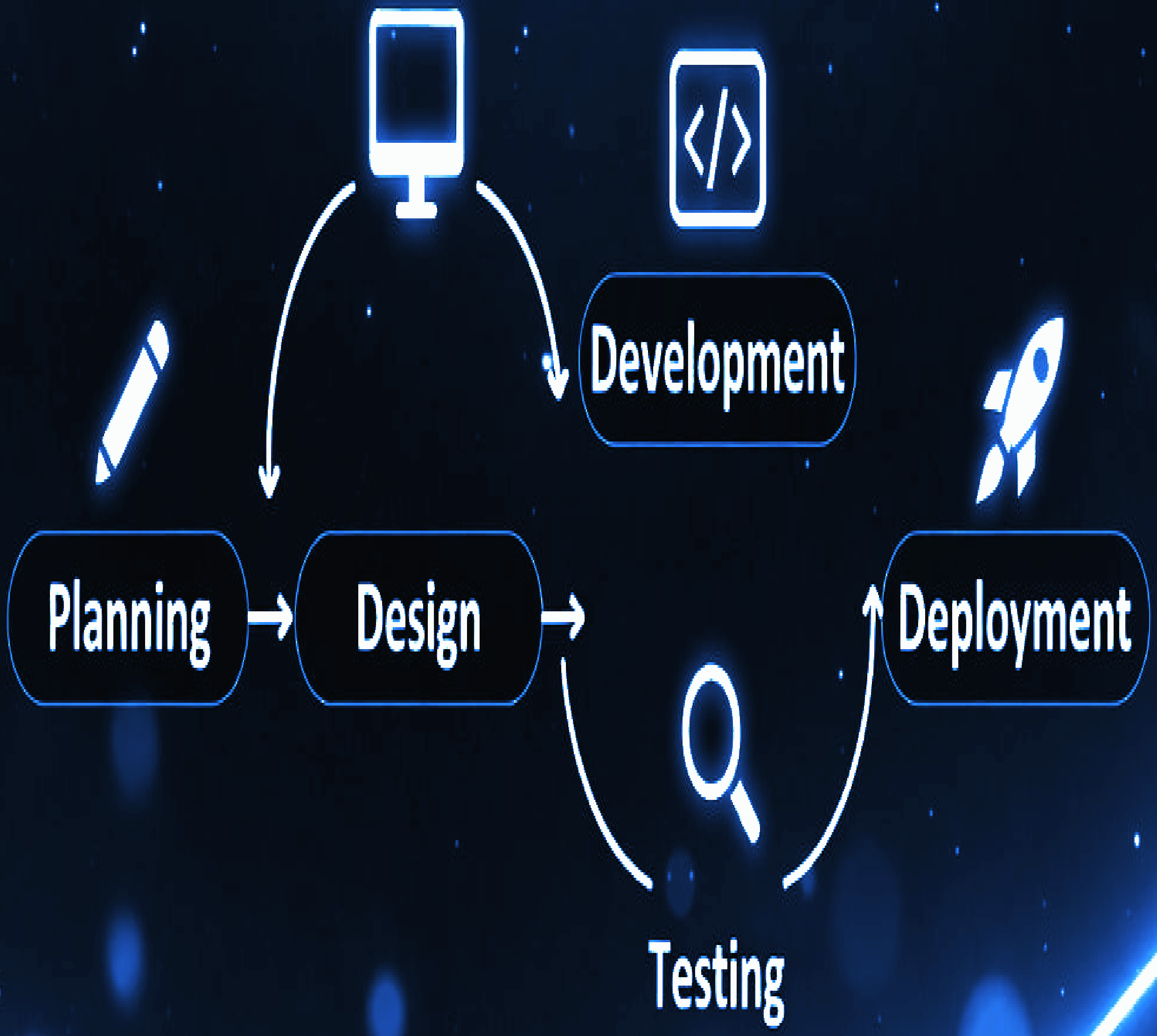
Introduction to Web Development

Day 14



Introduction to Website Development

Website development is the full process of planning, designing, coding, testing, launching, and maintaining websites or web applications that run in browsers.

Uses: Create anything from personal blogs and company sites to e-commerce stores, dashboards, or social platforms.

Working Procedure (simple flow):

1. Idea & planning → 2. Design (wireframes) → 3. Front-end coding → 4. Back-end coding + database → 5. Testing → 6. Deploy to server/hosting → 7. Maintain & update.

What is a Website?

Definition: A website is a collection of related web pages (usually under one domain) hosted on a server, accessible via internet using a URL (e.g., google.com), displaying content through browsers.

Uses: Share info (news/blogs), sell products, provide services, entertain, or build communities.

Working Procedure:

User enters URL → Browser requests page from server → Server sends HTML/CSS/JS + data → Browser renders visible page + interactivity.

Websites are digital storefronts/homes. Understand client → server flow – that's the magic behind every site you visit!

Front-End and Back-End

Web Development is two parts Front-End and Back-End

- **Front-end** (client-side): What users see and interact with directly in the browser.
- **Back-end** (server-side): Behind-the-scenes logic, data storage, security, and processing.

Uses:

Front-end → beautiful, responsive UI.

Back-end → secure data handling, authentication, calculations.

Working Procedure:
Front-end: Browser executes HTML/CSS/JS.

Back-end: Server (e.g., Node.js) processes requests, talks to database, sends response.

Key Differences:

Aspect	Front-End	Back-End
Runs on	User's device (browser)	Server
Main languages	HTML, CSS, JavaScript	Node.js, Python, Java, Go, PHP
Focus	Design, interactivity, UX	Logic, security, databases
Visible?	Yes (what you see)	No (hidden engine)
Common risks	XSS, UI spoofing	SQL injection, data breaches

Front = face & smile; Back = brain & heart.

Front-End Technologies

Tools/languages for building the visible, interactive part of websites.

The three pillars of the frontend are:

1. **HTML:** The "Skeleton" (provides structure).
2. **CSS:** The "Skin/Clothes" (provides style and layout).
3. **JavaScript:** The "Brain/Nervous System" (provides interactivity).

Uses: Responsive layouts, animations, forms, single-page apps (SPA).

Working Procedure: Structure (HTML) → Style (CSS) → Behavior (JS) → Bundle & optimize.

Front-end makes sites feel alive.

Back-End Technologies

Server-side tools handling data, logic, APIs, security.

Popular:

- Node.js (~45-50% usage)
- Python (Django/FastAPI) – huge AI growth
- Java (Spring Boot) – enterprise
- Go (fast & efficient)
- PHP (still strong for WordPress)

Uses: User login, payments, APIs, database ops.

Working Procedure: Receive request → Process (auth, query DB) → Return JSON/response.

Back-end is the secure engine.

Most Popular Web Technologies

Top (from surveys Stack Overflow):

Category	Leader	Approx. Usage
Language	JavaScript	~66%
Front-End	React	~40-45%
Back-End	Node.js	~45-50%
Styling	Tailwind CSS	Rising fast
Full-Stack	Next.js	Dominant
Others	TypeScript, Python	Strong growth

Tools Used to Build Websites

Popular:

- Editor: VS Code (king)
- Design: Figma
- Version control: Git + GitHub
- Hosting: Vercel, Netlify, AWS
- Testing: Cypress, Jest
- Others: Docker, Postman

Uses: Speed up coding, collaboration, deployment.

Introduction to Programming Languages

Definition: Formal languages to give instructions to computers.

Web-relevant: JS (browser + server), Python (easy back-end), TypeScript (typed JS).

Uses: Logic, automation, data handling.

Introduction to HTML

Definition: HyperText Markup Language – structures web content using tags.

HTML Living Standard (continuous updates, no "HTML6" – just evolving HTML).

Uses: Define headings, paragraphs, images, links.

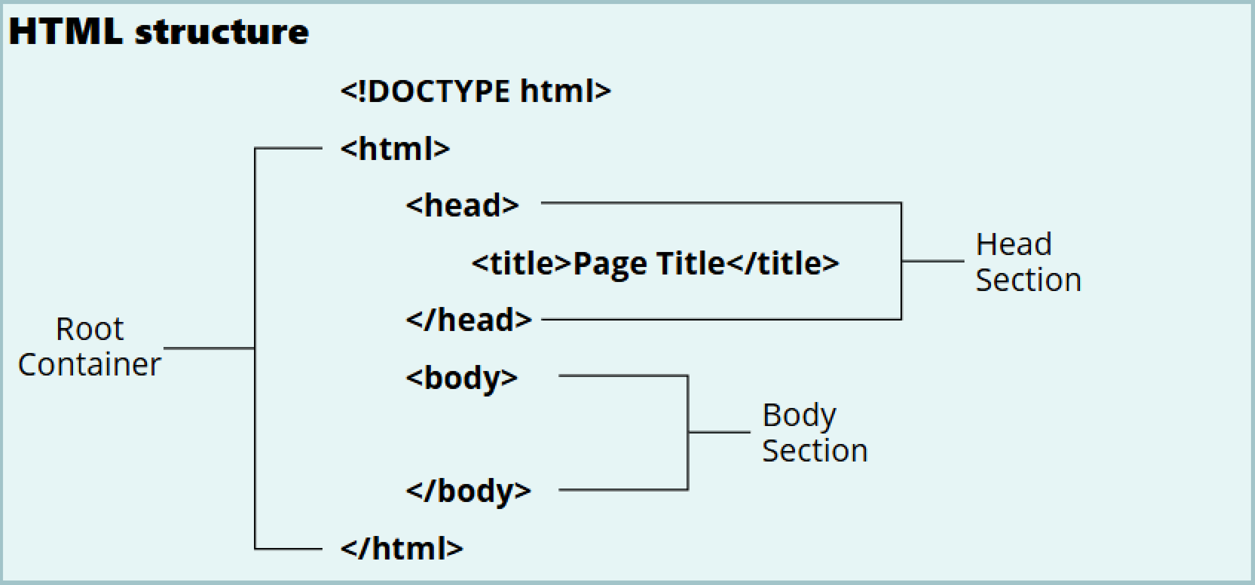
Working Procedure: Write tags in .html file → browser reads & displays.

HTML = bones of every page.

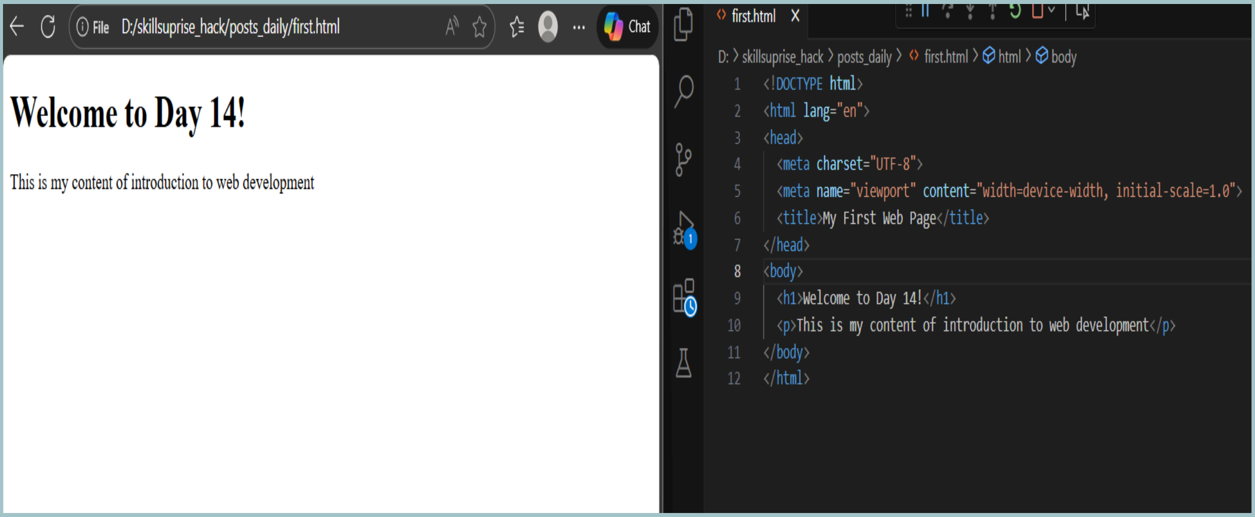
HTML Structure

Standard skeleton every HTML page follows.

Basic Structure (2026 standard):



Example program used vscode as code editor and msedge as browser


File D:/skillsuprise_hack/posts_daily/first.html
Chat
first.html X
D: > skillsuprise_hack > posts_daily > first.html > html > body

Welcome to Day 14!

This is my content of introduction to web development

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My First Web Page</title>
</head>
<body>
  <h1>Welcome to Day 14!</h1>
  <p>This is my content of introduction to web development</p>
</body>
</html>
```

Uses: Tells browser how to interpret document.

Key parts:

- <!DOCTYPE html> – modern standard
- <head> – metadata (title, links)
- <body> – visible content

Example: Copy above → save as index.html → open in browser!

Summary: Perfect structure = reliable pages. Always include DOCTYPE + viewport.