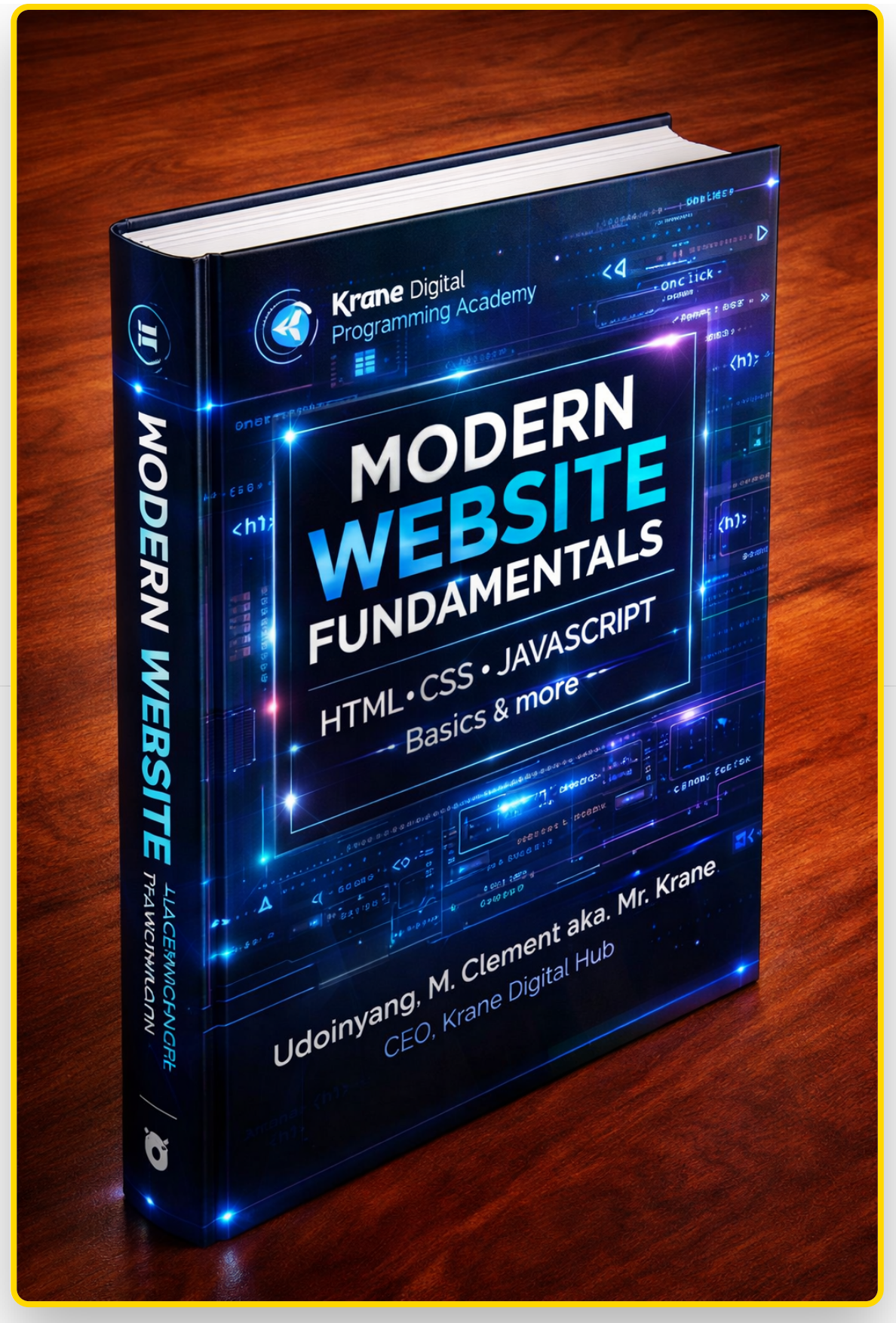
From Zero to Professional Web Developer


Krane Digital
Programming Academy
MODERN
WEBSITE
FUNDAMENTALS
HTML • CSS • JAVASCRIPT
Basics & more
Udoinyang, M. Clement aka. Mr. Krane
CEO, Krane Digital Hub
MODERN WEBSITE

From Zero to Professional Web Developer

A Comprehensive Guide to

HTML, CSS, JavaScript, PHP

and Modern Web Development

Udoinyang, M. Clement (Mr. Krane)

Krane Digital Programming Academy

Krane Digital Incorporation

First Edition, 2026

© 2026 Krane Digital Incorporation

Modern Website Fundamentals: From Zero to Professional Web Developer
By Udoinyang, M. Clement (Mr. Krane)

ISBN: 978-978-001-234-5 (Hardcover)
ISBN: 978-978-001-235-2 (PDF Edition)

Published by: Krane Digital Incorporation
Krane Digital Hub
https://kranedigitalhub.netlify.app
Email: kranedigitalsystemz@gmail.com

FIRST EDITION: First Edition, 2026

Library of Congress Cataloging-in-Publication Data: 2026954321

Printed in: Nigeria

Dedicated to

Every Beginner

who ever felt intimidated by code

May this book be your guiding light through the maze of web development

"Every expert was once a beginner. The only difference is they never stopped learning."

— Mr. Krane

Acknowledgments

This book would not have been possible without the support and encouragement of many individuals and organizations. The author would like to express sincere gratitude to:

- **The Krane Digital Hub Team** — for their unwavering support, late-night coding sessions, and dedication to the vision of "Empowering Dreams"
- **All Students** — whose questions, feedback, and enthusiasm shaped every page of this material. You are the reason this book exists.
- **The Open Source Community** — for creating and maintaining the incredible tools and libraries that make web development accessible to all
- **Family and Friends** — for their patience during countless writing hours, late nights, and weekends sacrificed for this project
- **Beta Readers** — who reviewed early drafts and provided invaluable feedback that improved every chapter
- **Mentors and Teachers** — at Springwood Information Technology and the University of Uyo, who inspired a lifelong love of learning and teaching

Special thanks to Springwood Information Technology, Uyo, for providing the foundational knowledge, and the University of Uyo, Faculty of Vocational Education, for shaping the pedagogical approach used throughout this book.

Udoinyang, M. Clement (Mr. Krane)
Uyo, Akwa Ibom State
2026

Contents

Part I: Foundation

Part II: HTML5 Fundamentals

Part III: CSS3 Mastery

Part IV: JavaScript Deep Dive

Part V: PHP & Backend

Part VI: Advanced Topics

Appendices

Back Matter

CHAPTER 1

How the Web Works

Key Terms in This Chapter:

Internet World Wide Web HTTP/HTTPS DNS IP Address Browser Server Client-Server Model TCP/IP URL

1.1 The Big Picture

The internet is not a single entity but a vast global network of interconnected computers communicating with each other. When you visit a website, a fascinating chain of events occurs in milliseconds—a ballet of data packets dancing across fiber optic cables, satellites, and wireless networks. Understanding this process is fundamental to becoming a proficient web developer.

What is the Internet?

The internet is a global network of billions of computers and other electronic devices. With the internet, it's possible to access almost any information, communicate with anyone else in the world, and do much more. You can think of the internet as the physical infrastructure—the cables, routers, servers, and devices—that connects the world.

The Journey of a Web Request:

1. **You type a URL** (like https://kranedigitalhub.netlify.app) in your browser
2. **DNS Lookup:** Your browser asks a DNS server "Where is this website?"
3. **IP Address:** DNS responds with an IP address (like 172.217.170.46)
4. **TCP Connection:** Your browser establishes a TCP connection with that server
5. **HTTP Request:** Browser sends an HTTP request for the page
6. **Server Processing:** Server processes the request (may involve databases, PHP, etc.)
7. **HTTP Response:** Server sends back HTML, CSS, and JavaScript
8. **Rendering:** Your browser renders the page

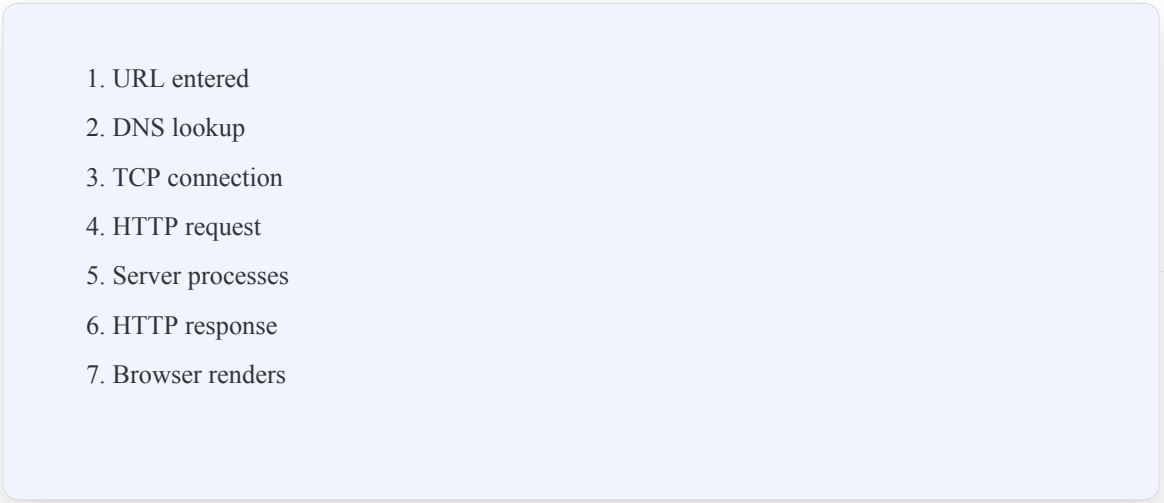


Figure 1.1: The complete journey of a web request from browser to server and back

Real-World Analogy: The Restaurant

Think of it like dining at a restaurant:

- **DNS lookup** — Finding the restaurant's address
- **Connect to server** — Walking to the restaurant
- **HTTP request** — Ordering your meal from the menu
- **Server processing** — Kitchen preparing your food
- **HTTP response** — Waiter bringing your meal
- **Browser rendering** — You eating and enjoying the meal

1.2 HTTP Deep Dive

HTTP (Hypertext Transfer Protocol) is the foundation of data communication on the web. It's a request-response protocol where a client (your browser) sends a request to a server, and the server sends back a response.

HTTP Request Methods:

Method	Description	Example Use Case
GET	Retrieve data	Loading a webpage
POST	Submit data	Form submission
PUT	Update data	Editing a profile
DELETE	Remove data	Deleting a post
PATCH	Partial update	Updating a field
HEAD	Get headers only	Checking if file exists
OPTIONS	Get allowed methods	CORS preflight

HTTP Status Codes:

Code Range	Category	Common Examples
1xx	Informational	100 Continue, 101 Switching Protocols
2xx	Success	200 OK, 201 Created, 204 No Content
3xx	Redirection	301 Moved Permanently, 304 Not Modified, 307 Temporary Redirect
4xx	Client Error	400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found
5xx	Server Error	500 Internal Server Error, 502 Bad Gateway, 503 Service Unavailable

⚠ Common HTTP Mistakes

The most famous HTTP error is **404 Not Found**. This occurs when you request a page that doesn't exist on the server. Always check your URLs and file paths!

Exercise 1.1: Trace a Web Request

Open Chrome DevTools (F12), go to the Network tab, and refresh this page. Observe the list of requests that load this very page. Identify:

1. The main HTML document and its status code
2. CSS files being loaded
3. JavaScript files
4. Images and other assets
5. HTTP status codes for each request
6. The total load time and page size

CHAPTER 2

Development Environment Setup

Key Terms:

IDE VS Code XAMPP Local Server Git GitHub Command Line Package Manager

2.1 Setting Up Your Development Environment

Before writing your first line of code, you need to set up your digital workshop. Think of this as gathering your tools before building furniture. A well-configured development environment will save you countless hours and headaches.

Essential Tools for Every Web Developer:

Tool	Purpose	Recommendation	Installation
Code Editor	Write and edit code	VS Code	`code.visualstudio.com`
Web Browser	View and test websites	Chrome DevTools	`google.com/chrome`
Local Server	Run PHP/MySQL locally	XAMPP	`apachefriends.org`
Version Control	Track code changes	Git + GitHub	`git-scm.com`
Package Manager	Manage dependencies	npm	Comes with Node.js
API Testing	Test APIs	Postman	`postman.com`
Design Tool	Create mockups	Figma	`figma.com`

2.1.1 Installing VS Code - Step by Step

1. Visit code.visualstudio.com
2. Download the version for your operating system (Windows, macOS, Linux)
3. Run the installer (keep all default settings)
4. Launch VS Code and install these essential extensions:
   - **Live Server** — Launch a local development server
   - **Prettier** — Code formatter

- **ESLint** — JavaScript linter
- **HTML CSS Support** — Intelligent CSS completion
- **JavaScript (ES6) code snippets** — Useful snippets
- **PHP IntelliSense** — PHP code completion
- **GitLens** — Supercharge Git

```
// VS Code Settings (settings.json)
{
    "editor.fontSize": 14,
    "editor.fontFamily": "'JetBrains Mono', monospace",
    "editor.fontLigatures": true,
    "editor.formatOnSave": true,
    "editor.tabSize": 4,
    "workbench.colorTheme": "Dark+",
    "files.autoSave": "onFocusChange"
}
```

2.1.2 Setting Up XAMPP for PHP Development

1. Download XAMPP from apachefriends.org
2. Run the installer (install to `C:\xampp` on Windows)
3. Open XAMPP Control Panel
4. Start Apache and MySQL services
5. Place your projects in `C:\xampp\htdocs\`
6. Access via `http://localhost/your-project/`

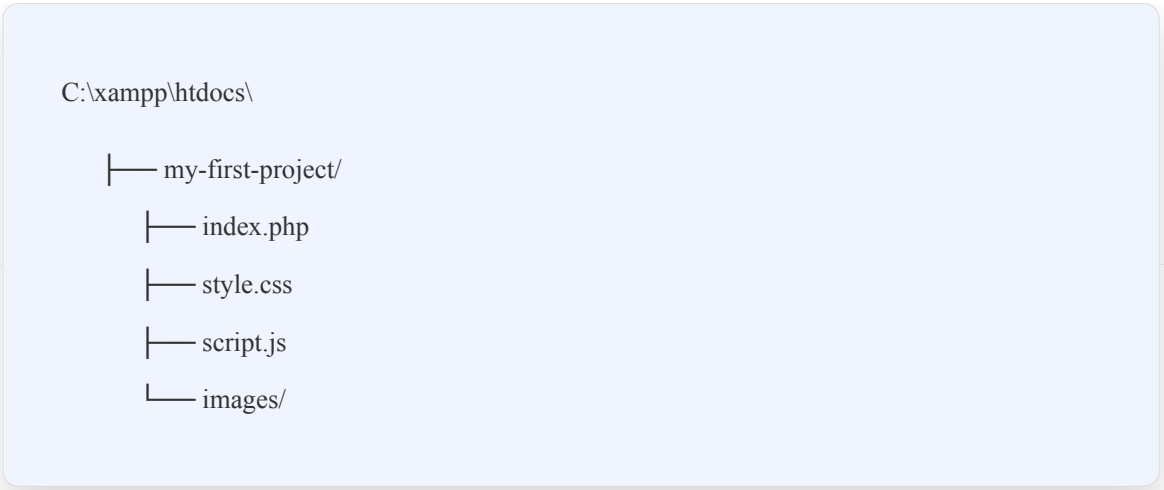


Figure 2.1: Typical XAMPP project folder structure in htdocs

Important Note

Always use descriptive folder names with lowercase letters and hyphens (kebab-case). For example: `my-first-website` not `My First Website`. Avoid spaces in folder and file names for web projects.

2.1.3 Introduction to Git and GitHub

Git is a version control system that tracks changes in your code. GitHub is a platform for hosting Git repositories. Together, they're essential for collaboration and backup.

```
# Git Commands Every Developer Should Know
# Initialize a repository
git init

# Check status
git status

# Add files to staging
git add .
git add index.html

# Commit changes
git commit -m "Initial commit"

# Connect to remote repository
git remote add origin https://github.com/username/repo.git

# Push to GitHub
git push -u origin main

# Pull latest changes
git pull
```

Exercise 2.1: Set Up Your Development Environment

1. Install VS Code and the recommended extensions
2. Install XAMPP and start Apache
3. Create a folder called "my-first-website" in `C:\xampp\htdocs\`
4. Create a file called `index.html` in that folder with "Hello World"
5. Open `http://localhost/my-first-website/` in your browser
6. Initialize a Git repository in your project folder
7. Create a GitHub account and push your code

CHAPTER 3

Your First Website

Key Terms:

HTML CSS JavaScript Element Tag Attribute DOCTYPE

3.1 Hello, World! in HTML

The tradition in programming is to start with "Hello, World!" — a simple program that displays this message. In web development, this is your first webpage.

```
<!-- This is a comment in 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 Webpage</title>
</head>
<body>
    <h1>Hello, World!</h1>
    <p>Welcome to my first website. I'm learning web development at Krane
Digital Academy!</p>
</body>
</html>
```

Breaking Down the Code:

- **<!DOCTYPE html>** — Declares this is an HTML5 document
- **<html>** — The root element of the page
- **<head>** — Contains meta information about the page
- **<meta charset="UTF-8">** — Sets character encoding
- **<title>** — Specifies a title for the page (shown in browser tab)
- **<body>** — Contains the visible page content
- **<h1>** — Defines a large heading
- **<p>** — Defines a paragraph

3.2 Adding Style with CSS

Now let's make it beautiful. CSS (Cascading Style Sheets) controls the presentation.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Styled Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            max-width: 800px;
            margin: 0 auto;
            padding: 40px;
            background: linear-gradient(135deg, #667eea, #764ba2);
            color: white;
            min-height: 100vh;
        }

        h1 {
            color: #FFD700;
            text-align: center;
            font-size: 3em;
            border-bottom: 3px solid #FFD700;
            padding-bottom: 20px;
        }

        p {
            font-size: 1.2em;
            line-height: 1.6;
            text-align: justify;
        }

        .card {
            background: rgba(255,255,255,0.1);
            padding: 30px;
            border-radius: 15px;
            margin: 30px 0;
            border: 1px solid rgba(255,255,255,0.2);
            backdrop-filter: blur(5px);
        }

        .card:hover {
            background: rgba(255,255,255,0.2);
            transform: translateY(-5px);
            transition: all 0.3s;
        }
```

```
        button {
            background: #FFD700;
            color: #2c3e50;
            border: none;
            padding: 15px 30px;
            border-radius: 50px;
            font-size: 1.1em;
            font-weight: bold;
            cursor: pointer;
            box-shadow: 0 5px 15px rgba(0,0,0,0.3);
            transition: all 0.3s;
        }

        button:hover {
            background: #ffed4a;
            transform: scale(1.05);
            box-shadow: 0 8px 20px rgba(0,0,0,0.4);
        }
    </style>
</head>
<body>
    <h1>Welcome to My Website!</h1>

    <div class="card">
        <h2>About Me</h2>
        <p>I'm learning web development at Krane Digital Academy! This is
my first complete webpage with HTML and CSS. I'm excited to learn more and
build amazing things.</p>
    </div>

    <div class="card">
        <h2>My Skills</h2>
        <ul>
            <li>✅ HTML5 - Building semantic structure</li>
            <li>✅ CSS3 - Creating beautiful designs</li>
            <li>⏳ JavaScript - Adding interactivity</li>
            <li>⏳ PHP - Server-side programming</li>
        </ul>
    </div>

    <div style="text-align: center;">
        <button onclick="alert('Hello from Krane Digital Academy!')">
            Click Me!
        </button>
    </div>

    <p style="text-align: center; margin-top: 40px;">
        Visit <a href="https://kranedigitalhub.netlify.app" style="color:
```

```
#FFD700;">Krane Digital Hub</a> for more resources!
    </p>
</body>
</html>
```

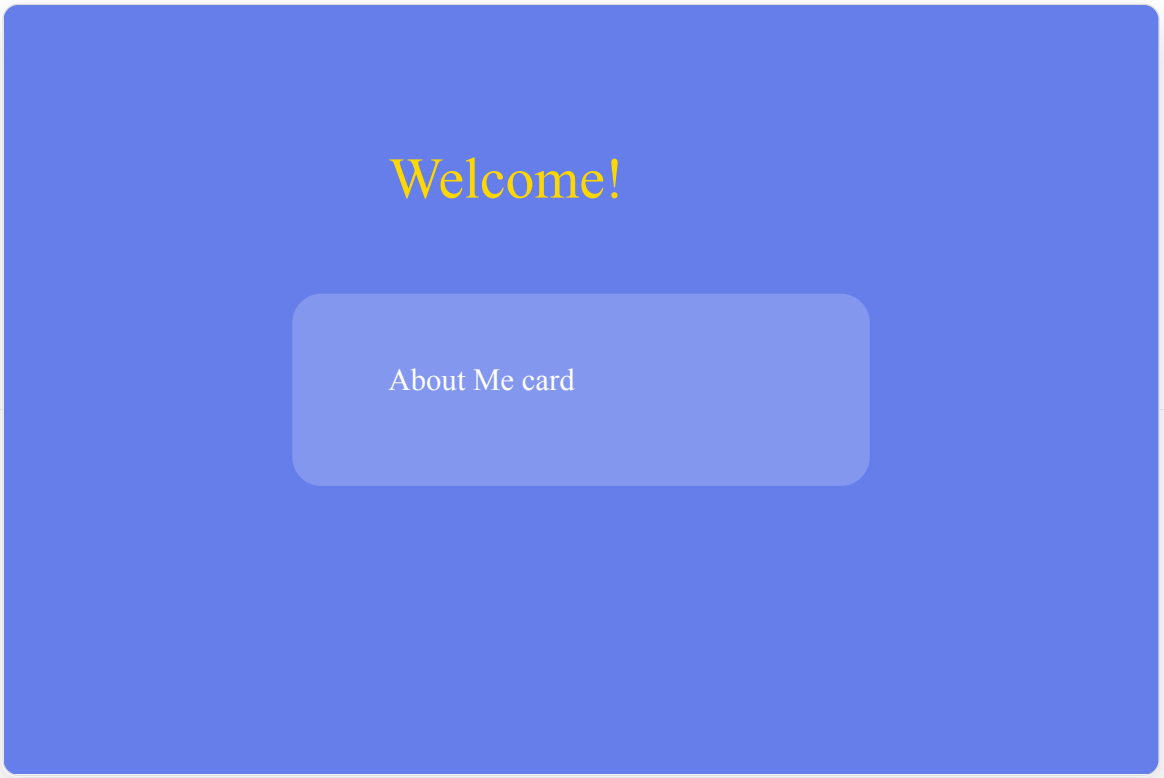


Figure 3.1: Preview of your first styled webpage

Exercise 3.1: Create Your Personal Page

1. Create a new file called `profile.html`
2. Add the complete HTML5 structure
3. Include a heading with your name
4. Add a paragraph about your goals
5. Create a list of technologies you want to learn
6. Add a link to Krane Digital Hub
7. Style it with CSS (colors, fonts, spacing)
8. Add a button that shows an alert

CHAPTER 4

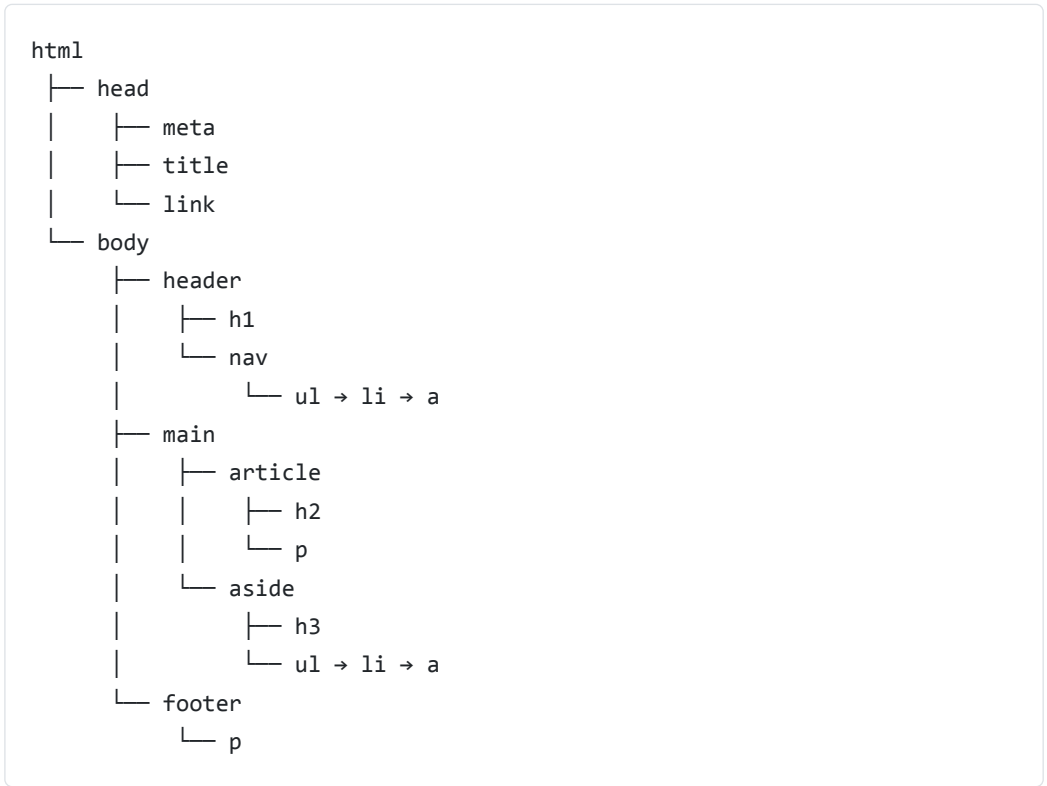
HTML Document Structure

4.1 The Anatomy of an HTML Document

Every HTML document follows a specific structure that browsers understand. Think of it as the blueprint of a house—it defines where everything goes.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="description" content="A complete guide to HTML">
    <meta name="author" content="Udoinyang, M. Clement (Mr. Krane)">
    <title>Document Title - Appears in Browser Tab</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <header>
        <h1>Main Website Title</h1>
        <nav>
            <ul>
                <li><a href="#home">Home</a></li>
                <li><a href="#about">About</a></li>
                <li><a href="#contact">Contact</a></li>
            </ul>
        </nav>
    </header>

    <main>
        <article>
            <h2>Article Title</h2>
            <p>Article content goes here...</p>
        </article>

        <aside>
            <h3>Related Content</h3>
            <ul>
                <li><a href="#">Related Link 1</a></li>
                <li><a href="#">Related Link 2</a></li>
            </ul>
        </aside>
    </main>

    <footer>
        <p>&copy; 2026 Udoinyang, M. Clement (Mr. Krane). All rights
reserved.</p>
    </footer>

    <script src="script.js"></script>
</body>
</html>
```

HTML Element Hierarchy:



Pro Tip

Always use semantic HTML5 elements (`<header>` , `<nav>` , `<main>` , `<article>` , `<section>` , `<aside>` , `<footer>`) instead of generic `<div>` s. This improves accessibility, SEO, and code readability.

Exercise 4.1: Build a Semantic HTML Structure

1. Create a new HTML file called `semantic-structure.html`
2. Include proper DOCTYPE and language attribute
3. Add meta tags for charset, viewport, description, and author
4. Create a header with navigation links
5. Add a main section with an article and an aside
6. Include a footer with copyright information
7. Validate your HTML using the W3C validator

CHAPTER 31

PHP: Hypertext Preprocessor

Key Terms:

PHP Server-Side Variables Arrays Functions Superglobals

31.1 Introduction to PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML. Unlike JavaScript which runs in the browser, PHP runs on the server, generating dynamic content before sending it to the client.

Why PHP?

PHP powers over 78% of all websites with known server-side programming languages, including Facebook, Wikipedia, and WordPress. It's particularly strong for database-driven websites, content management systems, and e-commerce platforms.

Your First PHP Script:

```
<!DOCTYPE html>
<html>
<head>
    <title>My First PHP Page</title>
</head>
<body>
    <h1>Welcome to PHP</h1>

    <?php
        // This is a PHP comment
        echo "<p>Hello from PHP! The time is " . date('H:i:s') . "</p>";

        // Variables in PHP
        $name = "Mr. Krane";
        $age = 35;
        $skills = ["HTML", "CSS", "JavaScript", "PHP"];

        echo "<p>Instructor: $name</p>";
        echo "<p>Experience: $age years</p>";
    ?>

    <h2>Skills Taught:</h2>
    <ul>
        <?php foreach($skills as $skill): ?>
            <li><?php echo $skill; ?></li>
        <?php endforeach; ?>
    </ul>
</body>
</html>
```

PHP Syntax Basics:

Concept	Syntax	Example
PHP Tags	`<?php ... ?>`	`<?php echo "Hello"; ?>`
Variables	`$variable_name`	`$name = "John";`
Strings	`"double"` or `'single'`	`$text = "Hello";`
Arrays	`array()` or `[]`	`$arr = [1,2,3];`
Associative Arrays	`["key" => "value"]`	`$user = ["name" => "John"];`
Concatenation	`.` (dot)	`"Hello " . "World"`
Echo/Print	`echo "text";`	`echo $variable;`

PHP Data Types:

```php
<?php
// Scalar Types
$string = "Hello World";        // String
$integer = 42;                  // Integer
$float = 3.14159;               // Float/Double
$boolean = true;                // Boolean

// Compound Types
$array = [1, 2, 3, 4, 5];       // Array
$object = new stdClass();       // Object
$callable = function() {        // Callable
    return "Hello";
};

// Special Types
$null = null;                   // NULL

// Check types
echo gettype($string);          // string
echo gettype($integer);         // integer
var_dump($array);               // Dumps complete info
?>
```

Exercise 31.1: Create a PHP Profile Page

1. Create a file called `profile.php`
2. Create variables for your name, age, location, and profession
3. Create an array of your skills
4. Use PHP to display this information in an HTML page
5. Add a loop to display your skills as a list
6. Use conditional statements to show different messages based on the time of day
7. Include the current date and time using PHP's date() function

31.2 Working with Forms in PHP

One of PHP's most powerful features is handling form data. PHP provides superglobal arrays like `$_GET`, `$_POST`, and `$_REQUEST` to access form data.

GET vs POST Methods:

Feature	GET	POST
Data visibility	Visible in URL	Not visible in URL
Bookmarkable	Yes	No
Data length limit	Yes (~2048 chars)	No limit
Security	Less secure	More secure
Use cases	Searches, filters	Logins, forms with sensitive data
Data type	Text only	Text + binary (files)

```
<!-- form.html -->
<form method="POST" action="process.php">
    <label for="name">Name:</label>
    <input type="text" name="name" id="name" required>

    <label for="email">Email:</label>
    <input type="email" name="email" id="email" required>

    <label for="message">Message:</label>
    <textarea name="message" id="message" rows="4" required></textarea>

    <button type="submit">Send Message</button>
</form>

<?php
// process.php
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    // Get and sanitize form data
    $name = htmlspecialchars($_POST['name'] ?? '');
    $email = filter_var($_POST['email'] ?? '', FILTER_SANITIZE_EMAIL);
    $message = htmlspecialchars($_POST['message'] ?? '');

    // Validate email
    if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        $error = "Invalid email format";
    } else {
        // Process the form (save to database, send email, etc.)
        echo "<h2>Thank you, $name!</h2>";
        echo "<p>Your message has been received.</p>";
        echo "<p>We'll contact you at $email shortly.</p>";
    }
}
?>
```

⚠ Security Warning

Never trust user input! Always validate and sanitize form data. Use `htmlspecialchars()` to prevent XSS attacks, `filter_var()` for validation, and prepared statements for database queries.

Conclusion: Your Journey as a Web Developer

Congratulations on completing this comprehensive journey through modern web development! You've covered an incredible amount of ground—from understanding how the web works at a fundamental level, to mastering HTML5, CSS3, JavaScript, PHP, and advanced concepts like security, performance, and deployment.

What You've Achieved:

- ✅ HTML5 Semantic Structure
- ✅ CSS3 Layouts (Flexbox & Grid)
- ✅ JavaScript Interactivity
- ✅ DOM Manipulation
- ✅ Form Validation
- ✅ API Integration
- ✅ PHP Server-Side Programming
- ✅ Database Operations
- ✅ Security Best Practices
- ✅ Performance Optimization
- ✅ Git & Version Control
- ✅ Deployment Strategies

Your Next Steps:

1. Build Projects

Apply your knowledge by building real projects:

- Personal portfolio website
- Blog with CMS
- E-commerce store
- Social media dashboard
- Task management app

2. Learn Frameworks

Expand your skills with modern frameworks:

- React / Vue / Angular (Frontend)
- Laravel / Symfony (PHP)
- Express (Node.js)
- Bootstrap / Tailwind (CSS)
- WordPress (CMS)

3. Join Communities

Connect with other developers:

- Stack Overflow
- GitHub
- Dev.to
- Local meetups
- Krane Digital Hub

> *"The expert in anything was once a beginner. The path from beginner to expert is paved with consistent practice, curiosity, and the courage to fail and learn. You've taken the first—and most important—step."*
>
> *— Mr. Krane*

Keep coding, stay curious, and never stop learning.

Krane Digital Programming Academy

Empowering Dreams

About the Author

Udoinyang, M. Clement

aka Mr. Krane · CEO, Krane Digital Hub

Udoinyang, M. Clement (professionally known as Mr. Krane) is a passionate software developer, web designer, teacher, and entrepreneur. His journey in technology began with a deep curiosity about how computers work and has evolved into a mission to empower others through education.

🎓 **Education:** Mr. Krane studied PC Support at Springwood Information Technology, Uyo, and later pursued Computer and Robotics Education at the Faculty of Vocational Education,

Library and Information Science, University of Uyo, Nigeria. This unique combination of technical and pedagogical training shapes his approach to teaching complex concepts in accessible ways.

📍 **Origin:** Ikot Ekpene Local Government Area, Akwa Ibom State, Nigeria — a region known as the "Raffia City" for its rich cultural heritage and craftsmanship.

💼 **Professional Journey:** Starting as a PC support technician, Mr. Krane quickly realized his passion for web development and education. He founded Krane Digital Hub to create a platform where aspiring developers could learn practical skills in a supportive environment. Today, Krane Digital Hub serves thousands of students across Nigeria and beyond.

🎯 **Hobbies & Interests:** When not coding or teaching, Mr. Krane enjoys playing chess (a game that sharpens his strategic thinking), listening to music, coding personal projects, and watching movies—particularly sci-fi and documentaries about technology.

- 👨‍💻 Software Developer
- 🎨 Web Designer
- 📚 Teacher
- 💼 Businessman
- 🏆 CEO, Krane Digital Hub
- 🎓 Educator
- ♟️ Chess Enthusiast

💡 **Teaching Philosophy:** "Every expert was once a beginner. My

goal is to make complex concepts simple and accessible to everyone. I believe in learning by doing—theory is important, but real understanding comes from building things yourself. This book is the culmination of years of teaching experience, distilled into a resource that I hope will guide you on your own journey."

> *"Empowering Dreams — one line of code at a time."*

References & Further Reading

📚 Books

[ ] Duckett, J. (2011). *HTML and CSS: Design and Build Websites*. Wiley. — The classic beginner's guide with beautiful visual examples.

[ ] Duckett, J. (2014). *JavaScript and jQuery: Interactive Front-End Web Development*. Wiley. — Perfect companion for learning JavaScript interactivity.

[ ] Flanagan, D. (2020). *JavaScript: The Definitive Guide*. O'Reilly Media. — The definitive reference for JavaScript, often called "The Rhino Book."

[ ] Haverbeke, M. (2018). *Eloquent JavaScript*. No Starch Press. — A modern introduction to JavaScript with a focus on programming fundamentals.

[ ] Zandstra, M. (2021). *PHP 8 Objects, Patterns, and Practice*. Apress. — Comprehensive guide to modern PHP development.

[ ] Meyer, E. (2018). *CSS: The Definitive Guide*. O'Reilly Media. — In-depth coverage of CSS from the experts.

[ ] McFarland, D. (2020). *JavaScript & jQuery: The Missing Manual*. O'Reilly Media. — Practical guide with real-world examples.

🌐 Online Documentation

- [ ] **MDN Web Docs:** developer.mozilla.org — The most comprehensive and authoritative resource for web technologies.
- [ ] **W3Schools:** w3schools.com — Beginner-friendly tutorials and references.
- [ ] **PHP Manual:** php.net/manual — Official PHP documentation with examples.
- [ ] **DevDocs:** devdocs.io — Fast, offline-capable API documentation browser.
- [ ] **CSS-Tricks:** css-tricks.com — Articles, guides, and snippets for CSS.
- [ ] **Stack Overflow:** stackoverflow.com — Q&A community for programmers.

🛠️ Tools & Resources

- [ ] **VS Code:** code.visualstudio.com — The code editor used throughout this book.
- [ ] **GitHub:** github.com — Platform for hosting and collaborating on code.
- [ ] **CodePen:** codepen.io — Social development environment for front-end code.
- [ ] **Can I Use:** caniuse.com — Browser compatibility tables.
- [ ] **Google Fonts:** fonts.google.com — Free, open-source fonts for web

projects.

- [ ] **Font Awesome:** fontawesome.com — Icon library used in this book.
- [ ] **Postman:** postman.com — API development and testing tool.

🎓 Courses & Tutorials

- [ ] **freeCodeCamp:** freecodecamp.org — Free, self-paced coding curriculum.
- [ ] **The Odin Project:** theodinproject.com — Full-stack curriculum with projects.
- [ ] **Krane Digital Hub:** kranedigitalhub.netlify.app — Courses and resources by Mr. Krane.
- [ ] **Udemy:** Various courses by Colt Steele, Angela Yu, and others.
- [ ] **Coursera:** University-level courses in web development.
- [ ] **YouTube Channels:** Traversy Media, The Net Ninja, Web Dev Simplified, freeCodeCamp.

📊 Standards & Specifications

- [ ] **HTML Specification:** html.spec.whatwg.org — Living standard for HTML.

- [ ] **CSS Specification:** w3.org/Style/CSS/ — W3C CSS specifications.
- [ ] **ECMAScript Specification:** ecma-international.org — JavaScript language standard.

Index

Colophon

This book was created using:

Typography: Playfair Display (headings) and Lato (body text) — two beautiful, open-source fonts from Google Fonts. Code examples are set in JetBrains Mono and Source Code Pro for optimal readability.

Tools: Written in PHP with embedded HTML and CSS. Generated as a web page optimized for PDF output using print stylesheets.

Icons: Font Awesome 6 — the web's most popular icon library.

Development Environment: Visual Studio Code with extensive use of the Live Server extension for real-time preview.

Version Control: Git and GitHub for tracking changes and collaboration.

Testing: Tested across Chrome, Firefox, Safari, and Edge browsers, with special attention to print output in Chrome's PDF engine.

About the Cover: The cover design features a gradient representing the spectrum of web technologies—from the deep blues of backend development to the vibrant golds of frontend creativity.

Production Notes: This book was written over several months, with countless revisions and improvements based on reader feedback. The goal was to create a resource that is both comprehensive and accessible—a book that could serve as both a textbook for beginners and a reference for experienced developers.

Edition Information: First Edition, 2026. Published by Krane Digital Incorporation in association with Krane Digital Programming Academy.

Printing: This book is designed to be printed on A4 paper with 2.5cm margins. When printed, it should be bound with the spine on the left.

Modern Website Fundamentals

From Zero to Professional Web Developer

A comprehensive guide that takes you from absolute beginner to confident web developer. Packed with examples, exercises, and real-world projects covering HTML, CSS, JavaScript, PHP, databases, security, and modern best practices.

40+ Chapters

100+ Exercises

Real Projects

ISBN: 978-978-001-234-5 (Hardcover) | 978-978-001-235-2 (PDF)

Published by: Krane Digital Incorporation

Empowering Dreams

Modern Website Fundamentals — First Edition, 2026

`978-978-001-234-5 | 978-978-001-235-2 (PDF)`

Published by Krane Digital Incorporation · Krane Digital Programming Academy