How static website get loaded in browser?

1.The browser first sends a DNS request with the hostname to the DNS server of your ISP (or your network if any)- DNS finds out the ip address from the hostname.

The DNS server replies with the IP address of the server

The browser then sends the HTTP request This is in the form e.g GET /index.html HTTP/1.1

The server responds with the format

The data is sent to the user

Usually if a webpage is requested, it is in HTML format(with javascript , css , etc). This is then parsed and processed by the browser to get the webpage we see.

**1. User Requests the Website**

* The process begins when a user types a URL into their browser or clicks a link. This URL points to a server hosting the static website.

**2. DNS Resolution**

* The browser contacts a Domain Name System (DNS) server to resolve the domain name (e.g., www.example.com) into an IP address of the server where the website is hosted.

**3. Establishing a Connection**

* Once the IP address is obtained, the browser establishes a connection to the server using the HTTP or HTTPS protocol. This often involves a TCP handshake.

**4. Sending the HTTP Request**

* The browser sends an HTTP GET request to the server, asking for the specific resource (like index.html or any other page).

**5. Server Response**

* The server receives the request and processes it. For a static website, it simply retrieves the requested files (HTML, CSS, JavaScript, images, etc.) from its storage.

**6. Returning the Response**

* The server sends back the requested files as an HTTP response. The response typically includes a status code (e.g., 200 OK for a successful request) and the content of the requested files.

**7. Rendering the Website**

* Once the browser receives the response, it begins rendering the webpage. It processes the HTML to build the Document Object Model (DOM), applies CSS for styling, and executes any JavaScript for interactivity.
* If the HTML file includes links to additional resources (like CSS files or images), the browser will make additional requests for those files.

**8. Displaying the Content**

* Finally, the browser displays the rendered webpage to the user, allowing them to interact with it.

**Key Characteristics of Static Websites**

* **Static Content**: The content does not change based on user interaction or requests. The same HTML, CSS, and JavaScript files are served to every user.
* **Faster Loading**: Since the files are pre-rendered and stored, static websites generally load faster than dynamic sites, which generate content on the fly.
* **Simpler Hosting**: Static websites can be hosted on simple web servers or even content delivery networks (CDNs) without requiring server-side processing.

4.WHAT IS DATATYPE AND EXPLAIN THEIR TYPES

: Data Types in JS:

In JavaScript, data types are broadly categorized into two groups: Primitive Types and Object Types. Understanding these

is fundamental, as they define the way JavaScript handles and stores data.

1. Primitive Types:

Primitive types are immutable (they cannot be altered). When you work with a primitive type, you are dealing

with the actual value itself.

The primary primitive types in JavaScript include:

➢ Number

Represents both integer and floating-point numbers.

Special values: NaN (Not-a-Number) and Infinity.

Example: let age = 25;

➢ String

Represents text, enclosed in single (' '), double (" "), or backticks (` `) for template literals.

Example: let name = "Alice";

➢ Boolean

Represents logical values: true or false.

Example: let isStudent = true;

➢ Undefined

A variable that has been declared but not assigned a value.

Example: let x; // x is undefined

➢ Null

Represents an intentional absence of any object value. It’s often used to signify "no value."

: ➢ Symbol (introduced in ES6)

Represents a unique and immutable identifier, often used as object property keys to avoid naming collisions.

Example: let id = Symbol('id');

➢ BigInt (introduced in ES11)

Allows representation of integers beyond the limit of the Number type (which is 2^53-1 for integers).

Created by appending n to an integer or using the BigInt() function.

Example: let bigNumber = 123456789012345678901234567890n;

2. Object Types

Object types in JavaScript are used to store collections of data and more complex entities.

➢ Object

The primary composite data type in JavaScript, used to store collections of properties and methods.

Example: let person = { name: "Alice", age: 25 };

➢ Array

A special type of object for storing ordered lists.

Example: let numbers = [1, 2, 3, 4, 5];

➢ Function

Functions in JavaScript are also objects and can have properties and methods.

Example:

function greet() {

console.log("Hello!");

}

➢ Date

Represents dates and times.

Example: let today = new Date();