

# Django 08 Authentication System 01

# • INDEX

---

- Cookie & Session
  - HTTP
  - 쿠키
  - 세션
- Django Authentication System
- Custom User model
  - User model 대체하기
- Login
- Logout
- Template with Authentication data

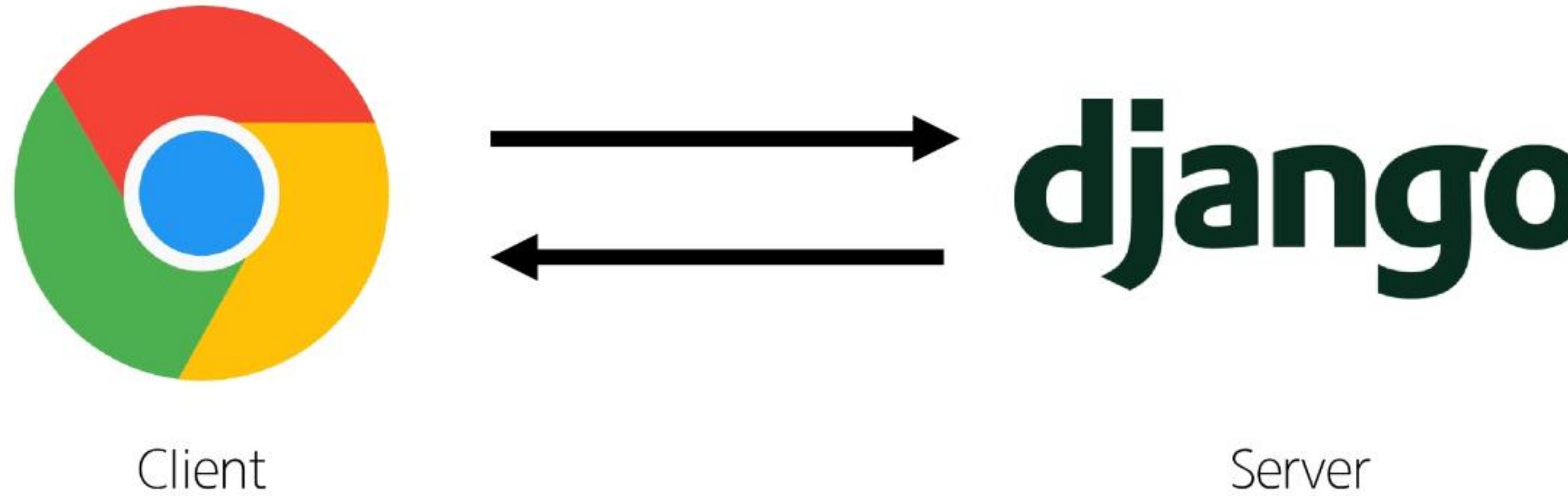
# • INDEX

---

- 참고
  - 쿠키의 수명
  - 쿠키와 보안
  - Django에서의 세션 관리
  - AuthenticationForm 내부 코드
  - AbstractUser class
  - User 모델 대체하기 Tip

# Cookie & Session

**HTTP**



**우리가 웹 페이지를 둘러볼 때  
우리는 서버와 서로 연결되어 있는 상태가 아니다.**

# HTTP

---

HTML 문서와 같은 리소스들을 가져올 수 있도록 해주는 규약  
웹(WWW)에서 이루어지는 모든 데이터 교환의 기초

## HTTP 특징

1. 비 연결 지향(connectionless)
  - 서버는 요청에 대한 응답을 보낸 후 연결을 끊음
2. 무상태(stateless)
  - 연결을 끊는 순간 클라이언트와 서버 간의 통신이 끝나며 상태 정보가 유지되지 않음

## 상태가 없다는 것은..

- 장바구니에 담은 상품을 유지할 수 없음
- 로그인 상태를 유지할 수 없음

쿠키

# 쿠키(Cookie)

---

서버가 사용자의 웹 브라우저에 전송하는 작은 데이터 조각

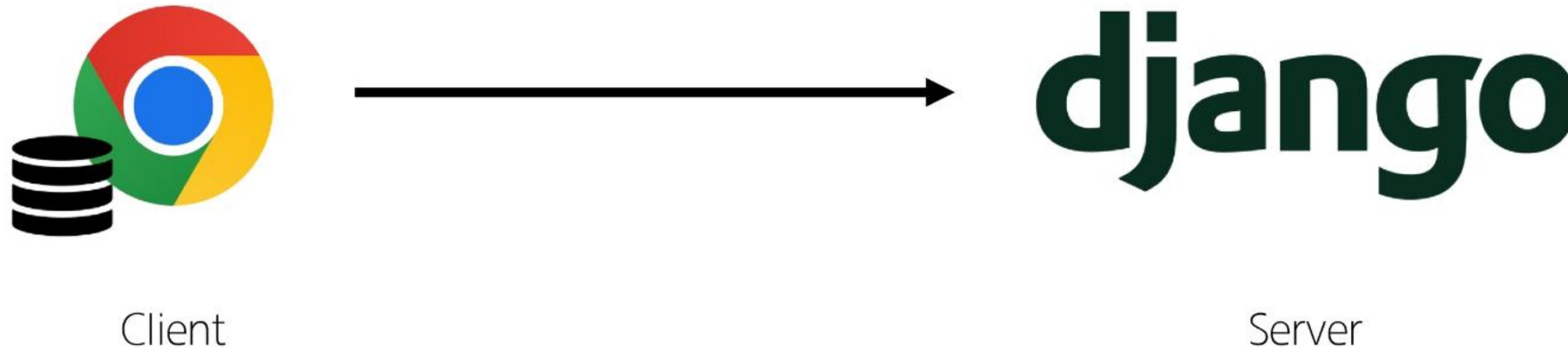
# 쿠키(Cookie)

-----  
서버가 사용자의 웹 브라우저에 전송하는 작은 데이터 조각

- 서버가 제공하여 클라이언트 측에서 저장되는 작은 데이터 파일
- 사용자 인증, 추적, 상태 유지 등에 사용되는 데이터 저장 방식

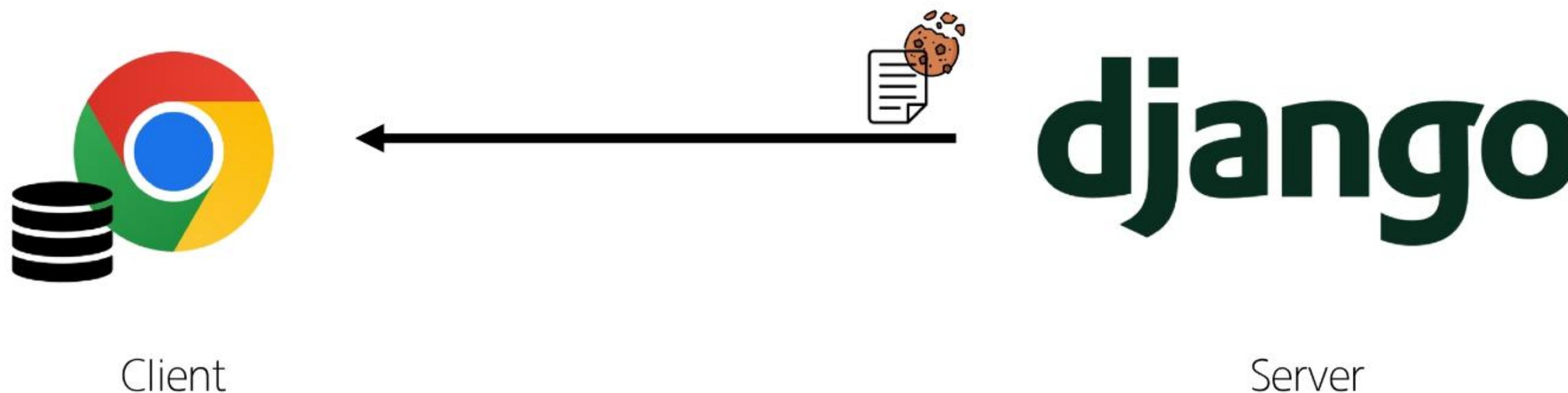
## 쿠키 동작 예시 (1/6)

1. 브라우저가 웹 서버에 웹 페이지를 요청



## 쿠키 동작 예시 (2/6)

2. 웹 서버는 요청된 페이지와 함께 쿠키를 포함한 응답을 브라우저에게 전송



## 쿠키 동작 예시 (3/6)

3. 브라우저는 받은 쿠키를 저장소에 저장  
쿠키의 속성(만료 시간, 도메인, 주소 등)도 함께 저장됨



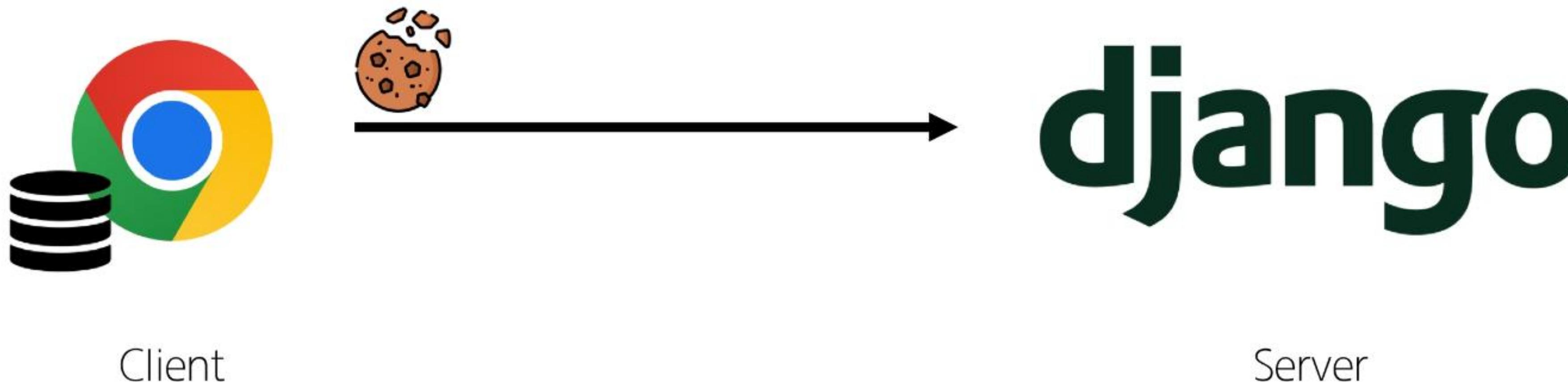
Client

**django**

Server

## 쿠키 동작 예시 (4/6)

4. 이후 브라우저가 같은 웹 서버에 웹 페이지를 요청할 때,  
저장된 쿠키 중 해당 요청에 적용 가능한 쿠키를 포함하여 함께 전송



## 쿠키 동작 예시 (5/6)

5. 웹 서버는 받은 쿠키 정보를 확인하고, 필요에 따라 사용자 식별, 세션 관리 등을 수행



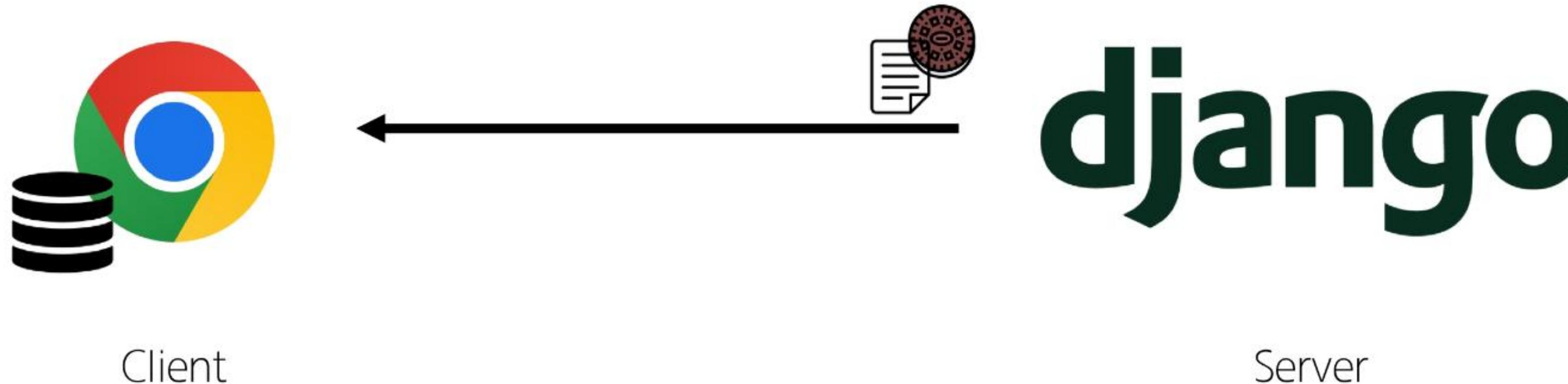
Client



Server

## 쿠키 동작 예시 (6/6)

6. 웹 서버는 요청에 대한 응답을 보내며, 필요한 경우 새로운 쿠키를 설정하거나 기존 쿠키를 수정할 수 있음



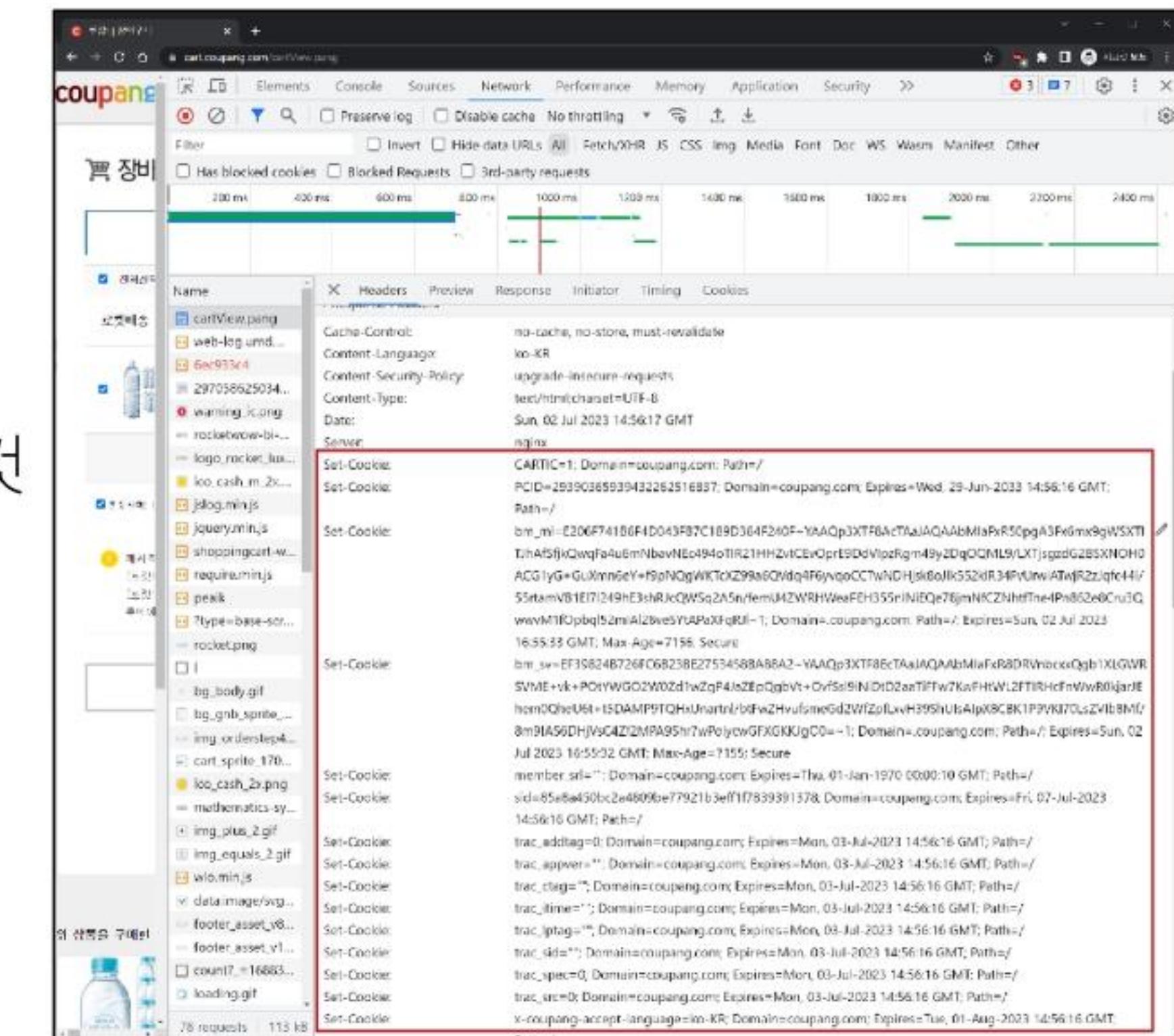
# 쿠키를 이용한 장바구니 예시 (1/5)

장바구니에 상품 담기

The screenshot shows the Coupang shopping cart interface. At the top, the Coupang logo is visible. Below it, the title "장바구니" (Cart) is displayed. To the right, a breadcrumb navigation shows the path: 01 옵션선택 > 02 장바구니 > 03 주문/결제 > 04 주문완료. The main area is divided into two tabs: "일반구매 (1)" (General Purchase (1)) and "정기배송 (0)" (Regular Delivery (0)). The "일반구매" tab is selected and highlighted with a blue border. Under this tab, there is a checkbox labeled "전체선택" (Select All) which is checked. To the right of the checkbox are columns for "상품정보" (Product Information), "상품금액" (Product Price), and "배송비" (Delivery Fee). Below this, a section for "로켓배송 상품" (Rocket Delivery Product) is shown, stating "무료배송 (19,800원 이상 구매가능)". A product item for "제주삼다수 그린 무라벨, 2L, 24개" is listed with an image of several water bottles. The quantity is set to 1, and the total price is 25,920원. The delivery option "내일(월) 7/3 도착 보장 (서울경기 기준)" is selected. The delivery fee is 25,920원, and the tip amount is 1,296원. The word "두료" (Duelio) is also present near the bottom right.

# 쿠키를 이용한 장바구니 예시 (2/5)

- 개발자 도구 - Network 탭 - cartView.pang 확인
- 서버는 응답과 함께 Set-Cookie 응답 헤더를  
브라우저에게 전송
  - 이 헤더는 클라이언트에게 쿠키를 저장하라고 전달하는 것



# 쿠키를 이용한 장바구니 예시 (3/5)

Cookie 데이터 자세히 확인

The screenshot shows the Network tab of the Chrome DevTools interface. The URL is `cart.coupaing.com/cart/view/pang`. The 'Cookies' tab is selected. The 'Request Cookies' section lists numerous cookies, many of which are highlighted in red. The 'Response Cookies' section also lists several cookies. The table columns include Name, Value, Do..., Path, Expl..., Size, Http..., Secu..., Sam..., Parti..., and Pr... . A red box highlights the 'Cookies' tab and the 'Request Cookies' table.

Name	Value	Do...	Path	Expl...	Size	Http...	Secu...	Sam...	Parti...	Pr...
sid	85a8a450bc2a4809be77921b3eff1f7...	cou...	/	202...	43					Med...
PCID	29390365939432262516837	cou...	/	202...	27					Med...
overrideAbTestGroup	%5B%5D	cou...	/	202...	25					Med...
MARKEID	29390365939432262516837	cou...	/	Sess...	31	✓	✓	None		Med...
x-coupaing-accept-langu...	ko-KR	cou...	/	202...	30					Med...
x-coupaing-target-market	KR	cou...	/	202...	25					Med...
ak_bmsc	70FCABD0E8BA297F554B63134A5D...	cou...	/	202...	475	✓				Med...
bm_sz	B868731896B8D4328FD60154E4438...	cou...	/	202...	358					Med...
searchKeyword	%EC%82%BC%EB%88%A4%EC%88%	cou...	/	202...	40					Med...
searchKeywordType	%7B%22%EC%82%BC%EB%88%A4%...	cou...	/	202...	60					Med...
FUN	"search":{ "reqUrl": "/search.pang", "is...	cou...	/	Sess...	58					Med...
bm_mi	E206F741B6F4D043FB7C189D364F2...	cou...	/	202...	372	✓				Med...
baby-isWide	small	cou...	/	202...	16					Med...
_abck	0C903C251C11E82BEEA3074C6DF8...	cou...	/	202...	485	✓				Med...
trac_src	0	cou...	/	202...	9					Med...
trac_spec	0	cou...	/	202...	10					Med...
trac_addtag	0	cou...	/	202...	12					Med...
trac_ctag	"	cou...	/	202...	11					Med...
trac_lptag	"	cou...	/	202...	12					Med...
trac_ltime	"	cou...	/	202...	12					Med...
trac_sid	"	cou...	/	202...	10					Med...
trac_appver	"	cou...	/	202...	13					Med...
CARTIC	1	cou...	/	Sess...	7					Med...
CCSID	0XGcVBVjCtoQ8BgVOzIP5Q%3D%3D	cou...	/	202...	33					Med...
SUBCARTIC	0	cou...	/	Sess...	10					Med...
bm_sv	EF39824B726FC6823BE2753458BA8...	cou...	/	202...	300	✓				Med...

Request Cookies

Response Cookies

# 쿠키를 이용한 장바구니 예시 (4/5)

메인 페이지 이동 - 장바구니 유지 상태 확인

The screenshot displays a browser window with the Coupang homepage loaded. The Network tab of the developer tools is open, showing a list of requests. One specific request to `https://www.coupang.com/` is selected, and its details are shown in the right panel. The 'Cookies' section of this panel is highlighted with a red border, displaying a long string of cookie data.

**Selected Request Details:**

- Request URL: `https://www.coupang.com/`
- Request Method: GET
- Status Code: 200
- Remote Address: 23.40.45.205:443
- Referrer Policy: strict-origin-when-cross-origin

**Selected Request Headers (20):**

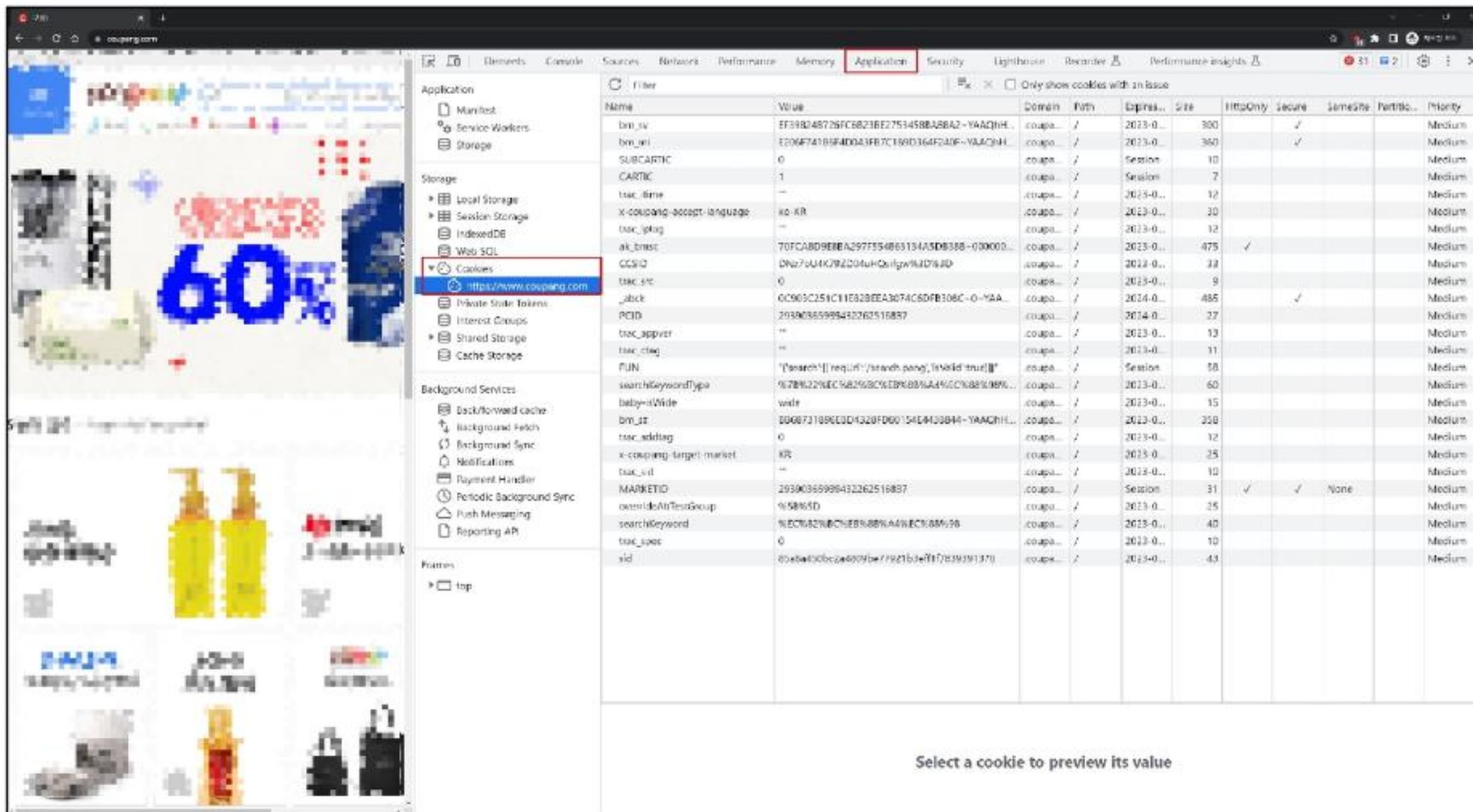
- :Authority: www.coupang.com
- :Method: GET
- :Path: /
- :Scheme: https
- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7
- Accept-Encoding: gzip, deflate, br
- Accept-Language: ko-KR,ko;q=0.9

**Selected Cookies:**

```
sid=85e8a450bc2a4809be77921b3eff1f7839391378; PCID=29390365939432262516837; overridesAbTestGroup=%5B%5D; MARKETID=29390365939432262516837; k-coupa...  
accept-language=ko-KR; x-coupa-target-market=KR;  
ak_bmsc=70FC4BD0E88A297F5548631345D8388-00000000000000000000000000000000  
~YAAQHIXTF0KpxASIAQAAy8QoFtxMwriTeuh00gqWoztCvO5Olxshsg607Jiq8or/6DfY  
BZx8wnUfJ9fOVlUjCqX+uJvhOf/3ODqz9EX+Wq8MdB/yTB1vub+F9GsjQAt1OwlXUMT  
ts8Ppl2G5e+GLq+xXVWiteTwLtd0s13EjlvVMP5qeFG068sDdyc4eKifG7JVCWqxGOVzO4chK  
8eI8QjAczNtsj0kHihdwqRdkbZ2ZBxt5uzWYg/Gpq5b3kpjhOUuv6hPTr3feLxL4E4Gmrx  
79HkDajdgRd5e4D4QzjMV2qj(DNqZSWGlw3duegFnARQxDH6kyVg@5Vjk9xGq)8ag7Yru  
swaNIkzoedRVTT+Zbow2AmkBrHqToysUstTsrgd1tw=ic  
bm_sz-BB66731896EBD4328FD60154E4438844~YAAQHIXTR0SqaASIAQAAyBQaFxQjD+B  
wG6w/uuMuuw128HSgsi+foLYovVQPNcvteCBoeLip5WNlLnw/vv95SVmC+eIr5jLz3BY  
BIO77imTNsFjCCVaqcwbsl79pxz9nFMVdkcPwkWW75hOMhba5BCx9te9MjsnB1vyPKG/Gc  
KmREON/VSPVFaadlw4xn3g91hlm8ixM3s2nTs1F29G+F8X2UMYNwWmlpB0X4ptUb1/x  
WCurUg7adHRCc0PWthvliQaNOjvn1GqNEsmV9bCFh9ouVTfrHj/j3DgOPg-4273458-44
```

# 쿠키를 이용한 장바구니 예시 (5/5)

- 개발자 도구 - Application 탭 - Cookies
- 마우스 우측 버튼 - Clear - 새로고침 - 장바구니가 빈 것을 확인



# 쿠키의 작동 원리와 활용 (1/2)

## 1. 쿠키 저장 방식

- 브라우저(클라이언트)는 쿠키를 KEY-VALUE의 데이터 형식으로 저장
- 쿠키에는 이름, 값 외에도 만료 시간, 도메인, 경로 등의 추가 속성이 포함 됨

## 2. 쿠키 전송 과정

- 서버는 HTTP 응답 헤더의 Set-Cookie 필드를 통해 클라이언트에게 쿠키를 전송
- 브라우저는 받은 쿠키를 저장해 두었다가, 동일한 서버에 재요청 시 HTTP 요청 Header의 Cookie 필드에 저장된 쿠키를 함께 전송

## 쿠키의 작동 원리와 활용 (2/2)

### 3. 쿠키의 주요 용도

- 두 요청이 동일한 브라우저에서 들어왔는지 아닌지를 판단할 때 주로 사용됨
  - 이를 이용해 사용자의 로그인 상태를 유지할 수 있음
  - 상태가 없는(stateless) HTTP 프로토콜에서 상태 정보를 기억시켜 주는 역할
- 서버에게 “나 로그인된 사용자야!”라는 인증 정보가 담긴 쿠키를 매 요청마다 계속 보내는 것

# 쿠키 사용 목적

1. 세션 관리 (Session management)
  - 로그인, 아이디 자동완성, 공지 하루 안 보기, 팝업 체크, 장바구니 등의 정보 관리
2. 개인화 (Personalization)
  - 사용자 선호 설정(언어 설정, 테마 등) 저장
3. 트래킹 (Tracking)
  - 사용자 행동을 기록 및 분석

# 세션

# 세션(Session)

-----  
서버 측에서 생성되어 클라이언트와 서버 간의 상태를 유지  
상태 정보를 저장하는 데이터 저장 방식

# 세션(Session)

-----  
서버 측에서 생성되어 클라이언트와 서버 간의 상태를 유지  
상태 정보를 저장하는 데이터 저장 방식

- 쿠키에 세션 데이터를 저장하여 매 요청시마다 세션 데이터를 함께 보냄

## 세션 작동 원리

1. 클라이언트가 로그인 요청 후 인증에 성공하면 서버가 **session** 데이터를 생성 후 저장
2. 생성된 **session** 데이터에 인증 할 수 있는 **session id**를 발급
3. 발급한 **session id**를 클라이언트에게 응답 (데이터는 서버에 저장, 열쇠만 주는 것)
4. 클라이언트는 응답 받은 **session id**를 쿠키에 저장
5. 클라이언트가 다시 동일한 서버에 접속하면 요청과 함께 쿠키(**session id**가 저장된)를 서버에 전달
6. 쿠키는 요청 때마다 서버에 함께 전송 되므로 서버에서 **session id**를 확인해 로그인 되어있다는 것을 계속해서 확인하도록 함

서버 측에서는  
세션 데이터를 생성 후 저장하고  
이 데이터에 접근할 수 있는 세션 ID를 생성

이 ID를 클라이언트 측으로 전달하고,  
클라이언트는 쿠키에 이 ID를 저장

이후  
**클라이언트가 같은 서버에 재요청 시마다  
저장해 두었던 쿠키도 요청과 함께 전송**

- 예를 들어 로그인 상태 유지를 위해  
로그인 되어있다는 사실을 입증하는 데이터를  
매 요청마다 계속해서 보내는 것

## 쿠키와 세션의 목적

---

클라이언트와 서버 간의 상태 정보를 유지하고  
사용자를 식별하기 위해 사용

# 이어서...

삼성 청년 SW 아카데미

# Django Authentication System

---

# Django Authentication System

---

사용자 인증과 관련된 기능을 모아 놓은 시스템

# Authentication

(인증)

사용자가 자신이 누구인지 확인하는 것  
(신원 확인)

## 사전 준비

- 두 번째 app **accounts** 생성 및 등록
  - ❖ auth와 관련한 경로나 키워드들을 django 내부적으로 accounts라는 이름으로 사용하고 있기 때문에 되도록 '**accounts**'로 지정하는 것을 권장

```
# accounts/urls.py

from django.urls import path
from . import views

app_name = 'accounts'
urlpatterns = [
    ...
]
```

```
# crud/urls.py

urlpatterns = [
    ...,
    path('accounts/', include('accounts.urls')),
]
```

# 이어서...

삼성 청년 SW 아카데미

# Custom User model

# User model 대체하기

# 기본 User Model의 한계

- 우리는 지금까지 별도의 User 클래스 정의 없이 내장된 auth 앱에 작성된 User 클래스를 사용했음
- Django의 기본 User 모델은 username, password 등 제공되는 필드가 매우 제한적
- 추가적인 사용자 정보(예: 생년월일, 주소, 나이 등)가 필요하다면 이를 위해 기본 User Model을 변경하기 어려움
  - 별도의 설정 없이 사용할 수 있어 간편하지만, 개발자가 직접 수정하기 어려움

## 내장된 auth 앱

```
# settings.py

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

# User Model 대체의 필요성

- 프로젝트의 특정 요구사항에 맞춰 사용자 모델을 확장할 수 있음
- 예를 들어 이메일을 username으로 사용하거나, 다른 추가 필드를 포함시킬 수 있음

# Custom User Model로 대체하기 (1/3)

- AbstractUser 클래스를 상속받는 커스텀 User 클래스 작성
  - 기존 User 클래스도 AbstractUser를 상속받기 때문에  
커스텀 User 클래스도 기존 User 클래스와 완전히 같은 모습을 가지게 됨

```
# accounts/models.py

from django.contrib.auth.models import AbstractUser

class User(AbstractUser):
    pass
```

# Custom User Model로 대체하기 (2/3)

- django 프로젝트에서 사용하는 기본 User 모델을 우리가 작성한 User 모델로 사용할 수 있도록 AUTH\_USER\_MODEL 값을 변경
  - 수정 전 기본 값은 ‘auth.User’

```
# settings.py

AUTH_USER_MODEL = 'accounts.User'
```

# Custom User Model로 대체하기 (3/3)

- admin site에 대체한 User 모델 등록
  - 기본 User 모델이 아니기 때문에 등록하지 않으면 admin 페이지에 출력되지 않기 때문

```
# accounts/admin.py

from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from .models import User

admin.site.register(User, UserAdmin)
```

# AUTH\_USER\_MODEL

---

Django 프로젝트의 User를 나타내는 데  
사용하는 모델을 지정하는 속성

※ 주의 ※

**프로젝트 중간에**

**AUTH\_USER\_MODEL을 변경 할 수 없음**

- ❖ 이미 프로젝트가 진행되고 있을 경우 데이터베이스 초기화 후 진행

# 사용하는 User 테이블의 변화

- > articles\_article
- > auth\_group
- > auth\_group\_permissions
- > auth\_permission
- > auth\_user
- > auth\_user\_groups
- > auth\_user\_user\_permissions
- > django\_admin\_log
- > django\_content\_type
- > django\_migrations
- > django\_session



- > accounts\_user
- > accounts\_user\_groups
- > accounts\_user\_user\_permissions
- > articles\_article
- > auth\_group
- > auth\_group\_permissions
- > auth\_permission
- > django\_admin\_log
- > django\_content\_type
- > django\_migrations
- > django\_session

# 프로젝트를 시작하며 반드시 User 모델을 대체해야 한다.

- Django는 새 프로젝트를 시작하는 경우 비록 기본 User 모델이 충분하더라도 커스텀 User 모델을 설정하는 것을 **강력하게 권장**하고 있음
- 커스텀 User 모델은 **기본 User 모델과 동일하게 작동(1)** 하면서도 필요한 경우 나중에 맞춤 설정(2)할 수 있기 때문
  - ❖ 단, User 모델 대체 작업은 프로젝트의 모든 migrations 혹은 첫 migrate를 실행하기 전에 이 작업을 마쳐야 함

# 이어서...

삼성 청년 SW 아카데미

# Login

---

# Login

---

로그인은 ??????을 Create하는 과정

# Login

---

로그인은 Session을 Create하는 과정

# AuthenticationForm()

---

로그인 인증에 사용할 데이터를 입력 받는  
built-in form

# 로그인 페이지 작성 (1/2)

```
# accounts/urls.py

app_name = 'accounts'
urlpatterns = [
    path('login/', views.login, name='login'),
]

<!-- accounts/login.html --&gt;

&lt;h1&gt;로그인&lt;/h1&gt;
&lt;form action="{% url 'accounts:login' %}" method="POST"&gt;
    {% csrf_token %}
    {{ form.as_p }}
    &lt;input type="submit"&gt;
&lt;/form&gt;</pre>
```

```
# accounts/views.py

from django.contrib.auth.forms import AuthenticationForm

def login(request):
    if request.method == 'POST':
        pass
    else:
        form = AuthenticationForm()
    context = {
        'form': form,
    }
    return render(request, 'accounts/login.html', context)
```

## 로그인 페이지 작성 (2/2)

Document

127.0.0.1:8000/accounts/login/

로그인

Username:

Password:

제출

AuthenticationForm

# 로그인 로직 작성

```
# accounts/views.py

from django.shortcuts import render, redirect
from django.contrib.auth import login as auth_login

def login(request):
    if request.method == 'POST':
        form = AuthenticationForm(request, request.POST)
        # form = AuthenticationForm(request, data=request.POST)
        if form.is_valid():
            auth_login(request, form.get_user())
            return redirect('articles:index')
    else:
        form = AuthenticationForm()
    context = {
        'form': form,
    }
    return render(request, 'accounts/login.html', context)
```

# **login(request, user)**

---

AuthenticationForm을 통해  
인증된 사용자를 로그인 하는 함수

# get\_user()

---

AuthenticationForm의 인스턴스 메서드

- 유효성 검사를 통과했을 경우 로그인 한 사용자 객체를 반환

# 세션 데이터 확인하기 (1/2)

- 로그인 후 발급받은 세션 확인
  - django\_session 테이블에서 확인

The screenshot shows a SQLite database browser interface. On the left, a sidebar lists tables in the database, including articles\_article, auth\_group, auth\_group\_permissions, auth\_permission, auth\_user, auth\_user\_groups, auth\_user\_user\_permissions, django\_admin\_log, django\_content\_type, django\_migrations, django\_session (which is currently selected and highlighted in blue), and sqlite\_sequence. The main area displays a query window with the following content:

```
Properties DATA
SELECT * FROM django_session LIMIT 100
```

The results pane shows one row of data from the django\_session table. The columns are session\_key (varchar(40)), session\_data (TEXT), and expire\_date (datetime). The session\_key value is highlighted with a red box. The session\_data value is truncated at the end.

	*	session_key	session_data	expire_date
	> 1	rryidpralk8p63l7g8a9htmqh9e	.eJxVjEEOwiAQRe_C2pASBgd...	2023-10-24 10:15:45

# 세션 데이터 확인하기 (2/2)

## 2. 브라우저에서 확인

- 개발자도구 - Application - Cookies

The screenshot shows the Chrome DevTools interface with the Application tab selected. On the left, the Storage section is expanded, showing Local Storage, Session Storage, IndexedDB, Web SQL, and Cookies. Under Cookies, there are two entries: 'sessionid' and 'csrfToken'. Both entries have their values highlighted in red. The 'sessionid' value is 'qj258qt9vr2mdrygj79ic2m87vse5tj' and it expires on 2023-07-12. The 'csrfToken' value is 'kHptIbyYap8tk5Pq2xaaxQnR0l6FLUfU' and it expires on 2024-07-12. A message at the bottom of the DevTools window says 'Select a cookie to preview its value'.

Name	Value	Dom	Path	Expire	Size	Http	Secure	SameSite	Parti	Pri
sessionid	qj258qt9vr2mdrygj79ic2m87vse5tj	127....	/	2023...	41	✓	Lax		Med...	
csrfToken	kHptIbyYap8tk5Pq2xaaxQnR0l6FLUfU	127....	/	2024...	41		Lax		Med...	

# 로그인 링크 작성

- 메인 페이지에 로그인 페이지로 갈 수 있는 링크 작성

```
<!-- articles/index.html -->

<h1>Articles</h1>
Login
NEW
<hr>
```

# 이어서...

삼성 청년 SW 아카데미

# Logout

---

# Logout

---

로그아웃은 ??????을 Delete하는 과정

# Logout

---

로그아웃은 Session을 Delete하는 과정

# logout(request)

---

1. DB에서 현재 요청에 대한 Session Data를 삭제
2. 클라이언트의 쿠키에서도 Session Id를 삭제

# 로그아웃 로직 작성 (1/2)

```
# accounts/urls.py

urlpatterns = [
    path('login/', views.login, name='login'),
    path('logout/', views.logout, name='logout'),
]
```

```
<!-- articles/index.html -->

<h1>Articles</h1>
<a href="{% url 'accounts:login' %}">Login</a>
<form action="{% url 'accounts:logout' %}" method="POST">
    {% csrf_token %}
    <input type="submit" value="Logout">
</form>
```

```
# accounts/views.py

from django.contrib.auth import logout as auth_logout

def logout(request):
    auth_logout(request)
    return redirect('articles:index')
```

# 로그아웃 로직 작성 (2/2)

로그아웃 진행 및 세션 데이터 삭제 확인

The screenshot shows the Chrome DevTools Application tab for a web application titled "Articles". On the left, there's a sidebar with links for "Login", "Logout", and "NEW". The main area displays a table of session storage items. One item is highlighted: "csrfmiddlewaretoken" with the value "kHptbyYbp8tix5Rq2xaQnRn6JFLRU". The table includes columns for Name, Value, Domain, Path, Expires, Size, HttpOnly, Secure, SameSite, and Priority.

Name	Value	Dom...	Path	Expir...	Size	Http...	Secur...	Same...	Priority
csrfmiddlewaretoken	kHptbyYbp8tix5Rq2xaQnRn6JFLRU	127....	/	2024...	41			Lax	Med...

Select a cookie to preview its value

The screenshot shows a SQLite database browser interface connected to a database at "/Users/har.../db.sqlite3". The browser displays the schema of the "django\_session" table, which has 12 columns: session\_key (varchar(40)), session\_data (TEXT), and expire\_date (datetime). A query is shown: "SELECT \* FROM django\_session LIMIT 100". The browser also shows the structure of other tables in the database, including accounts\_user, auth\_group, auth\_permission, django\_admin\_log, django\_content\_type, django\_migrations, and sqlite\_sequence.

Properties DATA

Tables (12)

- accounts\_user
- accounts\_user\_groups
- accounts\_user\_user\_permissions
- articles\_article
- auth\_group
- auth\_group\_permissions
- auth\_permission
- django\_admin\_log
- django\_content\_type
- django\_migrations
- django\_session
- sqlite\_sequence

Properties DATA

Tables (12)

- accounts\_user
- accounts\_user\_groups
- accounts\_user\_user\_permissions
- articles\_article
- auth\_group
- auth\_group\_permissions
- auth\_permission
- django\_admin\_log
- django\_content\_type
- django\_migrations
- django\_session
- sqlite\_sequence

Properties DATA

Tables (12)

- accounts\_user
- accounts\_user\_groups
- accounts\_user\_user\_permissions
- articles\_article
- auth\_group
- auth\_group\_permissions
- auth\_permission
- django\_admin\_log
- django\_content\_type
- django\_migrations
- django\_session
- sqlite\_sequence

# 이어서...

삼성 청년 SW 아카데미

# Template with Authentication data

---

# 템플릿과 인증 데이터

# Template with Authentication data

---

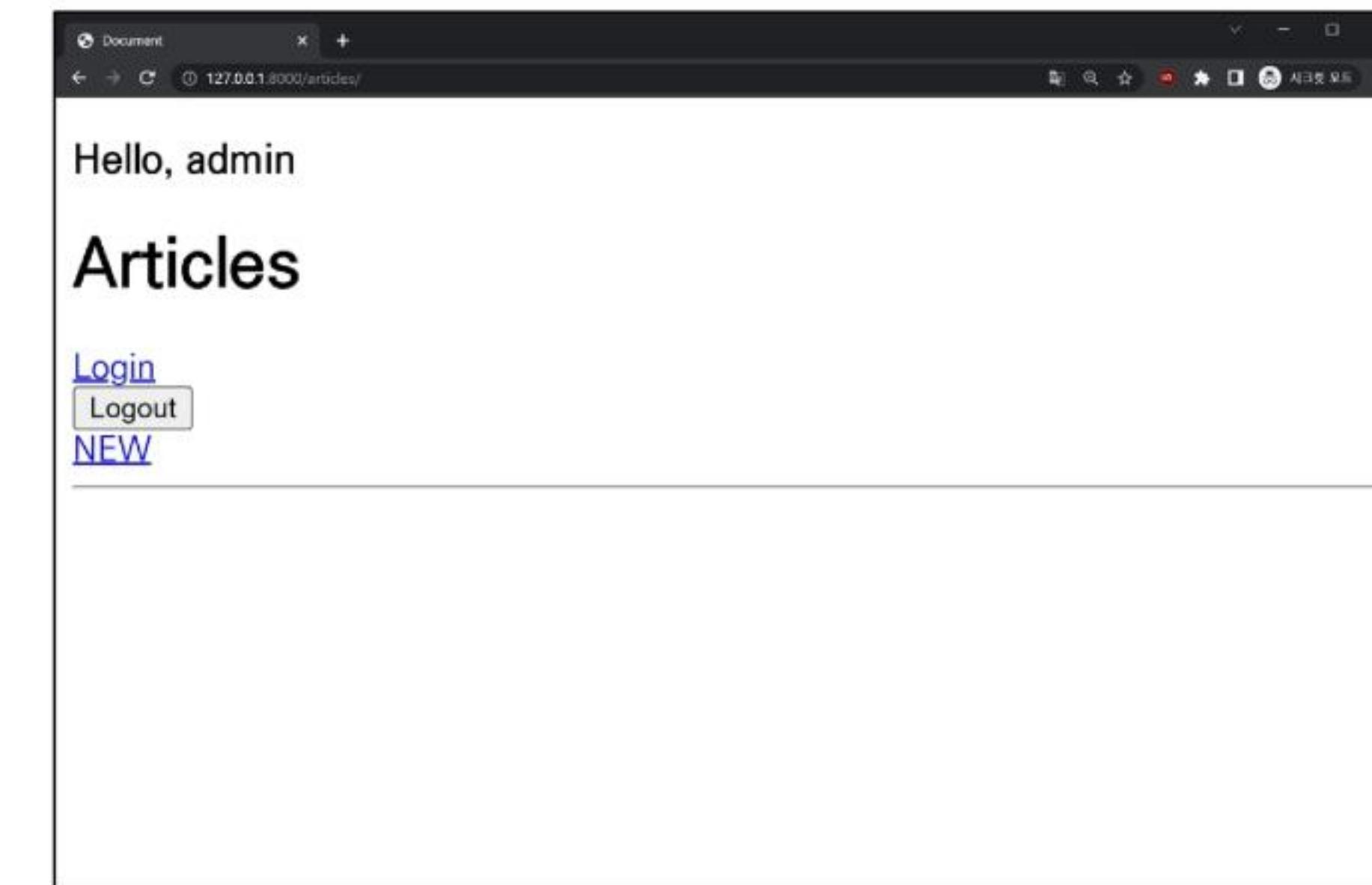
템플릿에서 인증 관련 데이터를 출력하는 방법

# 현재 로그인 되어있는 유저 정보 출력하기

user라는 context 데이터를 사용할 수 있는 이유는?

- django가 미리 준비한 context 데이터가 존재하기 때문(context processors)

```
<!-- articles/index.html -->  
  
<h3>Hello, {{ user.username }}</h3>
```



# context processors

- 템플릿이 렌더링 될 때 호출 가능한 컨텍스트 데이터 목록
  - 작성된 컨텍스트 데이터는 기본적으로 템플릿에서 사용 가능한 변수로 포함됨
- django에서 자주 사용하는 데이터 목록을 미리 템플릿에 로드 해 둔 것

```
# settings.py

TEMPLATES = [
    {
        ...
        "OPTIONS": {
            "context_processors": [
                "django.template.context_processors.debug",
                "django.template.context_processors.request",
                "django.contrib.auth.context_processors.auth",
                "django.contrib.messages.context_processors.messages",
            ],
        },
    },
]
```

# 이어서...

삼성 청년 SW 아카데미

# 참고

# 쿠키의 수명

# 쿠키 종류별 Lifetime (수명)

## 1. Session cookie

- 현재 세션(current session)이 종료되면 삭제됨
- 브라우저 종료와 함께 세션이 삭제됨

## 2. Persistent cookies

- Expires 속성에 지정된 날짜 혹은 Max-Age 속성에 지정된 기간이 지나면 삭제됨

# 쿠키와 보안

# 쿠키의 보안 장치

- 제한된 정보
  - 쿠키에는 보통 중요하지 않은 정보만 저장. (사용자 ID나 세션 번호 같은 것)
- 암호화
  - 중요한 정보는 서버에서 암호화해서 쿠키에 저장
- 만료 시간
  - 쿠키에는 만료 시간을 설정 시간이 지나면 자동으로 삭제
- 도메인 제한
  - 쿠키는 특정 웹사이트에서만 사용할 수 있도록 설정할 수 있음

# 쿠키와 개인정보 보호

- 많은 국가에서 쿠키 사용에 대한 사용자 동의를 요구하는 법규를 시행
- 웹사이트는 쿠키 정책을 명시하고, 필요한 경우 사용자의 동의를 얻어야 함

# Django에서의 세션 관리

# 세션 in Django

- Django는 ‘database-backed sessions’ 저장 방식을 기본 값으로 사용
  - session 정보는 DB의 **django\_session** 테이블에 저장
  - Django는 요청안에 특정 session id를 포함하는 쿠키를 사용해서 각각의 브라우저와 사이트가 연결된 session 데이터를 알아냄
- Django는 우리가 session 메커니즘(복잡한 동작원리)에 대부분을 생각하지 않게끔 많은 도움을 줌

# AuthenticationForm

## 내부 코드

# django github 코드 참고

- **AuthenticationForm()**
  - <https://github.com/django/django/blob/main/django/contrib/auth/forms.py#L286>
- **AuthenticationForm()의 get\_user() 인스턴스 메서드**
  - <https://github.com/django/django/blob/main/django/contrib/auth/forms.py#L356>

# AbstractUser class

## ‘AbstractUser’ class

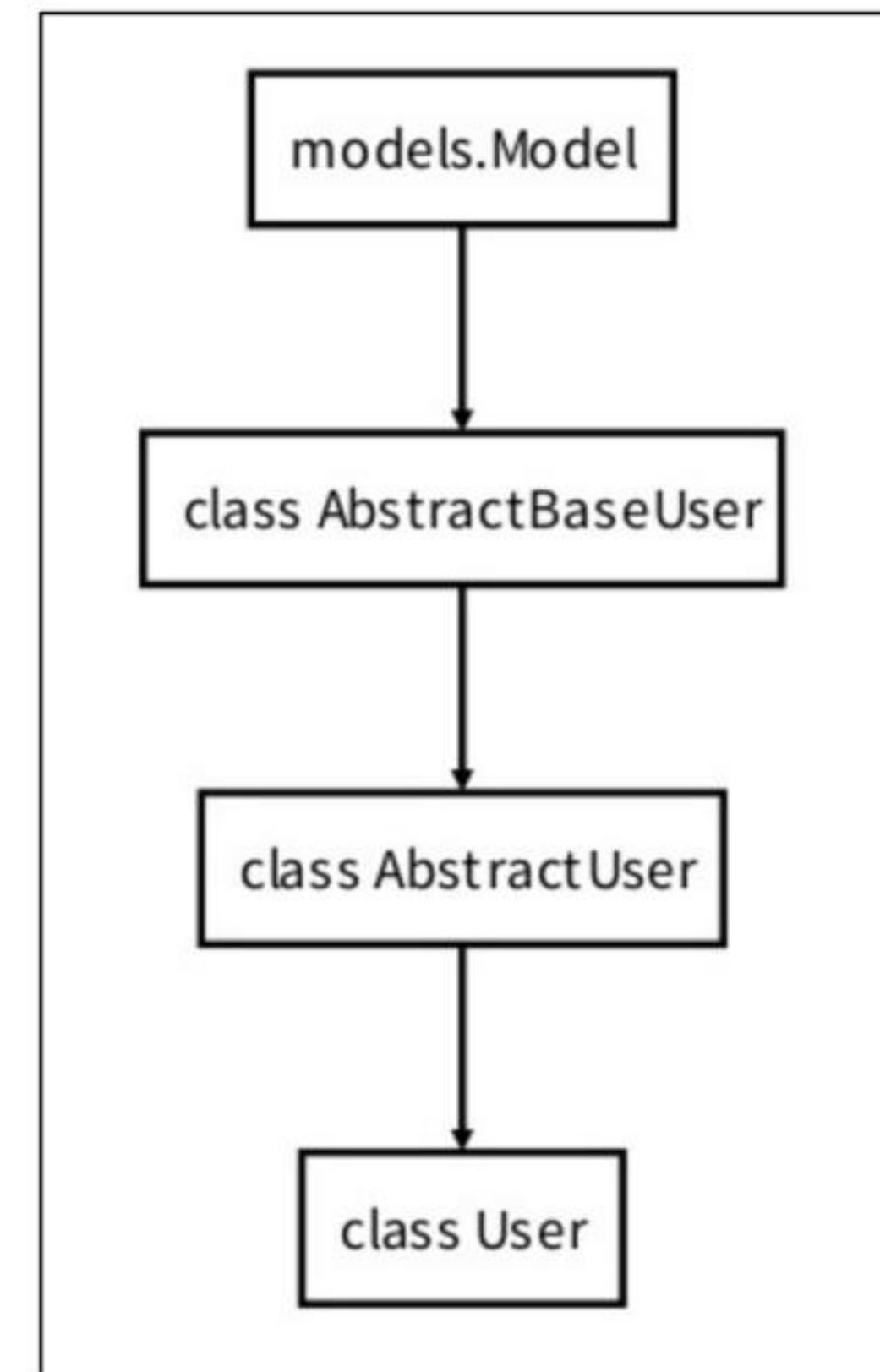
---

“관리자 권한과 함께 완전한 기능을 가지고 있는  
User model을 구현하는 추상 기본클래스”

# Abstract base classes (추상 기본 클래스)

- 몇 가지 공통 정보를 여러 다른 모델에 넣을 때 사용하는 클래스
- 데이터베이스 테이블을 만드는데 사용되지 않으며, 대신 다른 모델의 기본 클래스로 사용되는 경우 해당 필드가 하위 클래스의 필드에 추가됨

# User 모델 상속 관계



## User 모델 대체하기 Tip

# User 모델 대체하기 Tip

- User 모델을 대체하는 순서를 숙지하기 어려울 경우 해당 공식문서를 보면 순서대로 진행하는 것을 권장

다음 시간에  
만나요!

삼성 청년 SW 아카데미