⚡HTML[Frontend-Interview Checklist

🔹 1. Basics

- What is HTML & why is it used?
- Structure of an HTML document (`<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`)
- HTML tags vs. elements vs. attributes
- Block-level vs. Inline elements
- Semantic vs. Non-semantic elements
- Difference between HTML, HTML4, and HTML5
- Difference between HTML & XHTML

🔹 2. Common Tags

- Headings (`<h1>` → `<h6>`)
- Paragraphs (`<p>`)
- Links (`<a>`, absolute vs relative paths)
- Images (`<img>`, `alt`, `srcset`)
- Lists (`<ol>`, `<ul>`, `<dl>`)
- Tables (`<table>`, `<tr>`, `<td>`, `<th>`, `<caption>`, `<thead>`, `<tbody>`, `<tfoot>`)
- Inline-formatting(`<b>`,`<strong>`,`<i>`,`<em>`,`<mark>`,`<sup>`,`<sub>` ,etc.)

🔹 3. Semantic HTML5 Tags

- `<header>`
- `<footer>`
- `<nav>`
- `<article>`
- `<section>`
- `<aside>`
- `<main>`
- `<figure>` & `<figcaption>`
- Why semantic tags matter (SEO & accessibility)

🔹 4. Forms (Very Important 🚨)

- `<form>`,`<input>`,`<textarea>`,`<button>`, `<select>`, `<label>`
- Input types (`text`, `email`, `password`, `number`, `date`, `color`, `file`, etc.).
- HTML5 new input types (`datetime-local`, `month`, `week`, `search`, `tel`, `url`)
- Attributes: `required`, `readonly`, `disabled`, `placeholder`, `autofocus`, `autocomplete`, `pattern`, `min`, `max`, `step`, `multiple.`
- `<datalist>` for auto-suggestions.
- `<output>` element.
- Built-in form validation vs JavaScript validation.

🔹 5. Media & Graphics

- `<audio>`&`<video>`(attributes:`controls`,`autoplay`,`loop`,`muted`, `poster`).
- `<canvas>` (drawing graphics via JS)
- `<svg>` (Scalable Vector Graphics)
- Difference between `<canvas>` vs `<svg>`

🔹 6. HTML5 APIs

- Geolocation API (`navigator.geolocation`)
- Local Storage vs Session Storage vs Cookies
- Web Storage API advantages
- Drag & Drop API
- Web Workers (background scripts)
- WebSockets (real-time communication)
- History API (`pushState`, `replaceState`)

🔹 7. Accessibility & SEO

- `alt` attribute for images
- ARIA roles & attributes (`role`, `aria-label`, etc.)
- Heading hierarchy (`h1` → `h6`)
- `title` vs `alt` vs `aria-label`
- `tabindex` & keyboard navigation
- Accessible forms (`<label>` with `for`)

🔹 8. Meta & Head Elements

- `<title>`
- `<meta charset="UTF-8">`
- `<meta name="viewport">` (responsive design)
- `<meta name="description">` (SEO)
- `<meta name="keywords">`
- `<link>` (CSS, favicon)
- `<script>` (`async` vs `defer`)

🔹 9. HTML5 New Features

- Simplified doctype: `<!DOCTYPE html>`
- Audio & video support without plugins
- Local Storage & Session Storage
- New semantic elements(`<header>`,`<section>`,`<article>`, etc.)
- New form input types & attributes
- Canvas & SVG
- Custom data attributes (`data-*`)

🔹 10. Other Essentials

- Difference between `id` vs `class`
- Inline vs Internal vs External CSS
- Relative vs Absolute vs Fixed vs Sticky paths
- Difference between `<link>` vs `@import`
- Progressive enhancement vs Graceful degradation
- `rel="noopener noreferrer"` (security in links)

- Difference between `<script async>` vs `<script defer>`
- Favicon setup

🎨 CSS Interview & Job Checklist

🔹 1. CSS Basics

- Inline, Internal, External CSS
- CSS Selectors (element, class, id, universal `*`)
- Attribute selectors (`[type="text"]`, `[href^="https"]`)
- Combinators (`div p`, `div > p`, `div + p`, `div ~ p`)
- Pseudo-classes (`:hover`, `:first-child`, `:nth-child()`, etc.)
- Pseudo-elements (`::before`, `::after`, `::first-letter`)
- Difference between relative, absolute, fixed, and sticky units/positioning.

🔹 2. Box Model & Layout

- Box Model (content, padding, border, margin)
- Difference between `inline`, `block`, `inline-block.`
- `display` values (`none`, `inline`, `block`, `inline-block`, `flex`, `grid`).
- CSS units: px, em, rem, %, vh, vw, fr
- `overflow` (`hidden`, `scroll`, `auto`).
- `z-index` and stacking context.

🔹 3. Flexbox & Grid

- Flexbox basics (`display: flex`)
- Flex container properties (`justify-content`, `align-items`, `flex-wrap`, `gap`)
- Flex item properties (`flex-grow`, `flex-shrink`, `flex-basis`, `align-self`)
- CSS Grid basics (`display: grid`)
- Grid-properties(`grid-template-columns`,`grid-template-rows`, `gap`, `grid-area`)
- Difference between Flexbox and Grid

🔹 4. Styling & Effects

- Colors (hex, rgb, rgba, hsl)
- Background-properties(`background-image`, `background-size`, `background-clip`)
- Borders,rounded corners, shadows (`box-shadow`, `text-shadow`)
- Gradients (`linear-gradient`, `radial-gradient`)
- CSS transitions (`transition: all 0.3s ease`)
- CSS animations (`@keyframes`, `animation` shorthand)
- Transformations(`translate`,`rotate`,`scale`,`skew`

🔹 5. Responsive Design

- Media queries (`@media screen and (max-width: 768px)`)
- Mobile-first vs Desktop-first design
- Responsive units (`%`, `vh`, `vw`, `em`, `rem`)

- Viewport meta tag (`<meta name="viewport">`)
- CSS frameworks (Bootstrap, Tailwind) basics

🔹 6. Advanced CSS

- CSS variables (`--primary-color: #333`)
- `calc()` function
- Positioning (`relative`, `absolute`, `fixed`, `sticky`)
- Clip-path & masking
- Filters (`blur`, `brightness`, `contrast`)
- Responsive images (`object-fit`, `srcset`, `picture`)

🔹 7. CSS Interview Essentials

- Difference between `relative`, `absolute`, `fixed`, `sticky`
- Difference between `inline`, `block`, `inline-block`
- Specificity hierarchy (inline > id > class > element)
- CSS cascade & inheritance rules
- Difference between `em` and `rem`
- Difference between `absolute` units (px) and relative units (em, rem, %)
- Difference between CSS Grid and Flexbox

⚡ JavaScript Interview & Job Checklist

🔹 1. JS Basics

- Variables (`var`, `let`, `const`)
- Data types (primitive & reference)
- Type coercion & type conversion
- Operators (arithmetic, comparison, logical, ternary)
- Template literals (backticks ``)
- `==` vs `===`

🔹 2. Functions & Scope

- Function declaration vs function expression
- Arrow functions
- Parameters vs arguments
- Default parameters
- Scope (global, local, block)
- Hoisting (variables & functions), Closures
- Higher-order functions (map, filter, reduce, forEach)

🔹 3. DOM & Events

- DOM selection (`getElementById`, `querySelector`, etc.)
- DOM manipulation (`innerHTML`, `textContent`, `setAttribute`)
- Event handling (`addEventListener`, event bubbling vs capturing)
- Event delegation
- Difference between `target` vs `currentTarget`
- Preventing default actions (`event.preventDefault()`)

🔹 4. Objects & Arrays

- Object literals, properties, and methods
- Object destructuring & array destructuring
- Spread & rest operators
- Array methods (`map`, `filter`, `reduce`, `find`, `some`, `every`)
- `for...in` vs `for...of` loops

🔹 5. Advanced JS Concepts

- Prototype & Prototypal inheritance
- `this` keyword in different contexts
- Call, Apply, Bind
- ES6+ features (let/const, classes, modules, promises, async/await)
- Destructuring & template literals
- Nullish coalescing (`??`) & optional chaining (`?.`)
- Event loop & concurrency model

🔹 6. Asynchronous JavaScript

- Callbacks
- Promises (`.then`, `.catch`)
- Async/await
- Fetch API
- `try...catch` error handling
- AJAX basics

🔹 7. Browser APIs

- LocalStorage, SessionStorage, Cookies
- Geolocation API and History API
- Web Workers basics
- Intersection Observer API

🔹 8. OOP in JavaScript

- Object creation patterns (constructor functions, classes)
- `class` and `extends` (inheritance)
- Encapsulation using closures & private fields
- Static methods

🔹 9. JS Interview Must-Knows

- Difference between `==` and `===`
- Difference between `null`, `undefined`, and `NaN`
- Difference between deep copy vs shallow copy
- Event loop: microtasks vs macrotasks
- Synchronous vs asynchronous code
- Debouncing & Throttling
- Difference b/w`localStorage`,`sessionStorage`,cookies
- Hoisting & temporal dead zone (TDZ)
- Difference between `call`, `apply`, `bind`
- Arrow function vs normal function
- `this` binding rules.

Overview

✅ Week 1 — HTML Foundations

- Learn structure: `<!DOCTYPE>`, `<html>`, `<head>`, `<body>`
- Headings, paragraphs, links, images, lists, tables
- Semantic tags: `<header>`, `<main>`, `<footer>`, `<section>`
- Accessibility basics: `alt` in images, correct links
- Practice: Build a simple portfolio page

✅ Week 2 — HTML Forms & Media

- Learn forms: `<form>`, inputs (`text`, `email`, `password`, etc.)
- Attributes: `required`, `pattern`, `placeholder`
- `<label>` with `for`
- Multimedia: `<audio>`, `<video>`, `<canvas>`, `<svg>`
- Practice: Contact form + embedded video/audio

✅ Weeks 3–4 — CSS Basics & Layout

- CSS selectors, specificity

- Box model: margin, padding, border, content
- Display: `inline`, `block`, `inline-block`, `flex`, `grid`
- Positioning: static, relative, absolute, fixed, sticky
- Flexbox container & item properties

- Practice: Style portfolio page with Flexbox layout

✅ Weeks 5–6 — Responsive Design & Grid

- Media queries (`@media`)
- Mobile-first vs desktop-first
- CSS Grid basics: `grid-template-rows`, `grid-template-columns`
- Responsive units: `%`, `em`, `rem`, `vh`, `vw`
- Responsive images: `object-fit`, `srcset`
- Project: Responsive blog layout with Grid + Flexbox fallback

---

✅ Week 7 — Styling, Effects & Accessibility

- Colors (hex, rgb, rgba, hsl)
- Shadows: `box-shadow`, `text-shadow`
- Gradients: `linear-gradient`, `radial-gradient`
- Transitions: `transition: all 0.3s ease`
- Animations: `@keyframes`, `animation` shorthand
- CSS variables (`--primary-color`)
- Accessibility: ARIA basics, keyboard navigation, focus styles

---

✅ Weeks 8–10 — JavaScript Fundamentals

- Variables: `let`, `const`, `var`

- Data types: primitive vs reference
- Functions (normal, arrow, default params)
- Scope & closures
- Arrays & objects (methods: `map`, `filter`, `reduce`)
- DOM selection & manipulation
- Events, bubbling & delegation
- Practice:
  - To-Do app (add, edit, delete, localStorage)
  - Modal + tabs components

✅ Week 11 — Asynchronous JS & APIs

- Callbacks & promises (`.then`, `.catch`)
- Async/await with `try...catch`
- Fetch API basics
- Error handling & loading states
- Practice:
  - Weather app (fetch public API)
  - GitHub user search app

✅ Week 12 — Advanced JS & Mini Projects

- ES6+ features: destructuring, spread/rest, template literals
- `this`, `call`, `apply`, `bind`
- Event loop basics: microtasks vs macrotasks
- Modules (import/export)
- OOP: classes, inheritance

- Practice:
    - Photo gallery with lightbox + filters
    - Dashboard with charts (Chart.js)

🎯 Ongoing — Interview Prep

- Daily practice: array & string coding questions
- Weekly mock interviews (explain projects out loud)

  Review common questions:

- Event loop & async handling
- Differences: `var/let/const`, `==` vs `===`, arrow vs normal functions
- CSS specificity, box model, flex vs grid

Internet & Web Basics Interview Cheat Sheet

🌍 Internet Basics

Q1. What is the Internet?
👉 A global system of interconnected computer networks that communicate using TCP/IP.

Q2. What is the difference between the Internet and WWW?
👉 Internet = infrastructure (networks, protocols), WWW = service (web pages via HTTP).

Q3. What is an IP address?
👉 A unique numerical label (IPv4/IPv6) that identifies a device on the Internet.

Q4. What is DNS?
👉 Domain Name System – translates domain names (google.com) into IP addresses.

Q5. What is TCP vs UDP?
👉 TCP = reliable, ordered, connection-based; UDP = faster, connectionless, less reliable.

🌐 Web & Browser

Q6. What happens when you type a URL in a browser?
👉 DNS lookup → TCP/HTTPS handshake → Send HTTP request → Server responds → Browser renders.

Q7. What is HTTP vs HTTPS?
👉 HTTP = communication protocol; HTTPS = HTTP + SSL/TLS encryption (secure).

Q8. What is a CDN?
👉 Content Delivery Network – distributes content across global servers for faster delivery.

Q9. What is a cookie?
👉 Small piece of data stored in the browser for sessions, preferences, or tracking.

Q10. What is caching in browsers?
👉 Storing static files locally to reduce load times and server requests.

⚡ Networking Concepts

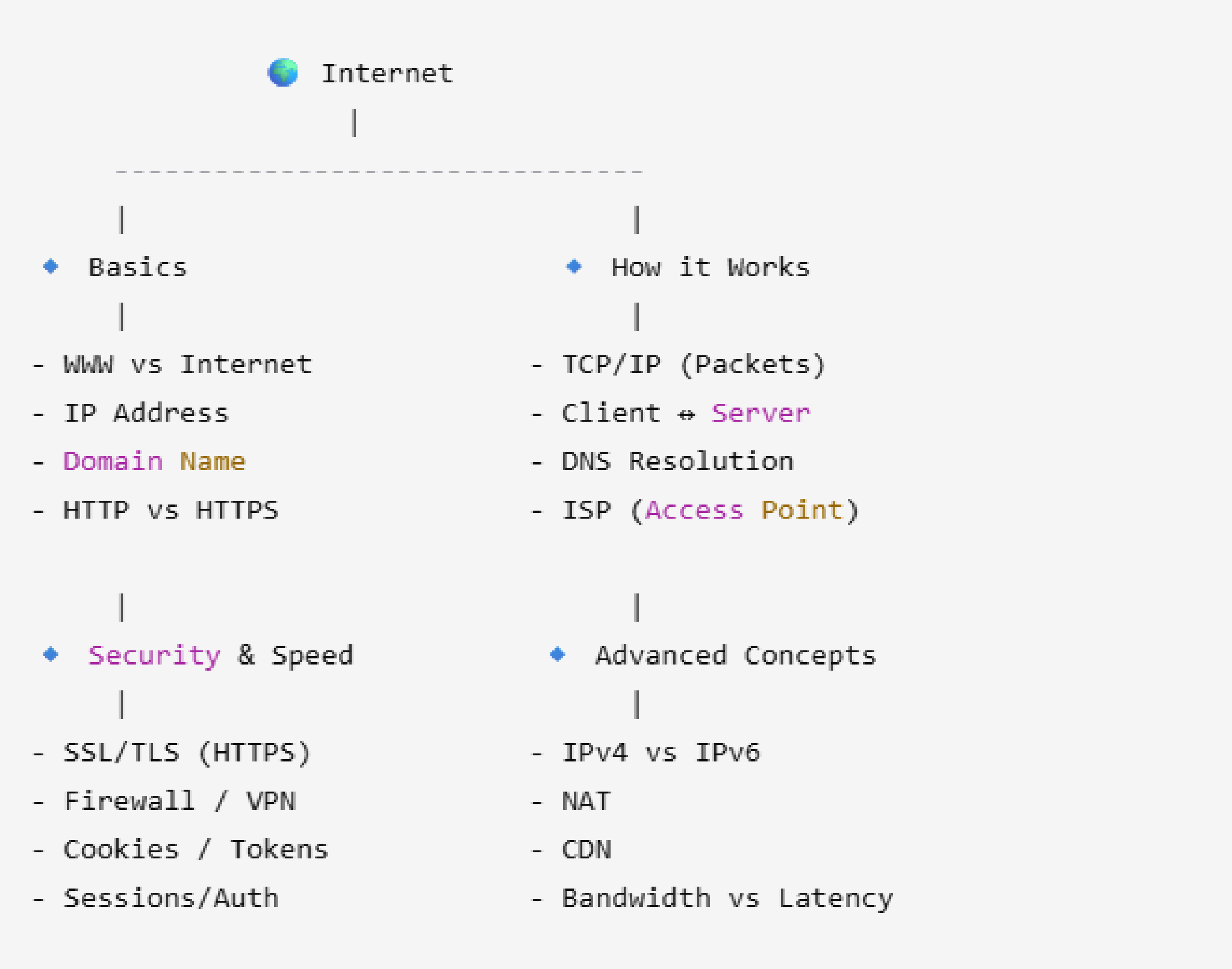
Q11. What is latency vs bandwidth?
👉 Latency = delay (response time), Bandwidth = data transfer capacity per second.

Q12. What is a VPN?
👉 Virtual Private Network – encrypts traffic and hides IP/location.

Q13. What is packet switching?
👉 Breaking data into packets and routing them independently across networks.

Q14. What is NAT?
👉 Network Address Translation – allows multiple private devices to share one public IP.

Q15. What is a firewall?
👉 A security system that monitors and controls incoming/outgoing network traffic.



```
              🌍 Internet
                   |
      ---------------------------------
       |                              |
  🔹 Basics                        🔹 How it Works
       |                              |
  - WWW vs Internet               - TCP/IP (Packets)
  - IP Address                    - Client ↔ Server
  - Domain Name                   - DNS Resolution
  - HTTP vs HTTPS                 - ISP (Access Point)

       |                              |
  🔹 Security & Speed             🔹 Advanced Concepts
       |                              |
  - SSL/TLS (HTTPS)               - IPv4 vs IPv6
  - Firewall / VPN                - NAT
  - Cookies / Tokens              - CDN
  - Sessions/Auth                 - Bandwidth vs Latency
```

Top Most Questions asked In Interview-

Questions-What happens when you type a URL in the browser?

🌍 Browser to Website Journey

1️⃣URL Entry -

- You type https://www.google.com in the browser.
- Browser parses protocol (https), domain(google.com), and path (/).

2️⃣DNS Lookup-

- Browser checks cache (browser → OS → router → ISP).
- If not found → query DNS server.
- Domain → IP address (e.g., 142.250.183.100).

3️⃣TCP Handshake-

- Browser establishes a TCP connection with the server (3-way handshake: SYN → SYN-ACK → ACK).

4️⃣TLS/SSL Handshake (if HTTPS)-

- Browser & server exchange certificates.
- Encryption keys are set up → Secure channel created.

5️⃣HTTP Request-

- Browser sends HTTP request (GET / POST).

  Example:

```
GET / HTTP/1.1

Host: www.google.com

User-Agent: Chrome/140.0
```

6️⃣Server Processing-

- Server receives the request.
- Runs backend logic (Node, PHP, Java, etc.).
- Fetches data from the database if needed.

7️⃣HTTP Response-

- Server sends back HTTP responses (HTML, CSS, JS, JSON, etc.).

  Example:

```
HTTP/1.1 200 OK

Content-Type: text/html
```

8️⃣Browser Rendering-

- Browser parses HTML → builds DOM tree.
- Loads CSS → builds CSSOM.

- Combines → Render Tree.
- Executes JavaScript.
- Loads images, fonts, other resources.

9️⃣Painting & Display-

- The browser paints pixels on the screen.
- You see the fully rendered web page 🎉.

🔹 Basic Level

1. What is the Internet? How is it different from the World Wide Web?
2. What are IP addresses and why are they important?
3. What is DNS and how does it work?
4. What is HTTP vs HTTPS? Why is HTTPS important?
5. What is a domain name? How does it resolve to an IP address?

🔹 Intermediate Level

1. What is an ISP (Internet Service Provider)?
2. Explain how data is transmitted over the Internet (packets, routers, TCP/IP).
3. What is latency vs bandwidth?
4. What is NAT (Network Address Translation)? Why is it used?
5. What is a CDN (Content Delivery Network) and why do we use it?

Explain the concept of client-server architecture on the Internet.

🔹 Advanced Level

1. What is IPv4 vs IPv6? Why do we need IPv6?
2. Explain how SSL/TLS works in HTTPS.
3. How does TCP differ from UDP? Give examples of use cases.
4. What is a VPN and how does it work?

5. What are cookies, sessions, and tokens in the context of the Internet?
6. What is a firewall and how does it secure Internet communication?
7. What is traceroute/ping and how do they help diagnose Internet issues?