

2026 WEB DESIGN & DEVELOPMENT

Program Standards

CONTENT STANDARD 1.0: PROFESSIONAL ORGANIZATIONS AND LEADERSHIP

Performance Standard 1.1: Student Leadership in Career Technical Student Organizations (CTSO) and Professional Associations

- 1.1.1 Explore the role of professional organizations and/or associations in the Web Design and Development Industry.
- 1.1.2 Define the values roles, and opportunities provided through career technical student organizations.
- 1.1.3 Engage in career exploration and leadership development.

CONTENT STANDARD 2.0: WEB PAGE DEVELOPMENT

Performance Standard 2.1: File Management and Directory Structure

- 2.1.1 Create filenames and uniform resource locator (URL) addresses following industry standard conventions.
- 2.1.2 Save text documents as HyperText Markup Language (HTML) and Cascading Style Sheets (CSS) files.
- 2.1.3 Organize files in your web directory following industry standard conventions.
- 2.1.4 Identify the domain and path in a URL.
- 2.1.5 Explain the difference between HyperText Transfer Protocol (HTTP) and HyperText Transfer Protocol Secure (HTTPS).
- 2.1.6 Compare relative and absolute file paths and their applications.
- 2.1.7 Describe how relative paths can cause issues during deployment.

Performance Standard 2.2: HTML Structure

- 2.2.1 Identify tags and their functions in web design.
- 2.2.2 Create a basic boilerplate HTML page (e.g., `<!DOCTYPE html>`, `<html>`, `<head>`, `<body>`).
- 2.2.3 Describe the purpose of the `<!DOCTYPE html>`, `<html>`, `<head>`, and `<body>` tags.
- 2.2.4 Document HTML code, using comments and proper indentation according to industry standards.
- 2.2.5 Create headings `<h1>` through `<h6>`.
- 2.2.6 Create paragraphs `<p>` and line breaks `
`.
- 2.2.7 Create ordered lists (``), unordered lists (``), and list items (``).
- 2.2.8 Create absolute hyperlinks (`<a>`) to external websites.
- 2.2.9 Use target attributes in `<a>` tags (e.g., "self", "_blank").
- 2.2.10 Create relative hyperlinks (`<a>`) to internal web pages.
- 2.2.11 Create hyperlinks to identification (ID) attributes (``).
- 2.2.12 Create hyperlinks to email addresses and phone numbers.
- 2.2.13 Add images to a web page (``).
- 2.2.14 Create responsive images with the picture element (`<picture>`) and `srcset` attribute.
- 2.2.15 Create diagrams (`<figure>`) and images with captions (`<figcaption>`).
- 2.2.16 Create hyperlinks within images.
- 2.2.17 Design content, using block level elements (`<div>`).
- 2.2.18 Design content, using inline elements (``).
- 2.2.19 Layout a web page, using semantic elements (e.g., `<header>`, `<section>`, `<article>`, `<footer>`).
- 2.2.20 Create tables (e.g., `<table>`, `<th>`, `<tr>`, `<td>`).

- 2.2.21 Create navigation (<nav>).
- 2.2.22 Create HTML forms.
- 2.2.23 Create appropriate labels and form controls (e.g., text, radio buttons, checkboxes, list boxes).
- 2.2.24 Describe techniques for HTML Form validation.

Performance Standard 2.3: Head Section

- 2.3.1 Create a page title.
- 2.3.2 Link a CSS sheet to a design.
- 2.3.3 Link a favicon to a design.
- 2.3.4 Create a meta description.
- 2.3.5 Create a meta viewport.
- 2.3.6 Describe meta charset.
- 2.3.7 Declare the language on the HTML tag.

Performance Standard 2.4: Format Web Pages with Styles

- 2.4.1 Describe the functions of HTML and CSS.
- 2.4.2 Write CSS syntax rules.
- 2.4.3 Create inline, embedded, and external styles.
- 2.4.4 Describe best practices of using external stylesheets over embedded or inline styles.
- 2.4.5 Design content, using font properties (e.g., font, font-size, font-family, color).
- 2.4.6 Control line spacing and white space (line-height, white-space).
- 2.4.7 Change the markers on ordered and unordered lists.
- 2.4.8 Change backgrounds (e.g., images, colors, gradients).
- 2.4.9 Control color properties with CSS (e.g., RGBA, HSL, HEX).
- 2.4.10 Design, using CSS tag styling.
- 2.4.11 Design, using CSS class styling.
- 2.4.12 Design, using CSS ID styling.
- 2.4.13 Determine when to use an ID and when to use a class.
- 2.4.14 Design, using external font libraries (e.g., Google Fonts, FontAwesome).
- 2.4.15 Describe the function of the <important> declaration in CSS and when to use.
- 2.4.16 List the order in which CSS rules are applied according to the cascade.

Performance Standard 2.5: Advanced CSS Selectors and Properties

- 2.5.1 Compare standalone and contextual selectors (e.g., .example vs. p.example).
- 2.5.2 Align text on a web page (text-align).
- 2.5.3 Design, using CSS pseudo-elements (e.g., ::before, ::after).
- 2.5.4 Identify the pseudo-class selectors (e.g., :focus, :hover, :active).
- 2.5.5 Select any descendent of an element.
- 2.5.6 Select any direct child of an element.
- 2.5.7 Select any sibling of an element.
- 2.5.8 Select, using the universal selector (*).
- 2.5.9 Select any attribute of an element.
- 2.5.10 Design, using CSS variables.
- 2.5.11 Design, using nested CSS styles.
- 2.5.12 Design, using CSS transform and translate properties.
- 2.5.13 Create CSS keyframe animations

CONTENT STANDARD 3.0: WEB PAGE DESIGN AND LAYOUT

Performance Standard 3.1: User Experience/User Interface (UX/UI) Theory for Web Design and Development

- 3.1.1 Design using industry standard UX/UI principles.
- 3.1.2 Create web pages, using responsive design and mobile-first design strategy.
- 3.1.3 Compare and define graphics file formats and extensions (e.g., vector [.svg] versus raster [.png, .jpg]).

- 3.1.4 Resize and optimize images, using graphics editors.
- 3.1.5 Describe typography and its effects as an element of web design.
- 3.1.6 Describe elements that require user interaction and benefit from feedback (e.g., button: hover).
- 3.1.7 Describe the functions of JavaScript and how it interacts with the Document Object Model (DOM).

Performance Standard 3.2: Page Layout and Positioning

- 3.2.1 Describe the function of the box model.
- 3.2.2 Assign content, padding, border, and margin properties.
- 3.2.3 Control overflow for a box.
- 3.2.4 Layout a page, using CSS Grid.
- 3.2.5 Layout a page, using Flexbox.
- 3.2.6 Compare CSS Grid and Flexbox.
- 3.2.7 Differentiate between positioning values (e.g., static, relative, absolute, fixed, sticky).
- 3.2.8 Create responsive design, using media queries.
- 3.2.9 Describe the float property and its legacy use for text wrapping.
- 3.2.10 Design using industry standard navigation principles (e.g., placement, consistency, simplicity, hierarchy, logo, clarity, mobile-friendly).

CONTENT STANDARD 4.0: WEB RELATED PLANNING AND ORGANIZATIONAL STANDARDS

Performance Standard 4.1: Website Architecture and Planning

- 4.1.1 Describe the client-server relationship.
- 4.1.2 Describe the web deployment process (e.g., domain name and acquisition, web hosting).
- 4.1.3 Describe the Web Development Life Cycle (WDLC).
- 4.1.4 Describe the purpose of using an HTML and CSS validator.
- 4.1.5 Describe different collaborative methodologies within a professional web design team (e.g., Agile, Scrum).
- 4.1.6 Initialize or clone a repository, using source control (e.g., git).
- 4.1.7 Commit and push code, using source control (e.g., git).
- 4.1.8 Pull code, using source control (e.g., git).
- 4.1.9 Test web pages on various browsers.

Performance Standard 4.2: Accessibility Standards

- 4.2.1 Implement Web Content Accessibility Guidelines (WCAG).
- 4.2.2 Access and describe Section 508 Standards.
- 4.2.3 Include descriptive alt attributes on images.
- 4.2.4 Include Accessible Rich Internet Applications (ARIA) attributes (e.g., aria-label, aria-labelledby).

CONTENT STANDARD 5.0: WEB MARKETING

Performance Standard 5.1: Multimedia on the Web

- 5.1.1 Embed external elements on a web page (e.g., maps, videos).
- 5.1.2 Add multimedia elements (e.g., <audio>, <video>) to a web page.
- 5.1.3 Describe the formats available for web-based video and the factors that determine which one to use.

Performance Standard 5.2: Brand Management

- 5.2.1 Describe the function of Search Engine Optimization (SEO).
- 5.2.2 Describe the legal and ethical issues involved in copyrighting, trademarking, and licensing.
- 5.2.3 Describe the legal, ethical, and cultural issues related to working in a global environment.
- 5.2.4 Describe the pros and cons of using Content Management Systems (CMS) (e.g., WordPress, Drupal, Joomla, Webflow, Squarespace, headless CMS platforms).

