

---

---

---

---

---

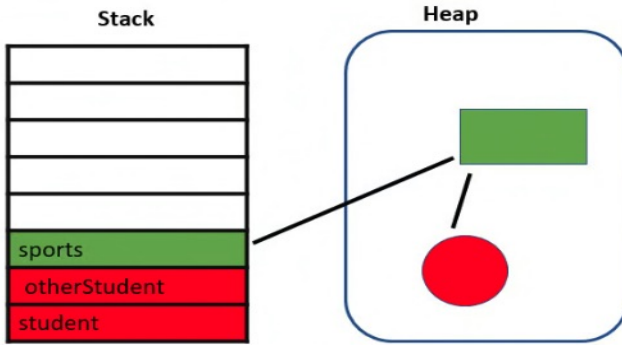


Class will start  
at 7:05AM

# Agenda

- \* Memory leaks in JS
  - \* Detached window
  - \* Closures
- \* Optimisation techniques
  - \* fragments, event delegation
  - \* Web-worker
- HTTP protocol in details
  - Response request cycle
  - \* Http methods
  - Status codes
  - How to send a network request
    - fetch
    - XMLHttpRequest
- caching your request

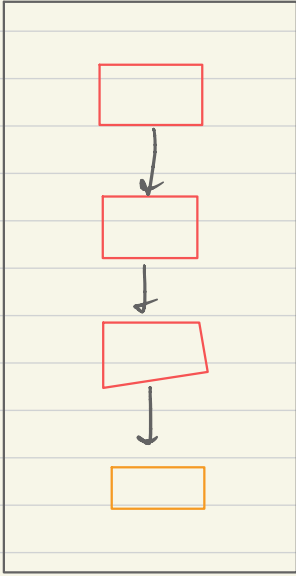
```
let student = {  
  name: 'Joe',  
  age: 15,  
  sports: ['soccer', 'chess']  
}  
let otherStudent = student;  
const sports = student.sports;  
student = null;  
otherStudent = null;
```



① Start → memory , Ended → freed up

② Optimization → when program is deleted

↪ main thread

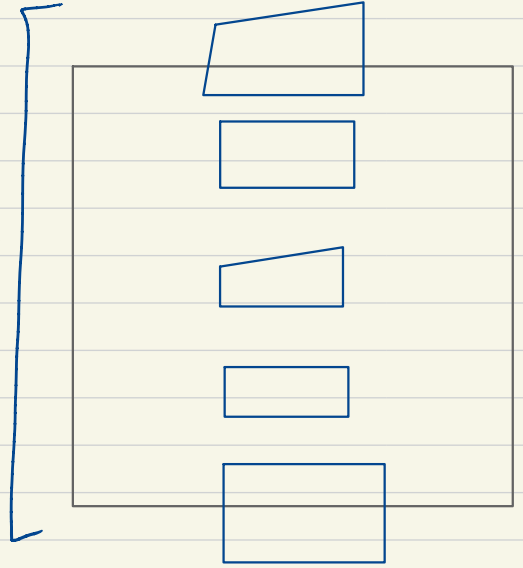


worker thread

React

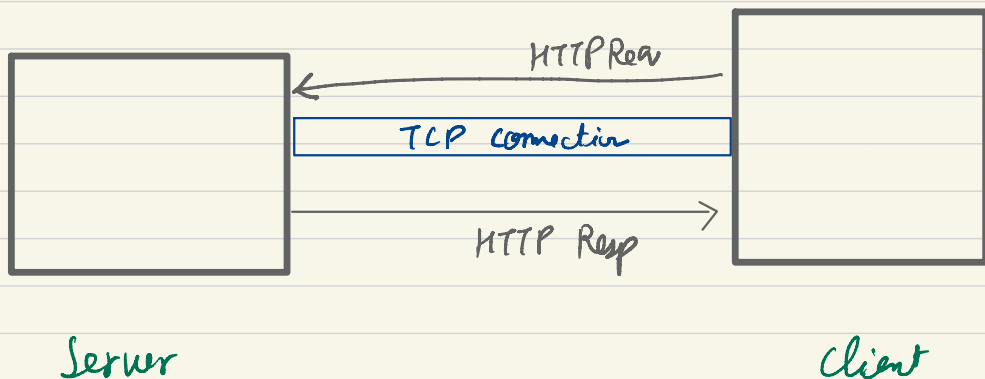
① Windowing →

5 dom  
part



② Lazy Loading → loading the content when  
it is required

# HTTP protocol



## ① HTTP packets

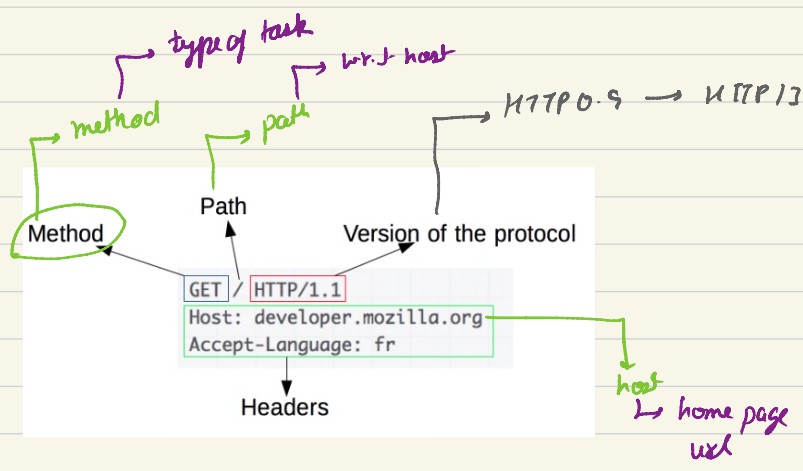
- ↳ head → metadata about the data
- ↳ body → part that contains actual

## ② Version

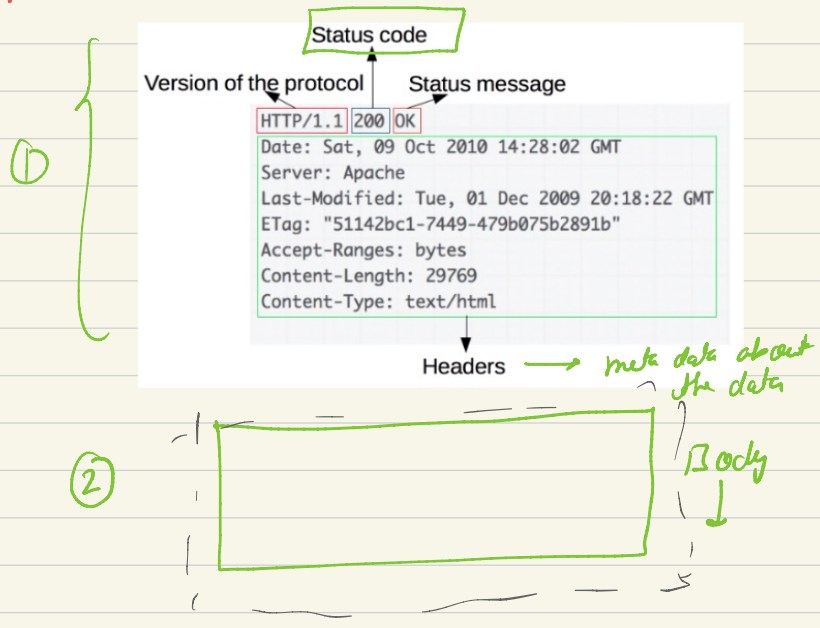
- ↳ 0.9 → contains data
- ↳ 1.1 → header were also introduced
- ↳ 2 → https was the default
- ↳ 3 → it is faster than http2

TCP/UDP

① HTTP Request → header



② HTTP Response →





Req methods → given by client to server

|               |                                   |
|---------------|-----------------------------------|
| <b>GET</b>    | retrieve information              |
| <b>HEAD</b>   | retrieve resource headers         |
| <b>POST</b>   | submit data to the server.        |
| <b>PUT</b>    | save an object at the location    |
| <b>DELETE</b> | delete the object at the location |

PATCH → update something on the server

OPTION → get's you all the methods allowed

Res codes

| HTTP STATUS CODES |                     |
|-------------------|---------------------|
| 2xx Success       |                     |
| 200               | Success / OK        |
| 3xx Redirection   |                     |
| 301               | Permanent Redirect  |
| 302               | Temporary Redirect  |
| 304               | Not Modified        |
| 4xx Client Error  |                     |
| 401               | Unauthorized Error  |
| 403               | Forbidden           |
| 404               | Not Found           |
| 405               | Method Not Allowed  |
| 5xx Server Error  |                     |
| 501               | Not Implemented     |
| 502               | Bad Gateway         |
| 503               | Service Unavailable |
| 504               | Gateway Timeout     |

## # HTTP Headers

- \* **User-Agent:** Identifies the client making the request, such as the browser or software used.
- \* **Content-Type:** Indicates the media type of the resource being sent in the response (e.g., HTML, JSON, XML).
- \* **Cache-Control:** Specifies caching policies, determining how long a resource should be cached.
- \* **Accept:** Specifies the media types that are acceptable for the response, telling the server what kind of content the client can understand.
- \* **Cookie:** Stores information on the client side, maintaining stateful sessions between the server and the browser.