

ICS 385

Web Development and Administration

Week 2 Study Guide

HTML and CSS Fundamentals

Course:	ICS 385 - Web Development and Administration
Week:	Week 2 (01/19/26 - 01/25/26)
Instructor:	Dr. Debasis Bhattacharya
Institution:	University of Hawaii Maui College
Semester:	Spring 2026

Week 2 Learning Objectives

1. Learn to write semantic HTML and CSS code following industry standards
2. Use CodePen to create and test HTML and CSS code interactively
3. Use Visual Studio Code for professional web development workflow
4. Upload code from VS Code to GitHub Public Repository for version control
5. Create a GitHub Page for Week 2 assignment and establish course portfolio
6. Understand HTML5 semantic structure and accessibility principles
7. Implement CSS styling techniques including layout with Grid and Flexbox
8. Develop responsive design skills for mobile-first approach

Week 2 Deliverables (3 Points Total)

Important: All assignments build upon each other - complete them in order!

Assignment	Due Date	Platform	Description
HW1	Tuesday 11 PM	CodePen	Basic HTML/CSS practice pen
HW2	Friday 11 PM	GitHub Repo	Complete website project with HTML/CSS
HW3	Sunday 11 PM	GitHub Pages	Deploy website as live GitHub Page

HTML Fundamentals

What is HTML?

HyperText Markup Language (HTML) is the standard markup language for creating web pages. HTML describes the structure and content of web pages using elements and tags.

HTML5 Document Structure

Every HTML5 document follows this basic structure:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document Title</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<!-- Your content goes here -->
</body>
</html>
```

Key HTML Elements

Element	Purpose	Example
<header>	Page/section header	<header><h1>Site Title</h1></header>
<nav>	Navigation links	<nav>Home</nav>
<main>	Main content area	<main><article>...</article></main>
<section>	Thematic grouping	<section><h2>About</h2></section>
<article>	Standalone content	<article><h3>Blog Post</h3></article>
<aside>	Sidebar content	<aside><p>Related links</p></aside>
<footer>	Page/section footer	<footer>© 2026 Company</footer>

CSS Fundamentals

What is CSS?

Cascading Style Sheets (CSS) is used to describe the presentation of HTML elements. CSS controls layout, colors, fonts, spacing, and responsive behavior.

CSS Syntax

```
selector {
  property: value;
  another-property: another-value;
}

/* Example */
h1 {
  color: #66bfbf;
  font-size: 2.5rem;
  text-align: center;
}
```

CSS Selectors

Selector Type	Syntax	Example	Selects
Element	element	h1	All <h1> elements
Class	.class-name	.intro	Elements with class='intro'
ID	#id-name	#header	Element with id='header'
Descendant	parent child	div p	<p> inside <div>
Pseudo-class	element:state	a:hover	Links when hovered

Layout Techniques

CSS Grid (Modern 2D Layout)

```
.parent {
  display: grid;
  grid-template-columns: 0.5fr 1fr;
  grid-template-rows: repeat(2, 1fr);
  gap: 2rem;
}
```

Flexbox (1D Layout)

```
.container {
  display: flex;
  justify-content: center;
  align-items: center;
  gap: 1rem;
}
```

Sample Code Analysis

Understanding the Sample Project

The provided sample code (bio.html and styles.css) demonstrates professional web development practices. Let's analyze key components:

HTML Structure Analysis

Key structural elements in bio.html:

1. DOCTYPE declaration:
 - Tells browser this is HTML5 document
2. Head section contains:
 - Meta charset for UTF-8 encoding
 - Title for browser tab/SEO
 - External CSS links (Google Fonts)
 - Local stylesheet link
3. Body structure:
 - .top-container (Hero section with clouds, title)
 - .middle-container (Profile, skills, contact)
 - .bottom-container (Footer with links)

CSS Techniques Used

Professional CSS techniques demonstrated:

1. CSS Reset:

```
body { margin: 0; text-align: center; }
```
2. Typography:
 - Custom Google Fonts (Merriweather, Sacramento)
 - Responsive font sizing with rem units
 - Font hierarchy (h1, h2, h3, p)
3. Layout:
 - CSS Grid for skills section (.parent)
 - Absolute positioning for decorative elements
 - Flexbox concepts implied in grid structure
4. Color Scheme:
 - Consistent color palette (#66bfbf, #40514e, #11999e)
 - Color psychology for professional appearance
5. Interactive Elements:
 - Hover effects on buttons and links
 - Gradient backgrounds for buttons
 - Smooth transitions (implied)

Development Tools and Workflow

CodePen Setup

CodePen is an online code editor perfect for experimenting with HTML and CSS:

1. Create free account at codepen.io
2. Start with a new 'pen'
3. Use HTML panel for markup
4. Use CSS panel for styling
5. See live preview instantly
6. Save and share your work

Visual Studio Code Setup

VS Code is a professional code editor. Essential setup:

Extension	Purpose
GitHub Copilot	AI-powered code completion
GitHub Copilot Chat	AI coding assistance
Live Server	Local development server
HTML CSS Support	Enhanced autocomplete
Prettier	Code formatting

GitHub Workflow

Professional development requires version control:

1. Create GitHub repository

```
git init
git add .
git commit -m "Initial commit"
git remote add origin [your-repo-url]
git push -u origin main
```
2. Enable GitHub Pages
 - Go to repository Settings
 - Scroll to Pages section
 - Select source: Deploy from a branch
 - Choose main branch
 - Your site will be at: `username.github.io/repository-name`

Web Development Best Practices

HTML Best Practices

- **Semantic Markup:** Use elements for their meaning (header, nav, main, article, aside, footer)
- **Accessibility:** Include alt attributes for images, proper heading hierarchy
- **Validation:** Ensure HTML validates at validator.w3.org
- **SEO:** Use descriptive titles, meta descriptions, heading structure
- **Performance:** Minimize HTTP requests, optimize images, use CDNs for fonts

CSS Best Practices

- **Mobile-First Design:** Start with mobile styles, add desktop with media queries
- **Responsive Units:** Use rem, em, %, vw/vh instead of fixed px values
- **CSS Architecture:** Organize styles logically, use consistent naming conventions
- **Performance:** Minimize CSS, avoid excessive nesting, optimize selectors
- **Maintainability:** Comment complex code, use CSS custom properties (variables)

Accessibility Guidelines

Follow WCAG 2.1 AA standards for inclusive web design:

Guideline	Implementation	Example
Color Contrast	4.5:1 ratio minimum	Dark text on light backgrounds
Keyboard Navigation	All interactive elements	Tab through all links/buttons
Alt Text	Descriptive image text	alt='Person typing on laptop'
Heading Structure	Logical h1-h6 hierarchy	h1 → h2 → h3 (no skipping)
Focus Indicators	Visible focus states	:focus { outline: 2px solid blue; }

Required Study Resources

Primary Resources

- **Udemy Course:** Web Development Bootcamp by Angela Yu - Sections 1-3 (HTML), 5-7 (CSS), 12-13 (Capstone Project)
- **W3Schools:** <https://www.w3schools.com/> - Comprehensive reference for HTML/CSS
- **VS Code Tutorial:** <https://code.visualstudio.com/docs/introvideos/basics>
- **GitHub Tutorials:** Account creation and Hello World project
- **GitHub Pages:** <https://docs.github.com/en/pages/quickstart>

Preparation Checklist

Complete before Tuesday class:

- Set up CodePen account and test basic functionality
- Install Visual Studio Code with required extensions
- Create GitHub account with student developer pack
- Complete VS Code basics tutorial
- Review Udemy sections 1-3, 5-7, 12-13
- Create local folder structure (c:/ics385spring2026/week2 or
~/Downloads/ics385spring2026/week2)
- Test GitHub repository creation and basic git commands
- Review W3Schools HTML and CSS fundamentals

Assignment Success Strategies

HW1: CodePen Assignment

Create a professional-looking pen demonstrating HTML and CSS skills:

- Use semantic HTML5 elements (header, nav, main, section, article, footer)
- Implement responsive design principles
- Apply consistent color scheme and typography
- Include hover effects and basic animations
- Ensure accessibility with proper alt text and headings
- Test across different screen sizes

HW2: GitHub Repository

Develop the CodePen project into a complete website:

- Create proper file structure (index.html, styles.css, assets/)
- Use external CSS file linked properly
- Implement advanced layout techniques (Grid/Flexbox)
- Add multiple pages with consistent navigation
- Include contact form with HTML5 validation
- Optimize for performance and SEO

HW3: GitHub Pages Deployment

Deploy your website as a live GitHub Page:

- Enable GitHub Pages in repository settings
- Ensure all links and resources work correctly
- Test website functionality on live URL
- Update README.md with project description
- Include live site URL in repository description
- Verify mobile responsiveness on actual devices

Common Issues and Solutions

Issue	Solution
CSS not loading	Check file path in link tag, ensure correct spelling
Images not displaying	Verify image paths are relative to HTML file
GitHub Pages not updating	Wait 5-10 minutes, check repository settings
Mobile layout broken	Use viewport meta tag, test with browser dev tools
Colors not showing	Use valid CSS color formats (#hex, rgb(), color names)
Grid not working	Check browser support, use fallback layouts

Course Support and Resources

Getting Help

When you encounter challenges, use these resources in order:

1. **Course Materials:** Review this study guide, Udemy videos, and W3Schools documentation
2. **Class Discussion:** Attend Tuesday Zoom session for live help and clarification
3. **Office Hours:** Friday 11 AM - noon, Kaaie 114, or by appointment
4. **Email Support:** debasisb@hawaii.edu (response within 1 business day)
5. **Online Communities:** Stack Overflow, MDN Web Docs, GitHub Discussions

Important Reminders

- **Late Policy:** 10% penalty per day, maximum 2 days late
- **Academic Integrity:** Use AI tools (cite usage), but work must be your own
- **Attendance:** Mandatory Tuesday Zoom sessions at 4:30 PM
- **Communication:** No instructor availability weekends - ask questions by Friday 5 PM

Success Tips

Web development is a cumulative skill. Each concept builds on the previous one. Don't rush - take time to understand the fundamentals. Practice regularly, experiment with different approaches, and don't hesitate to ask questions. Remember, even professional developers constantly learn new techniques!

Good luck with Week 2! ■