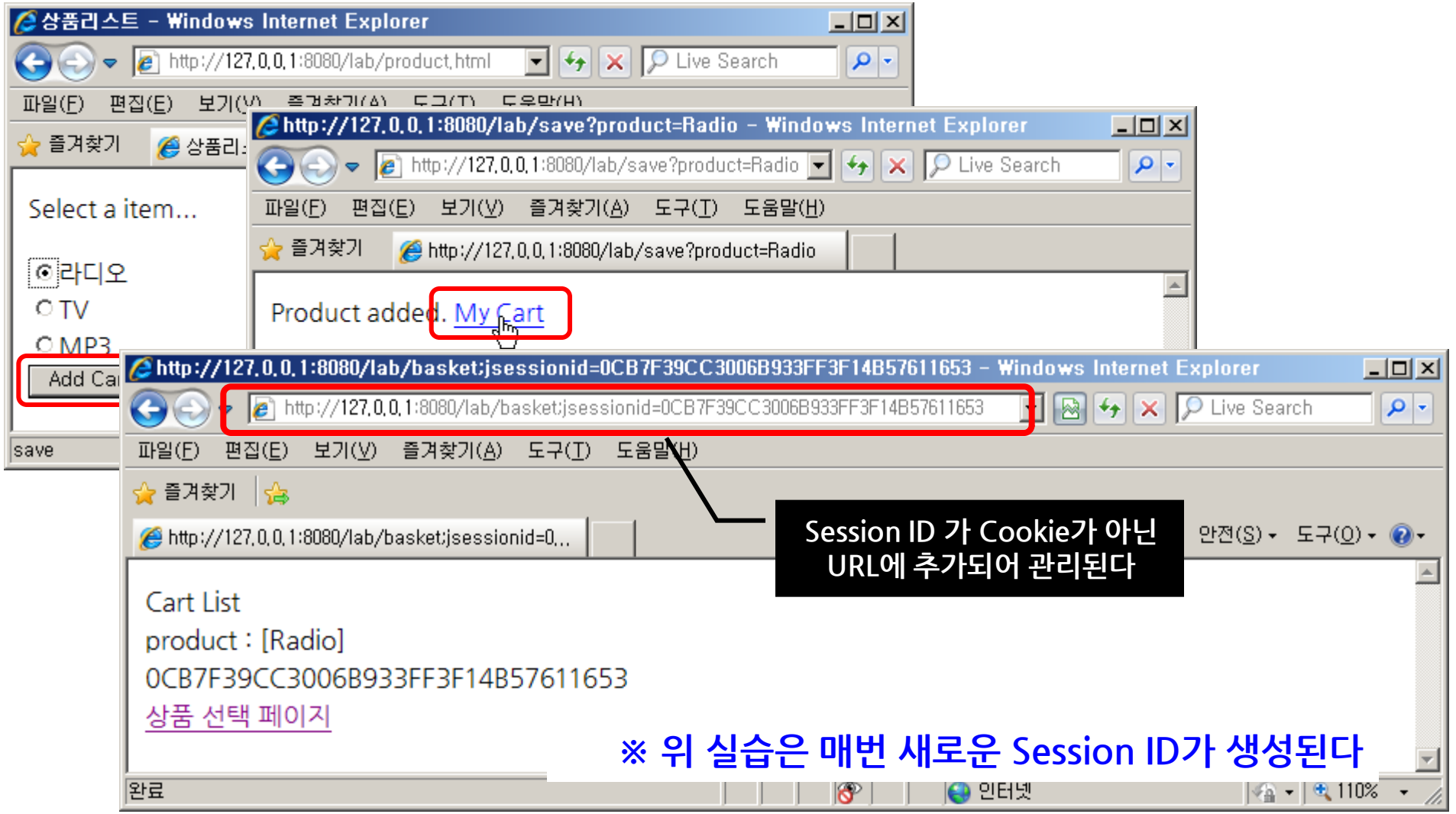


SaveServlet.java 수정 후

```
out.println( "<a href=" + response.encodeURL("basket") + ">My Cart</a>" );
```

[실습] product.html 실행

❑ http://127.0.0.1:8080/lab/product.html 로 접속한 후 테스트한다.



❖ 쿠키 차단 해제 (도구 → 인터넷 옵션 → 개인 정보)