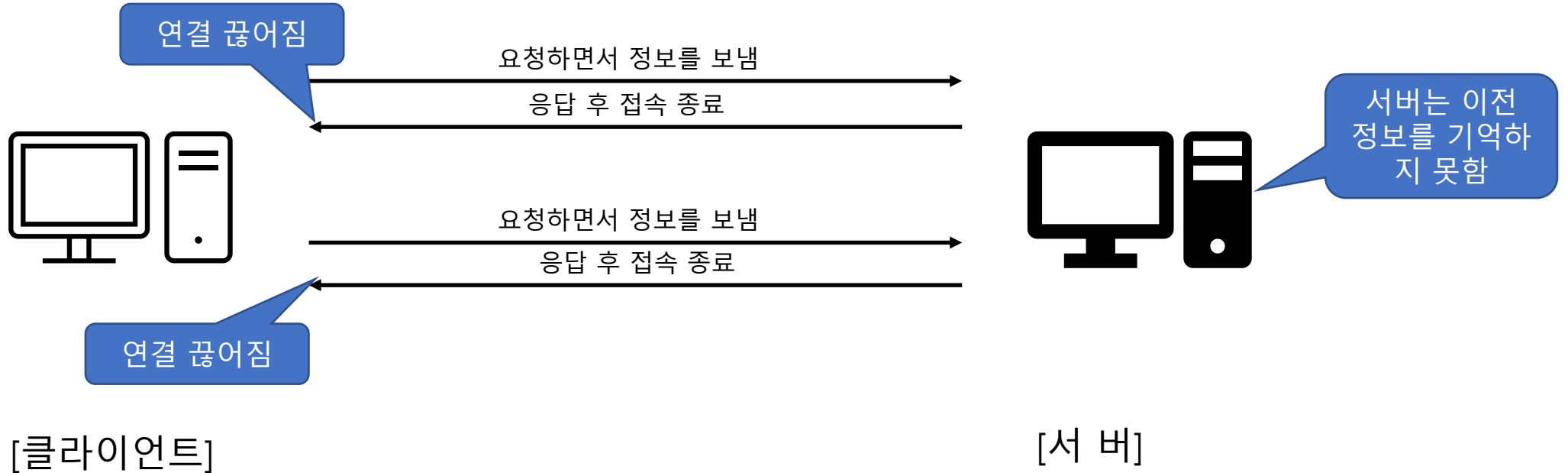


05강 쿠키와 세션

쿠키(cookie)

- 웹 서비스를 제공하는 핵심 프로토콜 HTTP는 비연결 지향형 통신 프로토콜이다.
- 특징은 클라이언트가 요청을 하면 서버는 응답을 한 후 연결을 끊어버린다.
- 클라이언트가 어떤 정보를 요청할 때마다 서버는 이전 정보는 기억하지 않고 새로운 연결을 생성한 후 응답을 진행한다.



쿠키(cookie)

- HTTP가 비연결형 통신 프로토콜이기 때문에 로그인 상태를 유지한다던가, 장바구니에 담은 내용을 유지한다거나 하는 일이 불가능하다. 왜냐하면 웹 브라우저는 현재의 상태 정보를 기억하고 있지 않기 때문이다.
- 이런 단점 극복 하기 위해서 쿠키라는 기술을 사용한다.
- 쿠키은 기본적으로 클라이언트의 컴퓨터에 파일 형태로 저장을 하는 정보이다.
- 사용자 정보를 지속적으로 유지하기 위해서 사용한다.
- 클라이언트의 일정 정보를 클라이언트 하드 디스크에 저장해 두었다가 웹 페이지를 요청하면 저장해둔 쿠키에 대한 정보를 서버에 넘겨 주어 해당 정보를 사용할 수 있게 한다.
- 장점은 서버의 부하를 감소 시키면서도 사용자의 정보를 적절하게 활용할 수 있다는점
- 단점은 보안에 대단히 취약하다는 점이다.

쿠키(cookie)

- 쿠키는 기본적으로 4KB로 크기가 제한 되어있고 300개의 데이터 정보 배열을 저장할 수 있다.
- 오래된 정보를 자동으로 삭제가 된다.
- 쿠키의 기본 사용방법
 - 쿠키 정보를 저장하는 방법
 - 쿠키 객체를 생성한다.
 - 쿠키에 속성 값을 설정한다.
 - response 객체에 addCookie()메소드를 호출해서 쿠키를 추가한다.

쿠키 객체 생성하기

```
Cookie info =new Cookie("쿠키이름","쿠키의 값");
```

쿠키(cookie)

- 쿠키 관련 메소드

메소드	설명
void setComment(String)	쿠키에 대한 설명을 설정
void setDomain(String)	쿠키의 유효한 도메인을 설정
void setMaxAge(int)	쿠키의 유효한 기간을 설정
void setPath(String)	쿠키의 유효한 디렉토리를 설정
void setSecure(String)	쿠키의 보안을 설정
void setValue()String)	쿠키의 값을 설정

쿠키(cookie)

- 쿠키 관련 메소드

메소드	설명
String getComment()	쿠키에 대한 설명을 알려줌
String getDomain()	쿠키의 유효한 도메인 정보를 알려줌
int getMaxAge()	쿠키가 사용할 수 있는 기간에 대한 정보를 알려줌
String getName()	쿠키의 이름을 알려줌
String getPath()	쿠키의 유효한 디렉토리 정보를 알려줌
Boolean getSecure()	쿠키의 보안이 어떻게 설정되어 있는지를 알려줌
String getValue()	쿠키에 설정된 값을 알려줌
int getVersion()	쿠키의 버전을 알려줌

- 유효시간은 초단위로 숫자를 입력한다. 예를 들어 365일이라면 $365 \times 24 \times 60 \times 60$ 을 입력하면 된다.

쿠키(cookie)

- 이렇게 생성된 쿠키를 클라이언트 측에 보내 저장해 두어야만 나중에 다시 가져와서 사용할 수 있다.

- 클라이언트에 보내는 법

```
response.addCookie(info);
```

- 예제1

- 크롬은 쿠키도 암호화 해서 저장하기 때문에 쿠키내용을 보려면 크롬 브라우저에서 직접 확인한다.

쿠키(cookie)

- 반대로 저장된 쿠키를 다시 서버가 얻어오는 방법을 알아보자
 - 쿠키 객체를 얻어온다.
 - 쿠키 객체에 설정된 값을 알아낸다.
 - 쿠키가 여러 개이므로 배열형태로 가져온다.

```
Cookie[] cookies = request.getCookies();
```

- 쿠키 하나하나의 이름과 값을 알아 올리려면 반복문을 사용한다.

예제2

쿠키(cookie)

- 저장된 쿠키를 삭제하기 위해서는 유효시간을 0으로 만들면 해당 쿠키가 삭제가 된다.

```
cookie.setMaxAge(0);
```

예제3

세션(session)

- 다른 페이지로 이동하더라도 그 정보를 유지하는 상태를 만들기 위해 특별한 기법이 필요하다.
- 다만 쿠키는 쿠키가 저장된 디렉터리만 검색하면 얼마든지 정보를 유출할 수 있으므로 보안에 대단히 취약하다. 게다가 저장할 수 있는 데이터도 1.2MB정도로 한계가 있다.
- 이런 단점을 극복하기 위해 나온 상태 유지 기법중 하나는 세션이다.
 - 세션의 장점
 - 서버에서만 접근이 가능하므로 보안 유지가 강력하다.
 - 저장할 수 있는 데이터의 한계가 없다 (서버의 메모리가 허용하는 한 저장 가능)

세션(session)

- 세션이란 서버상에 존재하는 객체로서 브라우저당 하나를 할당한다.
- HTTP는 기본적으로 비연결 상태 프로토콜이므로 이전의 정보를 저장하고 있지 않는 데 서버의 세션 객체가 해당 브라우저와의 정보를 기억했다가 다음 요청이 저장된 정보가 있다면 객체 내에서 꺼내서 사용할 수 있게 한다.
- 대표적인 예제 로그인 인증이다.

세션(session)

- 세션 사용하기 - 내장 객체 session을 사용한다.

메소드	설명
Object getAttribute(String name)	이름에 해당하는 객체 값을 가져온다. 없을 경우 null반환, 반환 타입은 Object이므로 반드시 형변환이 필요
Enumeration getAttributeNames()	세션에서 모든 객체의 이름을 Enumeration형으로 얻어준다.
long getCreationTime()	세션이 만들어진 시간을 반환
String getId()	해당 세션을 가리키는 고유 id값을 String형으로 변환
long getLastAccessedTime()	해당 세션이 클라이언트가 마지막으로 요청을 보낸 시간을 long 형으로 반환
int getMaxInactiveInterval()	사용자가 다음 요청을 보낼 때까지 세션을 유지하는 최대 시간 (초단위)을 반환(기본 30분, -1입력시 무한대)
Boolean isNew()	해당 세션이 처음 생성되었다면 true, 이전에 생성된 세션이라면 false를 반환

세션(session)

- 세션 사용하기 - 내장 객체 session을 사용한다.

메소드	설명
void removeAttribute(String name)	지정된 이름에 해당하는 객체를 세션에서 제거한다.
void setAttribute(String name, Object value)	세션에 지정된 이름에 객체를 추가한다.
void setMaxInactiveInterval(int interval)	사용자가 다음 요청을 보낼 때 까지 세션을 유지하는 최대 시간 (초단위)을 설정한다. 이 시간을 넘기면 서블릿 컨테이너는 세션을 종료한다.
void invalidate()	해당 세션을 없애고 세션에 속한 값을 없앤다.

세션(session)

- 세션 사용하기
- 상태 유지를 위해서는 세션에 값을 저장한 후 이것을 가져와서 사용한다.
- 세션에 값을 저장하기

```
session.setAttribute(String name, Object value);
```

- 세션에서 값을 얻어오기

```
[문자열] (String) session.getAttribute(String name);  
[숫자] (Integer) session.getAttribute(String name);
```

- 예제 4

세션(session)

- 세션 사용하기
- 세션에 저장된 속성을 모를 때 한꺼번에 가져오는 법

```
session.setAttributeNames();
```

- 위 메소드는 모든 세션은 가져와서 Enumeration 객체로 얻어온다.
- Enumeration객체는 여러 개의 데이터를 집합처럼 관리하는 인터페이스이다.
- 두가지 메소드를 가지는데

메소드	설명
Boolean hasMoreElements()	Enumeration 객체에 뽑아 올 데이터가 있는지를 검사하여 있다면 true, 없다면 false를 반환
E nextElements()	다음 아이템이 존재한다는 전제 아래에 아이템을 하나씩 뽑아 오는 역할을 한다.

세션(session)

- 세션 사용하기
- 세션에 저장된 속성을 모를 때 한꺼번에 가져오는 법

```
session.setAttributeNames();
```

- 위 메소드는 모든 세션은 가져와서 Enumeration 객체로 얻어온다.
- Enumeration객체는 여러 개의 데이터를 집합처럼 관리하는 인터페이스이다.
- 두가지 메소드를 가지는데

메소드	설명
Boolean hasMoreElements()	Enumeration 객체에 뽑아 올 데이터가 있는지를 검사하여 있다면 true, 없다면 false를 반환
E nextElements()	다음 아이템이 존재한다는 전제 아래에 아이템을 하나씩 뽑아 오는 역할을 한다.

세션(session)

- 세션을 이용한 로그인 처리 예제

로그인 페이지에서
회원 정보를 입력한 후
확인버튼 클릭

