HTML

Beginner / Basics

1. **Introduction to HTML**

   - Structure of an HTML document (`<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`).

2. **Basic HTML Tags**

   - Headings (`<h1>` … `<h6>`)

   - Paragraphs (`<p>`)

   - Line breaks (`<br>`)

   - Horizontal rules (`<hr>`)

3. **Text Formatting**

   - Bold (`<b>`, `<strong>`)

   - Italic (`<i>`, `<em>`)

   - Underline (`<u>`), Strike (`<s>`), Subscript (`<sub>`), Superscript (`<sup>`)

   - Quotations (`<q>`, `<blockquote>`), Code snippets (`<code>`)

4. **Lists**

   - Ordered lists (`<ol>`)

   - Unordered lists (`<ul>`)

   - Definition lists (`<dl>`)

5. **Links and Navigation**

- Anchor tag (`<a href="">`

6. **Images**

    - `<img>` tag

    - `src`, `alt`, `title`, `width`, `height` attributes

Intermediate

7. **Tables**

    - `<table>`, `<tr>`, `<td>`, `<th>`

    - Table attributes (`colspan`, `rowspan`, `border`)

    - Table captions (`<caption>`)

8. **Forms**

    - `<form>` tag and attributes (`action`, `method`)

    - Input types (`text`, `password`, `email`, `checkbox`, `radio`, `file`, `date`, `color`, `range`)

    - `<label>`, `<select>`, `<option>`, `<textarea>`, `<button>`

    - Form validation attributes (`required`, `min`, `max`, `pattern`, `maxlength`)

9. **Semantic HTML**

    - `<header>`, `<nav>`, `<main>`, `<section>`, `<article>`, `<aside>`, `<footer>`

    - Importance of semantic structure for SEO & accessibility

10. **Multimedia**

- `<audio>` and `<video>`

- Attributes (`controls`, `autoplay`, `loop`, `muted`)

- Embedding YouTube videos (`<iframe>`)

11. Block vs Inline Elements

- Block elements (`<div>`, `<p>`, `<h1>` …)

- Inline elements (`<span>`, `<a>`, `<img>`)

12. HTML Attributes

- Global attributes (`id`, `class`, `style`, `title`)

Advanced

14. HTML5 Features

- New form input types (`email`, `date`, `number`, `url`)

- `<canvas>` for graphics

- `<svg>` for scalable graphics

- `<progress>`, `<meter>`, `<details>`, `<summary>`

- `<dialog>`

15. Accessibility

- `alt` attributes for images

- ARIA roles (`role="button"`, `aria-label`)

16. Meta Information

- `<meta charset="UTF-8">`

- Viewport for responsive design (`<meta name="viewport">`)

- SEO meta tags (`description`, `keywords`, `author`)

17. HTML Best Practices

- Clean & semantic code

- Indentation & comments

- Using external CSS & JS instead of inline code

18. Latest HTML Updates

- New attributes (e.g., `loading="lazy"` for images, `autocomplete`)

- Native `<dialog>` & `<template>` support

- Shadow DOM (with Web Components)

CSS

Beginner / Basics

1. **Introduction to CSS**

   - What is CSS?

   - Inline, Internal, External CSS

   - Syntax (selectors, properties, values)

2. **Selectors**

   - Universal (`*`)

   - Element/tag selector

   - Class (`.classname`) & ID (`#idname`)

   - Grouping selectors

   - Attribute selectors (`[type="text"]`)

3. **Colors & Backgrounds**

   - Color formats (name, HEX, RGB, RGBA, HSL, HSLA)

   - Background color, image, repeat, position, size

   - Gradients (linear, radial, conic)

4. **Text & Fonts**

   - Font family, size, weight, style

   - Line height, letter spacing, word spacing

   - Text decoration, alignment, transform, shadow

- ○ Google Fonts & custom fonts (`@font-face`)

5. **Box Model**

   - ○ Content, Padding, Border, Margin

   - ○ Border styles (solid, dotted, dashed, radius)

   - ○ Box sizing (`content-box`, `border-box`)
   - ○ CSS variables (`--primary-color`)

Intermediate

6. **Display & Visibility**

   - ○ `block`, `inline`, `inline-block`, `none`

   - ○ `visibility: hidden` vs `display: none`

7. **Positioning**

   - ○ Static, Relative, Absolute, Fixed, Sticky

   - ○ `z-index` stacking order

8. **Flexbox**

   - ○ `display: flex`

   - ○ `justify-content`, `align-items`, `align-self`

   - ○ `flex-wrap`, `flex-grow`, `flex-shrink`, `flex-basis`

   - ○ Building layouts with flexbox

9. **Grid Layout**

- • `display: grid`

- Rows, columns, gap

- `grid-template-rows`, `grid-template-columns`

- `grid-area`, `grid-template-areas`

- Auto-fit & auto-fill

11. **Lists & Tables Styling**

- Removing list styles (`list-style-type: none`)

- Custom bullet points

- Table borders, hover effects

12. **Forms Styling**

- Input, select, button styles

- Placeholder styling

- Checkbox & radio button custom styles

Advanced

13. **Transitions & Animations**

- CSS transitions (`transition: all 0.3s ease`)

- `@keyframes` animations

- Animation properties (`duration`, `delay`, `iteration-count`)

14. **Transforms**

- 2D (`translate`, `rotate`, `scale`, `skew`)

- 3D transforms (perspective, rotateX, rotateY)

15. **Responsive Design**

- Media queries (`@media screen and (max-width: 768px)`)

- Mobile-first vs desktop-first approach

- Responsive images (`object-fit`, `max-width`)

16. **Advanced Selectors**

- `:not(), :nth-child(), :nth-of-type(), :is()`

- Combinators (`>`, `+`, `~`)

17. **CSS Functions**

- `calc(), min(), max(), clamp()`

- `var()` for CSS variables

18. **Shadows & Effects**

- Box-shadow

- Text-shadow

- Filter effects (`blur`, `grayscale`, `brightness`)

- Glassmorphism, Neumorphism

19. **CSS Layout Techniques**

- Multi-column layout

- CSS shapes (`shape-outside`)

- Sticky & fixed headers/footers

- Centering techniques

JavaScript

Beginner / Basics

1. **Introduction**

   - What is JavaScript?

   - How to include JS in HTML (inline, internal, external script)

2. **JavaScript Basics**

   - `console.log()` and debugging

   - Variables (`var`, `let`, `const`)

   - Data types (string, number, boolean, null, undefined, symbol, bigint)

3. **Operators**

   - Arithmetic operators (`+`, `-`, `*`, `/`, `%`, `**`)

   - Assignment operators (`=`, `+=`, `-=`, etc.)

   - Comparison operators (`==`, `===`, `!=`, `!==`, `>`, `<`, `>=`, `<=`)

   - Logical operators (`&&`, `||`, `!`)

   - Ternary operator (`condition ? value1 : value2`)

4. **Strings**

   - String creation (`""`, `''`, template literals)

   - String methods (`length`, `toUpperCase()`, `slice()`, `split()`, `includes()`, etc.)

   - Template literals & interpolation

5. **Numbers & Math**

   - Number methods (`toFixed`, `toString`, `isNaN`)

   - Math object (`Math.round`, `Math.floor`, `Math.ceil`, `Math.random`, `Math.max`, `Math.min`)

6. **Control Flow**

   - `if`, `else if`, `else`

   - `switch` statement

   - Loops (`for`, `while`, `do while`, `for...in`, `for...of`)

   - `break` & `continue`

Intermediate

7. **Functions**

   - Function declaration & expression

   - Arrow functions (`() => {}`)

   - Callback functions

8. **Arrays**

   - Array creation & indexing

   - Array methods (`push`, `pop`, `shift`, `unshift`, `splice`, `slice`, `concat`, `indexOf`, `includes`)

   - Iteration methods (`forEach`, `map`, `filter`, `reduce`, `find`, `some`, `every`, `sort`)

   - Spread & rest operators (`...`)

9. **Objects**

- Object creation (`{}`)

- Properties & methods

- Object methods (`keys`, `values`, `entries`, `assign`)

- Destructuring objects

- `this` keyword

10. **Date & Time**

- `Date()` object

- Getting & setting time (`getDate`, `getMonth`, `getFullYear`, etc.)

- Formatting dates

11. **Events**

- Event handling (`onclick`, `onchange`, `onmouseover`, etc.)

- `addEventListener`

- Event bubbling & capturing

- Preventing default actions (`event.preventDefault()`)

12. **DOM Manipulation**

- `document.getElementById`, `getElementsByClassName`, `querySelector`

- Changing content (`innerHTML`, `textContent`)

- Changing styles (`element.style`)

- Creating & removing elements (`createElement`, `appendChild`, `removeChild`)

- Traversing the DOM (`parentNode`, `children`, `nextElementSibling`)

Advanced JavaScript

13. **ES6+ Features**

- `let`, `const`, arrow functions

- Template literals

- Destructuring (arrays & objects)

- Modules (`import`, `export`)

- Default & named exports

- `...spread` & `...rest` operators

14. **Closures & Scope**

- Block scope, function scope, global scope

- Hoisting (variables & functions)

- Closures and practical uses

15. **Objects in Depth**

- Object prototypes

- `Object.create()`

- Constructor functions

- ES6 Classes (`class`, `constructor`, `extends`, `super`)

- Getters & setters

16. **Asynchronous JavaScript**

- Synchronous vs asynchronous

- `setTimeout`, `setInterval`

- Callbacks & callback hell

- Promises (`.then`, `.catch`, `.finally`)

- Async/Await

- `Promise.all`, `Promise.race`

17. Error Handling

- `try...catch`

- `throw` keyword

- Custom error messages

18. Advanced Array & Object Handling

- Higher-order functions

- `reduce()` advanced use cases

- Deep vs shallow copy (`Object.assign`, `structuredClone`)

19. Regular Expressions

- Creating regex (`/pattern/`)

- Regex methods (`test`, `match`, `replace`)

- Common regex patterns

20. Local Storage & Session Storage

- `localStorage.setItem`, `getItem`, `removeItem`, `clear`

- `sessionStorage`

21. JSON

- `JSON.stringify()`

- `JSON.parse()`

- Fetching & handling JSON data

22. **Modules & Import/Export**

- ES modules (`import`/`export`)

- Dynamic imports

- CommonJS (`require`, `module.exports`)

Expert / Professional-Level JS

24. **JavaScript Engines & Internals**

- How JS works (V8 engine, memory heap, call stack)

- Execution context

- Event loop & task queue

- Microtasks vs macrotasks

25. **Functional Programming Concepts**

- Pure functions

- Immutability

- Higher-order functions

26. **OOP in JavaScript**

- Encapsulation, Abstraction, Inheritance, Polymorphism

- Classes vs Prototypes

27. **Advanced Asynchronous Patterns**

- Event loop in depth

- Async iterators

- Observables (RxJS basics)

28. **Advanced Error Handling**

- Custom error classes

- Global error handling (`window.onerror`)

JQuery

Beginner / Basics

1. **Introduction**

   - What is jQuery?

   - Why use jQuery (advantages over plain JS)?

   - Adding jQuery to a webpage (CDN, local file).

   - `$(document).ready()` function.

2. **Selectors**

   - Basic selectors (`#id`, `.class`, `element`)

   - Universal selector (`*`)

   - Grouping selectors (`div, p, span`)

   - Attribute selectors (`[href]`, `[type="text"]`, `[name^="a"]`)

   - Hierarchy selectors (`parent > child`, `ancestor descendant`, `next + prev`, `next ~ siblings`)

3. **CSS Manipulation**

   - `.css()` to get/set CSS properties

   - `.addClass()`, `.removeClass()`, `.toggleClass()`, `.hasClass()`

   - `.width()`, `.height()`, `.innerWidth()`, `.outerWidth()`

4. **Content & Attributes**

   - `.html()` vs `.text()`

- ○ `.attr()` & `.removeAttr()`

- ○ `.val()` for form fields

- ○ `.prop()` vs `.attr()`

Intermediate

5. **DOM Manipulation**

   - ○ Inserting elements: `.append()`, `.prepend()`, `.after()`, `.before()`

   - ○ Removing elements: `.remove()`, `.empty()`

   - ○ Cloning: `.clone()`

   - ○ Wrapping elements: `.wrap()`, `.unwrap()`, `.wrapAll()`

6. **Traversing the DOM**

   - ○ `.parent()`, `.parents()`, `.closest()`

   - ○ `.children()`, `.find()`

   - ○ `.siblings()`, `.next()`, `.prev()`

   - ○ `.first()`, `.last()`, `.eq()`

   - ○ `.filter()`, `.not()`

7. **Events**

   - ○ Event binding: `.on()`, `.off()`, `.one()`

- Common events: `.click()`, `.dblclick()`, `.hover()`, `.focus()`, `.blur()`, `.keyup()`, `.keydown()`, `.submit()`, `.change()`

- Event object (`event.preventDefault()`, `event.stopPropagation()`)

- Event delegation

8. **Effects & Animations**

- Show/Hide: `.show()`, `.hide()`, `.toggle()`

- Fading: `.fadeIn()`, `.fadeOut()`, `.fadeToggle()`, `.fadeTo()`

- Sliding: `.slideDown()`, `.slideUp()`, `.slideToggle()`

- Custom animation: `.animate()`

- Stopping animations: `.stop()`, `.finish()`

9. **Chaining**

- Using multiple methods in one statement

- Example:
  `$("#box").css("color","red").slideUp(2000).slideDown(2000);`

Advanced

10. **AJAX with jQuery**

- `.load()` for partial HTML loading

- `$.get()` & `$.post()`

- `$.ajax()` (full AJAX control)

- Handling JSON data

- `.done()`, `.fail()`, `.always()` callbacks

11. **Utilities**

- `$.each()` vs `.each()`

- `$.trim()`

- `$.isNumeric()`, `$.type()`

- `$.extend()` (merging objects)

- `$.noConflict()`

12. **Form Handling**

- Getting and setting input values with `.val()`

- Form validation basics

- `.serialize()` and `.serializeArray()`

13. **Selectors & Filters (Advanced)**

- `:first`, `:last`, `:even`, `:odd`

- `:eq()`, `:gt()`, `:lt()`

- `:contains()`, `:has()`, `:empty`, `:visible`, `:hidden`

14. **Plugins**

- What are jQuery plugins?

- Installing & using plugins

- Creating custom plugins

15. **Deferred & Promises**

- Introduction to deferred objects

- `.done()`, `.fail()`, `.always()`

- `$.when()`

16. **jQuery UI (optional add-on library)**

- Widgets (datepicker, dialog, tabs, accordion)

- Effects (bounce, explode, highlight, shake)

- Interactions (draggable, droppable, resizable, sortable)

React Js

Beginner / Basics

1. **Introduction**

   - What is React.js?

   - SPA (Single Page Applications)

   - Virtual DOM

2. **Environment Setup**

   - Node.js, npm,

   - Create React App (CRA), Vite, Next.js starter

   - Project structure

3. **JSX (JavaScript XML)**

   - Embedding expressions in JSX

   - JSX rules (`className`, self-closing tags)

   - Fragments (`<> </>`)

4. **Components**

   - Functional components

   - Class components (legacy)

   - Props (passing data to components)

   - Children props

   - Default props & PropTypes

5. **Rendering**

   ○ Rendering lists (`.map()`)

   ○ Keys in lists

   ○ Conditional rendering (`if`, ternary, `&&`)

Intermediate

6. **State Management (Core React)**

   ○ `useState` hook

   ○ Updating state correctly

   ○ Props vs state

7. **Events Handling**

   ○ onClick, onChange, onSubmit, etc.

   ○ Passing arguments to event handlers

8. **Forms**

   ○ Controlled components

   ○ Uncontrolled components (`useRef`)

   ○ Handling inputs, checkboxes, radios, selects

9. **React Hooks (Core)**

   ○ `useState`

   ○ `useEffect` (side effects, lifecycle replacement)

- ○ `useRef`

- ○ `useContext`

10. **React Lifecycle (Class Components)**

- `componentDidMount`, `componentDidUpdate`, `componentWillUnmount`

- Why hooks replaced lifecycle methods

11. **Styling**

- Inline styles

- CSS Modules

- Styled Components

Advanced

12. **React Router**

- `<BrowserRouter>`, `<Routes>`, `<Route>`

- `<Link>` & `useNavigate`

- URL Params (`useParams`)

- Query Params (`useSearchParams`)

- Nested & Protected routes

13. **React Hooks (Advanced)**

- `useReducer`

- `useMemo`

- `useCallback`

- `useLayoutEffect`

- Custom hooks

14. **Context API**

- Create & provide context

- `useContext` for consuming
-
16. **Data Fetching**

- Fetch API

- Axios

- Handling loading & error states

18. **Performance Optimization**

- `React.memo`

- `useMemo`, `useCallback`

- Lazy loading components (`React.lazy`, `Suspense`)

- Code splitting

---

19. **Forms (Advanced)**

- Form

- React Hook Form

- Validation

20. Animations

- CSS animations in React

21. React with Backend

- REST APIs