STRANDS AND STANDARDS

WEB DEVELOPMENT 2

Learning that works for Utah

CTE®

Course Description

Web Development 2 is a course designed to guide students in a project-based environment in the development of up-to-date concepts and skills that are used in the development of today's websites. Students will learn the fundamentals of how the Internet works. They will learn and use the basic building blocks of the World Wide Web: HTML5 coding, Cascading Style Sheets (CSS), and JavaScript. They follow the steps to create a website by planning, designing, developing, deploying, and maintaining of the website projects. Students will learn and use different scripting technologies to create more dynamic and interactive websites. They will learn what it takes for a career in Web Development as they complete projects and create their own web site.

Intended Grade Level	9-12
Units of Credit	0.5
Core Code	35.02.00.00.065
Concurrent Enrollment Core Code	None
Prerequisite	Web Development 1
Skill Certification Test Number	894, 993, 9451, 9454
Test Weight	0.5
License Type	CTE and/or Secondary Education 6-12
Required Endorsement(s)	
Endorsement 1 or	Web Development
Endorsement 2	Programming & Software Development

ADA Compliant: June 2020

STRAND 1

Fundamentals of HTML5: Students will demonstrate creation of “well formed” web pages. (Suggested time proportion 20%)

Standard 1

Use advanced HTML5 elements to create web pages.

- Incorporate SVG image maps on web page.
- Incorporate forms on web page. Examples: radio, checkbox, text, text area.
- Understand the action that will run on the server on form submit button. (PHP file)
- Embed iframes or other content on a web page.
- Incorporate Javascript on a web page.
- Use a GUI-based HTML editing software to create web pages.

STRAND 2

Cascading Style Sheets (CSS): Students will format web pages using CSS.
(Suggested time proportion 35%)

Standard 1

Understand the use of various CSS selectors.

- Understand the cascade order for inline, internal, and external style sheet.
- Implement element selections to modify HTML elements (tag, ID, & classes)
- Implement an id selector to modify a single element on a page.
- Implement class selectors to modify several class elements.
- Implement contextual selectors to modify nested elements.
- Implement selectors a: link, a:visited, a:active a:hover

Standard 2

Format page layout with advanced CSS.

- Use width, height, or auto to adjust the size of the elements.
- Use float to position elements.
- Use absolute, relative, fixed, and static to position elements.
- Use text align, margin, and padding.

Standard 3

Frameworks to layout web pages.

- Format web pages using frameworks like: BootStrap, JQuery, etc.
- Explore new frameworks.
- Use frameworks to create responsive pages.

Standard 4

Code animation and graphics with advanced CSS.

- Buttons. (ie, rounded, colored, etc.)
- Create Navigation Bars.
- Image overlay hover. (ie. Fade-in, slide-out, etc.)
- Image slider.

STRAND 3

Site Planning and Design: Students will plan, design, implement, and maintain website(s). (Suggested time proportion 15%)

Standard 1

Create a website plan and pitch for a client.

- Identify basic principles of website usability, readability, and accessibility.
- Plan a website by using sketches, website hierarchy, wireframe, or a site map.
- Communicate with others (such as peers and clients) about design and content plans.
- Produce website designs that work on various devices and browser versions/configurations.
- Plan, communicate, or present a client's website before, during or after website development.

Standard 2

Create content for website.

- Create and prepare 2D images. .gif, .png, .jpg, .svg
- Prepare rich media; such as, video, sound, or animation.
- Identify when to use various image and digital media file formats.
- Optimize images for web content, such as resize, resolution, compress, thumbnails.
- Understand the use of favicons.
- Identify how to avoid violating copyright rules.
- Demonstrate the use of semantic elements such as: audio, video, track, mark, picture, figure, source, and canvas.

Standard 3

Uploading and maintaining a site.

- Understand and be able to describe the capabilities of web servers.
- Upload pages to a web server. * Optional if security allows.
- Differentiate between types of IP addresses.
- Describe a static IP address.
- Describe a Dynamic IP address.
- Differentiate between ipv4 and ipv6.
- Conduct basic technical tests such as validating the website (wc3 compliant), accessibility, SEO, etc.
- Present web pages to others for quality assurances (QA) such as team members and clients for feedback and evaluation on technical merits and usability.
- Identify methods for collecting site feedback, such as using counters, feedback forums, Google Analytics, Google Webmaster Tools.
- Provide site maintenance using bug reports, backups, and promotion.
- Document all aspects of website maintenance.
- Identify internet protocols: http, https, ftp, tcp/ip.

Standard 4

Use version control in projects.

- Utilize tools like Git/Github, Tortoise SVN, ect.
- Understand basic terms and procedures of version control.

Standard 5

Work as a team to create a website.

- Use good oral and written communication skills as a team member.
- Use Agile/Scrum project management tools to help the teamwork.

STRAND 4

Advanced Web Concepts: Students will explore advanced web concepts.
(Suggested time proportion 5%)

Standard 1

Demonstrate the use of scripting and other interactive tools.

- Use HTML5 tags
- Add interactivity to your website using JavaScript.
- Understand the difference between client side and server side scripting languages.

Standard 2

Understand other web technologies.

- Understand when to use a database.
- Describe what a Web 2.0 site is and how wikis, blogs, and forums work.
- Understand the purpose of content management systems (CMS) such as Wordpress, Drupal, Weebly, etc.

STRAND 5

JavaScript has become an essential web technology along with HTML and CSS, as most browsers implement JavaScript. Students will understand basic JavaScript in front-end and back-end development, mobile app development, desktop app development, and game development. (Suggested time proportion 20%)

Standard 1

Intro to JavaScript.

- Introduce JavaScript and how it is used in web development.
- Learn a few basic JavaScript commands.
- Create a simple webpage using JavaScript.

Standard 2

Variables in JavaScript.

- Introduce variables and uses.
- Learn about variable scope.
- Use input in programs.

Standard 3

Input in JavaScript.

- Review variables.
- Learn how to gather input from the user.
- Create a simple webpage using JavaScript.

Standard 4

Conditionals in JavaScript.

- Review if statements.
- Learn how to create a custom function.
- Se input in programs.

Standard 5

Functions in JavaScript.

- Review functions.
- Learn syntax for conditionals.
- Use if statements in programs.

STRAND 6

Exploration & Preparation for Careers in Web Development: Students will explore careers in Web (Suggested time proportion 5%)

Standard 1

Explore IT Web Development careers.

- Identify job roles in the Information Technology (IT) industry as they apply to web development. Full-stack Developer, Front-end back-end developer.
- Understand the responsibilities, tasks, and skills each job requires.

Standard 2

Create a Student portfolio.

- Prepare an electronic portfolio of projects developed in the class.

Standard 3

Participate in a CTSO, Utah Digital Media Arts, or competition.

Performance Skills

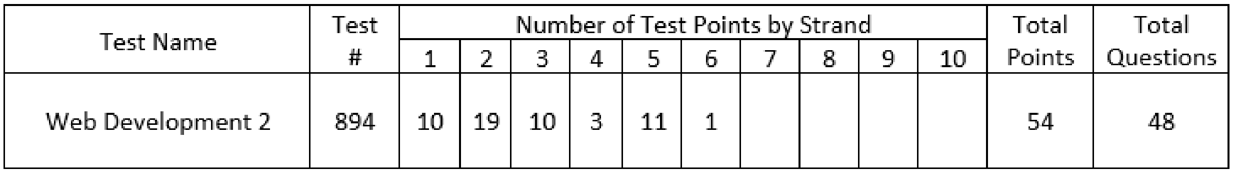
- Validate website HTML
- Validate Website CSS

Workplace Skills

The following workplace skills should be discussed, taught, and re-enforced in the course:

- Communication
- Teamwork
- Critical and Creative Thinking
- Problem Solving
- Dependability
- Legal requirements / expectations

Skill Certificate Test Points by Strand



Test Name	Test #	Number of Test Points by Strand										Total Points	Total Questions
		1	2	3	4	5	6	7	8	9	10		
Web Development 2	894	10	19	10	3	11	1					54	48

Vocabulary

Strand 1 – HTML	
SVG (Scalable Vector Graphics)	A file extension for scalable vector graphics
iframe	(Short for inline frame) Is an HTML element that allows an external webpage to be embedded in an HTML document.
Embed	Defines a container for an external resource, such as a web page, a picture, a media player, or a plug-in application.
GUI	Stands for "Graphical User Interface" and is pronounced "gooey." It is a user interface that includes graphical elements, such as windows, icons and buttons.

Strand 2 - CSS	
Class Selectors	Used to select multiple elements on a webpage.
ID Selectors	Used to select one element on a webpage.
Contextual Selectors	Are more specific than a single element selector Example #content h1 {background-color: red;}
Frameworks	Provides a foundation on which developers can build programs for websites. Example Bootstrap
Responsive Website	Will appear one way on a computer, another way on a tablet, and still another way on smartphone.
CSS Animations	To use CSS animation, you must first specify keyframes. An animation lets an element gradually change from one style to another.
CSS Graphic	Is essentially designing a vector graphic but instead of using vector illustration software you are using CSS code

Strand 3 – Site Planning	
Usability	Assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process.
Readability	The ease with which a person can understand a passage of text. The complexity of your content, as well as the typographic elements used on your site.
Accessibility	The inclusive practice of ensuring there are no barriers that prevent interaction with, or access to, websites on the World Wide Web by people with physical disabilities, situational disabilities, and socio-economic restrictions on bandwidth and speed.
Validation	Validating a website is the process of ensuring that the pages on the website conform to the norms or standards defined by various organizations
Wireframe	A visual prototype of a web page that focuses on content, layout, and behavior.
Site Map	A list of pages of a web site.
Cross-Browser Compatibility	If your website is not tested and debugged on different platforms and browsers, it won't work the same on all of them.
Image Resolution	The detail an image holds. The term applies to raster digital images, film images, and other types of images. Higher resolution means more image detail. Image resolution can be measured in various ways.
Image Optimization	To optimize an image so that it will run at its utmost capacity in terms of speed and efficiency.
IP Address	Internet Protocol Address. A uniquely assigned numeric address for each computer connected to the Internet.
IPv4	An ip address is in the format of 4 sets of decimal numbers separated by dots. Example 70.89.31.226
IPv6	Typically written in hexadecimal separated by colon. Example 2002:4559:1FE2::4559:1FE2
Static IP	A permanent number assigned to a computer by an Internet service provider (ISP).
Dynamic IP	A temporary IP address that is assigned to a computing device or node when it's connected to a network.
TCP/IP	Transmission Control Protocol/Internet Protocol. A network communication protocol used on the internet and computer networks to transmit data using packets.
Domain	The "address" or URL of a particular Web site.
Protocol	The common set of standards and rules ensure that all computers connected to the network have a common ground from which to work.
Web Host (Server)	A computer that stores and sends requested web pages and other files.
Favicon	A file containing one or more small icons, associated with a website or web page. Browser will display a page's favicon in the browser's address bar. Can be used to have a textless favorites site, saving space.

Strand 4 – Advanced Web Concepts	
Client-Side Scripting Languages	Runs in the user interface with which the user interacts in the web. It is mostly a browser or in the user's machine, that runs the code. Examples JavaScript, HTML, and CSS
Server-Side Scripting Languages	Deals with dynamic content and runs on the server. Examples PHP and Python
Database	A data structure that stores organized information. Most databases contain multiple tables, which may each include several different fields.
CMS (Content Management Systems)	Stands for "Content Management System." A CMS is a software tool that allows you to create, edit, and publish content to the web.

Strand 5 JavaScript	
JavaScript	A web-scripting code that interacts with HTML code to create dynamic content, such as rollovers or interactive forms on a web page; also called JS.
Variables	Containers for storing data values and must be identified with unique names.
Data Types	Numbers, strings, objects
Functions	Are executed when "something" invokes it (calls it)
Angle Brackets	<>
Brace Brackets	{}
Parentheses	()
Square Brackets	[]

Stand 6 Career Exploration	
Front End Developer	Developer who uses HTML, CSS, JavaScript, Databases, PHP to develop Web sites.
Back End Developer	Developer who uses CSS and image editing software to design the front end or public side of the website.
Full Stack Developer	Developers who are comfortable working with both back-end and front-end technologies. To be more specific, it means that the developer can work with databases, PHP, HTML, CSS, JavaScript and everything in between, also, venturing as far as converting Photoshop designs to front-end code.
UI Developer	Designers that are primarily concerned with how the product feels. A given design problem has no single right answer. These designers explore many different approaches to solving a specific user problem.
UX Developer	Designers that are particular about how the product is laid out. They are in charge of designing each screen or page with which a user interacts and ensuring that the design visually communicates the path that a UX designer has laid out.

Code List

<audio></audio>	The <audio> tag defines sound, such as music or other audio streams. Currently, there are 3 supported file formats for the <audio> element: MP3, Wav, and Ogg	**CSS - Continued**	
<video></video>	The HTML5 <video> element specifies a standard way to embed a video in a web page.	a:visited	Visited link
<canvas></canvas>	The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript. The <canvas> element is only a container for graphics. You must use JavaScript to actually draw the graphics. Canvas has several methods for drawing paths, boxes, circles, text, and adding images	a:active	Selected Link
<figure></figure>	The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.	a:hover	Selects links on mouse over
<mark></mark>	Use the <mark> tag if you want to highlight parts of your text.	Layout	The position property specifies the type of positioning method used for an element (relative, fixed, or absolute)
<picture></picture>	The <picture> element works similar to the <video> and <audio> elements. You set up different sources, and the first source that fits the preferences is the one being used.	width, height, auto	To format the page
<script></script>	Inserts Javascript	float	Positioning on the page
<source></source>	The <source> tag is used to specify multiple media resources for media elements, such as <video>, <audio>, and <picture>.	absolute positioning	The element is positioned relative to its first positioned (not static) ancestor element
<track></track>	The <track> tag specifies text tracks for media elements (<audio> and <video>).	relative positioning	The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position
<form></form>	defines a form that is used to collect user input Examples Radio button, checkbox, text, submit, and reset.	fixed positioning	The element is positioned relative to the browser window
<svg> </svg>	SVG is used to define vector-based graphics for the Web using XML. SVG has some predefined shape elements that can be used by developers	understand static positioning	Browser defaults
<rect />	Rectangle	align	The text-align property is used to set the horizontal alignment of text. A text can be left or right aligned, centered, or justified.
<circle />	Circle	margin	Margin properties are used to create space around elements, outside of any defined border.
<ellipse />	Ellipse	padding	Padding properties are used to generate space around an element's content, inside of any defined borders.
<line />	Line	buttons	shape that is created using CSS that the includes interactivity (User Driven Events)
<meta>	Defines metadata about an HTML document. Metadata is data (information) about data. <meta> tags always go inside the <head> element, and are typically used to specify character set, page description, keywords, author of the document, and viewport settings.	rounded corners	border-radius - A shorthand property for setting all the four borders
		colored button	Example: button1 {background-color: #4CAF50;} /* Green */
CSS		Navigation using CSS	list-style-type: none; - Removes the bullets. A navigation bar does not need list markers
internal stylesheet code	Defined inside the <style> element, inside the <head>< section of an HTML page	fade in/out	ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
external stylesheet link	<link rel="stylesheet" href="mystyle.css">		linear - specifies a transition effect with the same speed from start to end
id #	#firstname { background-color: yellow; }		ease-in - specifies a transition effect with a slow start
class .	.intro { background-color: yellow; }		ease-out - specifies a transition effect with a slow end
contextual selectors	example footer H1	image slider	Code that creates a "slideshow" on the web page
ID #	Used 1 time per page		
Class .	Used multiple times per page		
a:link	Unvisited link		