what is CDN? why do we use it

A **CDN (Content Delivery Network)** is a network of servers distributed across various locations worldwide that work together to deliver digital content (such as web pages, images, videos, and other resources) to users in a faster and more efficient manner. The main purpose of a CDN is to reduce latency and improve the loading speed of websites or applications.

Why Do We Use a CDN?

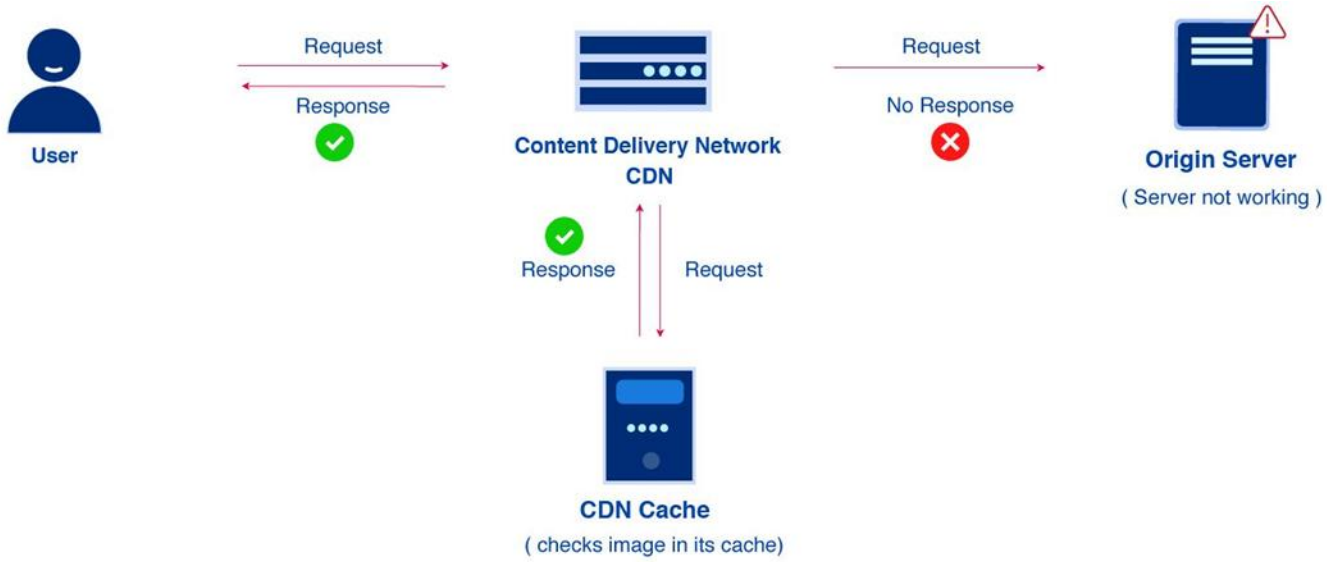
1. **Faster Load Times:**
    - CDNs store copies of content on multiple servers in different geographic locations. When a user requests content from a website, the CDN serves the content from the server closest to the user's location, reducing the time it takes to load the page.
2. **Reduced Latency:**
    - By reducing the physical distance between the user and the server, CDNs help lower the latency, leading to faster access to content.
3. **Improved Website Performance:**
    - CDNs can offload traffic from the origin server, which can help prevent slowdowns, especially during high traffic periods or when serving large files (like images, videos, or software downloads).
4. **Scalability:**
    - CDNs can easily handle spikes in traffic by distributing the load across multiple servers. This makes it easier for websites to scale without needing to upgrade their origin server infrastructure.
5. **Reliability and Redundancy:**
    - CDNs provide redundancy by having multiple copies of the content across various servers. If one server fails or experiences issues, the CDN can quickly redirect requests to another server, improving reliability and uptime.
6. **Global Reach:**
    - A CDN helps businesses reach global audiences more effectively by providing fast content delivery, regardless of the user's geographical location.
7. **Better Security:**

- CDNs can offer security features like DDoS (Distributed Denial of Service) protection, secure sockets layer (SSL) encryption, and web application firewalls (WAF), which enhance the overall security of websites.

8. **Reduced Bandwidth Costs:**
    - By caching content and reducing the number of requests that reach the origin server, CDNs can help reduce bandwidth usage and associated costs.

In Summary:

CDNs are used to optimize website speed, performance, and security by delivering content from the server closest to the user, reducing latency, and distributing traffic efficiently. This is essential for providing a better user experience and ensuring the website can handle high traffic loads.



What is diference between React and ReactDOM

React and **ReactDOM** are closely related but serve different purposes:

- **React** is the core library that allows you to build components and manage the application's state. It provides the building blocks for creating UI components using JavaScript.
- **ReactDOM** is a separate library that specifically handles the interaction between React and the DOM (Document Object Model). It is responsible for rendering React components to the web page and updating the DOM when the component state changes.

Key Difference:

- **React** is used to define components and logic.
- **ReactDOM** is used to render those components into the browser's DOM.

What is difference between react.development.js and react.production.js files via CDN?

When using React via CDN, you'll encounter two main versions of the library: react.development.js and react.production.js. Here's a breakdown of their key differences:

Development Build (react.development.js)

- **Purpose:** Designed for development environments.
- **Features:**
  - Includes helpful warnings and error messages to aid in debugging.
  - Contains extra code for development tools and features.
  - Unminified, making it easier to read and debug.
- **Performance:** Slower and larger in file size due to the additional features and unminified code.

Production Build (react.production.js)

- **Purpose:** Optimized for production environments.
- **Features:**
  - Removes all warnings and error messages to reduce file size and improve performance.
  - Excludes development-specific code and tools.
  - Minified and compressed for faster loading times.
- **Performance:** Faster and smaller in file size, resulting in improved performance for end-users.

Which one to use:

- **Development:** Use react.development.js for local development and testing.
- **Production:** Use react.production.js when deploying your application to a live environment.

What is async and defer?

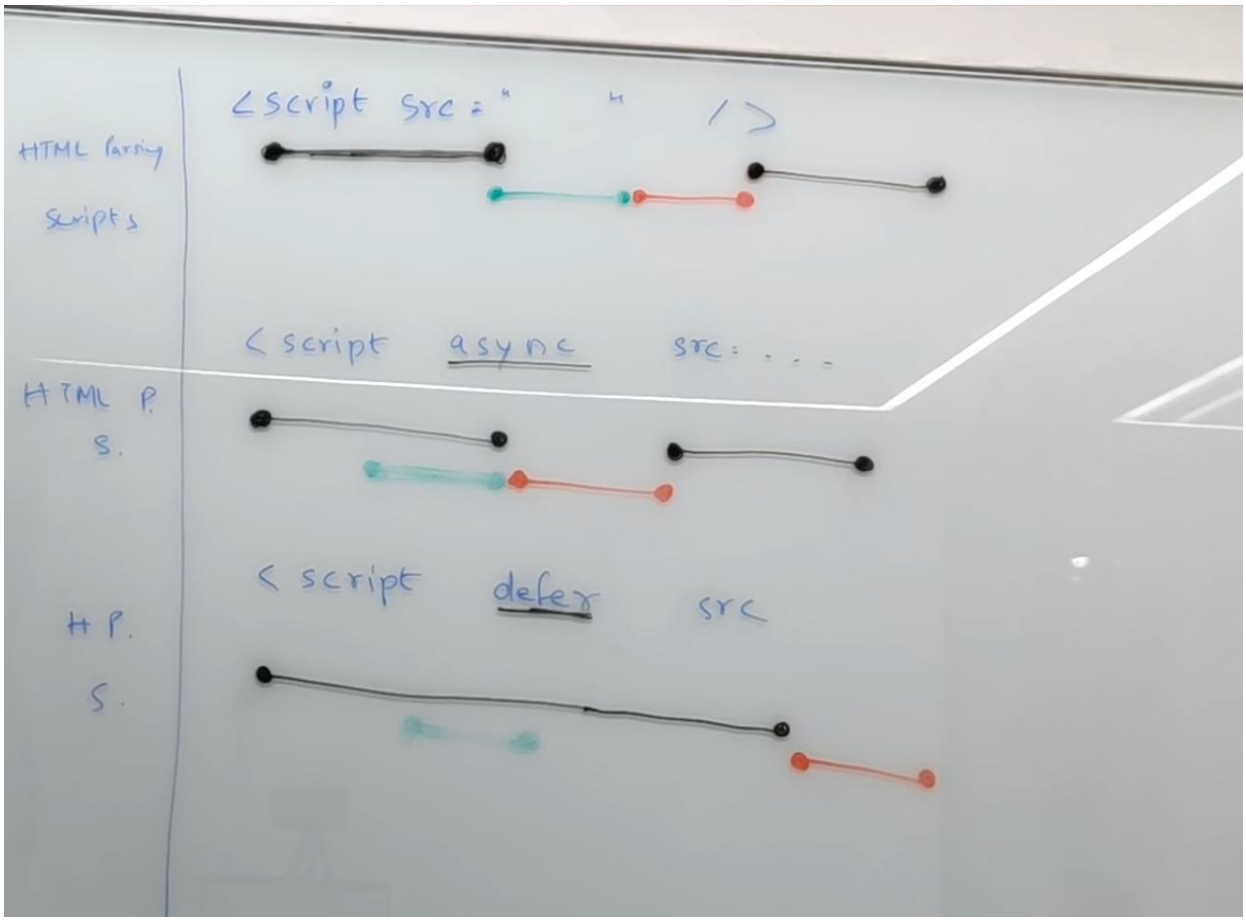
1. async:

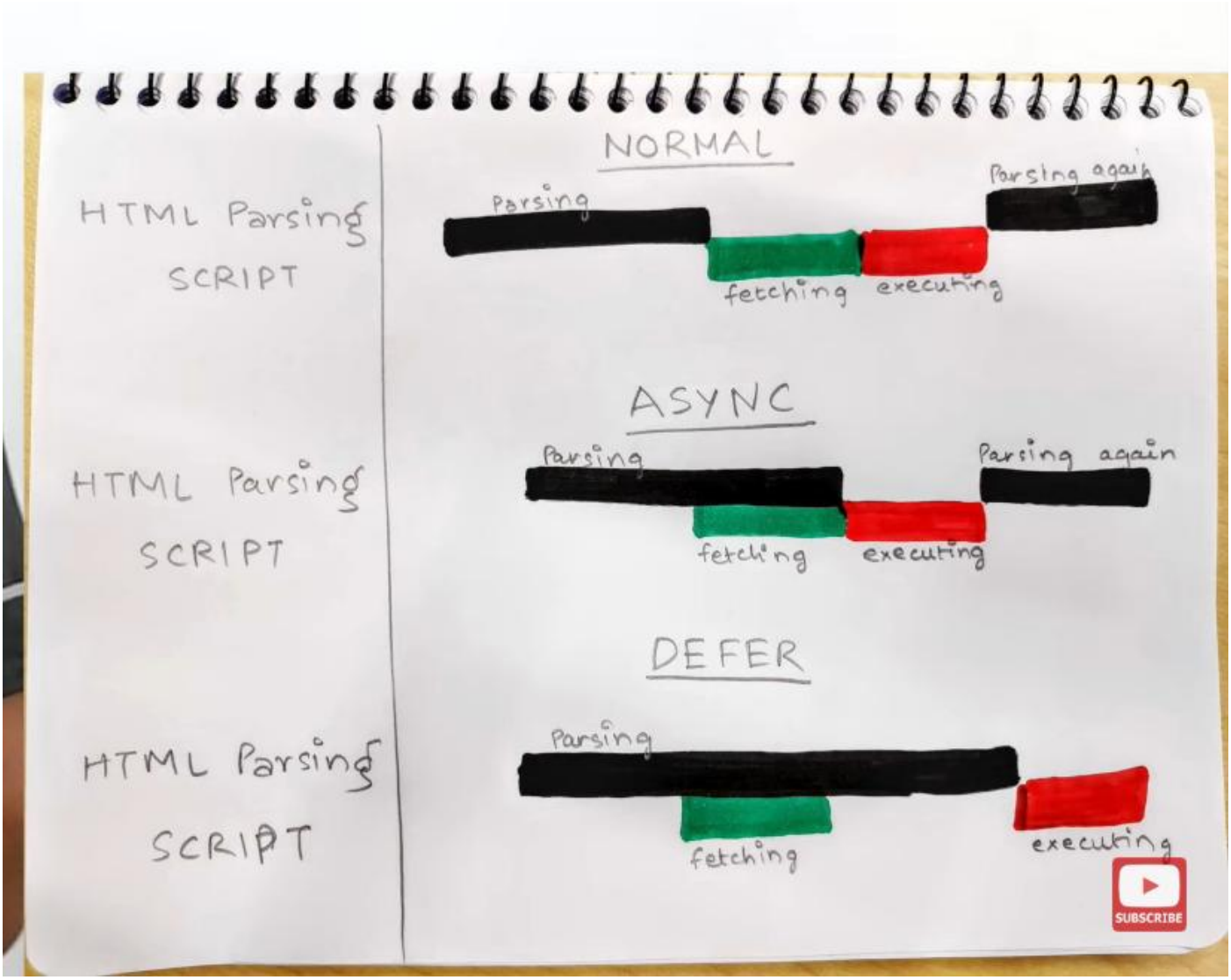
- Scripts are loaded **asynchronously**, meaning the browser doesn't block the rest of the page from rendering while the script is being fetched.

- The script is executed **immediately after it is downloaded**, regardless of the order in which the scripts appear in the HTML.
- Ideal for scripts that don't depend on other scripts or need to block rendering.

2. defer:

- Scripts are loaded **asynchronously**, but the script will only execute **after the HTML document has been completely parsed** (just before the DOMContentLoaded event).
- Maintains the order of scripts as they appear in the HTML.
- Ideal for scripts that depend on the DOM being fully loaded




NORMAL
HTML Parsing
SCRIPT
Parsing
Parsing again
fetching
executing
ASYNC
HTML Parsing
SCRIPT
Parsing
Parsing again
fetching
executing
DEFER
HTML Parsing
SCRIPT
Parsing
fetching
executing
SUBSCRIBE