

Django(프레임워크)

웹 애플리케이션



Django(프레임워크)

웹 애플리케이션(MVT, MTV)

response

url

response
(View)

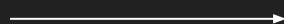
잘못된 요청 -> error

로그인 페이지 -> 로그인
페이지 GET

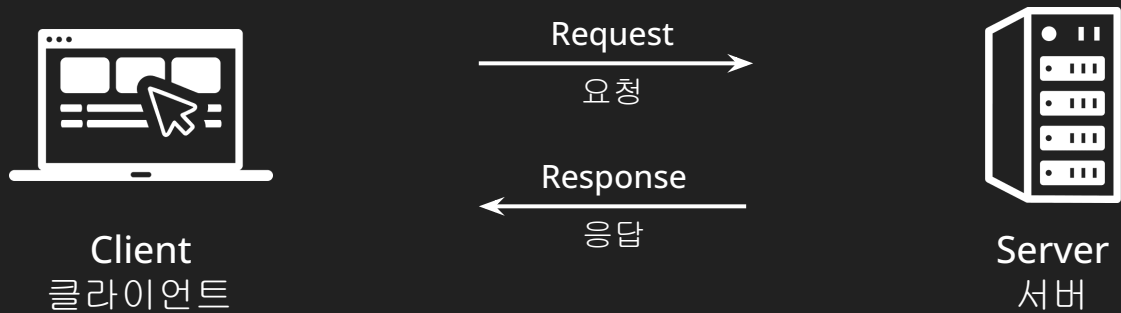
로그인 -> 로그인 이후 페이지
POST

Template

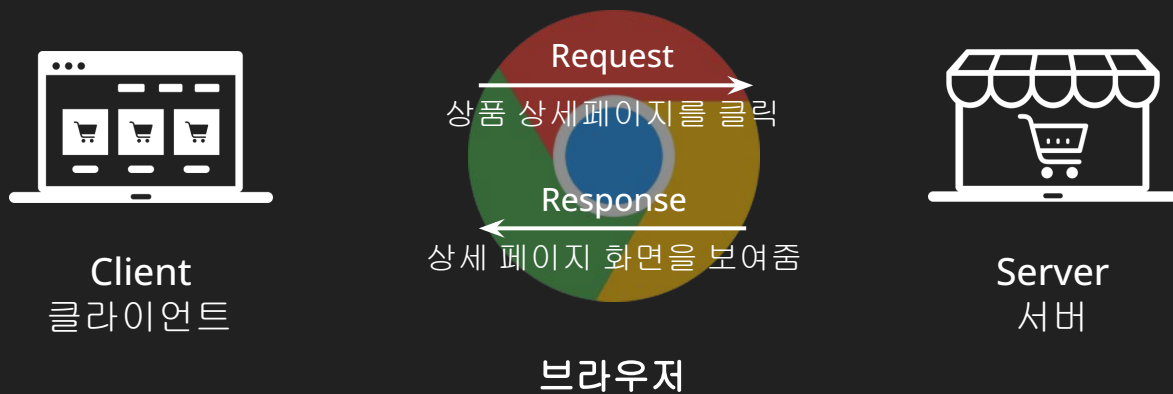
django-Model
database



인터넷 흐름 이해하기 - Client, Server



인터넷 흐름 이해하기 - Client with Browser



인터넷 흐름 이해하기 - Client with Browser



HTML, CSS,
JavaScript



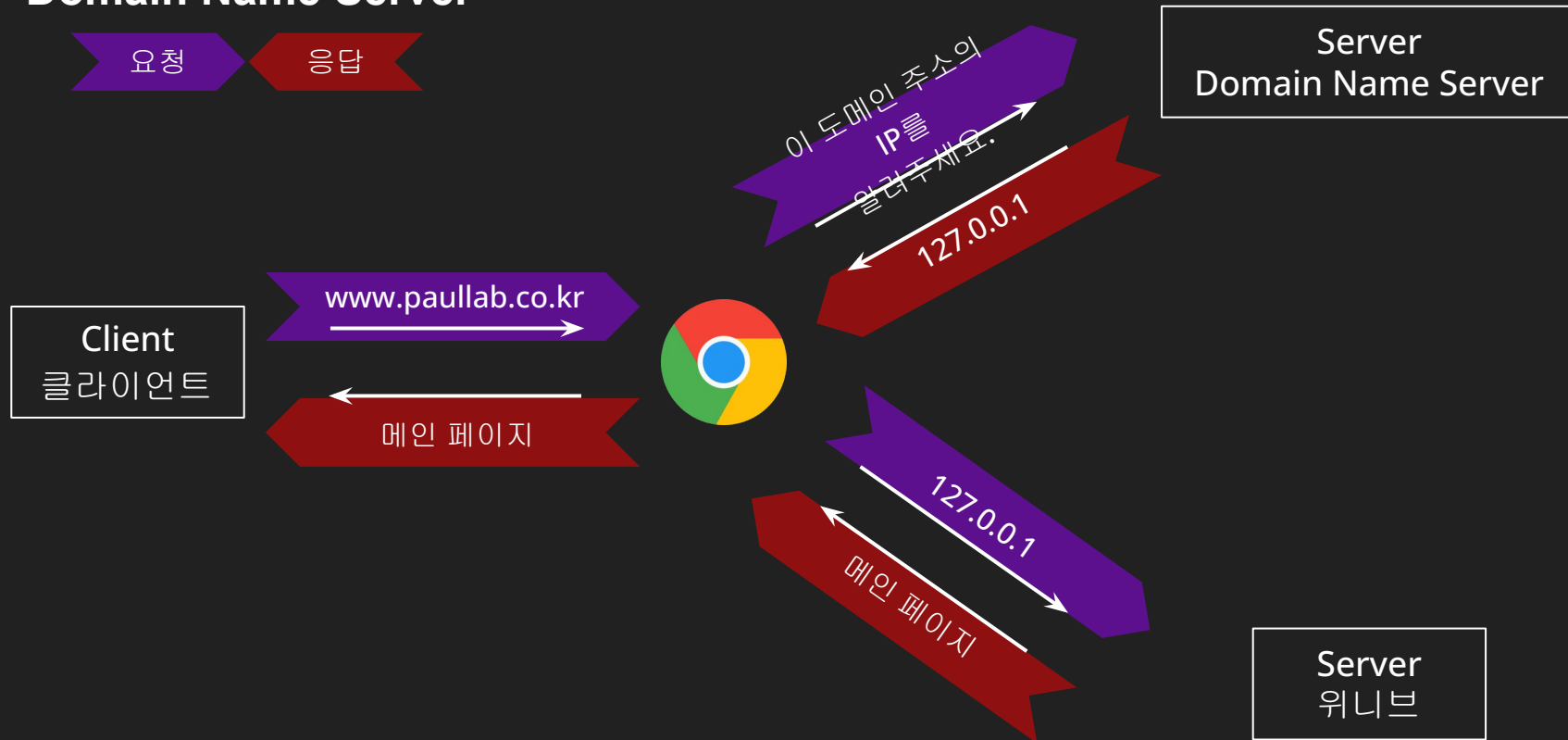
브라우저



화면

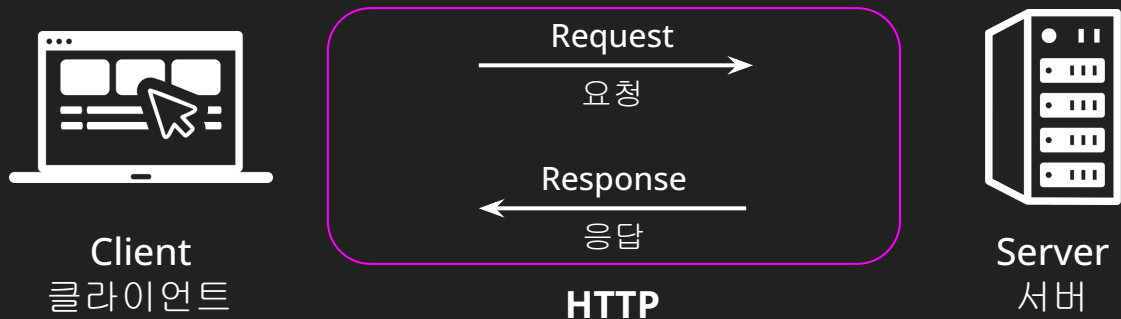
인터넷 흐름 이해하기 - Client with Browser

- Domain Name Server



인터넷 흐름 이해하기 - 프로토콜

어떻게? -> 정해진 규칙(규약)에 따라 -> 프로토콜



HTTP - HTTP 개념

HTTP (Hyper Transfer Protocol)

: 인터넷 상에서 데이터를 주고받을 때 사용하는
프로토콜
통신마다 독립적으로 관리

Start(Status) Line
HTTP Header
(Blank Line)
HTTP Body

HTTP Request

Client
클라이언트



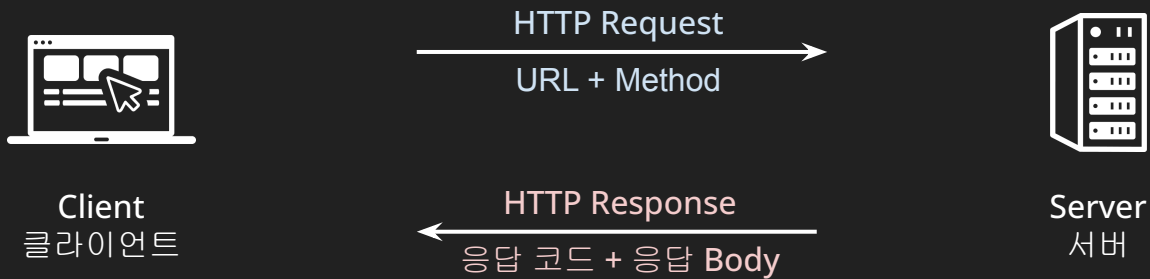
HTTP Response

Server
서버

HTTP - HTTP 사용

	Where 어디에	How 어떻게
<u>HTTP Request</u>	URL	요청 메서드 이용 : <u>GET (요청)</u> , <u>POST (생성)</u> , PUT (수정), DELETE (삭제)
<u>HTTP Response</u>	요청을 한 곳	응답 코드 : 성공, 실패 (2XX, 3XX, 4XX, 5XX)

HTTP - HTTP 사용



HTTP - HTTP Request (요청)

Request - Methods

메서드	특징
GET	<ul style="list-style-type: none">- 데이터 조회 요청- 요청 정보가 URL에 포함
POST	<ul style="list-style-type: none">- 데이터 생성 요청- 요청 정보가 URL에 불포함- 요청 정보가 Body에 포함
PUT	<ul style="list-style-type: none">- 데이터 수정 요청 (덮어쓰기)- 요청 정보가 URL에 불포함- 요청 정보가 Body에 포함
DELETE	<ul style="list-style-type: none">- 데이터 삭제 요청

같은 **URL** 주소일지라도
메서드에 따라 다른 동작을 수행할 수
있음

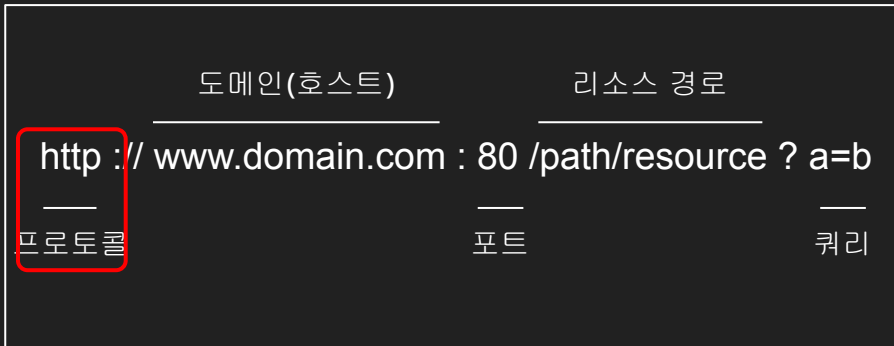
http://localhost:8000/product

- **GET:** 상품 정보 요청
- **POST:** 상품 등록 요청
- **PUT:** 상품 수정 요청
- **DELETE:** 상품 삭제 요청

HTTP - HTTP Request (요청)

Request - URL

URL 구조



예시

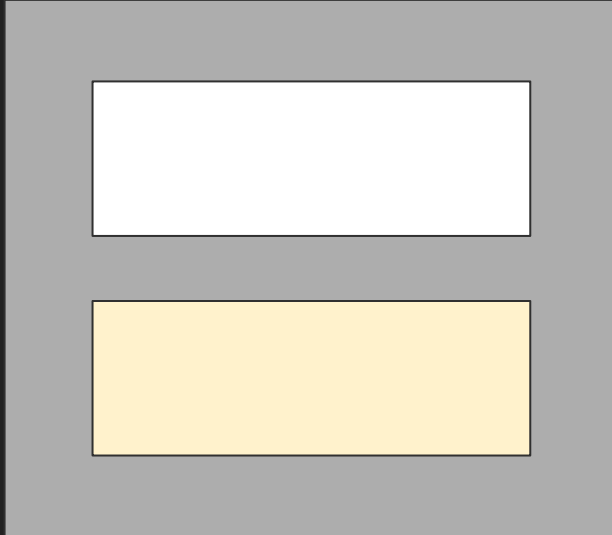
<https://shopping.naver.com/home>

www.shopping.naver.com/home

-> 네이버 쇼핑 홈을 보여줘

www.shopping.naver.com/beauty/home

-> 네이버 쇼핑 중 뷰티 카테고리의 홈을 보여줘



Ajax

비동기적인 웹 애플리케이션의 제작을 위해
아래와 같은 조합을 이용하는 웹 개발 기법

HTTP/HTTPS

HTTPS

HTTP(평문) -> 보안 기능을 넣음
TLS, SSL (암호화 과정)

암호화

평문 -> 암호문

암호문 -> 평문

복호화

Client, Server

Request, Response

HTTP/HTTPS

GET/POST

Browser, Rendering

DNS

URL

WireShark