



# HTTP

the Internet protocol

Mohammad ali Ali panah - Fall 2023

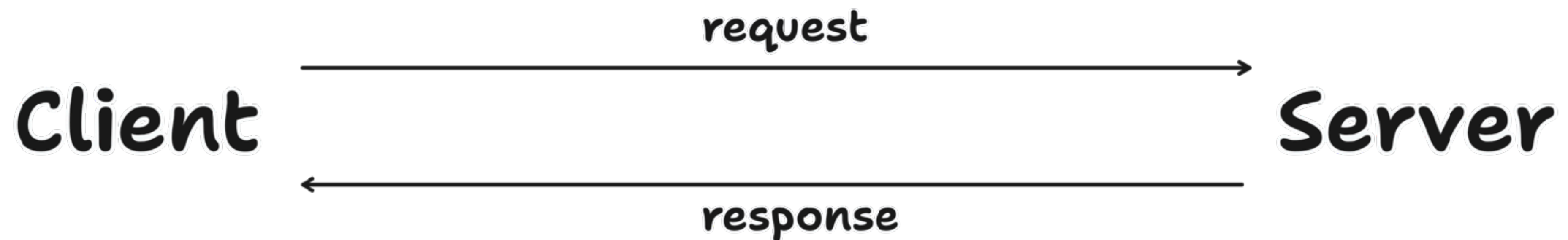


# Introduction

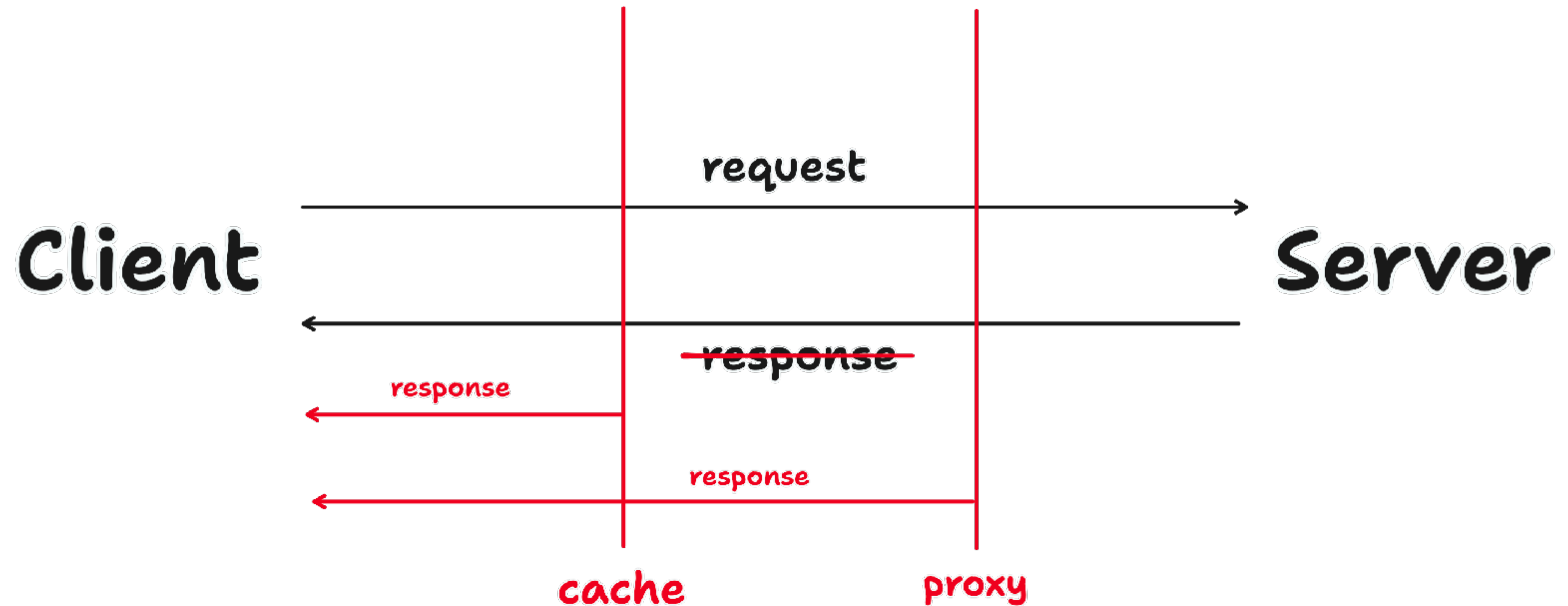
- HTTP(Hyper Text Transfer Protocol) is the transfer protocol for web applications
- HTTP 1.0([RFC 1945](#)), HTTP 1.1([RFC 2068](#)), HTTP 2([RFC 7540](#))
- It can be used to transfer anything(not only hyper text)
  - Text Documents e.g. HTML, XML, JSON, etc.
  - Multimedia e.g. JGP, GIF, MP4, MKV, etc.
  - Application e.g. PDF, ZIP, etc.



- HTTP uses the client/server paradigm
  - HTTP server provides resource
  - HTTP client (usually web browser) gets resource

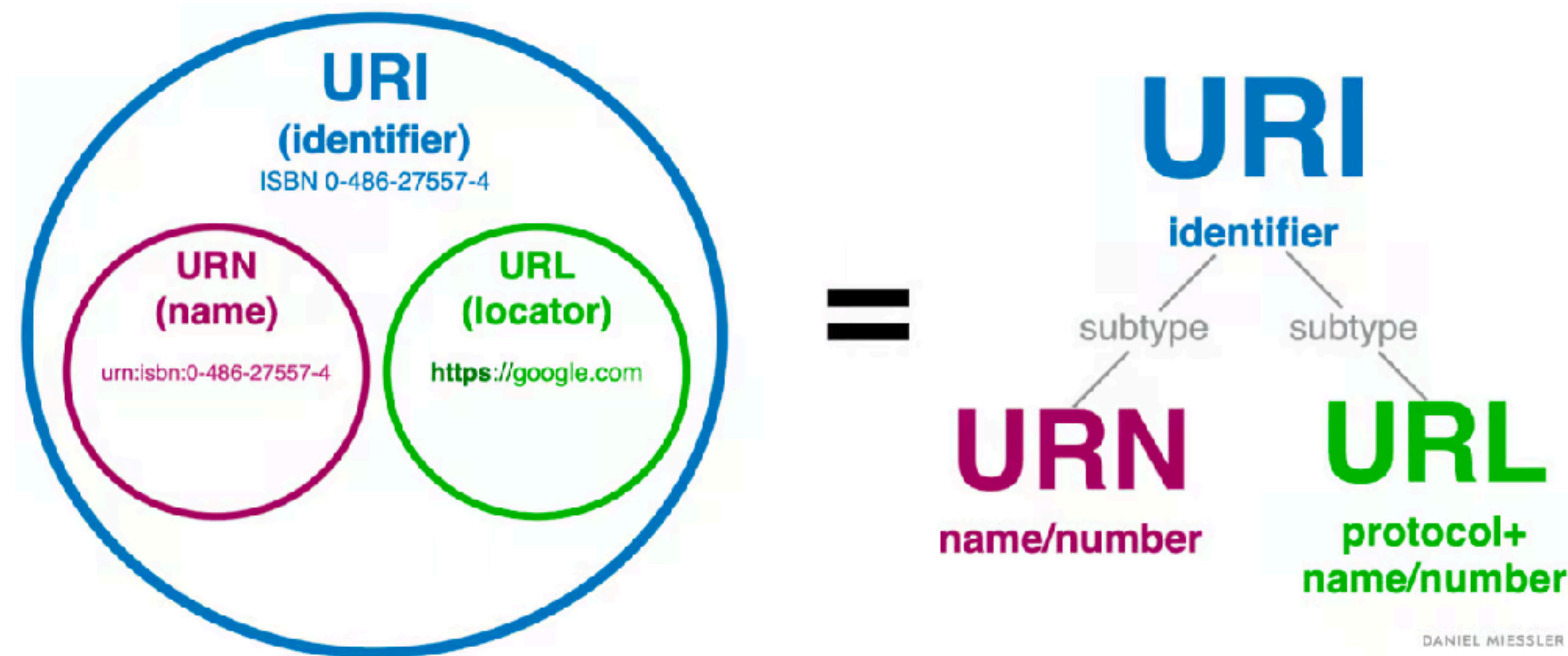


- But it's not always direct communication

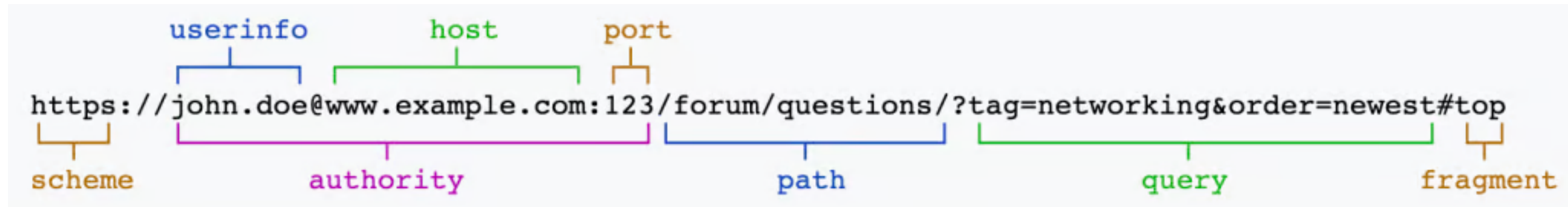


- HTTP is an application layer protocol
  - HTTP assumes reliable communication
  - over TCP
  - default (server) port: 80
- HTTP is Stateless
  - Server does not keep history/state of client
  - High performance & Low Complexity
  - How to identify the user(and keep the user login)?
    - Cookies
    - JSON Web Tokens

- Each resource must be identified/located uniquely
- URIs(Uniform Resource Identifier) identify and URLs(Uniform Resource Locator) locate; however, locators are also identifiers, so every URL is also a URI, but there are URIs which are not URLs.



- <https://www.zoomit.ir/>
- <https://www.zoomit.ir/ai-articles/412007-crisis-at-openai-sam-altman/#655da74f646a1d48d4ae7eb9>
- `file://home/tur1ng/Documents/programming/nit/x`
- <ftp://soft.nit.ac.ir>





- **Scheme**: The application layer protocol
  - HTTP: The web protocol
  - HTTPS: Secure HTTP
  - FTP: File Transfer Protocol
  - File: Access to a local file
  - javascript: Run javascript code
  - mailto: Send mail to given address

- **Path**: The path of the object on host filesystem
  - E.g. web server root directory is /var/www/
    - `http://www.example.com/1.html` → `/var/www/1.html`
    - `http://www.example.com/1/2/3.jpg` → `/var/www/1/2/3.jpg`
    - `http://www.example.com/1/2/../3.jpg` → `/var/www/1/3.jpg`



- **Query**: A mechanism to pass information from client to active pages or forms
  - Fill information in a form
  - Ask Google! to search a phrase
- Starts with "?"
- name=value format
- "&" is the border between multiple parameters

- **Frag**: A name for a part of resource
  - A section in a document
- Handled by browser
  - Browser gets whole resource from server
  - It jumps to the specific part



- Domain names are case insensitive
- The rest of URL is sent to the server can be case sensitive or not
- URL is encoded by client before transmission
- How: Each byte is divided into two 4-bit group, hexadecimal of the 4-bits are prefixed by %
  - ~ → 126 → %7E

- HTTP is text-based protocol
  - Human readable headers
  - The header is composed of some lines

## Structure of HTTP Message

Request Line	Status Line
General Header	
Request Header	Response Header
Entity Header	
Empty Line	
Message Body (entity body or encoded entity body)	



- **GET**: Retrieve resource from server
- **HEAD**: Similar to GET but the resource itself is not retrieved, just the HTTP response header
  - Useful for debugging or some other applications
- **POST**: Submit data to be processed by the specified resource
  - Data itself is enveloped in message body
- **DELETE**: Remove the resource
  - Not popular in web, can be used in other applications
- **PUT**: Add message body as the specified resource to server
- **PATCH**: A PATCH request is considered a set of instructions on how to modify a resource. Contrast this with PUT; which is a complete representation of a resource.
- **TRACE**: Server echoes back the received message
  - For troubleshooting & debugging
- **OPTIONS**: Request the list of supported methods by server on the resource

- 2xx
  - Successful responses
  - 200: OK
  - 201: Created
  - 204: No Content
- 4xx
  - Client errors
  - 400: Bad Request
  - 401: Unauthorized (Authorization required)
  - 403: Forbidden
  - 404: Not Found
  - 405: Method Not Allowed



- 5xx
  - Server errors
  - 500: Internal Server Error
  - 501: Not Implemented
  - 503: Service Unavailable
- 3xx
  - Redirects
  - 301: Moved Permanently
  - 302: Found
    - redirect status response code indicates that the resource requested has been temporarily moved to the URL given by the Location header
  - 307: Moved Temporarily
    - Resource has been moved, Redirection
    - Location header contains the new location of resource
  - 304: Not Modified

- 1xx
  - Informational responses
  - 101: Switching Protocol
    - This code is sent in response to an Upgrade request header from the client, and indicates the protocol the server is switching to
    - e.g. WebSocket connection upgrade from HTTP request

- Headers are additional information that is sent by client to server and vice versa
  - Most (almost all) are optional
- Which headers?
  - Information about client
  - Information about server
  - Information about the requested resource
  - Information about the response
  - Security/Authentication



- General headers
  - Appear both on request & response messages
- Request headers
  - Information about request
- Response headers
  - Information about response
- Entity headers
  - Information about body (size, ...)
- Extension headers
  - New headers (not standard)

- **Date**: Date & Time that message is created
- **Connection**: Close or Keep-Alive
  - Close: Non-persistent connection
  - Keep-Alive: Persistent connection
- **Via**: Information about the intermediate nodes between two sides
  - Proxy servers

- **Host**: The name of the server (required, why?)
- **Referer**: URL that contains requested URL
- Information about the client
  - User-Agent: The client program
  - Accept: The acceptable media types
  - Accept-Encoding: The acceptable encoding
  - Accept-Language: The acceptable language

- **Range**: Specific range (in byte) of resource
- **Authorization**: Response to the authenticate
- **Cookie**: To return back the cookies
- **If-Modified-Since**: Request is processed if the object is modified since the specified time(Used in Web Caching).



- **Server**: Information about server
- **WWW-Authenticate**: Used to specify authentication parameters by server
- **Proxy-Authenticate**: Used to specify authentication parameters by proxy
- **Set-Cookie**: To send a cookie to client
- **Location**: The location of entity to redirect client
- **Last-Modified**: The date and time of last modification of entity
- **Content-Range**: Range of this entity in the entire resource
- **Expires**: The date and time at which the entity will expire

- **Content-Length**: The length of body (in byte)
- **Content-Type**: The type of entity
  - MIME types: text/xml, image/gif
- **Allow**: The allowed request methods can be performed on the entity
  - This is in response of OPTIONS method

- Custom proprietary headers have historically been used with an X- prefix, but this convention was deprecated in June 2012
  - X-Device
  - X-Client-Ip
  - X-Cache-Status

- Parham Alavi Internet Engineering slides(<https://github.com/1995parham-teaching/ie-lecture>)
- <https://danielmiessler.com/p/difference-between-uri-url/>



# End

Thanks everybody  
esp. ygn <3