

1. Wk 0 Prep: Course Overview and Syllabus

1.1 Welcome to CSSE280 Fall 2020



Question set

1.1.1 Course expectations

1.2 Course Description and Learning Objectives



Question set

1.2.1 background

1.3 Class Meetings, Contact Info, Outside Help Class



Figure

1.3.1 Dr. Jason Yoder – Assistant Professor of Computer Science and Software Engineering



Figure

1.3.2 Professor Eliza Marcum - Visiting Assistant Professor of Computer Science and Software Engineering



Figure

1.3.3 manage team



Figure

1.3.4 Members of Team Channel



Figure

1.3.5 Click the members list for the TAsAndInstructors tag



Figure

1.3.6 From here you can start a chat to get private help from anyone available



Question set

1.3.1 Questions?

1.4 Resources



Question set

1.4.1 schedule

1.5 Course Structure and Grading



Question set

1.5.1 Policy questions

1.6 Academic Integrity

 **Question set**

1.6.1 academic integrity

1.7 Professional Behavior and Diversity Statements

 **Question set**

1.7.1 Channel for Concerns

2. Wk 0 Prep: Introduction to Web Programming (~58)

2.1 Web history

Internet

The Internet began as four networked computers in 1969 but is today the largest and most popular computer network spanning the entire globe.

FTP

FTP (File Transfer Protocol) was an early way for transferring files over the Internet.

File Transfer Protocol

FTP (File Transfer Protocol) was an early way for transferring files over the Internet.

World Wide Web

In the early 1990s, Tim Berners-Lee was working at a Swiss research institute named CERN and developed a more convenient way for computers to communicate files over the Internet. Berners-Lee named his creation the World Wide Web, or simply "the web".

web

In the early 1990s, Tim Berners-Lee was working at a Swiss research institute named CERN and developed a more convenient way for computers to communicate files over the Internet. Berners-Lee named his creation the World Wide Web, or simply "the web".

web page

A web page is a document that is viewed in a web browser.

website

A collection of related web pages are organized into a website.

web server

A web server is a program that serves web pages to web browsers.

HyperText markup language

HyperText markup language (HTML) is the standard markup language for web documents.

HTML

HyperText markup language (HTML) is the standard markup language for web documents.

Hypertext

Hypertext is text that has links to other text (and today to images, videos, and more).

markup

Document markup is special markings in the document that provide additional information about links, formatting, and images.

web browser

A web browser is a program that downloads an HTML document from a web server, displays the document to the user with the appropriate formatting, and allows the user to interact with the document, such as clicking hyperlinks to access other documents.

World Wide Web Consortium

The World Wide Web Consortium (W3C) is the international standards organization that traditionally has controlled a number of web standards, including HTML.

W3C

The World Wide Web Consortium (W3C) is the international standards organization that traditionally has controlled a number of web standards, including HTML.

Web Hypertext Application Technology Working Group

The Web Hypertext Application Technology Working Group (WHATWG), an organization that develops a variety of web standards and whose members include the major browser vendors.

WHATWG

The Web Hypertext Application Technology Working Group (WHATWG), an organization that develops a variety of web standards and whose members include the major browser vendors.

HTML Living Standard

WHATWG produces the HTML Living Standard, a continually evolving standard without version numbers that replaces HTML5.

Figure

2.1.1 NASA public FTP site.

Animation

2.1.1 The web uses HTML files, web browsers, and HTTP to provide a more convenient way for computers to communicate on the web.

Figure

2.1.2 The web's name comes from the interconnections of computers being like the interconnections of a spider's web.

Figure

2.1.3 One of the first web pages.

▀▀▀ Aside

Hypertext history

██ Question set

2.1.2 The web.

▀▀▀ Table

2.1.1 Number of websites and Internet users per year.

██ Question set

2.1.3 Websites per year.

▀▀▀ Figure

2.1.4 Web browser timeline.

▀▀▀ Aside

W3C vs. WHATWG

██ Question set

2.1.4 Significant points in web history.

██ Question set

2.1.5 Separation of structure, presentation, and interaction.

▀▀▀ Aside

Great example

2.2 IP addresses, domain names, and URLs

packet

An Internet packet contains To and From IP addresses, the information to communicate, and other configuration information.

IP address

An IP address (short for Internet Protocol address) is a computer's unique address on the Internet (like a house's unique address in the world), usually represented numerically like 198.51.100.7.

Internet Protocol

An IP address (short for Internet Protocol address) is a computer's unique address on the Internet (like a house's unique address in the world), usually represented numerically like 198.51.100.7.

IPv4

The original Internet Protocol, known as IPv4, has 32-bit addresses.

IPv6

A new version of the Internet Protocol, IPv6, uses 128-bit addresses, capable of representing 2^{128} addresses.

domain name

A domain name is a name for an IP address, such as the name wikipedia.org for the IP address 198.35.26.96; the name is easier to remember and type.

DNS server

When a computer sends a packet using a domain name over the Internet, the first step is to contact a DNS server to convert the domain name to an IP address.

DNS

DNS is short for Domain Name System.

root servers

Thirteen main DNS servers (called root servers) exist in the world, and a computer's operating system or an ISP keeps a reference to the root servers' IP addresses.

domain name registrar

Anyone may register an unused domain name with a domain name registrar.

top-level domains

A domain name belongs to one of numerous top-level domains (TLD), such as .com, .net, .org, .edu, and .gov.

TLD

A domain name belongs to one of numerous top-level domains (TLD), such as .com, .net, .org, .edu, and .gov.

country code top-level domain

Each country is assigned a unique two-letter country code top-level domain (ccTLD) like .uk (United Kingdom), .ru (Russia), and .de (Germany).

ccTLD

Each country is assigned a unique two-letter country code top-level domain (ccTLD) like .uk (United Kingdom), .ru (Russia), and .de (Germany).

ICANN

ICANN, the organization that manages TLDs, now allows companies and organizations to create customized TLDs, like .church, .pizza, and .music.

second-level domain

Immediately after a top-level domain comes a second-level domain, such as wikipedia in wikipedia.org.

URL

A URL (Uniform Resource Locator) is the location of a web resource on the web, such as <http://www.cdc.gov/alcohol/faqs.htm>.

Uniform Resource Locator

A URL (Uniform Resource Locator) is the location of a web resource on the web, such as <http://www.cdc.gov/alcohol/faqs.htm>.

web resource

A web resource is any retrievable item, like an HTML file, image, video, CSS style sheet, etc.

Scheme

Scheme - Characters at the beginning of a URL followed by a colon ":" or a colon and double slashes "://".

Hostname

Hostname - The complete domain name following the scheme in a URL.

Path

Path - All characters to the right of the hostname in a URL.

404

If a web server is reached but the specific requested page isn't found, the server returns a 404 status code, which is a code number for page not found.

Linkrot

Linkrot is the general name for a once valid link that now return a 404 status code.

Animation

2.2.1 IP addresses.

Question set

2.2.2 IP addresses.

Animation

2.2.3 A DNS server first looks up a domain name's IP address.

Aside

Registering a domain name

Question set

2.2.4 IP addresses and domain names.

Try

2.2.1 IP addresses and domain names.

 **Animation**

2.2.5 Common top-level domains.

 **Question set**

2.2.6 Domain hierarchy.

 **Aside**

Note

 **Figure**

2.2.1 Chrome hides the scheme and "www" in hostname.

 **Animation**

2.2.7 A URL can specify the location of subfolders and files.

 **Question set**

2.2.8 URLs.

 **Figure**

2.2.2 An example web page that returns a message with the code 404 shown.

 **Aside**

Linkrot

 **Question set**

2.2.9 Web page errors.

2.3 HTTP

HyperText Transfer Protocol (HTTP)

The HyperText Transfer Protocol (HTTP) is a networking protocol that runs over TCP/IP and governs communication between web browsers and web servers.

Transmission Control Protocol/Internet Protocol (TCP/IP)

Transmission Control Protocol/Internet Protocol (TCP/IP) is a protocol suite that governs how packets of data are transferred over the Internet from one machine to another.

HTTP/1.1

HTTP/1.1 is the HTTP standard used for most of the web's lifetime.

HTTP/2

Many websites are adopting HTTP/2, a new HTTP standard that speeds-up the transfer of information between web browsers and web servers.

DNS lookup

The web browser performs a DNS lookup by sending the domain name to the local DNS and getting back the IP address of the web server hosting the domain name.

HTTP request

An HTTP request is a message sent from the web browser to the web server. Often the request asks the web server to send back a web resource like an HTML file, image, CSS style sheet, JavaScript file, or video.

HTTP response

An HTTP response is a message sent from the web server back to the web browser in response to an HTTP request. Often the response contains the requested web resource.

status line

The status line specifies the HTTP version being used. A request status line includes a request type and path; a response status line includes a status code.

header field

A header field is a keyword followed by a colon and a value. Header fields supply additional information about the request or response.

message body

A message body contains data being transferred between a web browser and web server. In a request, the message body may be empty or contain submitted form data. In a response, the message body may contain the requested resource.

request method

An HTTP request method indicates the desired action to perform on a resource.

status code

An HTTP response status code is a three digit number that indicates the status of the requested resource.

browser redirect

A browser redirect is when the web server returns a 301 or 302 status code with a Location header indicating the URL the browser should load next.

URL shortening

URL shortening is a technique to create shorter URLs that redirect to longer URLs. Ex: http://en.wikipedia.org/wiki/URL_shortening has a short URL of <http://tinyurl.com/urlwiki>.

browser cache

A browser cache is an area on the computer's disk where web content can be stored by the web browser for quick retrieval later.

entity tag (ETag)

An entity tag (ETag) is an identifier for a specific version of a web resource. Ex: 34905a3e285dd11. When the resource changes, so should the ETag associated with the resource.

If-Modified-Since

If-Modified-Since is used with the Last-Modified date/time to request the web server only send the requested resource if the resource has changed since the specified date/time.

Expires

Expires contains a date/time indicating when the requested resource is considered "stale".

Cache-Control

Cache-Control is used to specify a number of caching directives.

network sniffer

A network sniffer is software that monitors network traffic and allows users to inspect HTTP requests and responses.

HTTPS

HTTPS encrypts HTTP traffic between a browser and web server so a network sniffer cannot intercept sensitive information in the HTTP traffic like passwords, credit card numbers, financial transactions, etc.

Transport Layer Security

HTTPS uses a protocol called Transport Layer Security (TLS), which uses asymmetric public keys to encrypt data between the browser and web server.

TLS

HTTPS uses a protocol called Transport Layer Security (TLS), which uses asymmetric public keys to encrypt data between the browser and web server.

digital certificate

A website wanting to use HTTPS must acquire a digital certificate, issued by a trusted certificate authority, that contains a public key used by TLS to encrypt data.

certificate authority

A website wanting to use HTTPS must acquire a digital certificate, issued by a trusted certificate authority, that contains a public key used by TLS to encrypt data.

Aside

HTTP/1.1 and HTTP/2

Animation

2.3.1 HTTP requests and responses.

 **Question set**

2.3.2 HTTP requests and responses.

 **Figure**

2.3.1 HTTP request with no message body.

 **Figure**

2.3.2 HTTP response with HTML in message body.

 **Question set**

2.3.3 HTTP request and response headers.

 **Aside**

List of HTTP headers

 **Aside**

Chrome DevTools for watching HTTP traffic.

 **Table**

2.3.1 Common HTTP request methods.

 **Table**

2.3.2 Common HTTP response status codes.

 **Aside**

URL shortening

 **Question set**

2.3.4 HTTP request methods and response status codes.

 **Animation**

2.3.5 Requesting cached resources with ETags.

 **Question set**

2.3.6 Requesting cached content.

 **Aside**

Other caching mechanisms

 **Aside**

Viewing 304 responses in Chrome DevTools

 **Figure**

2.3.3 Chrome, Firefox, and Edge browsers showing a padlock symbol when HTTPS is used.

Question set

2.3.7 Steps in an HTTPS transaction.

2.4 Web trends

mobile device

A mobile device is a handheld computer, like a smartphone or tablet.

Internet of Things

The Internet of Things (abbreviated as IoT) is the global collection of communicating devices that sense and control technology on behalf of humans.

Web accessibility

Web accessibility is the ability of users with disabilities to access and use a web page with reasonable effort.

Cognitive computing

Cognitive computing is the use of artificial intelligence techniques and access to vast amounts of data to simulate human problem solving in complex situations with ambiguity, changing data, and even conflicting information.

Separation of concerns

Separation of concerns is the design principle of breaking up web content using distinct languages and documents that overlap as little as possible.

Aside

Making predictions

Figure

2.4.1 Global browser usage trends.

Question set

2.4.1 Implications of the market share browsers.

Figure

2.4.2 Global browser usage trends: Desktop vs. mobile.

Question set

2.4.2 Important mobile development topics.

Example

2.4.1 Example Internet of Things devices.

 **Question set**

2.4.3 Internet of Things.

 **Figure**

2.4.3 Web accessibility timeline.

 **Question set**

2.4.4 Accessibility fact check.

 **Figure**

2.4.4 IBM Watson cognitive computing system.

 **Question set**

2.4.5 Review trends in web technology.

2.5 Introduction to HTML

HTML

HTML (HyperText Markup Language) is a textual language for creating web pages.

tag

A tag has a descriptive name surrounded by < and > characters that the web browser uses to display content.

link

A link on a web page is a clickable item that causes the web browser to jump to another web page when clicked.

 **Figure**

2.5.1 An HTML file and the rendered web page.

 **Table**

2.5.1 Common HTML tags.

 **Learning tool**

2.5.1 Practice with HTML tags.

 **Question set**

2.5.2 Match HTML basic formatting.

 **Learning tool**

2.5.3 Creating a link in HTML.

 **Question set**

2.5.4 HTML links.

2.6 Introduction to CSS

Cascading Style Sheets

Cascading Style Sheets (CSS) is a textual language for describing how a web page is styled for visual presentation.

CSS

Cascading Style Sheets (CSS) is a textual language for describing how a web page is styled for visual presentation.

CSS rule

A CSS rule specifies styling properties for specific HTML elements.

CSS color

A CSS color can be a pre-defined name like blue, or an rgb value like `rgb(50, 100, 255)`.

rgb

Rgb is short for red, green, blue; all colors can be formed by combining those three colors.

Figure

2.6.1 HTML code without and with CSS style rules.

Table

2.6.1 Common CSS properties.

Learning tool

2.6.1 Styling a web page with CSS.

Table

2.6.2 Some CSS colors using rgb.

Question set

2.6.2 CSS colors.

Learning tool

2.6.3 CSS colors.

2.7 Introduction to JavaScript

JavaScript

JavaScript is a programming language that runs in a browser, enabling web pages supporting actions like responding to a button click.

function

A JavaScript function is a named group of statements that can be run by referring to that name.

variable

A variable stores a value or a link to an element of a web page.

Learning tool

2.7.1 JavaScript to change colors.

Question set

2.7.2 The colorable JavaScript example.

Learning tool

2.7.3 Updating user ratings.

Question set

2.7.4 JavaScript for updating user ratings.

Question set

2.7.5 JavaScript clock example.

3. Wk 1 Prep: HTML (~63)

3.1 HTML document structure

HTML

HTML (HyperText Markup Language) is a textual language for creating web pages.

element

An element is a single HTML structure.

tag

An HTML tag is a markup instruction identified by <, a tag name, and a >.

opening tag

An opening tag indicates the starting point in the document where the tag takes effect.

closing tag

A closing tag indicates the ending point in the document where the tag stops having an effect.

<!DOCTYPE html>

The <!DOCTYPE html> declaration instructs the web browser about what type of document follows.

<html>

The <html> opening and closing tags enclose everything but the <!DOCTYPE html> declaration.

<meta>

The <meta> tag specifies metadata, which is data that describes the document's data.

<title>

The <title> opening and closing tags enclose the name of the document. The title is usually displayed in the titlebar of the browser, is used by search engines, and is used for bookmarking.

<body>

The <body> opening and closing tags enclose all elements and content to be rendered in the browser.

attribute

A tag attribute is a value that provides additional information about a particular tag and is included after the tag name but before the > in the tag.

void element

A void element is an element that only needs an opening tag.

Quirks mode

Quirks mode is a browser rendering mode that renders a web page using non-standard layout rules emulating older web browsers.

standards mode

A web page with the DOCTYPE is rendered using standards mode, which uses the HTML Living Standard.

UTF-8

UTF-8 is a common character encoding that describes characters used by most of the world's languages.

<head>

The <head> tag is used to contain the document title, document metadata, and various other elements that are typically not displayed.

deprecated

A deprecated feature is part of a language that is officially discouraged because newer or better features exist, or because the deprecated feature is no longer considered safe.

HTML validator

An HTML validator checks that an HTML document conforms to the standard.

Question set

3.1.1 Understanding the HTML acronym.

Figure

3.1.1 HTML tags and element.

Question set

3.1.2 Valid HTML markup.

Figure

3.1.2 HTML page source.

Question set

3.1.3 View source in action.

Animation

3.1.4 Minimum HTML document structure.

Aside

Notes

Question set

3.1.5 Matching basic document tag order.

Aside

View elements in Chrome DevTools

Question set

3.1.6 HTML standardization and validation.

3.2 Basic HTML tags

paragraph

A paragraph is the basic unit of text in an HTML document. A paragraph is enclosed in HTML by the `<p>` opening and closing tags. Browsers visually separate paragraphs from other text with spacing above and below.

<p>

A paragraph is the basic unit of text in an HTML document. A paragraph is enclosed in HTML by the `<p>` opening and closing tags. Browsers visually separate paragraphs from other text with spacing above and below.

whitespace

A whitespace character is an unprinted character such as the spaces between words and lines of text.

**
**

The `
` tag creates a line break in a paragraph, such that the content after the line break appears on a new line.

line break

The `
` tag creates a line break in a paragraph, such that the content after the line break appears on a new line.

section

A section is a collection of related paragraphs, images, and other content organized into units such as chapters, subchapters, tabbed content in a tabbed container, and numbered sections of a thesis. Sections are surrounded by the `<body>` and `<section>` tags, or are implicitly created using a heading tag.

`<section>`

A section is a collection of related paragraphs, images, and other content organized into units such as chapters, subchapters, tabbed content in a tabbed container, and numbered sections of a thesis. Sections are surrounded by the `<body>` and `<section>` tags, or are implicitly created using a heading tag.

heading

A heading provides a title for a section of content.

`<h1>`

The first level heading is surrounded by the `<h1>` opening and closing tags, and is appropriate for chapter titles.

`<h2>`

The second level heading is surrounded `<h2>` tags, and is appropriate for subchapter titles.

`<h3>`

The third through sixth level headings tags are `<h3>`, `<h4>`, `<h5>`, and `<h6>`.

`<h4>`

The third through sixth level headings tags are `<h3>`, `<h4>`, `<h5>`, and `<h6>`.

`<h5>`

The third through sixth level headings tags are `<h3>`, `<h4>`, `<h5>`, and `<h6>`.

`<h6>`

The third through sixth level headings tags are `<h3>`, `<h4>`, `<h5>`, and `<h6>`.

``

The `` tag indicates emphasized text, such as text having an emphasized pronunciation when spoken, and is italicized by default.

emphasized

The `` tag indicates emphasized text, such as text having an emphasized pronunciation when spoken, and is italicized by default.

``

The `` tag indicates text that has strong importance, and is bolded by default.

strong importance

The `` tag indicates text that has strong importance, and is bolded by default.

<cite>

The `<cite>` tag denotes a title, such a book or song title, and is italicized by default.

<mark>

The `<mark>` tag denotes important content that should be semantically highlighted and is rendered with a yellow background by default.

The `` tag indicates text that needs attention, like key words in a document abstract or product names in a review, and renders the text in bold.

bold

The `` tag indicates text that needs attention, like key words in a document abstract or product names in a review, and renders the text in bold.

<i>

The `<i>` tag indicates text in an alternative voice, such as a word or phrase in a foreign language, and is rendered using italics.

italics

The `<i>` tag indicates text in an alternative voice, such as a word or phrase in a foreign language, and is rendered using italics.

<u>

The `<u>` tag denotes text that should appear differently from normal text, such as misspelled words, and is underlined by default.

Animation

3.2.1 <p> tags enclose a paragraph.

Learning tool

3.2.2 Add <p> tags to enclose the paragraphs.

Animation

3.2.3 Whitespace in HTML.

Learning tool

3.2.4 Line break.

Question set

3.2.5 Whitespace and line breaks.

 **Animation**

3.2.6 Section headings.

 **Question set**

3.2.7 Match the HTML tag with the tag's meaning.

 **Question set**

3.2.8 Sections and headings.

 **Learning tool**

3.2.9 Headings activity.

 **Question set**

3.2.10 Hierarchical structure of documents.

 **Question set**

3.2.11 Valid or invalid HTML.

 **Aside**

Note

 **Table**

3.2.1 Text-formatting tags.

 **Aside**

Note

 **Question set**

3.2.12 Valid or invalid HTML.

 **Progression**

3.2.1 Basic HTML tags.

3.3 Comments

comment

An HTML comment is a portion of the document that is not displayed by the browser.

 **Learning tool**

3.3.1 Comments.

 **Question set**

3.3.2 Valid or invalid comments.

3.4 Lists

unordered list

An unordered list is a collection of items, usually indented and shown using bullets, surrounded by the opening and closing tags.

An unordered list is a collection of items, usually indented and shown using bullets, surrounded by the opening and closing tags.

list item

Each list item is surrounded by opening and closing tags.

Each list item is surrounded by opening and closing tags.

ordered list

An ordered list is a sequenced collection of items, usually indented and shown using numbers or letters, surrounded by the opening and closing tags.

An ordered list is a sequenced collection of items, usually indented and shown using numbers or letters, surrounded by the opening and closing tags.

type

The tag's numbering scheme is specified with the type attribute.

list-style-type

The CSS property list-style-type provides the ability to change the bullet used in an unordered list and offers more numbering options in an ordered list.

nested list

A nested list is a list within a list item of another list.

Animation

3.4.1 Unordered list.

Question set

3.4.2 Unordered lists and list items.

Question set

3.4.3 Unordered list tags.

Table

3.4.1 Ordered list types.

Aside

CSS list-style-type

Learning tool

3.4.4 An ordered list: High demand majors.

Question set

3.4.5 Ordered list.

Learning tool

3.4.6 Nested lists.

Question set

3.4.7 Nested lists.

Progression

3.4.1 Lists.

3.5 Tables

table

A table is an HTML structure surrounded by the `<table>` opening and closing tags that allows data to be organized in rows and columns.

`<table>`

A table is an HTML structure surrounded by the `<table>` opening and closing tags that allows data to be organized in rows and columns.

cell

A cell is a location in the table at a specific row and column.

`<tr>`

The `<tr>` opening and closing tags create a table row, which contains all the row's cells.

table row

The `<tr>` opening and closing tags create a table row, which contains all the row's cells.

`<th>`

The `<th>` opening and closing tags create a new table cell containing table header information about the data.

table header

The `<th>` opening and closing tags create a new table cell containing table header information about the data.

`<td>`

The `<td>` opening and closing tags create a new table cell containing a table datum.

table datum

The <td> opening and closing tags create a new table cell containing a table datum.

caption

A table caption defines a short descriptive text for a table and is created with <caption> tags.

<caption>

A table caption defines a short descriptive text for a table and is created with <caption> tags.

colspan

A single table cell occupies a single row and column by default, but a cell may span multiple columns or multiple rows using the colspan and rowspan attributes.

rowspan

A single table cell occupies a single row and column by default, but a cell may span multiple columns or multiple rows using the colspan and rowspan attributes.

<thead>

The <thead> tag specifies the table header.

<tbody>

The <tbody> tag specifies the table body.

<tfoot>

The <tfoot> tag specifies the table footer.

Aside

Note

Animation

3.5.1 HTML tables organize data into rows and columns.

Question set

3.5.2 Table structure.

Question set

3.5.3 Creating tables.

Learning tool

3.5.4 Add table row.

Learning tool

3.5.5 Add table column.

Aside

Note

Question set

3.5.6 Identify missing table tags.

Aside

Ugly tables

Learning tool

3.5.7 Spanning multiple columns and rows.

Question set

3.5.8 Spanning multiple columns and rows.

Progression

3.5.1 Modifying a table.

Question set

3.5.9 Head, body, footer.

3.6 Images

The tag displays an image in a web page.

src attribute

The src attribute specifies the URL location of the image file to display.

alt attribute

The alt attribute provides a text description to use as an alternative to displaying the image.

width

The width and height attributes are optional attributes supported by tags that tell the browser how many pixels the image should occupy.

height

The width and height attributes are optional attributes supported by tags that tell the browser how many pixels the image should occupy.

aspect ratio

An image aspect ratio is the ratio of the image width to the image height.

JPEG

The JPEG (Joint Photographic Experts Group) image format is commonly used for digital photographs.

Joint Photographic Experts Group

The JPEG (Joint Photographic Experts Group) image format is commonly used for digital photographs.

PNG

The PNG (Portable Network Graphics) image format is commonly used for line art and screenshots or images requiring transparency.

Portable Network Graphics

The PNG (Portable Network Graphics) image format is commonly used for line art and screenshots or images requiring transparency.

GIF

The GIF (Graphics Interchange Format) image format is commonly used for simple animated images.

Graphics Interchange Format

The GIF (Graphics Interchange Format) image format is commonly used for simple animated images.

lossy compression

JPEG images use lossy compression, meaning that some of the original picture information is lost when compressed.

lossless compression

PNG and GIF images use lossless compression, meaning no image quality is lost.

true color

JPEG and PNG images can display true color (24-bit color), which is approximately 16 million different colors.

WebP

Google created the WebP image format that supports true color, transparency, and animation.

<link>

A <link> tag, which defines a link between a web page and an external resource.

Animation

3.6.1 Loading images in HTML.

Question set

3.6.2 Images.

Question set

3.6.3 Reasons for using alt attribute for images.

 **Learning tool**

3.6.4 Image sizing.

 **Question set**

3.6.5 Image width and height attributes.

 **Question set**

3.6.6 Valid HTML code.

 **Table**

3.6.1 Three popular image formats.

 **Aside**

WebP

 **Question set**

3.6.7 Image formats.

 **Aside**

Favicon

 **Progression**

3.6.1 Images.

3.7 Links

anchor tag

The anchor tag `<a>` defines a hyperlink in a web page.

`<a>`

The anchor tag `<a>` defines a hyperlink in a web page.

hyperlink

A hyperlink, or link, specifies where other information is located and what action the web browser should perform when a user clicks the hyperlink.

href

The href attribute specifies the hyperlink's URL.

fragment

A URL can point to a section, or fragment, of a document by adding a hash tag (#) and a fragment identifier at the end of the URL.

id attribute

Adding the id attribute to any HTML tag creates a fragment identifier, thus permitting URLs to link directly to the id's location in the document.

fragment identifier

Adding the id attribute to any HTML tag creates a fragment identifier, thus permitting URLs to link directly to the id's location in the document.

graphical hyperlink

A graphical hyperlink or image link uses an image inside a hyperlink instead of text.

target

An anchor tag's target attribute indicates how the browser should display the link when clicked.

absolute URL

An absolute URL is a complete URL.

relative URL

A relative URL specifies the relative path to the web resource with no scheme or hostname.

Question set

3.7.1 URL schemes.

Question set

3.7.2 Hyperlink actions.

Learning tool

3.7.3 Create a hyperlink to a web page.

Learning tool

3.7.4 Create a hyperlink to a web page fragment.

Figure

3.7.1 The id attribute creates fragment identifiers for two paragraphs.

Question set

3.7.5 Section links.

Figure

3.7.2 Graphical hyperlink: An image used within a hyperlink.

Figure

3.7.3 Target attribute: browser will open the link in a new tab.

Question set

3.7.6 Graphical hyperlinks and the target attribute.

Animation

3.7.7 Using relative URLs to link to files on the same website.

Question set

3.7.8 Relative URLs.

Question set

3.7.9 Hyperlink structure.

Progression

3.7.1 Links.

3.8 <script> and <style>

<script>

The <script> tag allows a web page to include interactive content, which the browser assumes to be JavaScript unless indicated otherwise.

<style>

The <style> tag allows the web page to introduce presentational directives, usually CSS.

Example

3.8.1 Two ways of using <script> tags.

Example

3.8.2 Using the <style> tag.

Aside

Note

Question set

3.8.1 <script> and <style> tags.

3.9 Special characters

entity

An entity is a mechanism for writing special characters or symbols in HTML, such as mathematical symbols, characters in most languages, and many other symbols.

non-breaking character

A non-breaking character is an inter-word character that permits treating the words on both sides to be one word.

non-breaking hyphen

A non-breaking hyphen, ‑, looks like a regular hyphen but acts like a normal character in the middle of a word.

non-breaking space

A non-breaking space, , looks like a single space but acts like a normal character in the middle of a word.

Table

3.9.1 HTML entity formats.

Question set

3.9.1 HTML entities for special symbols.

Question set

3.9.2 HTML entities.

Question set

3.9.3 HTML entities for spaces and hyphens.

3.10 Example: Band web page

wireframe

A wireframe is a blueprint, showing where the future content will be arranged.

Figure

3.10.1 Wireframe for the band web page.

Figure

3.10.2 Initial HTML for band.html web page.

Question set

3.10.1 Initial design.

Question set

3.10.2 Photo and navigation links.

Figure

3.10.3 HTML for Members section.

Question set

3.10.3 Members section.

Figure

3.10.4 HTML for Concert section.

 **Question set**

3.10.4 Concerts section.

 **Figure**

3.10.5 HTML for contacting the band.

 **Question set**

3.10.5 Contact section.

3.11 LAB: News article (HTML)

 **Lab activity**

3.11.1 LAB: News article (HTML)

3.12 LAB: Hometown web page (HTML)

 **Lab activity**

3.12.1 LAB: Hometown web page (HTML)

3.13 LAB: Photo gallery table (HTML)

 **Lab activity**

3.13.1 LAB: Photo gallery table (HTML)

3.14 LAB: Poem (HTML)

 **Lab activity**

3.14.1 LAB: Poem (HTML)

3.15 LAB: HTML practice

 **Lab activity**

3.15.1 LAB: HTML practice

4. Wk 1 Prep: Basic CSS (~58)

4.1 Using CSS in HTML

CSS

CSS (Cascading Style Sheets) is a textual language for describing how a web page is styled for visual presentation.

CSS rule

A CSS rule consists of a selector, an opening curly brace ({}), a declaration block, and then a closing curly brace ()).

selector

A CSS selector specifies the HTML elements to which the specific style rule applies.

declaration block

A declaration block contains one or more declarations separated by semicolons (;).

declaration

A CSS styling declaration is a CSS property followed by a colon (:) and the property value.

Inline style

Inline style places CSS declarations inside a tag's style attribute.

style attribute

Inline style places CSS declarations inside a tag's style attribute.

Embedded stylesheet

Embedded stylesheet places CSS rules in an HTML document's head using <style> tags.

External stylesheet

External stylesheet places CSS rules in a separate file and imports the stylesheet into an HTML document using <link href="stylesheet.css" rel="stylesheet">.

inheritance

The style declarations from a parent element cascade down and are applied to any child elements, a concept called inheritance.

conflict

When two style declarations with identical properties apply to the same element, a conflict occurs.

specificity

The browser uses specificity, a scoring system that resolves more complex CSS conflicts, to determine what style declarations apply to an element.

!important

The !important rule may be used on a style declaration to override other declarations and disregard specificity.

inherit

The inherit keyword, which specifies that a property should inherit the parent element's value.

Question set

4.1.1 Understanding the CSS acronym.

Figure

4.1.1 Parts of a CSS rule.

Aside

Semicolons.

 **Question set**

4.1.2 Valid CSS.

 **Animation**

4.1.3 Three ways to apply CSS.

 **Question set**

4.1.4 Applying CSS.

 **Aside**

Specificity and !important

 **Animation**

4.1.5 Applying styles to HTML elements.

 **Question set**

4.1.6 Inheritance and conflicts.

 **Question set**

4.1.7 Applying CSS declarations.

 **Aside**

Viewing CSS with Chrome DevTools

 **Progression**

4.1.1 Writing CSS rules.

4.2 Basic selectors

element selector

The element selector matches elements with the specified element names.

class selector

The class selector, specified with a period character followed by the class name, matches elements that have the specified class name.

ID selector

The ID selector, specified with a hash character followed by the ID name, matches the element that has the specified ID.

descendant selector

The descendant selector, specified with a selector followed by a space and another selector, matches elements that are contained in other elements.

pseudo-class selector

The pseudo-class selector, specified with a colon character followed by a pseudo-class name, matches elements based on user behavior or element metainformation.

class attribute

An HTML tag's class attribute specifies the classes to which the tag belongs, with each class name separated by a space.

Animation

4.2.1 Applying styles to HTML elements.

Question set

4.2.2 Basic selectors.

Question set

4.2.3 Descendant matching.

Question set

4.2.4 Basic CSS selector matching.

Question set

4.2.5 Practice identifying CSS selectors.

Question set

4.2.6 Practice using CSS selectors.

Learning tool

4.2.7 Element, class, and ID selectors.

Learning tool

4.2.8 Pseudo-class and descendant selectors.

Progression

4.2.1 Basic selectors.

4.3 Advanced selectors

universal selector

The universal selector, specified using an asterisk character (*), matches *all* elements in the web page.

multiple selector

The multiple selector, specified using a comma (,) to separate selectors, matches all listed elements to apply a style rule.

child selector

The child selector, specified using a greater than character (>) between two selectors, matches any elements where the second element is a direct child of the first element.

Sibling elements

Sibling elements are elements that share the same parent element.

general sibling selector

The general sibling selector, specified using a tilde character (~) between two selectors, matches the second element if the second element occurs after the first element and both elements are siblings.

adjacent sibling selector

The adjacent sibling selector, specified using a plus character (+) between two selectors, matches an element that immediately follows another element, where both elements have the same parent.

Combinators

Combinators are CSS selectors that match specific relationships between other selectors.

attribute selector

The attribute selector, specified with an attribute name and optional value comparison enclosed in square brackets ([and]), matches elements with the specified attribute or the specified attribute and value.

pseudo element selector

The pseudo element selector, specified with two colon characters (:) followed by a pseudo-element, matches parts of elements.

Figure

4.3.1 Comparing separate and multiple selectors.

Animation

4.3.1 Styling background color and image.

Question set

4.3.2 List Item Selectors.

Question set

4.3.3 Matching specific elements.

Learning tool

4.3.4 Child and multiple selectors.

Figure

4.3.2 General sibling selector.

 **Figure**

4.3.3 Adjacent sibling selector.

 **Aside**

Combinators.

 **Question set**

4.3.5 Identify CSS selector names.

 **Table**

4.3.1 Common attribute selector comparators.

 **Question set**

4.3.6 Attribute selectors.

 **Table**

4.3.2 Common pseudo-element selectors.

 **Question set**

4.3.7 Pseudo-element selectors.

 **Progression**

4.3.1 Advanced selectors.

4.4 Common properties

color

The color CSS property changes the color of the text to a specified color value.

RGB color value

An RGB color value specifies a color using the `rgb(red, green, blue)` function by indicating the red, green, and blue intensities.

hexadecimal color

A hexadecimal color specifies a color using the `#RRGGBB` format by indicating the red, green, and blue intensities.

HSL color value

An HSL color value specifies a color using the `hsl(hue, saturation, lightness)` function by indicating the hue, saturation, and lightness values.

RGBA color value

The RGBA color value specifies a color using the `rgba(red, green, blue, alpha)` function by indicating the red, green, blue, and alpha intensities.

HSLA color value

The HSLA color value specifies a color using the `hsla`(hue, saturation, lightness, alpha) function by indicating the hue, saturation, lightness, and alpha intensities.

background-color

The background-color property specifies the background color.

background-image

The background-image property specifies a background image.

background

The background property is shorthand for setting several of the element's background properties at the same time.

float

The float property specifies whether the element will float to the right or left of the element's parent container, allowing text to flow around the element.

clear

The clear property can stop elements from floating.

display

The display property controls the layout of the element on a web page.

CSS variable

A CSS variable is a custom CSS property that defines a value. A CSS variable is declared in a CSS selector that defines the variable's scope.

:root

A CSS variable can have global scope by declaring the variable in the :root selector, which targets the highest DOM element: the `<html>` element.

var()

A CSS variable is accessed with the `var()` function. Ex: `var(--my-variable);`

Question set

4.4.1 Color values and names.

Learning tool

4.4.2 CSS text color.

Animation

4.4.3 Styling background color and image.

Question set

4.4.4 Determining background properties.

Learning tool

4.4.5 Floating images.

Question set

4.4.6 Float property values.

Question set

4.4.7 Display CSS values.

Animation

4.4.8 Using CSS variables.

Question set

4.4.9 CSS variables.

Progression

4.4.1 Common properties.

4.5 Font and text properties

font-family

The font-family property specifies the font family, such as "Times New Roman" or serif.

font-size

The font-size property changes the font size, such as 120%, small, or 12px.

font-weight

The font-weight property specifies the font weight, such as normal or bold.

font-style

The font-style property changes the text style, such as normal, italic, or oblique.

font-variant

The font-variant property specifies the variant of the text, such as normal or small-caps.

font

The font property is shorthand for setting several font properties at the same time.

web font

A web font is a CSS feature that allows custom fonts to be downloaded to the web browser.

absolute size

An absolute size is a size that is fixed and independent of other CSS sizes.

relative size

A relative size is a size that is relative to another size.

text-align

The text-align property changes the horizontal alignment of text for an element.

text-decoration

The text-decoration property can add or remove text decorations like underlining or a ~~line-through~~.

text-transform

The text-transform property converts letters to UPPERCASE, lowercase, or Capitalizes Initial Letters.

text-indent

The text-indent property specifies the first line's indentation amount.

Example

4.5.1 Generic family names.

Aside

Web fonts

Figure

4.5.1 Changing the font size for the entire web page.

Question set

4.5.1 Computing font properties.

Table

4.5.1 Text alignment examples.

Aside

Underlined links

Question set

4.5.2 Text alignment.

Question set

4.5.3 Text properties.

Progression

4.5.1 Font and text properties.

4.6 Box model

box model

The box model describes the size of each element as a series of nested boxes.

Content

Content: The innermost box contains the content of the element, such as text and images.

Padding

Padding: The padding box contains the content box and adds a transparent area around the content.

Border

Border: The border box contains the padded content and adds an optionally colored area around the padding.

Margin

Margin: The margin box contains all three boxes and adds a transparent area around the border.

padding

The padding property specifies the padding thickness.

border

The border property specifies the border's thickness, style, and color.

margin

The margin property specifies the margin thickness.

width

The width property specifies the content's width.

height

The height property specifies the content's height.

border-width

The border-width property specifies the border's width.

border-style

The border-style property specifies the border's style.

border-color

The border-color property specifies the border's color.

Figure

4.6.1 Box model with labeled boxes.

 **Animation**

4.6.1 Box model displayed.

 **Question set**

4.6.2 Choose the correct display color.

 **Example**

4.6.1 Example with non-uniform margin and padding thickness.

 **Example**

4.6.2 Example specifying one side of the box.

 **Question set**

4.6.3 Box model properties.

 **Example**

4.6.3 Example using width and height properties.

 **Example**

4.6.4 Example using width and height properties on an inline element.

 **Question set**

4.6.4 Compute the size.

 **Aside**

Chrome DevTools

 **Table**

4.6.1 Common border styles.

 **Learning tool**

4.6.5 Modify the padding, border, and margin.

 **Example**

4.6.5 Collapsed margins.

 **Example**

4.6.6 Horizontal centering.

 **Question set**

4.6.6 Margins.

 **Progression**

4.6.1 Box model.

4.7 Example: Styled band web page

 **Question set**

4.7.1 Page colors.

 **Figure**

4.7.1 CSS text and font properties.

 **Question set**

4.7.2 Font and text styles.

 **Figure**

4.7.2 CSS to float unordered lists next to member images.

 **Aside**

Float alternatives

 **Question set**

4.7.3 Rules that handle floating.

 **Figure**

4.7.3 CSS to add spacing.

 **Question set**

4.7.4 CSS spacing.

4.8 LAB: News article (CSS)

 **Lab activity**

4.8.1 LAB: News article (CSS)

4.9 LAB: Auto loan (CSS)

 **Lab activity**

4.9.1 LAB: Auto loan (CSS)

4.10 LAB: Sleep recommendation list (CSS)

 **Lab activity**

4.10.1 LAB: Sleep recommendation list (CSS)

4.11 LAB: Best-selling video games table (CSS)

 **Lab activity**

4.11.1 LAB: Best-selling video games table (CSS)

4.12 LAB: Style the form with advanced selectors

Lab activity

4.12.1 LAB: Style the form with advanced selectors

5. Wk 2 Prep: Basic JS, Intro to JS in the Browser; Modern Tools: Bootstrap (~132)

5.1 Syntax and variables

ECMAScript

The ECMAScript standard has been improved over the years, and JavaScript is an implementation of ECMAScript.

ECMAScript 2019

The latest version of ECMAScript is version 10 (ECMAScript 2019).

interpreter

An interpreter executes programming statements without first compiling the statements into machine language.

JavaScript engines

Modern JavaScript interpreters (also called JavaScript engines) use just-in-time (JIT) compilation to compile the JavaScript code at execution time into another format that can be executed quickly.

just-in-time (JIT) compilation

Modern JavaScript interpreters (also called JavaScript engines) use just-in-time (JIT) compilation to compile the JavaScript code at execution time into another format that can be executed quickly.

variable

A variable is a named container that stores a value in memory.

variable declaration

A variable declaration is a statement that declares a new variable with the keyword var followed by the variable name.

var

A variable declaration is a statement that declares a new variable with the keyword var followed by the variable name.

assignment

An assignment assigns a variable with a value, like score = 2.

initializing

A variable may also be assigned a value on the same line when the variable is declared, which is called initializing the variable.

identifier

A name created for an item like a variable is called an identifier.

constant

A constant is an initialized variable whose value cannot change.

const

A JavaScript constant is declared with the const keyword and is usually named with an identifier in all capital letters.

Dynamic typing

Dynamic typing determines a variable's type at run-time.

comment

A comment is any text intended for humans that is ignored by the JavaScript interpreter.

prompt()

The prompt() function prompts the user with a dialog box that allows the user to type a single line of text and press OK or Cancel.

console.log()

Output may be produced using the function console.log(), which displays text or numbers in the console.

console

The console is a location where text output is displayed. Web browsers have a console (accessible from the browser's development tools) that displays output from code the browser executes. This chapter's activities display the console output in the web page.

syntax error

A syntax error is the incorrect typing of a programming statement. Ex: Forgetting to place "quotes" around a string value is a syntax error.

Aside

ECMAScript name

Question set

5.1.1 JavaScript background.

Animation

5.1.2 Declaring variables and assigning values.

 **Aside**

let keyword

 **Question set**

5.1.3 Declaring and naming variables.

 **Table**

5.1.1 JavaScript data types.

 **Question set**

5.1.4 Variable data types.

 **Figure**

5.1.1 Comments.

 **Figure**

5.1.2 Using semicolons.

 **Question set**

5.1.5 Detect the error.

 **Animation**

5.1.6 Prompting for input and displaying output.

 **Question set**

5.1.7 prompt() and console.log().

 **Learning tool**

5.1.8 JavaScript practice.

 **Progression**

5.1.1 prompt() and console.log().

5.2 Arithmetic

expression

An expression is a combination of items like variables, numbers, operators, and parentheses, that evaluates to a value like $2 * (x + 1)$.

arithmetic operator

An arithmetic operator is used in an expression to perform an arithmetic computation.

compound assignment operator

A compound assignment operator combines an assignment statement with an arithmetic operation.

String concatenation

String concatenation appends one string after the end of another string, forming a single string.
Ex: "back" + "pack" is "backpack".

operand

An operand is the value or values that an operand works on, like the number 2 or variable x.

parseInt()

The JavaScript functions parseInt() and parseFloat() convert strings into numbers.

parseFloat()

The JavaScript functions parseInt() and parseFloat() convert strings into numbers.

NaN

NaN is a JavaScript value that means Not a Number. Ex: parseInt ("dog") is NaN.

isNaN()

The JavaScript function isNaN() returns true if the argument is not a number, false otherwise.

Table

5.2.1 JavaScript arithmetic operators.

Question set

5.2.1 Arithmetic practice.

Table

5.2.2 Compound assignment operators.

Question set

5.2.2 Practice with compound assignment operators.

Animation

5.2.3 Type conversion in arithmetic operations.

Question set

5.2.4 Arithmetic practice with numbers and strings.

Progression

5.2.1 Arithmetic operators.

5.3 Conditionals

if statement

An if statement executes a group of statements if a condition is true.

if-else statement

An if-else statement executes a block of statements if the statement's condition is true, and executes another block of statements if the condition is false.

comparison operator

A comparison operator compares two operands and evaluates to a Boolean value, meaning either true or false.

identity operator

The identity operator `==` performs strict equality. Two operands are strictly equal if the operands' data types and values are equal.

strict equality

The identity operator `==` performs strict equality. Two operands are strictly equal if the operands' data types and values are equal.

non-identity operator

The non-identity operator `!=` is the opposite of the identity operator.

Unicode

Unicode is a computing industry standard that assigns a unique number to characters in over one hundred different languages, including multiple symbol sets and emoji.

nested

An if or if-else statement that appears inside another if or if-else statement is called a nested statement.

else-if

The else-if statement is an alternative to nested if-else statements that produces an easier-to-read list of statement blocks.

complex condition

Multiple `&&` and `||` conditions may be combined into a single complex condition.

Construct

5.3.1 if statement.

Animation

5.3.1 Evaluating if statements.

Question set

5.3.2 If statement.

Construct

5.3.2 if-else statement.

 **Animation**

5.3.3 Evaluating if-else statements.

 **Question set**

5.3.4 If-else statements.

 **Aside**

Using {} around if and else blocks

 **Table**

5.3.1 Comparison operators.

 **Aside**

What is Unicode?

 **Question set**

5.3.5 Comparison operators.

 **Animation**

5.3.6 Nested if-else statement example.

 **Figure**

5.3.1 Nested if-else statements vs. else-if statements.

 **Learning tool**

5.3.7 Nested if-else practice.

 **Question set**

5.3.8 Nested if and if-else statements.

 **Progression**

5.3.1 Conditionals.

 **Table**

5.3.2 Logical operators.

 **Question set**

5.3.9 Evaluating complex conditions.

5.4 More conditionals

truthy

A truthy value is a non-Boolean value that evaluates to true in a Boolean context. Ex: if (18) evaluates to true because non-zero numbers are truthy values.

falsy

A falsy value is a non-Boolean value that evaluates to false in a Boolean context. Ex: if (null) evaluates to false because null is a falsy value.

conditional operator

The conditional operator (or ternary operator) has three operands separated by a question mark (?) and colon (:). If the condition evaluates to true, then the value of expression1 is returned, otherwise the value of expression2 is returned.

ternary operator

The conditional operator (or ternary operator) has three operands separated by a question mark (?) and colon (:). If the condition evaluates to true, then the value of expression1 is returned, otherwise the value of expression2 is returned.

switch statement

A switch statement compares an expression's value to several cases using strict equality (==) and executes the first matching case's statements. If no case matches, an optional default case's statements execute.

break statement

The break statement stops executing a case's statements and causes the statement immediately following the switch statement to execute.

Table

5.4.1 Truthy values.

Table

5.4.2 Falsy values.

Question set

5.4.1 Truthy and falsy values.

Construct

5.4.1 Conditional (ternary) operator.

Animation

5.4.2 Evaluating the conditional operator.

Question set

5.4.3 Conditional operator.

Construct

5.4.2 switch statement.

 **Animation**

5.4.4 Evaluating the switch statement.

 **Question set**

5.4.5 switch statement.

 **Learning tool**

5.4.6 Practice with the switch statement.

5.5 Loops

while loop

The while loop is a looping structure that checks if the loop's condition is true before executing the loop body, repeating until the condition is false.

loop body

The loop body is the statements that a loop repeatedly executes.

infinite loop

An infinite loop is a loop that never stops executing.

do-while loop

The do-while loop executes the loop body before checking the loop's condition to determine if the loop should execute again, repeating until the condition is false.

for loop

A for loop executes the initialization expression, evaluates the loop's condition, and executes the loop body if the condition is true. After the loop body executes, the final expression is evaluated, and the loop condition is checked to determine if the loop should execute again.

break

The break statement breaks out of a loop prematurely.

continue

The continue statement causes a loop to iterate again without executing the remaining statements in the loop.

 **Construct**

5.5.1 while loop.

 **Animation**

5.5.1 Executing a while loop.

 **Aside**

JavaScript infinite loop with Chrome's "Page Unresponsive" message

 **Question set**

5.5.2 while loop.

 **Construct**

5.5.2 do-while loop.

 **Animation**

5.5.3 Executing a do-while loop.

 **Question set**

5.5.4 do-while loop.

 **Learning tool**

5.5.5 Practice with the while and do-while loops.

 **Construct**

5.5.3 for loop.

 **Animation**

5.5.6 Executing a for loop.

 **Question set**

5.5.7 For loops.

 **Aside**

break and continue statements

 **Progression**

5.5.1 Loops.

5.6 LAB: JavaScript loops

 **Lab activity**

5.6.1 LAB: JavaScript loops

5.7 Functions

function

A function is a named group of statements.

parameter

A parameter is a variable that supplies the function with input.

return

A function may return a single value using the return statement.

function call

Invoking a function's name, known as a function call, causes the function's statements to execute.

argument

An argument is a value provided to a function's parameter during a function call.

function expression

A function expression is identical to a function declaration, except the function name is omitted.

Hoisting

Hoisting is JavaScript's behavior of moving variable declarations to the top of the current scope.

anonymous function

An anonymous function is a function that does not have a name.

self-invoking function

A self-invoking function is an anonymous function that invokes (calls) itself.

arrow function

An arrow function is an anonymous function that uses an arrow \Rightarrow to create a compact function.

let

In addition to declaring variables with `var`, a variable can be declared with the `let` keyword.

scope

A JavaScript variable's scope is the context in which the variable can be accessed.

local scope

A variable declared inside a function has local scope, so only the function that defines the variable has access to the local variable.

local variable

A variable declared inside a function has local scope, so only the function that defines the variable has access to the local variable.

global scope

A variable declared outside a function has global scope, and all functions have access to a global variable.

global variable

A variable declared outside a function has global scope, and all functions have access to a global variable.

block scope

A variable declared inside a function with let has block scope: the variable is accessible only within the enclosing pair of braces.

the global object

Before developer code is run, JavaScript implementations create the global object: an object that stores certain global variables, functions, and other properties.

Construct

5.7.1 Function declaration.

Animation

5.7.1 Declaring and calling functions.

Question set

5.7.2 Functions and return values.

Learning tool

5.7.3 Function practice.

Figure

5.7.1 Assigning a function expression to a variable.

Animation

5.7.4 Hoisting variables and functions.

Figure

5.7.2 A self-invoking function.

Question set

5.7.5 Function expressions.

Animation

5.7.6 Arrow functions that sum two numbers and square a number.

Question set

5.7.7 Arrow functions.

Progression

5.7.1 Functions.

Animation

5.7.8 var vs. let scoping.

Question set

5.7.9 Local and global variables.

Question set

5.7.10 var vs. let scoping.

Figure

5.7.3 Example with accidental global variable.

Question set

5.7.11 Variable scope and functions.

5.8 LAB: JavaScript password strength

Lab activity

5.8.1 LAB: JavaScript password strength

5.9 Arrays

array

An array is an ordered collection of values called elements.

elements

An array is an ordered collection of values called elements.

index

Each array element is stored in a numeric location called an index.

Array object

An array is an Array object that defines numerous methods for manipulating arrays.

method

A method is a function that is attached to an object and operates on data stored in the object.

Methods are called by prefacing the method with the object. Ex: `myArray.method()` .

length

The array property length contains the number of elements in the array.

for-of

The for-of statement is a simplified for loop that loops through an entire array.

forEach()

The Array method forEach() also loops through an array.

indexOf()

The array methods `indexOf()` and `lastIndexOf()` search an array and return the index of the first found value or -1 if the value is not found.

lastIndexOf()

The array methods `indexOf()` and `lastIndexOf()` search an array and return the index of the first found value or -1 if the value is not found.

sort()

The array method `sort()` sorts an array in ascending (increasing) order.

Figure

5.9.1 Array called `points` is initialized with 4 numbers.

Animation

5.9.1 Add, modify, and display array elements.

Question set

5.9.2 Adding and displaying array elements.

Table

5.9.1 Array methods for adding and removing array elements.

Learning tool

5.9.3 Adding and removing array elements.

Figure

5.9.2 Looping through an array with a for loop.

Figure

5.9.3 Looping through an array with a for-of loop.

Figure

5.9.4 Looping through an array with the `forEach()` method.

Question set

5.9.4 Looping through an array.

Learning tool

5.9.5 Practice looping.

Figure

5.9.5 Searching for array elements.

Question set

5.9.6 Searching an array.

Learning tool

5.9.7 Practice searching an array.

Figure

5.9.6 Sorting an array of numbers.

Question set

5.9.8 Sorting arrays.

Progression

5.9.1 Arrays.

5.10 LAB: JavaScript arrays

Lab activity

5.10.1 LAB: JavaScript arrays

5.11 Objects

object

An object is an unordered collection of properties.

property

An object property is a name-value pair, where the name is a string and the value is any data type.

object literal

An object literal (also called an object initializer) is a comma-separated list of property name and value pairs.

object initializer

An object literal (also called an object initializer) is a comma-separated list of property name and value pairs.

getter

A getter is a function that is called when an object's property is retrieved. Syntax to define a getter: `get property() { return someValue; }`.

setter

A setter is a function that is called when an object's property is set to a value. Syntax to define a setter: `set property(value) { ... }`.

accessor property

An accessor property is an object property that has a getter or a setter or both.

associative array

An associative array or map is a data structure that maps keys to values.

map

An associative array or map is a data structure that maps keys to values.

for-in loop

The for-in loop iterates over an object's properties in arbitrary order.

Object.keys()

The Object.keys() method returns an array of an object's property names and is often used to determine the number of elements in an associative array.

delete operator

The delete operator removes a key/property from an associative array or object.

in operator

The in operator returns true if an object contains the given property and returns false otherwise.

Map object

The Map object is an alternative to the associative array for storing key/value pairs.

Animation

5.11.1 Creating an object with an object literal.

Question set

5.11.2 Accessing object properties.

Figure

5.11.1 Defining a method in an object literal.

Figure

5.11.2 Defining a method for an existing object.

Question set

5.11.3 Object methods.

Learning tool

5.11.4 Practice creating objects and methods.

Figure

5.11.3 Defining an accessor property called 'area'.

Question set

5.11.5 Accessor properties.

Learning tool

5.11.6 Practice creating accessor properties.

Figure

5.11.4 Defining an associative array of state capitals.

Construct

5.11.1 for-in loop.

Figure

5.11.5 Looping through an associative array with for-in.

Figure

5.11.6 Using Object.keys() to determine an associative array's size.

Question set

5.11.7 Associative arrays.

Learning tool

5.11.8 Practice with associative arrays.

Figure

5.11.7 Using the "delete" and "in" operators.

Question set

5.11.9 in and delete operators.

Aside

Map object

Progression

5.11.1 Objects.

5.12 LAB: JavaScript associative arrays

Lab activity

5.12.1 LAB: JavaScript associative arrays

5.13 Exception handling

exception

An exception is an error that disrupts the normal flow of program execution.

Exception handling

Exception handling is the process of catching and responding to an exception.

throw

The throw statement throws a user-defined exception. Syntax: `throw expression`.

try-catch

A try-catch statement has a `try` block to execute code that may throw an exception and a `catch` block that executes when an exception is thrown.

polyfill

A polyfill is code that provides functionality a web browser is currently lacking but may support in the future.

finally

A `finally` block may follow a `try` or `catch` block. The `finally` block executes regardless of whether an exception was thrown or not.

Error object

The `Error` object represents a runtime error, which is an error that occurs when the program is executing.

RangeError

`RangeError` - Thrown when a numeric variable or parameter is outside the valid range.

InternalError

`InternalError` - Thrown when an internal error occurs in the JavaScript interpreter.

TypeError

`TypeError` - Thrown when a variable or parameter is not the expected data type.

Figure

5.13.1 An exception is generated when calling a non-existing method.

Construct

5.13.1 try-catch.

Animation

5.13.1 Throwing and catching an exception.

Question set

5.13.2 Exception handling.

Learning tool

5.13.3 Practice throwing and handling an exception.

 **Construct**

5.13.2 try-catch-finally.

 **Animation**

5.13.4 try-catch-finally execution.

 **Figure**

5.13.2 finally without a catch.

 **Figure**

5.13.3 finally block executes when the catch block throws an exception.

 **Question set**

5.13.5 try-catch-finally.

 **Figure**

5.13.4 findAverage() throws an Error, TypeError, and RangeError.

 **Question set**

5.13.6 Error object.

 **Learning tool**

5.13.7 Practice throwing an Error object.

 **Progression**

5.13.1 Throwing and catching exceptions.

5.14 Document Object Model (DOM)

node

A node is an individual object in the DOM tree.

document.documentElement

The JavaScript object `document.documentElement` is the root of the DOM tree (the "top" node).

childNodes

The `childNodes` property is an array-like collection of objects for each of the node's children.

parentNode

Each node has a property `parentNode` that refers to the node's parent element.

children

The DOM provides the `children` property for a node, which is similar to the `childNodes` property except that the `children` property only contains element nodes and no textual content nodes.

document.getElementById()

The `document.getElementById()` method returns the DOM node whose id attribute is the same as the method's parameter.

document.getElementsByTagName()

The `document.getElementsByTagName()` method returns an array containing all the DOM nodes whose type is the same as the method's parameter.

document.getElementsByClassName()

The `document.getElementsByClassName()` method returns an array containing all the DOM nodes whose class attribute matches the method's parameter.

document.querySelectorAll()

The `document.querySelectorAll()` method returns an array containing all the DOM nodes that match the CSS selector passed as the method's parameter.

document.querySelector()

The `document.querySelector()` method returns the first element found in the DOM that matches the CSS selector passed as the method's parameter.

nextSibling

`NextSibling` is a node property that refers to the node with the same parent following the current node in the document.

prevSibling

`PrevSibling` is a node property that refers to the node with the same parent preceding the current node in the document.

removeAttribute()

Each DOM node provides a method named `removeAttribute()`, which takes a name of an attribute as a parameter. The `removeAttribute()` method removes the corresponding attribute from the node.

nodeValue

The `nodeValue` property sets or gets the value of text nodes.

innerHTML

The `innerHTML` property sets or gets a DOM node's content, including all of the node's children, as a string instead of as a tree.

appendChild()

The `appendChild()` method appends a DOM node to the child nodes of the method's caller.

insertBefore()

The insertBefore() method inserts a DOM node as a child node before an existing child node of the method's caller.

removeChild()

The removeChild() method removes a node from the method's caller's children.

createElement()

The createElement() method creates a DOM node from a string parameter for an HTML element.

createTextNode()

The createTextNode() method creates a DOM node containing the text specified by a string argument.

cloneNode()

The cloneNode() method creates a DOM node or tree identical to the tree rooted at the method's caller. The method's boolean argument indicates whether the method should clone the node's children.

Animation

5.14.1 Creating the DOM from HTML.

Figure

5.14.1 Visualizing the DOM tree.

Figure

5.14.2 Complete DOM tree visualization with whitespace text nodes.

Question set

5.14.2 DOM structure.

Question set

5.14.3 DOM children and parents.

Aside

Viewing the DOM in Chrome

Figure

5.14.3 Example HTML for searching the DOM.

Aside

Note

Question set

5.14.4 DOM traversal.

Figure

5.14.4 Example HTML for sibling methods.

Question set

5.14.5 Using nextSibling, prevSibling, and parentNode.

Question set

5.14.6 Reading and modifying DOM node attribute values.

Animation

5.14.7 firstChild.nodeValue vs. innerHTML.

Learning tool

5.14.8 Modify the DOM nodes.

Question set

5.14.9 Adding and removing DOM nodes.

Question set

5.14.10 Creating new DOM nodes.

Progression

5.14.1 Using the Document Object Model.

5.15 Using JavaScript with HTML

Document Object Model

The Document Object Model (or DOM) is a data structure that represents all parts of an HTML document.

DOM

The Document Object Model (or DOM) is a data structure that represents all parts of an HTML document.

document

The JavaScript object document represents the entire DOM and is created from the document's HTML.

document.writeln()

The document.writeln() method, which outputs HTML into the document and alters the DOM.

window

JavaScript running in a web browser has access to the window object, which represents an open browser window.

window.location

Window.location is a location object that contains information about the window's current URL.

window.navigator

Window.navigator is a navigator object that contains information about the browser.

window.innerHeight

Window.innerHeight and window.innerWidth are the height and width in pixels of the window's content area.

window.innerWidth

Window.innerHeight and window.innerWidth are the height and width in pixels of the window's content area.

window.alert()

Window.alert() displays an alert dialog box.

window.confirm()

Window.confirm() displays a confirmation dialog box with OK and Cancel buttons. confirm() returns true if OK is pressed and false if Cancel is pressed.

window.open()

Window.open() opens a new browser window.

console

Modern browsers provide a console that allows the JavaScript code to produce informational and debugging output for the web developer, which does not affect the functionality or presentation of the web page.

API

An API (Application Programming Interface) is a specification of the methods and objects that defines how a programmer should interact with software components.

Application Programming Interface

An API (Application Programming Interface) is a specification of the methods and objects that defines how a programmer should interact with software components.

console.log()

Console.log() displays informational data to the console.

console.warn()

Console.warn() displays warnings to the console.

console.error()

Console.error() displays errors to the console.

console.dir()

Console.dir() displays a JavaScript object to the console.

async

The async attribute allows the browser to process the web page concurrently with loading and processing the JavaScript.

defer

The defer attribute allows the browser to load the web page concurrently with loading the JavaScript, but the JavaScript is not processed until the web page is completely loaded.

Minification

Minification or minimization is the process of removing unnecessary characters (like whitespace and comments) from JavaScript code so the code executes the same but with fewer characters.

minimization

Minification or minimization is the process of removing unnecessary characters (like whitespace and comments) from JavaScript code so the code executes the same but with fewer characters.

obfuscator

A JavaScript obfuscator is software that converts JavaScript into an unreadable form that is very difficult to convert back into readable JavaScript.

Learning tool

5.15.1 Writing JavaScript within the body of an HTML file.

Question set

5.15.2 JavaScript Basics.

Learning tool

5.15.3 Using the window object.

Question set

5.15.4 Window object.

Figure

5.15.1 Chrome console showing a syntax error on line 9 of test.html.

Figure

5.15.2 console.log() output example.

Question set

5.15.5 Match terms with definitions.

 **Example**

5.15.1 Loading JavaScript from an external file.

 **Animation**

5.15.6 Loading JavaScript from an HTML file.

 **Question set**

5.15.7 Downloading JavaScript files.

 **Example**

5.15.2 Loading JavaScript with the `async` attribute.

 **Animation**

5.15.8 Using the `async` attribute with the `<script>` tag.

 **Example**

5.15.3 Loading JavaScript with the `defer` attribute.

 **Animation**

5.15.9 Using the `defer` attribute with the `<script>` tag.

 **Question set**

5.15.10 Loading JavaScript.

 **Aside**

Minification and obfuscation

 **Progression**

5.15.1 JavaScript with HTML.

5.16 Modifying CSS with JavaScript

CSS Object Model

The CSS Object Model (CSSOM) is a set of APIs that allow JavaScript to manipulate CSS properties of a web page.

CSSOM

The CSS Object Model (CSSOM) is a set of APIs that allow JavaScript to manipulate CSS properties of a web page.

style

Every element in the DOM has a `style` property that holds the inline styles set on the element.

CSSStyleDeclaration

The `style` object implements the CSSOM interface `CSSStyleDeclaration`, which provides methods for accessing, modifying, and removing CSS properties.

getPropertyValue()

The `getPropertyValue()` method returns the value of an element's CSS property.

setProperty()

The `setProperty()` method sets the value of an element's CSS property.

removeProperty()

The `removeProperty()` method removes an element's CSS property.

document.styleSheets

The `document.styleSheets` object is a list of the stylesheets used in the web page.

CSSStyleSheet

Each stylesheet in `document.styleSheets` is a `CSSStyleSheet` object, which maintains a list of the stylesheet's CSS rules in the property called `cssRules`.

cssRules

Each stylesheet in `document.styleSheets` is a `CSSStyleSheet` object, which maintains a list of the stylesheet's CSS rules in the property called `cssRules`.

insertRule()

The `insertRule()` method inserts a new rule into the stylesheet.

deleteRule()

The `deleteRule()` method deletes a rule at a given index number from the stylesheet.

classList

Every DOM node has a `classList` property that lists the classes assigned to the node.

add()

The `add()` method adds a class to the node's `classList`.

remove()

The `remove()` method removes a class from the node's `classList`.

toggle()

The `toggle()` method adds the class to the node's `classList` if the class is not present. If the class is already present, the class is removed.

className

A DOM node's class list can also be modified directly using the `className` property, which is a space-delimited list of the classes assigned to the node.

Animation

5.16.1 Modifying inline CSS style.

Question set

5.16.2 Modifying inline style.

Aside

Security issue

Learning tool

5.16.3 Insert, modify, and delete CSS rules.

Question set

5.16.4 Modifying stylesheet rules.

Aside

className property

Learning tool

5.16.5 Add and remove classes.

Question set

5.16.6 Adding and removing classes.

5.17 Bootstrap

Bootstrap

Bootstrap is one of the most popular frameworks for creating responsive websites because Bootstrap simplifies the task of creating fluid layouts and provides many common website components.

Reboot

Bootstrap provides consistent styling for all browsers using CSS selector rules called Reboot.

grid system

Bootstrap uses a responsive grid system that allows developers to create layouts for a variety of screen sizes.

components

Bootstrap provides many components for developing interactive user interfaces.

Aside

History of Bootstrap

Figure

5.17.1 Accessing Bootstrap from a CDN.

Learning tool

5.17.1 Practice with Bootstrap.

Question set

5.17.2 Bootstrap library.

Table

5.17.1 Bootstrap breakpoints.

Figure

5.17.2 Viewport meta tag required by Bootstrap grid system.

Animation

5.17.3 Bootstrap grid system.

Learning tool

5.17.4 Practice with Bootstrap's grid system.

Question set

5.17.5 Bootstrap grid system.

Figure

5.17.3 Bootstrap image shapes.

Learning tool

5.17.6 Fluid image.

Question set

5.17.7 Bootstrap images.

Learning tool

5.17.8 Practice with Modal and Alert.

Question set

5.17.9 Bootstrap Modals and Alerts.

5.18 LAB: Bootstrap for a vacation website

Lab activity

5.18.1 LAB: Bootstrap for a vacation website

6. Wk 3 Prep: Advanced CSS (~88)

6.1 Page layout with flexbox

fixed layout

Some web pages use a fixed layout, which uses a fixed-width container to envelop the web page contents. Ex: The figure below shows a web page at the top using a fixed layout where the entire contents fit inside 960px. Resizing the browser does not change the width of the web page contents.

fluid layout

A web page that uses a fluid layout allows the page contents to fill the browser, sometimes by using percentages for widths. Ex: The figure below shows a web page at the bottom using a fluid layout where the contents always fit the browser's width. Fluid layouts make better use of the available space than fixed layouts and do not produce a horizontal scroll bar when the browser is resized.

wireframe

The web page below displays a wireframe, a blueprint for a web page that shows how future content will be arranged.

Flexible Box

The Flexible Box or flexbox is a CSS layout mode that provides an efficient way to lay out elements in a container so the elements behave predictably when the container is resized or viewed on different screen sizes.

flexbox

The Flexible Box or flexbox is a CSS layout mode that provides an efficient way to lay out elements in a container so the elements behave predictably when the container is resized or viewed on different screen sizes.

flex container

A flex container is an element that has the CSS property `display` set to `flex` to create a block-level flex container or `inline-flex` to create an inline flex container. Ex:

`<div style="display: flex">`. Flex containers hold flex items.

flex item

A flex item is an element inside a flex container that is positioned and sized according to various CSS flexbox properties.

Figure

6.1.1 Typical web page layout.

Figure

6.1.2 Fixed vs. fluid layout.

Learning tool

6.1.1 Create a fluid layout.

 **Question set**

6.1.2 Fluid layout.

 **Figure**

6.1.3 Flexbox container and items.

 **Question set**

6.1.3 Flexbox properties.

 **Animation**

6.1.4 Flexbox layout.

 **Question set**

6.1.5 Flexbox layout example.

 **Learning tool**

6.1.6 Practice with flexbox layout.

 **Progression**

6.1.1 Flexbox.

6.2 Page layout with grid layout

Grid layout

Grid layout is a CSS layout mode that divides a web page into a rectangular grid in which to position page elements.

grid container

A grid container is an element that has the CSS property display set to grid to create a block-level grid container or grid-in-line to create an inline grid container.

grid item

A grid item is a child element of a grid container that is by default placed into a single grid cell.

grid-template-columns

The grid-template-columns property defines the grid container's number of columns and optionally the width of each column.

grid-gap

The grid-gap property defines the gap between each grid column and row.

grid-template-rows

The grid-template-rows property defines the height of each row.

justify-content

The justify-content property horizontally aligns the grid items inside the grid container.

align-content

The align-content property vertically aligns the grid items inside the grid container.

grid-row

The grid-row property lists the grid item's starting and ending row line numbers.

grid-column

The grid-column property lists the grid item's starting and ending column line numbers.

grid-area

The grid-area property lists the grid item's starting and ending row and column numbers.

grid-template-areas

The grid container's grid-template-areas property specifies the grid layout using the named grid items.

Animation

6.2.1 Grid layout example.

Question set

6.2.2 Grid layout basics.

Question set

6.2.3 Modify the grid container.

Figure

6.2.1 Row and column lines.

Learning tool

6.2.4 Rearrange the grid.

Question set

6.2.5 Modify the grid layout.

Figure

6.2.2 Layout goal for the Participation Activity below.

Learning tool

6.2.6 Grid layout using named grid items.

Question set

6.2.7 Named grid items.

6.3 Positioning elements

position

The CSS position property gives developers more control over where elements should appear in the browser.

Static positioning

Static positioning is the default positioning.

Relative positioning

Relative positioning positions the element relative to the element's default position.

Fixed positioning

Fixed positioning positions the element relative to the viewport in a fixed location.

Absolute positioning

Absolute positioning positions the element relative to the nearest positioned ancestor.

viewport

A viewport is the visible area of a web page.

z-index

However, the CSS z-index property is used to specify a relative distance that orders the appearance of elements.

Animation

6.3.1 Relative positioning.

Question set

6.3.2 Relative and static positioning.

Animation

6.3.3 Fixed positioning.

Question set

6.3.4 Fixed and absolute positioning.

Figure

6.3.1 Cheer is absolute positioned inside a positioned ancestor (left) and relative to the document body (right).

Question set

6.3.5 Absolute positioning.

Figure

6.3.2 No z-index is used on the left, but z-index changes the rendered order on the right.

Question set

6.3.6 z-index.

Learning tool

6.3.7 Positioning practice.

Progression

6.3.1 Positioning elements.

6.4 Special effects

text-shadow

Shadows are added to text using the CSS property `text-shadow`.

box-shadow

The CSS property `box-shadow` adds a shadow to the box around an element.

border-radius

An element border's corners can be rounded using the CSS property `border-radius`.

border-image

The CSS property `border-image` renders an element's border using sections of an image.

vendor prefix

A vendor prefix is a prefix added to an experimental or nonstandard CSS property that only works on a specific browser type.

linear-gradient(color1, color2)

The CSS function `linear-gradient(color1, color2)` creates a linear gradient that transitions from `color1` to `color2` when moving from the top edge to the bottom edge.

repeating-linear-gradient()

The `repeating-linear-gradient()` function repeats a linear gradient where the color values are supplied an optional percent.

radial-gradient(color1, color2)

A radial gradient is created with the CSS function `radial-gradient(color1, color2)`, which creates an ellipse-shaped gradient that begins with `color1` in the center and ends with `color2` on the perimeter.

Figure

6.4.1 Examples of different text-shadow values.

Question set

6.4.1 Text shadows.

Figure

6.4.2 Examples of different box-shadow values.

 **Question set**

6.4.2 Box shadows.

 **Learning tool**

6.4.3 Shadow practice.

 **Question set**

6.4.4 Rounded corners.

 **Learning tool**

6.4.5 Try different border-image values.

 **Question set**

6.4.6 Border images.

 **Aside**

CSS3 browser support

 **Figure**

6.4.3 Examples of linear gradients.

 **Question set**

6.4.7 Linear gradients.

 **Figure**

6.4.4 Examples of radial gradients.

 **Question set**

6.4.8 Radial gradient.

 **Question set**

6.4.9 Positioned radial gradients.

 **Learning tool**

6.4.10 Gradient practice.

 **Progression**

6.4.1 Special effects.

6.5 Animation

CSS animation

A CSS animation transforms an element's styles over a set time period, producing an animation.

@keyframes

The @keyframes rule defines a keyframe list.

keyframe list

A keyframe list has a name and contains the keyframes or the properties and values that will be animated.

from

From - The animation starting state that lists the CSS properties and values that apply when the animation begins.

to

To - The animation ending state that lists the CSS properties and values that the "from" values become by the time the animation ends.

animation-name

Animation-name - Names the keyframe list associated with the animation.

animation-duration

Animation-duration - Length of the animation.

animation-delay

An animation begins immediately when the browser renders the web page unless an animation-delay is used to delay the start of the animation.

animation-timing-function

The animation-timing-function property controls an animation's speed between keyframes.

animation-iteration-count

Animation-iteration-count - Indicates the number of times the animation will run.

animation-direction

Animation-direction - Indicates animation direction.

animation

Animation - Shorthand property indicating the animation name, duration, timing function, delay, iteration count, and direction.

CSS transition

A CSS transition animates an element's transition from one state to another when an element's CSS property changes value.

transition

Transitions are defined with the transition property.

transition-timing-function

The transition-timing-function property controls the speed of the transition.

transition-delay

The transition-delay property delays the transition's start.

transform

The transform property applies a 2D or 3D transformation to an element.

transformation

A transformation is a graphical operation that alters the position, shape, or orientation of an object.

Animation

6.5.1 Animating the background color.

Question set

6.5.2 Keyframes and animation.

Learning tool

6.5.3 Percentages for keyframes.

Aside

CSS3 animation vs. JavaScript animation

Question set

6.5.4 Keyframes and timing functions.

Animation

6.5.5 Transitioning the width and height when hovering.

Question set

6.5.6 Transitions.

Table

6.5.1 Selected 2D transformation functions.

Animation

6.5.7 translate(), scale(), and rotate() transformation functions.

Learning tool

6.5.8 Animations, transitions, and transformations practice.

 **Question set**

6.5.9 Transformations in transitions and animations.

6.6 Styling forms

 **Figure**

6.6.1 HTML form without CSS styling and an improved form with styling.

 **Learning tool**

6.6.1 Create a styled form.

 **Question set**

6.6.2 Form styles.

 **Aside**

Paragraphs vs. divs

 **Aside**

Proper use of field labels

 **Learning tool**

6.6.3 Augmenting an input.

 **Question set**

6.6.4 Augmented inputs.

 **Animation**

6.6.5 Styling radio buttons.

 **Learning tool**

6.6.6 Style the checkboxes.

 **Question set**

6.6.7 Styling radio buttons and checkboxes.

 **Progression**

6.6.1 Styling forms.

6.7 Sass

Sass

Sass is a popular CSS preprocessor that uses CSS-like syntax to build complex CSS stylesheets.

Sassy CSS

Sass version 3 introduced a new syntax called Sassy CSS (SCSS), which uses semicolons and brackets like CSS.

SCSS

Sass version 3 introduced a new syntax called Sassy CSS (SCSS), which uses semicolons and brackets like CSS.

SassScript

SassScript is a set of extensions to CSS that allow properties to use variables, arithmetic, and functions. SassScript also provides basic control directives for performing conditional logic and looping.

mixin

A mixin is set of reusable styles and is defined by the @mixin directive.

@mixin

A mixin is set of reusable styles and is defined by the @mixin directive.

directive

A directive is an extension to the CSS at-rules, which are statements that begin with the @ character.

@include

Mixins are included in a document using the @include directive.

Animation

6.7.1 Compiling SCSS into CSS.

Question set

6.7.2 Sass CSS preprocessor.

Figure

6.7.1 Selector nesting.

Figure

6.7.2 Referencing the parent with &;

Figure

6.7.3 Property nesting.

Question set

6.7.3 Nested selectors and properties.

Animation

6.7.4 Performing SassScript arithmetic.

Question set

6.7.5 Variables and arithmetic.

Table

6.7.1 Some SassScript functions.

Question set

6.7.6 SassScript functions.

Animation

6.7.7 Including mixins.

Question set

6.7.8 Mixins.

Aside

Control directives and expressions

6.8 Media queries

Graceful degradation

Graceful degradation: Design the desktop website first and modify the design to fit smaller screens.

Progressive enhancement

Progressive enhancement: A "mobile first" design methodology that begins with designing the website for the smallest device and then adapts the design for larger screens.

media query

A CSS media query is a combination of media type and optionally one or more expressions that evaluate to true or false based on various media features like width, height, and orientation.

breakpoint

A breakpoint is the screen width that activates a media query.

Figure

6.8.1 Designing for three platforms.

Question set

6.8.1 Targeting different screens.

Figure

6.8.2 A fluid design for desktop should be altered for mobile.

Table

6.8.1 Media types used in media queries.

Table

6.8.2 Common media features used in media queries.

Animation

6.8.2 Evaluating media queries.

Question set

6.8.3 Media queries.

Figure

6.8.3 Mobile, tablet, and desktop wireframes for the Participation Activity below.

Learning tool

6.8.4 Create a responsive layout.

Question set

6.8.5 Breakpoints.

6.9 Example: Styled Restaurant Reviews

Question set

6.9.1 Home page.

Figure

6.9.1 HTML and CSS for the advertisement.

Figure

6.9.2 CSS for expanding advertisement.

Question set

6.9.2 Advertisement.

Figure

6.9.3 CSS for styling the reviews and overall rating stars.

Figure

6.9.4 Orange grid lines show the grid layout for each review.

Figure

6.9.5 CSS for the user rating grid layout.

Question set

6.9.3 Detail page.

Question set

6.9.4 Styling the form.

6.10 LAB: CSS practice

Lab activity

6.10.1 LAB: CSS practice

6.11 LAB: Tic-tac-toe game with grid layout

Lab activity

6.11.1 LAB: Tic-tac-toe game with grid layout

6.12 LAB: Position the playing cards

Lab activity

6.12.1 LAB: Position the playing cards

6.13 LAB: Animating the answer

Lab activity

6.13.1 LAB: Animating the answer

6.14 LAB: Recipe with flexbox

Lab activity

6.14.1 LAB: Recipe with flexbox

7. Wk 3 Prep: JavaScript Classes and Objects (~55)

7.1 Classes

class

A JavaScript class is a special function, called a constructor function, that defines properties and methods from which an object may inherit.

constructor function

A constructor function is a function that initializes a new object when an object is instantiated with the `new` operator.

this

The `this` keyword refers to the current object and is used to access properties inside the class.

prototype

The prototype object contains properties that an associated object inherits when the associated object is created.

private property

A private property is a property that is only accessible to object methods but is not accessible from outside the class.

closure

A closure is a special object that is automatically created and maintains a function's local variables and values after the function has returned.

Inheritance

Inheritance creates a new child class that adopts properties of a parent class. Ex: A Student class (child) may inherit from a Person class (parent), so a Student class has the same properties of a Person and may add even more properties.

Animation

7.1.1 Creating a Person class with a constructor function.

Question set

7.1.2 JavaScript classes.

Figure

7.1.1 Assigning methods to the prototype object is more memory efficient.

Question set

7.1.3 The prototype object.

Learning tool

7.1.4 Practice with classes.

Figure

7.1.2 Creating a private property called "secret" with a getter and setter.

Question set

7.1.5 Private properties and closures.

Animation

7.1.6 Student class inherits properties from the Person class.

Question set

7.1.7 Inheritance.

Learning tool

7.1.8 Practice with inheritance and private properties.

Progression

7.1.1 Classes.

 **Aside**

Classes and inheritance in EcmaScript 6

7.2 Classes (ES6)

class keyword

A class is declared by using the class keyword followed by a class name.

constructor()

The method name constructor() is reserved for the class constructor.

extends keyword

The extends keyword allows one class to inherit from another.

super()

In the inheriting class' constructor, calling the super() function calls the parent class' constructor.

get keyword

A class method declaration preceded by the get keyword defines a getter method for a property.

set keyword

A class method declaration preceded by the set keyword defines a setter method for a property.

static method

A static method is a method that can be called without creating an instance of the class.

static

A static method is declared with the static keyword preceding the method name.

 **Animation**

7.2.1 CityState ES6 class.

 **Question set**

7.2.2 Classes.

 **Figure**

7.2.1 Person class and inheriting Student class.

 **Question set**

7.2.3 Inheritance.

 **Animation**

7.2.4 Getters and setters can be used to convert an angle between degrees and radians.

Question set

7.2.5 Getter and setter methods.

Animation

7.2.6 StringOps class with static methods.

Question set

7.2.7 StringOps class and static method concepts.

7.3 JavaScript Object Notation (JSON)

JavaScript Object Notation

JavaScript Object Notation, or JSON, is an efficient, structured format for data based on a subset of the JavaScript language.

JSON

JavaScript Object Notation, or JSON, is an efficient, structured format for data based on a subset of the JavaScript language.

JSON object

A JSON object is an unordered list of zero or more name/value pairs separated by commas and enclosed within braces ({}).

JSON array

A JSON array is an ordered list of zero or more JSON values separated by commas and enclosed within brackets ([]).

JSON value

A JSON value can be one of seven possibilities: a JSON object, a JSON array, a string, a number, a boolean (true, false), or null.

JSON.parse()

The JSON.parse() method creates a JavaScript object from a string containing JSON.

JSON.stringify()

The JSON.stringify() method creates a string from a JavaScript object.

reviver function

A reviver function is used to modify parsed values before being returned, and is helpful when a JSON string represents a data type not available in JSON.

Question set

7.3.1 JSON basics.

Figure

7.3.1 An example JSON data structure.

 **Question set**

7.3.2 JSON data types.

 **Animation**

7.3.3 JSON.parse and JSON.stringify example.

 **Question set**

7.3.4 Using JSON.parse and JSON.stringify.

 **Animation**

7.3.5 Reviver function for JSON.parse.

 **Question set**

7.3.6 Customizing JSON.parse and JSON.stringify.

 **Progression**

7.3.1 JavaScript and JSON.

7.4 String object

String

The String object defines methods to manipulate strings, extract substrings, test for string inclusion, etc.

charAt()

The String method charAt() returns the character at the specified index as a string.

length

The String property length returns the number of characters in a string.

indexOf()

The indexOf() method returns the index of the search string's first occurrence inside the String object or -1 if the search string is not found.

lastIndexOf()

The lastIndexOf() method returns the index of the search string's last occurrence inside the String object or -1 if the search string is not found.

replace()

The replace() method replaces one string with another and returns the string with the replacement string inside.

template literal

A template literal is a string literal enclosed by the back-tick (`) that allows embedding expressions with a dollar sign and braces (\${expression}).

 **Animation**

7.4.1 Counting spaces in a string.

 **Question set**

7.4.2 String object.

 **Figure**

7.4.1 Searching for a string with `indexOf()` and `lastIndexOf()`.

 **Figure**

7.4.2 Replacing a string with `replace()`.

 **Question set**

7.4.3 Search and replace.

 **Table**

7.4.1 Common String methods.

 **Question set**

7.4.4 String methods.

 **Learning tool**

7.4.5 Practice with String methods.

 **Animation**

7.4.6 Template literal simplifies syntax.

 **Question set**

7.4.7 Template literal.

 **Progression**

7.4.1 Strings.

7.5 Date object

Date

A Date object represents a single moment in time, based on the number of milliseconds since the Unix Epoch (January 1, 1970 UTC).

constructor

A constructor is a function that creates an instance of an object.

 **Animation**

7.5.1 Date object constructor.

 **Question set**

7.5.2 Date object constructor.

 **Table**

7.5.1 Date object getter and setter methods.

 **Learning tool**

7.5.3 Practice with the Date object.

 **Progression**

7.5.1 Using Date methods.

7.6 Math object

Math

The Math object provides properties for mathematical constants and methods to perform mathematical functions.

Math.random()

The Math.random() method returns a pseudo-random number between 0 and 1.

pseudo-random number

A pseudo-random number is a number generated by an algorithm that approximates randomness, but is not truly random.

 **Question set**

7.6.1 Math properties.

 **Table**

7.6.1 Common Math object methods.

 **Question set**

7.6.2 Math methods.

 **Figure**

7.6.1 Display 5 random numbers with Math.random().

 **Question set**

7.6.3 Random numbers.

 **Figure**

7.6.2 Display 5 random numbers between 1 and 10.

 **Learning tool**

7.6.4 Practice with random numbers.

7.7 Event-driven programming

event

An event is an action, usually caused by a user, that the web browser responds to.

Event-driven programming

Event-driven programming is a programming style where code runs only in response to various events.

event handler

Code that runs in response to an event is called an event handler or event listener.

event listener

Code that runs in response to an event is called an event handler or event listener.

change

A change event is caused by an element value being modified.

input

An input event is caused when the value of an input or textarea element is changed.

load

A load event is caused when the browser completes loading a resource and dependent resources. Usually load is used with the body element to execute code once all the web page's CSS, JavaScript, images, etc. have finished loading.

DOMContentLoaded

A DOMContentLoaded event is caused when the HTML file has been loaded and parsed, although other related resources such as CSS, JavaScript, and image files may not yet be loaded.

focus

A focus event is caused when an element becomes the current receiver of keyboard input.

blur

A blur event is caused when an element loses focus and the element will no longer receive future keyboard input.

submit

A submit event is caused when the user submits a form to the web server.

addEventListener()

Using the JavaScript addEventListener() method to register an event handler for a DOM object.

event object

Every handler has an optional event object parameter that provides details of the event.

event capturing

In the event capturing phase, the browser traverses the DOM tree from the root to the event target node, at each node calling any event-specific handlers that were explicitly registered for activation during the capturing phase.

at target

In the at target phase, the browser calls all event-specific handlers registered on the target node.

event bubbling

In the event bubbling phase, the browser traverses the DOM tree from the event target node back to the root node, at each node calling all event-specific handlers registered for the bubbling phase on the current node.

stopPropagation()

The event capturing and bubbling process can be stopped by calling the stopPropagation() method on the event object provided to the handler.

preventDefault()

The event object's preventDefault() method stops the web browser from performing the built-in handler.

Animation

7.7.1 Focus and blur event handling.

Question set

7.7.2 Event-driven programming.

Question set

7.7.3 Mouse and keyboard events.

Question set

7.7.4 Other common browser events.

Animation

7.7.5 Registering event handlers with addEventListener().

Learning tool

7.7.6 Registering event handler using addEventListener.

Question set

7.7.7 Registering event handlers.

Animation

7.7.8 Capturing and bubbling.

 **Question set**

7.7.9 Capturing and bubbling.

 **Question set**

7.7.10 Bubbling and capturing.

 **Progression**

7.7.1 Event-driven programming.

7.8 LAB: JavaScript game object

 **Lab activity**

7.8.1 LAB: JavaScript game object

8. Wk 4 Prep: More HTML (~62) *Forms*

8.1 HTML containers

container

A container is any part of a web document body that has opening and closing tags.

parent container

A parent container is the container in which another element resides.

block

A block element (sometimes called a block-level element) fills the width of the element's parent container and can contain other block elements, inline elements, and text.

block-level

A block element (sometimes called a block-level element) fills the width of the element's parent container and can contain other block elements, inline elements, and text.

<div>

A <div> element is a generic element for creating block containers to facilitate managing page content and is the only block element with no semantic meaning, unlike other block elements such as <p> and <table>.

inline

An inline element fills the minimum space possible in the element's parent container and can only contain text or other inline elements.

The element is the generic element for creating inline containers to facilitate managing content on the page.

 **Table**

8.1.1 Common HTML containers.

 **Animation**

8.1.1 Containers and parent containers.

 **Question set**

8.1.2 Parent containers.

 **Question set**

8.1.3 Container structure.

 **Aside**

Note

 **Example**

8.1.1 Block elements with visible containers.

 **Question set**

8.1.4 Block tags.

 **Example**

8.1.2 Span tags with visible containers.

 **Question set**

8.1.5 Block and inline elements.

 **Question set**

8.1.6 Hierarchy of block and inline elements.

 **Aside**

Block vs. inline in Chrome DevTools

 **Learning tool**

8.1.7 Block vs. inline.

 **Progression**

8.1.1 HTML containers.

8.2 Forms

<form>

The <form> tag allows the web browser to submit information from the user to the server.

action

The action attribute indicates the URL where the form data should be sent.

method

The method attribute indicates the HTTP request type the browser will use to communicate with the server.

GET method

The GET method is a technique used by a web browser to submit information to a web server by altering the URL of the HTTP request.

query string

The query string is a set of name=value pairs separated by the ampersand character (&). Each name is specified as an attribute of the HTML field, and the value is the user-entered data.

POST method

The POST method is a technique used by a web browser to submit information to a web server by sending the information in the HTTP request body.

enctype

The <form> tag's enctype attribute value "multipart/form-data" indicates the web browser should split a POST request into multiple parts, where each input field is sent as a separate part of the HTTP request message.

escaped

If a user enters characters like &, ?, =, or white space characters like space, newline, or tab, the characters must be escaped, meaning the characters must be transformed into other representations.

unescapes

The web server unescapes the form data to determine what the original values are.

widget

A widget is an interactive component (usually graphical) that the browser uses to interact with a user. Ex: Buttons, drop-down menus, and data entry fields.

<input>

The <input> tag allows the user to enter information into a web page.

type

The type attribute indicates the widget type. Common types include text, password, submit, and button.

name

The name attribute names the widget and sends the widget's value when the widget's form is submitted.

id

The id attribute is used to give a widget a unique identifier.

placeholder

The placeholder attribute specifies text that first appears in a text widget, typically for giving the user a hint as to the expected value.

value

The value attribute specifies a default value for a widget.

text box

A text box widget is an `input` element with the `type` attribute of "text" that allows users to enter a single line of text.

submit button

The web browser displays a submit button widget for an `<input>` tag with the `type` attribute of "submit", which sends the associated form's data to the server when clicked.

<label>

The `<label>` tag displays descriptive text associated with a specific widget.

for

A label has a `for` attribute whose value should match the `id` attribute for the widget being labeled.

text area

A text area widget is an `input` element specified by `<textarea>` opening and closing tags that allows users to enter multiple lines of text.

<textarea>

A text area widget is an `input` element specified by `<textarea>` opening and closing tags that allows users to enter multiple lines of text.

rows

A `<textarea>` tag has optional `rows` and `cols` attributes to specify the initial size of the text area.

cols

A `<textarea>` tag has optional `rows` and `cols` attributes to specify the initial size of the text area.

Figure

8.2.1 Partial HTML form sending data to the Twitter home page using a secure HTTP POST request.

Animation

8.2.1 Submitting form data to a server.

 **Question set**

8.2.2 <form> tags.

 **Animation**

8.2.3 Using the GET method to submit form data to a server.

 **Aside**

Warning

 **Question set**

8.2.4 GET method.

 **Figure**

8.2.2 Chrome DevTools showing form data in POST request.

 **Animation**

8.2.5 Using the POST method to submit form data to a server.

 **Question set**

8.2.6 POST method.

 **Aside**

Escaping form data

 **Figure**

8.2.3 Complete HTML form sending status to the Twitter home page using a secure HTTP POST request.

 **Question set**

8.2.7 <input> attributes.

 **Figure**

8.2.4 HTML for a label associated with a text box.

 **Figure**

8.2.5 HTML for a textarea.

 **Learning tool**

8.2.8 Text inputs.

 **Question set**

8.2.9 Text inputs.

 **Progression**

8.2.1 Building forms.

8.3 Common form widgets

checkbox

A checkbox is a widget for input elements with the type attribute of "checkbox", which allows users to check, or select, a value.

checked

A checkbox initially appears selected if the checked attribute is set.

boolean attribute

A boolean attribute is an attribute that is true when present and false when absent. No value must be assigned to a boolean attribute.

radio button

A radio button is a widget for input elements with the type attribute of "radio", which allows users to select exactly one value from possibly many values.

<select>

The <select> opening and closing tags create a drop-down menu (or drop-down list), which allows users to select one of several predefined values.

drop-down menu

The <select> opening and closing tags create a drop-down menu (or drop-down list), which allows users to select one of several predefined values.

drop-down list

The <select> opening and closing tags create a drop-down menu (or drop-down list), which allows users to select one of several predefined values.

<option>

The <option> opening and closing tags create a value, or option, the user can select within a drop-down menu.

list box

A list box widget is created by specifying a size with the select element's size attribute.

button

A button widget can be created using the <button> opening and closing tags or with <input type="button">.

<button>

A button widget can be created using the <button> opening and closing tags or with <input type="button">.

password field

A password field is a widget for input elements with the type attribute of "password", which allows users to enter a password without the password contents being displayed on-screen.

<fieldset>

The <fieldset> tag groups related form widgets together and draws a box around the related widgets.

<legend>

The <legend> tag defines a caption for a <fieldset>.

Animation

8.3.1 Submitting checkboxes to the server.

Learning tool

8.3.2 Checkbox inputs.

Learning tool

8.3.3 Radio buttons.

Question set

8.3.4 Checkboxes and radio buttons.

Figure

8.3.1 Drop-down menu's default appearance (left) and when selecting an option (right).

Figure

8.3.2 List box that allows multiple options to be selected.

Figure

8.3.3 HTML buttons.

Aside

Styling widgets

Figure

8.3.4 Password field that limits the number of characters to 10.

Figure

8.3.5 Fieldset around related radio buttons.

Question set

8.3.5 Menus, buttons, and passwords.

Progression

8.3.1 Building common widgets.

8.4 Additional form widgets

picker

An input picker is a widget that allows the user to interactively pick a choice using a popup or other guided selection method.

date picker

The date picker is an input picker that allows the user to enter a date or choose a date from a calendar popup.

color picker

Clicking on the color picker creates a color selector popup that helps the user explore and choose a color.

number input

The number input ensures user input is a valid number.

range input

The range input widget allows the user to select a value by dragging a sliding control along the length of a line.

combo box

A combo box is the combination of a text box and drop-down menu into a single widget.

<datalist>

A combo box is created with an `<input>` element, which creates the text box, and a `<datalist>` element, which provides the drop-down list options.

fallback

A fallback is a mechanism that allows a web page element to function correctly even if the browser does not support a particular element.

polyfill

A polyfill is a fallback using JavaScript code that makes certain HTML features (Ex: the date picker) work on browsers that do not natively support those features.

Example

8.4.1 Date picker.

Example

8.4.2 Color picker.

Example

8.4.3 Number input.

Example

8.4.4 HTML5 range input.

Example

8.4.5 Combo box widget.

Example

8.4.6 Mobile browser keyboards for tel and email types.

Question set

8.4.1 Some additional widgets.

Question set

8.4.2 Input types.

Table

8.4.1 Input attributes.

Question set

8.4.3 Input attributes.

Example

8.4.7 A date polyfill.

Question set

8.4.4 Fallbacks and polyfills.

8.5 Audio and video

<embed>

Prior to HTML5, developers used `<embed>` and `<object>` elements to embed audio or video in a web page.

<object>

Prior to HTML5, developers used `<embed>` and `<object>` elements to embed audio or video in a web page.

plug-in

A browser plug-in is software that can properly read and interpret a file format that the browser cannot.

<audio>

The `<audio>` element plays an audio file in a web page.

<source>

The <source> element is used inside the <audio> tag to specify an audio file to play.

<video>

The <video> element displays a video in a web page.

<iframe>

The <iframe> element allows a web page to be embedded in a rectangular area of the current web page.

Figure

8.5.1 Flash example (left) and the same example when the Flash plug-in is unavailable (right).

Question set

8.5.1 Browser plug-ins.

Learning tool

8.5.2 Audio element.

Table

8.5.1 Common audio formats.

Question set

8.5.3 Audio element.

Learning tool

8.5.4 Playing a video.

Table

8.5.2 Common video formats.

Question set

8.5.5 Video element.

Aside

Controlling media playback with JavaScript

Learning tool

8.5.6 Embedding a YouTube video.

Question set

8.5.7 YouTube videos and <iframe>;

8.6 HTML developer guidelines

escaped

A special character such as the double quote can be escaped within a string by placing the backslash character (\) before the special character, causing that character to lose any special meaning.

boolean attribute

A boolean attribute is an attribute that is true when present and false when absent.

Question set

8.6.1 Use closing tags.

Question set

8.6.2 Identify correct use of quotes around attribute values.

Question set

8.6.3 Boolean attributes.

Question set

8.6.4 Use lowercase.

Question set

8.6.5 Start block elements on a new line.

Table

8.6.1 Good and bad examples of consistent indentation.

Question set

8.6.6 Indent nested elements.

Question set

8.6.7 Providing labels and placeholders.

Question set

8.6.8 Separate content from presentation and functionality.

Question set

8.6.9 Using the validator.

Example

8.6.1 Example web page implements all guidelines.

Question set

8.6.10 Consider all developer guidelines.

8.7 Example: Restaurant Reviews

index.html

Most web servers by default serve the index.html file when a URL does not specify an explicit filename.

Figure

8.7.1 Wireframes of restaurant review web pages.

Question set

8.7.1 Restaurant review web pages.

Question set

8.7.2 Home page.

Aside

Dynamically-built web pages

Question set

8.7.3 Detail pages.

Question set

8.7.4 Add Review page.

Question set

8.7.5 About page.

8.8 LAB: Fan web page (HTML)

Lab activity

8.8.1 LAB: Fan web page (HTML)

8.9 LAB: Form for joining a social network (HTML)

Lab activity

8.9.1 LAB: Form for joining a social network (HTML)

8.10 LAB: Happy birthday message creator (HTML)

Lab activity

8.10.1 LAB: Happy birthday message creator (HTML)

8.11 LAB: Starting lineup (HTML)

Lab activity

8.11.1 LAB: Starting lineup (HTML)

9. Wk 4 Prep: JavaScript Client/Server Interactions (~63)

9.1 Timers

timer

A timer is a general name for techniques to execute JavaScript code after some amount of time has occurred.

setTimeout()

The setTimeout() method takes two arguments: a function and a time delay in milliseconds (1/1000th of a second). The browser calls the function after the time delay.

clearTimeout()

The timeout can be cancelled by passing the identifier to clearTimeout().

setInterval()

The setInterval() method takes two arguments: a function and a time interval in milliseconds (t). The browser calls the function every t milliseconds until the interval is canceled.

clearInterval()

The interval identifier can be passed to the clearInterval() method to cancel the interval.

Animation

9.1.1 Showing a daily special with setTimeout().

Question set

9.1.2 Timeouts.

Animation

9.1.3 Animating a ball with setInterval().

Question set

9.1.4 Intervals.

Learning tool

9.1.5 Intervals.

Progression

9.1.1 Timers.

9.2 Form validation

value

Each textual input element in an HTML document has a value attribute that is associated with the user-entered text.

checked

Checkboxes and radio buttons have a checked attribute that is a boolean value indicating whether the user has chosen a particular checkbox or radio button.

isNaN(num)

The JavaScript function isNaN(num) returns true if the parameter, num, is *not* a number.

required

The required attribute indicates that the field must have a value (text or selection) prior to submitting the form.

max

The max and min attributes indicate the maximum and minimum values respectively that can be entered in an input field with ranges, such as a date or number.

min

The max and min attributes indicate the maximum and minimum values respectively that can be entered in an input field with ranges, such as a date or number.

maxlength

The maxlength and minlength attributes indicate the maximum and minimum length of input allowed by an input field.

minlength

The maxlength and minlength attributes indicate the maximum and minimum length of input allowed by an input field.

pattern

The pattern attribute provides a regular expression that valid input must match.

title

The title attribute can be used to provide a description of valid input when using the pattern attribute.

:valid

The :valid pseudo-class is active on an element when the element meets all the stated requirements in field attributes.

:invalid

The :invalid pseudo-class is active on an element when one or more of the attributes in the field are not fully met.

:required

The :required pseudo-class is active on an element if the element has the required attribute set.

:optional

The :optional pseudo-class is active on an element if the element does not have the required attribute set.

 **Animation**

9.2.1 Validating form input.

 **Question set**

9.2.2 Identify why the field is invalid.

 **Figure**

9.2.1 Ensuring a checkbox is selected before the form is submitted.

 **Learning tool**

9.2.3 Practice validating form prior to submission.

 **Aside**

Note

 **Figure**

9.2.2 Checking a ZIP code field as the user updates the field.

 **Question set**

9.2.4 Using JavaScript to validate input fields.

 **Aside**

Note

 **Figure**

9.2.3 Using HTML form validation.

 **Learning tool**

9.2.5 Practice with CSS pseudo-classes.

 **Question set**

9.2.6 Form validation questions.

 **Progression**

9.2.1 Form validation.

9.3 XMLHttpRequest (Ajax)

Ajax

Ajax (Asynchronous JavaScript and XML) is a technique to asynchronously communicate with a server and update a web page once the response is received, without reloading the whole web page.

Asynchronous JavaScript and XML

Ajax (Asynchronous JavaScript and XML) is a technique to asynchronously communicate with a server and update a web page once the response is received, without reloading the whole web page.

asynchronous request

An asynchronous request occurs when the web application sends a request to the server and continues running without waiting for the server response.

XMLHttpRequest

XMLHttpRequest is an object for communicating with web servers using Ajax.

cross-origin HTTP request

A cross-origin HTTP request is a request made to another domain.

load

The load handler is called when the exchange between the browser and server has completed.

error

The error handler is called when the browser does not receive an appropriate response to a request.

abort

The abort handler is called when the browser is told to stop a request/response that is still in progress.

timeout

The timeout handler is called if the browser takes too much time to fully receive a response to a request.

readystatechange

The readystatechange handler relates to any change in the XMLHttpRequest.

loadstart

The loadstart handler is called when the browser begins to send a request.

loadend

The loadend handler is called after the browser receives the response.

progress

The progress handler is called one or more times while a response is being received by the client.

status

The status attribute is the numeric status code returned in the response.

statusText

The statusText attribute is the descriptive text describing the status attribute.

response

The response attribute is the response body, which is parsed by the browser according to the responseType attribute.

responseText

The responseText attribute is the plain text version of the response.

responseXML

The responseXML attribute is the XML DOM version of the response.

responseType

The responseType attribute is set by the programmer to let the browser know the expected response data format.

upload

The XMLHttpRequest object's upload attribute is an object for monitoring the status of the request being sent to the server.

Animation

9.3.1 Asynchronous HTTP request.

Aside

Note

Question set

9.3.2 Identify steps in making an Ajax request.

Aside

Note

Question set

9.3.3 Match the event handlers to their descriptions.

Question set

9.3.4 XMLHttpRequest event handler order.

Table

9.3.1 Common HTTP response status codes.

Animation

9.3.5 Creating a query string and loading JSON.

 **Question set**

9.3.6 Ajax and JSON.

 **Learning tool**

9.3.7 Ajax practice.

 **Question set**

9.3.8 Updating the DOM via an Ajax request.

 **Progression**

9.3.1 XMLHttpRequest (Ajax).

 **Example**

9.3.1 Monitoring the progress of an uploaded file.

 **Question set**

9.3.9 Uploading files using XMLHttpRequest.

9.4 Using third-party web APIs (JavaScript)

third-party web API

A third-party web API is a public web API used by a web application to access data provided by a third party.

API key

To use a third-party web API, a developer usually registers with the third party to obtain an API key.

RESTful web API

A RESTful web API is a web API that is called with a URL that specifies API parameters and returns JSON or XML containing the API data.

SOAP-based web API

A SOAP-based web API is another type of web API that relies heavily on XML and is in general more complex to use than RESTful web APIs.

 **Aside**

SOAP

 **Figure**

9.4.1 Calling third-party web API from the web server or web browser.

 **Question set**

9.4.1 Third-party web APIs.

Figure

9.4.2 GET request to obtain the current weather for ZIP 90210.

Try

9.4.1 Try OpenWeatherMap's API in your web browser.

Question set

9.4.2 The Weather API.

Animation

9.4.3 Calling the Weather API with JavaScript.

Question set

9.4.4 Calling the Weather API from JavaScript.

9.5 Browser differences: JavaScript

polyfill

A polyfill is JavaScript code that provides missing standard functionality for older browsers.

Question set

9.5.1 JavaScript support in browsers.

Aside

Note

Example

9.5.1 An HTML5 date polyfill.

Example

9.5.2 Creating a polyfill for the Date.now() method.

Question set

9.5.2 Review browser JavaScript differences.

9.6 Example: Lights Out game

Animation

9.6.1 3x3 Lights Out game example.

Question set

9.6.2 Lights Out game concepts.

Figure

9.6.1 LightsOut.html file.

 **Question set**

9.6.3 LightsOut.html file.

 **Figure**

9.6.2 LightsOut.js excerpt: Game object declaration.

 **Question set**

9.6.4 JavaScript game object.

 **Figure**

9.6.3 LightsOut.js excerpt: allLightsOut() and toggle() functions.

 **Question set**

9.6.5 allLightsOut and toggle functions.

 **Figure**

9.6.4 LightsOut.js excerpt: input and UI functions.

 **Question set**

9.6.6 Game input and UI functions.

9.7 Example: Weather Comparison

 **Animation**

9.7.1 Using the OpenWeatherMap web API to retrieve a weather forecast.

 **Figure**

9.7.1 URL and JSON response containing Denver forecast.

 **Question set**

9.7.2 JSON forecast.

 **Question set**

9.7.3 HTML and CSS for Weather Comparison app.

 **Figure**

9.7.2 Compare button callback and supporting code.

 **Question set**

9.7.4 Compare button click.

 **Figure**

9.7.3 getWeatherForecast() function.

 **Question set**

9.7.5 Requesting the forecast.

Figure

9.7.4 responseReceived() and supporting functions.

Figure

9.7.5 getSummaryForecast() function.

Question set

9.7.6 Displaying the forecast.

9.8 LAB: Grade distribution

Lab activity

9.8.1 LAB: Grade distribution

9.9 LAB: Temperature conversion

Lab activity

9.9.1 LAB: Temperature conversion

9.10 LAB: JavaScript Tic-Tac-Toe

Lab activity

9.10.1 LAB: JavaScript Tic-Tac-Toe

9.11 LAB: Medical device vulnerability scoring

Lab activity

9.11.1 LAB: Medical device vulnerability scoring

9.12 LAB: Quote web API

Lab activity

9.12.1 LAB: Quote web API

9.13 LAB: Move the heart image with a timer

Lab activity

9.13.1 LAB: Move the heart image with a timer

10. IN-PROGRESS Wk 5 Prep: Mobile Web Development (62)

10.1 Mobile websites and browsers

mobile web browser

A mobile web browser is a web browser designed for mobile devices that can display web pages using HTML, CSS, and JavaScript.

mobile website

A mobile website is a website that is designed for mobile devices with smaller screen sizes and touch interfaces.

Responsive web design

Responsive web design (RWD) is a collection of techniques to create web pages that adapt to the browser's size.

RWD

Responsive web design (RWD) is a collection of techniques to create web pages that adapt to the browser's size.

adaptive website

An adaptive website adapts to the width of the browser at specific widths. Ex: A container is 400 pixels wide when the browser is wider than 500 pixels, but the container shrinks to 200 pixels when the browser is less than 500 pixels wide.

▀▀▀ Aside

Early mobile markup languages

▀▀▀ Figure

10.1.1 CNN.com for desktop and mobile browsers.

▀▀▀ Figure

10.1.2 Desktop vs. mobile access in North America.

▀▀▀ Figure

10.1.3 Top 9 mobile browsers in North America.

███ Question set

10.1.1 Mobile web browsers.

███ Question set

10.1.2 Designing mobile websites.

███ Animation

10.1.3 Dynamic serving.

▀▀▀ Table

10.1.1 Example user agent strings for mobile browsers.

███ Question set

10.1.4 Implementing mobile websites.

▀▀▀ Figure

10.1.4 Web page using responsive web design for desktop, tablet, and mobile.

Aside

Responsive vs. adaptive

Question set

10.1.5 Responsive web design.

10.2 Mobile development tools

screen emulator

A screen emulator is software that simulates how mobile device screens operate.

device pixel ratio (DPR)

The device pixel ratio (DPR) is the ratio between device pixels and logical pixels. Ex: A DPR of 2 means 1 logical pixel is 2 device pixels wide and 2 device pixels tall, 4 device pixels altogether.

logical pixel

A logical pixel is also called a device-independent pixel (DIP). The CSS "px" unit is a logical pixel unit. Ex: `<div style="width:20px">` is 20 logical pixels wide.

device-independent pixel (DIP)

A logical pixel is also called a device-independent pixel (DIP). The CSS "px" unit is a logical pixel unit. Ex: `<div style="width:20px">` is 20 logical pixels wide.

Retina display

A Retina display or high-density display is a screen that packs more pixels into a smaller area than traditional screens. Pixels are not visible at a viewing distance on a Retina display like pixels are on a standard display.

high-density display

A Retina display or high-density display is a screen that packs more pixels into a smaller area than traditional screens. Pixels are not visible at a viewing distance on a Retina display like pixels are on a standard display.

Figure

10.2.1 Chrome DevTools screen emulator.

Figure

10.2.2 Selecting a device to emulate.

Question set

10.2.1 Chrome screen emulator.

Figure

10.2.3 Logical and device pixels on standard and 2 DPR display.

Figure

10.2.4 DPR settings in Chrome's screen emulator.

Figure

10.2.5 Retina and standard display.

Question set

10.2.2 DPR and high-density displays.

Figure

10.2.6 Emulator configuration options.

Question set

10.2.3 Emulating network conditions.

Figure

10.2.7 Sensors to emulate geolocation and device orientation.

Question set

10.2.4 Sensors emulator.

10.3 Viewport

viewport

A viewport is the visible area of a web page.

viewport unit

A viewport unit (vw and vh) is a percentage of the browser viewport's width or height, where $1\text{vw} = 1\%$ of viewport width and $1\text{vh} = 1\%$ of viewport height.

vw

A viewport unit (vw and vh) is a percentage of the browser viewport's width or height, where $1\text{vw} = 1\%$ of viewport width and $1\text{vh} = 1\%$ of viewport height.

vh

A viewport unit (vw and vh) is a percentage of the browser viewport's width or height, where $1\text{vw} = 1\%$ of viewport width and $1\text{vh} = 1\%$ of viewport height.

Figure

10.3.1 Fitting the desktop viewport into a mobile viewport.

Figure

10.3.2 Viewport meta tag with standard parameters.

Figure

10.3.3 Viewport meta tag used in web page.

Question set

10.3.1 Viewport meta tag.

Animation

10.3.2 Viewport units.

Question set

10.3.3 Viewport units.

Learning tool

10.3.4 Practice with viewport units.

Progression

10.3.1 Viewport.

10.4 Responsive images

responsive image

A responsive image is an image that scales to fit different layouts in a responsive website.

Scalable Vector Graphics (SVG) image

A Scalable Vector Graphics (SVG) image is a vector image that is defined with XML.

vector image

A vector image is an image defined with lines, curves, and points that scale nicely at any resolution and always appear crisp.

srcset

The `` `srcset` attribute specifies which image should be displayed for specific DPR values.

sizes

The image's `sizes` attribute specifies one or more pairs of media conditions and relative image sizes.

Art direction

Art direction is the process of swapping out images with different proportions for different screen sizes.

<picture>

The `<picture>` element contains one or more `<source>` elements that specify a media condition and image. If the media condition is true, the associated image is downloaded and displayed.

<source>

The <picture> element contains one or more <source> elements that specify a media condition and image. If the media condition is true, the associated image is downloaded and displayed.

polyfill

A polyfill is code that supports some functionality that is missing from a browser.

Figure

10.4.1 Low DPR images appear pixelated on high DPR screens.

Question set

10.4.1 Responsive images.

Figure

10.4.2 SVG image file and <svg> element.

Question set

10.4.2 SVG images.

Figure

10.4.3 JPG images at different resolutions.

Animation

10.4.3 Browsers request images at correct resolution.

Question set

10.4.4 Image srcset attribute.

Animation

10.4.5 Browsers request images with optimal widths.

Question set

10.4.6 Image sizes and srcset attributes.

Figure

10.4.4 Art direction for various screen sizes.

Animation

10.4.7 Browsers request images with <picture> element.

Learning tool

10.4.8 Practice with the picture element.

Question set

10.4.9 Picture element.

10.5 Example: Responsive Band Web Page

Figure

10.5.1 Mobile, tablet, and desktop wireframes.

Question set

10.5.1 Mobile design.

Figure

10.5.2 Media query for tablets.

Question set

10.5.2 Tablet design.

Figure

10.5.3 Media query for desktop.

Question set

10.5.3 Desktop design.

Figure

10.5.4 Member images for various DPRs.

Figure

10.5.5 Images using srcset attribute.

Figure

10.5.6 Chrome DevTools emulating a 3 DPR device.

Question set

10.5.4 Final version.

10.6 LAB: Media queries for a vacation website

Lab activity

10.6.1 LAB: Media queries for a vacation website

10.7 LAB: Responsive images for a vacation website

Lab activity

10.7.1 LAB: Responsive images for a vacation website

11. IN-PROGRESS Wk 6,7 Prep: Advanced JavaScript (119) *Auth.*

11.1 Regular expressions

regular expression

A regular expression (often shortened to regex) is a string pattern that is matched against a string.

regex

A regular expression (often shortened to regex) is a string pattern that is matched against a string.

RegExp

Regular expressions may be defined with a RegExp object or between two forward slashes.

test(str)

The RegExp method test(str) returns true if the string str matches the regex, and false otherwise.

metacharacter

A metacharacter is a character or character sequence that matches a class of characters in a regular expression.

mode modifier

A mode modifier (sometimes called a flag) changes how a regex matches and is placed after the second slash in a regex.

flag

A mode modifier (sometimes called a flag) changes how a regex matches and is placed after the second slash in a regex.

exec(str)

The RegExp method exec(str) determines what part of the string str matches a regex. The exec() method returns a result array, or returns null if the pattern does not match.

match()

Match() returns an array of the matches made when matching the string against a regex.

replace()

Replace() returns a new string that replaces matching strings with a replacement string.

search()

Search() returns the first match between the regex and the given string.

split()

Split() returns an array of strings created by separating the string into substrings based on a regex.

Animation

11.1.1 Searching an array for the pattern 'ab'.

 **Question set**

11.1.2 Simple regular expressions.

 **Table**

11.1.1 Selected special characters in regex patterns.

 **Question set**

11.1.3 Regex with special characters.

 **Question set**

11.1.4 Regex with brackets.

 **Table**

11.1.2 Selected metacharacters in regex patterns.

 **Question set**

11.1.5 Metacharacters in regex.

 **Table**

11.1.3 Selected mode modifiers.

 **Question set**

11.1.6 Regex with mode modifiers.

 **Learning tool**

11.1.7 Using regular expressions to identify secret messages.

 **Progression**

11.1.1 Regular expressions.

 **Figure**

11.1.1 Using exec() to discover what characters matched the regex.

 **Figure**

11.1.2 Remembering matches in a regex.

 **Learning tool**

11.1.8 Extracting regex matches.

 **Question set**

11.1.9 The exec() method.

 **Progression**

11.1.2 Determining what matches.

▀▀▀ Aside

String methods that use regex

11.2 Inner functions, outer functions, and function scope

inner function

An inner function (nested function) is a function declared inside another function.

nested function

An inner function (nested function) is a function declared inside another function.

outer function

An outer function is a function containing an inner function.

filter()

An Array object's filter() method takes a filter function as an argument, calls the filter function for each array element, and returns a new array consisting only of elements for which the filter function returns true.

scope object

To store a collection of variables for a particular scope, JavaScript implementations commonly use a scope object: An object that stores a collection of variable names and corresponding values.

scope chain

A scope chain is a linked list of scope objects used by the JavaScript runtime to store and lookup variable values when executing code. When a variable is needed, a search begins at the scope object at the beginning of the scope chain. If the variable is found, the corresponding value is used. Otherwise, the next object in the scope chain is searched. If the search reaches a null object at the end of the scope chain, the variable is not found and a ReferenceError is thrown.

▀ Animation

11.2.1 Inner and outer functions.

▀ Question set

11.2.2 Inner and outer functions.

▀▀▀ Aside

A function declaration or function expression can be used to declare an inner function

▀ Animation

11.2.3 Filtering an array of grades to get only passing grades.

▀ Question set

11.2.4 Array filtering using inner functions.

 **Animation**

11.2.5 Scope objects.

 **Aside**

Avoid mixing 'var' and 'let' in practice

 **Question set**

11.2.6 Scope objects.

 **Animation**

11.2.7 Scope chain.

 **Aside**

Scope objects reference the next/outer scope object

 **Question set**

11.2.8 Scope chain.

11.3 Closures

execution context

An execution context is an object that stores information needed to execute JavaScript code, and includes, but is not limited to:

- information about code execution state, such as the line of code being executed and the line to return to when a function completes, and
- a reference to a scope chain.

current execution context

The current execution context (running execution context) is the execution context at the top of the execution stack.

running execution context

The current execution context (running execution context) is the execution context at the top of the execution stack.

current scope chain

The current scope chain is the scope chain of the current execution context.

closure

A closure is a combination of a function's code and a reference to a scope chain.

 **Animation**

11.3.1 A closure allows an inner function to access the outer function's scope.

■ Animation

11.3.2 Closures can access local variables from functions that have completed execution.

■ Animation

11.3.3 Changing a local variable after a closure's creation can affect functionality.

■ Question set

11.3.4 Execution context and closures.

■ Animation

11.3.5 A new block scope is created each loop iteration, storing block-scoped variables.

■ Question set

11.3.6 Closures and loops.

■ Animation

11.3.7 Variables declared with var have function scope, even if declared inside a loop.

■ Question set

11.3.8 Closures and loops.

■ Aside

JavaScript runtimes may optimize scope chains

11.4 Strict mode

Strict mode

Strict mode makes a JavaScript interpreter apply a set of restrictive syntax rules to JavaScript code.

■ Animation

11.4.1 Misspelling a variable name can be difficult to detect.

■ Animation

11.4.2 Strict mode causes misspelled variable assignment to throw an exception.

■ Question set

11.4.3 Strict errors.

■ Figure

11.4.1 A strict function and a regular function.

■ Question set

11.4.4 Strict errors in functions.

11.5 Web storage

Web Storage API

The Web Storage API provides storage objects that allow JavaScript programs to securely store key/value pairs in the web browser.

localStorage

The sessionStorage object stores key/value pairs for an origin that are only available for the duration of the session.

localStorage

The localStorage object stores key/value pairs for an origin that are stored indefinitely.

origin

An origin is a combination of scheme, hostname, and port number in a URL.

setItem(key, value)

setItem(key, value) stores the key string and associated value string in storage.

getItem(key)

GetItem(key) returns the value associated with the key in storage or null if the key does not exist.

removeItem(key)

RemoveItem(key) removes the key and associated value from storage.

clear()

Clear() removes all keys and associated values from storage.

Question set

11.5.1 Web storage.

Aside

Private browsing

Animation

11.5.2 Storing and retrieving values from localStorage.

Question set

11.5.3 Web storage methods.

Progression

11.5.1 Web storage.

Aside

Chrome DevTools displays web storage

11.6 Canvas drawing

<canvas>

The <canvas> tag defines a rectangular area of a web page where shapes, images, and text can be displayed using JavaScript.

origin

The canvas origin (0, 0) is located in the upper-left corner of the canvas.

context

The canvas' context object represents the drawing surface of the canvas. The 2D context is used for drawing two dimensional graphics.

strokeRect()

The context method strokeRect() draws an outlined rectangle.

fillRect()

The fillRect() method draws a filled rectangle.

path

A path is a list of points that are connected by lines.

beginPath()

The beginPath() method creates a new path.

moveTo()

The moveTo() method defines the path's starting point at coordinate (x, y).

lineTo()

The lineTo() method draws a line from the last point to a new point at coordinate (x, y).

closePath()

The closePath() method draws a line from the current point to the path's starting point.

arc()

The context method arc() adds an arc to a path.

drawImage()

The context method drawImage() draws an Image object with the image's top-left corner anchored at the given (x, y) coordinate. Optional width and height arguments draw the image with the given dimensions.

fillText()

The context method fillText() draws filled text on a canvas.

strokeText()

The `strokeText()` method draws the outline of text.

Figure

11.6.1 Canvas uses a coordinate system with $(0, 0)$ anchored in the upper-left corner.

Animation

11.6.1 Drawing rectangles on a canvas.

Table

11.6.1 Rectangle methods and properties.

Question set

11.6.2 Canvas and context.

Animation

11.6.3 Using a path to draw a triangle.

Table

11.6.2 Path methods and properties.

Question set

11.6.4 Drawing lines.

Learning tool

11.6.5 Practice drawing paths and rectangles.

Figure

11.6.2 `.arc(x, y, radius, startAngle, endAngle)` draws an arc or circle.

Figure

11.6.3 Two semicircles drawn with the `arc()` method.

Aside

Converting degrees to radians.

Table

11.6.3 Arc methods and properties.

Question set

11.6.6 Drawing arcs and circles.

Figure

11.6.4 Drawing an image with the `drawImage()` method.

Question set

11.6.7 Drawing images.

Figure

11.6.5 Drawing text with fillText() and strokeText() methods.

Table

11.6.4 Text methods and properties.

Question set

11.6.8 Drawing text.

Learning tool

11.6.9 Practice drawing images and text.

11.7 Canvas transformations and animation

transformation

A transformation is a graphical operation that alters the position, shape, or orientation of an object.

Translation

Translation - Moving the canvas' origin to another location to draw a graphic at a different location.

Rotation

Rotation - Rotating the canvas to draw a graphic at an angle.

Scaling

Scaling - Increasing or decreasing the canvas' grid to draw a graphic larger or smaller.

translate()

The translate() method translates the origin of the canvas to the given (x,y) coordinate, relative to the current origin.

rotate()

The context method rotate() rotates the canvas clockwise about the current origin by the given radian angle.

scale()

The scale() method uses horizontal and vertical multipliers to increase or decrease the size of graphics on a canvas.

clearRect()

ClearRect() clears the canvas of any previous graphics.

save()

Save() saves the current context settings.

restore()

Restore() restores the saved context settings.

animation frame

The human eye perceives smooth animation when the screen updates at least 60 times a second, so each animation frame is visible for only 1/60 of a second.

window.requestAnimationFrame(callback)

The window.requestAnimationFrame(callback) method is used to repeatedly redraw the canvas to produce an animation.

 **Figure**

11.7.1 Three transformations: Translation, rotation, and scaling.

 **Animation**

11.7.1 Translating the origin to draw 3 circles.

 **Question set**

11.7.2 Translating.

 **Learning tool**

11.7.3 Practice translating.

 **Figure**

11.7.2 Using rotate() to display rotated squares.

 **Figure**

11.7.3 Combining translate() and rotate() to rotate squares about the center of the squares.

 **Question set**

11.7.4 Rotating the canvas.

 **Figure**

11.7.4 Doubling the width and halving the height of a square with scale().

 **Question set**

11.7.5 Scaling.

 **Learning tool**

11.7.6 Practice scaling.

 **Animation**

11.7.7 Animating a circle.

 **Question set**

11.7.8 Canvas animation.

 **Learning tool**

11.7.9 Practice animating.

11.8 WebSockets

Polling

Polling is a technique where the client sends an Ajax request periodically to the web server asking if any new data is available.

WebSocket protocol

The WebSocket protocol runs over a single TCP connection and allows real-time communication between web servers and web browsers.

WebSocket handshake

The browser and server perform a WebSocket handshake to establish a WebSocket connection.

 **Figure**

11.8.1 WebSockets allow a server to notify the browser instantly when new data arrives.

 **Animation**

11.8.1 Using the WebSocket object to send and receive a message.

 **Question set**

11.8.2 Using the WebSocket object.

 **Figure**

11.8.2 Viewing a WebSocket handshake in Chrome DevTools.

 **Figure**

11.8.3 Viewing WebSocket messages in Chrome DevTools.

 **Question set**

11.8.3 WebSocket handshake.

 **Learning tool**

11.8.4 Create a parrot chat app with WebSockets.

11.9 Asynchronous JavaScript

synchronous function

A synchronous function is a function that completes an operation before returning.

asynchronous function

An asynchronous function is a function that starts an operation and potentially returns before the operation completes. The operation completes in the background, allowing other code to execute in the meantime.

Promise object

So asynchronous functions often return a Promise object: An object representing the eventual completion of the asynchronous operation.

Pending

Pending means that the asynchronous operation is still running.

Fulfilled

Fulfilled means that the asynchronous operation has completed successfully.

Rejected

Rejected means that the asynchronous operation has ended in failure to produce the intended result.

settled

Once reaching the fulfilled or rejected state, the Promise object is settled (or resolved), and the state will not change again.

resolved

Once reaching the fulfilled or rejected state, the Promise object is settled (or resolved), and the state will not change again.

then()

A Promise object's then() method can be called to request notifications about the state.

catch()

A Promise object's catch() method takes a single argument that is a function to call if the Promise is rejected or if the fulfilled handler throws an exception.

await operator

The await operator waits for a Promise object to resolve and invoke the appropriate callback function before proceeding to the next statement.

Animation

11.9.1 Asynchronous functions are commonly used to download data in the background.

Question set

11.9.2 Asynchronous function result data.

Question set

11.9.3 Synchronous and asynchronous functions.

Figure

11.9.1 Promise state transition diagram.

Question set

11.9.4 Promises.

Animation

11.9.5 Cloud storage APIs commonly return Promise objects for file operations.

Question set

11.9.6 Promise.then() method.

Animation

11.9.7 Promise fulfillment values and rejection reasons.

Question set

11.9.8 Promise fulfillment values and rejection reasons.

Table

11.9.1 Comparison of Promise object's then() and catch() usage scenarios.

Question set

11.9.9 Promise.catch() method.

Animation

11.9.10 await allows an async function to wait for completion of an asynchronous operation.

Question set

11.9.11 async and await.

11.10 Example: Lights Out game with canvas

Object.freeze()

Object.freeze() makes an object a frozen object: an object that cannot have any properties added, removed, or changed.

Figure

11.10.1 LightsOutRevisited.html file.

Question set

11.10.1 LightsOutRevisited.html file.

Aside

Object.freeze() creates frozen light objects, preventing external changes that violate game rules

Figure

11.10.2 LightsOutGame.js file.

Question set

11.10.2 LightsOutGame.js file.

Figure

11.10.3 LightsOutRevisited.js file.

Question set

11.10.3 LightsOutRevisited.js functions.

11.11 LAB: User registration and validation

Lab activity

11.11.1 LAB: User registration and validation

11.12 LAB: JavaScript SuperHero and SuperVillain classes

Lab activity

11.12.1 LAB: JavaScript SuperHero and SuperVillain classes

11.13 LAB: Grocery list

Lab activity

11.13.1 LAB: Grocery list

11.14 LAB: Snowman canvas

Lab activity

11.14.1 LAB: Snowman canvas

11.15 LAB: Frog image rotation

Lab activity

11.15.1 LAB: Frog image rotation

11.16 LAB: Circle with a Promise

Lab activity

11.16.1 LAB: Circle with a Promise

12. IN-PROGRESS Wk 8,9 Prep: Node.js (160)

12.1 Full-stack development (Node)

Client-side

Client-side (or front-end) refers to those technologies that run in the web browser like HTML, CSS, and JavaScript.

front-end

Client-side (or front-end) refers to those technologies that run in the web browser like HTML, CSS, and JavaScript.

Server-side

Server-side (or back-end) refers to those technologies that run on the web server like PHP, Python, Node.js, etc. and databases.

back-end

Server-side (or back-end) refers to those technologies that run on the web server like PHP, Python, Node.js, etc. and databases.

front-end developer

A front-end developer is a developer that is proficient in client-side technologies.

back-end developer

A back-end developer is a developer that is proficient in server-side technologies.

full-stack developer

A full-stack developer is a developer who has expertise in all aspects of a website or web application's development, including client technologies, server technologies, data modeling, and user interfaces.

web hosting company

A web hosting company is a company that hosts others' websites on the company's servers, usually for a fee.

virtual private server (VPS)

A virtual private server (VPS) is an autonomous server that is hosted on a physical server with other virtual servers.

Malware

Malware is malicious software designed to cripple a computer system or perform unwanted actions.

dynamic web page

A dynamic web page is a web page that is generated on the web server when requested, typically personalized to the user who requested the page.

Single Page Application (SPA)

A Single Page Application (SPA) is a web application that provides a similar user experience as a desktop application, all in a single web page.

web API

A web API is a collection of functions that are invoked using HTTP. Ex: An HTTP GET request to the URL <http://linkedin.com/api/contacts> may retrieve a list of all contacts from the web server.

isomorphic JavaScript application

An isomorphic JavaScript application is a web application that can run on the web browser and the web server.

relational database

A relational database stores data in relations (usually called tables).

Structured Query Language (SQL)

The Structured Query Language (SQL) is a language for creating, editing, selecting, and deleting data in a relational database.

Non-relational databases

Non-relational databases, sometimes called non-SQL or NoSQL databases, have become increasingly popular over the last few years. Non-relational databases use different methods to store and retrieve data using a variety of data access languages.

non-SQL

Non-relational databases, sometimes called non-SQL or NoSQL databases, have become increasingly popular over the last few years. Non-relational databases use different methods to store and retrieve data using a variety of data access languages.

NoSQL

Non-relational databases, sometimes called non-SQL or NoSQL databases, have become increasingly popular over the last few years. Non-relational databases use different methods to store and retrieve data using a variety of data access languages.

HTML preprocessor

An HTML preprocessor is a program that converts a markup language into HTML. The markup languages supported by HTML preprocessors are generally easier to use and read than HTML. Ex: Haml, Markdown, Slim, Pug.

CSS preprocessor

A CSS preprocessor is a program that converts a CSS-like language into CSS. CSS-like languages simplify the development of CSS stylesheets used in large projects. Ex: Sass, Less, Stylus.

CSS front-end framework

A CSS front-end framework is a framework that uses CSS or CSS pre-processors to aid in developing responsive websites that work well on every screen size. Ex: Bootstrap, YAML 4, Skeleton, Foundation.

compile-to-JavaScript language

A compile-to-JavaScript language is a programming language that is compiled into JavaScript.

JavaScript framework

A JavaScript framework is a JavaScript environment that dictates the organization of the application's JavaScript to simplify many programming tasks.

Figure

12.1.1 Front-end and back-end technologies.

Question set

12.1.1 Primary layers of the full stack.

Question set

12.1.2 Web hosting.

Question set

12.1.3 Server-side programming platforms and languages.

Animation

12.1.4 Cool Contacts SPA.

Question set

12.1.5 Server-side programming platforms.

Aside

Isomorphic JavaScript applications

Figure

12.1.2 Relational model and document model for student data.

Question set

12.1.6 Databases.

Figure

12.1.3 Example use of HTML and CSS preprocessors and compile-to-JavaScript.

Question set

12.1.7 Client-side technologies.

Question set

12.1.8 Testing a web application.

12.2 Getting started with Node.js

Node.js

Node.js is a JavaScript runtime environment that is primarily used to run server-side web applications.

Node.js module

A Node.js module is a self-contained collection of JavaScript code.

http module

The http module allows a Node.js application to create a simple web server.

Node.js project

A Node.js project is a collection of JavaScript files, packages, configuration files, and other miscellaneous files that are stored in a directory.

Node Package Manager (npm)

The Node Package Manager (npm) is the package manager for Node.js that allows developers to install and update packaged modules.

dependency

A dependency is a package that a Node.js project must be able to access to run.

nodemon module

The nodemon module saves developers time by restarting a Node.js application whenever the files in a project are modified.

underscore module

The underscore module is a library of helpful functions that extends some built-in JavaScript objects.

require()

The require() function imports a package for use in a Node.js program.

package.json

The package.json file contains JSON that lists the Node.js project's name, version, license, dependencies, and other information.

package-lock.json

A package-lock.json file is created or modified when project dependencies are added or removed. The file ensures that the same dependency versions are always used when the project is installed on different machines.

Semantic versioning

Semantic versioning is a popular software versioning scheme that uses a sequence of three digits: major.minor.patch.

Figure

12.2.1 Number of modules for the Node.js package manager (npm) in 2018-2019 far exceeds other languages.

Question set

12.2.1 Introduction to Node.js.

Figure

12.2.2 Node.js interactive shell.

Figure

12.2.3 Simple Node.js program.

Animation

12.2.2 A simple Node.js web server.

Question set

12.2.3 Node.js web server.

Figure

12.2.4 Example Node.js project with a single JavaScript file.

Figure

12.2.5 Display npm's version.

Figure

12.2.6 Get npm's prefix directory where global packages are installed.

Figure

12.2.7 Installing and running nodemon.

Figure

12.2.8 Installing "underscore" as a local package.

Figure

12.2.9 Using the underscore package to get random dice rolls.

Table

12.2.1 Summary of npm commands.

Question set

12.2.4 Using npm.

Figure

12.2.10 Example package.json file.

▀▀▀ **Aside**

Semantic versioning

▀▀▀ **Figure**

12.2.11 Files composing Node.js project.

█████ **Question set**

12.2.5 Node.js project's package.json file.

12.3 Express

Express

Express is a popular web application framework for Node.js because Express allows developers to create web servers with less code.

express module

The express module contains the Express framework and is installed using:

```
npm install express.
```

route

An Express route is a specific URL path and an HTTP request method to which a callback function is assigned.

request

A route's callback function has request and response parameters, which represent the HTTP request and response.

response

A route's callback function has request and response parameters, which represent the HTTP request and response.

req.query

Express automatically parses query string parameters (the values appearing after the "?" in a URL) and stores the parameters' names and values in the req.query object.

body-parser

The body-parser package is middleware that parses data that has been posted to a route.

middleware function

A middleware function (or just middleware) is a function that examines or modifies the request and/or response objects.

middleware

A middleware function (or just middleware) is a function that examines or modifies the request and/or response objects.

route parameter

A route parameter is a string near or at the end of the URL path that specifies a data value. A route parameter is defined in the route path with a colon (:) before the parameter name.

req.params

Route parameters are attached to the req.params object.

app.param()

Express has an app.param() function to define parameter middleware that executes before the route is called.

next()

The next() function allows other middleware functions to execute.

Animation

12.3.1 Simple Express web server.

Question set

12.3.2 Express server.

Figure

12.3.1 Example routes.

Question set

12.3.3 Express routes.

Animation

12.3.4 Extracting query string values.

Animation

12.3.5 Using the body-parser middleware to extract posted form data.

Table

12.3.1 Popular middleware for Express.

Question set

12.3.6 Express middleware.

Animation

12.3.7 Extracting route parameters.

Figure

12.3.2 Route parameter middleware example.

Question set

12.3.8 Express route parameters.

Aside

Express application generator.

12.4 Pug

template engine

A template engine creates dynamic web pages by replacing variables in a template file with specific values, creating HTML that is sent to the web browser.

Rendering

Rendering is the process of combining data with a template.

view

A view is the visual presentation of the application's data.

Pug

Pug (previously Jade) is a popular template engine for Express.

local

A local is a JavaScript variable defined outside of a template whose value is made available to a template.

mixin

A mixin is a Pug function that is defined with the `mixin` keyword.

block

The block statement names a block of Pug content that may be replaced by inherited content.

append

The append and prepend statements `append` and `prepend`, respectively, a Pug block to the inherited Pug block.

prepend

The append and prepend statements `append` and `prepend`, respectively, a Pug block to the inherited Pug block.

extend

A Pug template that inherits from another template uses the extend statement.

include

The include statement allows a Pug template to import the contents of a text file.

 **Animation**

12.4.1 Rendering Pug on the web server.

 **Question set**

12.4.2 Template engines.

 **Animation**

12.4.3 Indentation's effects on rendered HTML.

 **Table**

12.4.1 Summary of Pug syntax.

 **Question set**

12.4.4 Pug syntax basics.

 **Figure**

12.4.1 JavaScript in a Pug template.

 **Animation**

12.4.5 Locals passed to the Pug template.

 **Question set**

12.4.6 JavaScript in Pug templates.

 **Table**

12.4.2 Summary of Pug conditions and loops.

 **Question set**

12.4.7 Pug conditions and loops.

 **Table**

12.4.3 Pug mixins.

 **Question set**

12.4.8 Pug mixins.

 **Figure**

12.4.2 Pug inheritance example.

 **Figure**

12.4.3 Pug include example.

 **Question set**

12.4.9 Pug inheritance and includes.

12.5 Relational databases and SQL (Node)

table

A table is a collection of related data that is organized into columns and rows similar to a spreadsheet.

Structured Query Language

Developers use the Structured Query Language (SQL) to manipulate data in a relational database.

SQL

Developers use the Structured Query Language (SQL) to manipulate data in a relational database.

relational database management system

A relational database management system (RDBMS) is a system that manages the data for relational databases and executes SQL statements.

RDBMS

A relational database management system (RDBMS) is a system that manages the data for relational databases and executes SQL statements.

SQL statement

An SQL statement is a complete command and is terminated with a semicolon.

CREATE TABLE

The CREATE TABLE statement is an SQL statement that creates a table by specifying the table name, column names, and data types.

primary key

The primary key is the column or columns that uniquely identify each row in the table.

auto-increment column

An auto-increment column is a column that is assigned an automatically incrementing value.

AUTO_INCREMENT

MySQL uses the AUTO_INCREMENT keyword to define an auto-increment column.

INSERT

An INSERT statement inserts one or more rows into a table.

SELECT

The SELECT statement retrieves data from a table.

WHERE

The WHERE clause works like an if statement in a programming language, specifying conditions that must be true for a row to be selected.

ORDER BY

A SELECT statement may use the ORDER BY clause to sort selected rows in ascending (alphabetic) order.

COUNT()

The COUNT() function counts how many rows are returned by a SELECT statement.

UPDATE

The UPDATE statement changes the values of existing rows in a table.

DELETE

The DELETE statement removes rows from a table.

Animation

12.5.1 Students table with three rows.

Question set

12.5.2 Introduction to relational databases.

Animation

12.5.3 Creating a students table.

Table

12.5.1 Common data types in SQL.

Question set

12.5.4 SQL and creating tables.

Aside

Auto-increment columns

Aside

SQL Fiddle

Animation

12.5.5 INSERT statement.

Question set

12.5.6 INSERT statement.

Aside

Inserting with auto-incrementing columns

Animation

12.5.7 SELECT statement.

Table

12.5.2 WHERE condition operators.

Question set

12.5.8 SELECT statement.

Aside

Joining data from multiple tables.

Figure

12.5.1 UPDATE statement changes Alice's GPA to 3.9.

Figure

12.5.2 DELETE statement removes Billy from the students table.

Question set

12.5.9 UPDATE and DELETE statement.

12.6 MySQL (Node)

MySQL

MySQL is a popular RDBMS that offers an open source version and a few commercial versions that provide additional functionality.

database administrator

MySQL server has a user account assigned administrative roles that allows the database administrator to create user accounts and databases and perform other administrative activities.

mysql command-line tool

The mysql command-line tool (sometimes called the "terminal monitor" or "monitor") is a command-line program that allows developers to connect to a MySQL server, perform administrative functions, and execute SQL statements.

MySQL database

A MySQL database is a collection of tables.

CREATE DATABASE

The CREATE DATABASE command creates a new database.

SHOW DATABASES

The SHOW DATABASES command shows all the databases.

USE

The USE command selects a database to use.

SHOW TABLES

The SHOW TABLES command shows all the tables in the database.

QUIT

The QUIT command exits the mysql tool.

error code

MySQL server returns an error code and description when a SQL statement is syntactically incorrect or the database cannot execute the statement.

Question set

12.6.1 Introduction to MySQL.

Animation

12.6.2 Using mysql tool to connect to MySQL server.

Question set

12.6.3 Connecting with mysql tool.

Figure

12.6.1 Creating a database called "test" and using the "test" database.

Figure

12.6.2 Creating a table and showing all the tables in the test database.

Figure

12.6.3 Inserting, selecting, and updating a student.

Question set

12.6.4 Entering commands.

Aside

MySQL Workbench

Figure

12.6.4 Improperly formatted SELECT returns error code 1064, and INSERT with duplicate student ID returns error code 1062.

Question set

12.6.5 Common MySQL error codes.

12.7 mysql module (Node)

mysql module

The mysql module allows Node.js applications to interface with a MySQL database.

mysql.createConnection()

The mysql.createConnection() method creates a connection object using an object parameter that specifies the hostname of the computer running MySQL, the developer's username and password, and the database to select.

connection object

The connection object provides methods to interact with the MySQL server.

connect()

The connection object's connect() method attempts to establish the database connection.

query()

SQL statements can be executed using the connection object's query() method.

auto-increment field

An auto-increment field is a database field that increments by one each time a row is inserted into the table.

SQL injection attack

A SQL injection attack is when a malicious user enters input into a web application that alters the intent of a SQL statement. SQL injection attacks can lead to sensitive data being divulged or deleted.

Figure

12.7.1 Connecting to a MySQL server running on localhost.

Question set

12.7.1 Connecting to MySQL server.

Animation

12.7.2 Selecting rows from the students table.

Question set

12.7.3 SELECT query.

Aside

Connection pool

 **Animation**

12.7.4 Values substituted into SELECT query.

 **Figure**

12.7.2 Express route displaying students with GPAs between two values obtained from the query string.

 **Question set**

12.7.5 Select with values array.

 **Figure**

12.7.3 .query() uses INSERT with map.

 **Figure**

12.7.4 CREATE with AUTO_INCREMENT attribute creates unique student IDs for INSERT.

 **Figure**

12.7.5 Express route that inserts a new student into the students table and displays the student's ID.

 **Question set**

12.7.6 INSERT statement.

 **Figure**

12.7.6 .query() method executing an UPDATE statement.

 **Figure**

12.7.7 .query() method executing a DELETE statement.

 **Question set**

12.7.7 UPDATE and DELETE statements.

 **Aside**

SQL injection

12.8 MongoDB

MongoDB

MongoDB is the most popular NoSQL database used by Node.js developers. MongoDB stores data objects as documents inside a collection.

document

A document is a single data object in a MongoDB database that is composed of field/value pairs, similar to JSON property/value pairs.

collection

A collection is a group of related documents in a MongoDB database.

BSON document

A BSON document (Binary JSON) is a binary representation of JSON with additional type information.

MongoDB shell

The MongoDB shell is a command-line interface for creating and deleting documents, querying, creating user accounts, and performing many other operations in MongoDB. The `mongo` command starts the MongoDB shell.

primary key

A primary key is a field that uniquely identifies each document in a collection.

ObjectId

An ObjectId is a 12-byte BSON type that contains a unique value.

CRUD

CRUD stands for Create, Read/Retrieve, Update, and Delete. CRUD operations are the most common operations performed on stored data.

Figure

12.8.1 Example student document and collection.

Question set

12.8.1 MongoDB concepts.

Aside

Installing MongoDB

Figure

12.8.2 Creating a new student and finding all students.

Question set

12.8.2 MongoDB shell commands.

Figure

12.8.3 Inserting multiple students in bulk.

Question set

12.8.3 MongoDB `_id` field.

Figure

12.8.4 Find 'Sue' and students with GPA \geq 3.0.

Table

12.8.1 Common MongoDB query operators.

Question set

12.8.4 Querying the 'autos' collection.

Figure

12.8.5 Change Sue's GPA to 3.3, and set all students with GPA $>$ 3 to 1.

Figure

12.8.6 Remove all students with GPA $<$ 3.5 (Sue and Larry), and remove the first student with GPA $>$ 3.5 (Bob).

Question set

12.8.5 Updating and removing documents in the 'autos' collection.

12.9 Mongoose

mongodb module

A Node.js application using MongoDB may use the `mongodb` module to interact with a MongoDB database using many of the same functions supported in the mongo shell.

mongoose module

The `mongoose` module provides object data mapping (ODM) and structured schemas to MongoDB collections.

Object data mapping (ODM)

Object data mapping (ODM) is the conversion of data from a database into a JavaScript object.

schema

A schema defines the structure of documents within a MongoDB collection.

model

A model is a constructor compiled from a schema. Model instances represent MongoDB documents that can be saved to or retrieved from a MongoDB database.

Figure

12.9.1 Mongoose schema for a student.

Figure

12.9.2 Mongoose model for a 'students' collection.

Question set

12.9.1 Mongoose concepts.

Figure

12.9.3 Connecting to the 'mydb' database on MongoDB.

Animation

12.9.2 Mongoose saving a student to MongoDB.

Question set

12.9.3 Saving documents to MongoDB using Mongoose.

Table

12.9.1 Mongoose find methods.

Animation

12.9.4 Finding students by chaining Query methods.

Question set

12.9.5 Mongoose find methods.

Figure

12.9.4 Changing a student's GPA with save().

Table

12.9.2 Mongoose update methods.

Table

12.9.3 Mongoose remove methods.

Question set

12.9.6 Updating and removing documents with Mongoose.

Figure

12.9.5 Example Node.js project that adds new students entered in a form to a MongoDB database.

Question set

12.9.7 Layout of Node.js project.

12.10 Creating RESTful web APIs (Node)

client

A client is a program that sends web API requests.

REST (representational state transfer)

REST (representational state transfer) is an architectural style that describes how various components interact in a distributed hypermedia system.

RESTful web API

A RESTful web API is a web API that implements CRUD (create, read/retrieve, update, delete) operations on resources using the mechanics of HTTP.

resource

A resource is a data entity whose representation is accessible via a URL.

endpoint

An endpoint is a URL for a web API used to access a resource. Ex:

`http://yahoo.com/api/movies`.

Animation

12.10.1 Web application interacting with a web API to obtain song data.

Table

12.10.1 Example requests and responses for a Music API.

Question set

12.10.2 RESTful web APIs.

Aside

Web services and SOAP

Animation

12.10.3 Express router sends JSON-encoded song to web browser.

Figure

12.10.1 Example POST of JSON-encoded song to Music API.

Question set

12.10.4 Express router.

Aside

Developer tools for interacting with web APIs

Figure

12.10.2 Music API using Mongoose to access/save data in MongoDB.

Figure

12.10.3 Organizing routes and APIs in separate modules: Example Music API project.

 **Question set**

12.10.5 Web API project.

 **Figure**

12.10.4 GET request can limit results by genre using the query string.

 **Figure**

12.10.5 GET request to find specific song by ID.

 **Question set**

12.10.6 Web API GET parameters.

 **Figure**

12.10.6 Update song's title and popularity with PUT verb in Music API.

 **Figure**

12.10.7 Delete a song by ID with DELETE verb in Music API.

 **Question set**

12.10.7 Update and delete.

12.11 Using RESTful web APIs (Node)

 **Table**

12.11.1 Music API operations.

 **Animation**

12.11.1 Using `$.get()` to request all songs from the Music API.

 **Question set**

12.11.2 GET requests to the Music API.

 **Figure**

12.11.1 jQuery `$.ajax()` method POSTing a new song.

 **Question set**

12.11.3 POST requests to the Music API.

 **Figure**

12.11.2 jQuery `$.ajax()` method sending PUT request to update an existing song.

 **Figure**

12.11.3 jQuery `$.ajax()` method sending DELETE request to remove a song.

 **Question set**

12.11.4 PUT and DELETE requests to the Music API.

 **Question set**

12.11.5 Keeping the UI updated.

12.12 Third-party web APIs (Node)

third-party web API

A third-party web API is a public web API that is used by a web application to perform some operation.

Cross-Origin Resource Sharing

Cross-Origin Resource Sharing (CORS) is a W3C specification for how web browsers and web servers should communicate when making cross-origin requests.

CORS

Cross-Origin Resource Sharing (CORS) is a W3C specification for how web browsers and web servers should communicate when making cross-origin requests.

JSON with Padding

JSON with Padding (JSONP) is a technique to circumvent cross-origin restrictions by injecting `<script>` elements dynamically into a web page. Script elements have no cross-origin restrictions.

JSONP

JSON with Padding (JSONP) is a technique to circumvent cross-origin restrictions by injecting `<script>` elements dynamically into a web page. Script elements have no cross-origin restrictions.

 **Figure**

12.12.1 Calling third-party web API from the web browser or web server.

 **Question set**

12.12.1 Third-party web APIs.

 **Figure**

12.12.2 GET request to obtain the current weather for ZIP 90210.

 **Try**

12.12.1 Try OpenWeatherMap's API in your web browser.

 **Question set**

12.12.2 The Weather API.

 **Animation**

12.12.3 Calling the Weather API from an Express server with the 'request' module.

Question set

12.12.4 Calling the Weather API from the web server.

Figure

12.12.3 Making a request to the Weather API with CORS.

Question set

12.12.5 Calling a third-party web API with CORS.

Learning tool

12.12.6 Calling the Weather API from the web browser using CORS.

Animation

12.12.7 Calling the Weather API using JSONP.

Question set

12.12.8 Calling the Weather API from the web browser.

12.13 Token-based user authentication (Node)

User authentication

User authentication is the process of verifying that a user is who the user claims to be.

Token based authentication

Token based authentication is a technique where the client uses a signed token to "prove" to a server that the client has successfully authenticated.

signed token

A signed token is a string of characters produced with a secret key that uniquely identifies the entity that created the token.

JSON Web Token (JWT)

A JSON Web Token (JWT), pronounced "jot", is a string that encodes JSON data and is signed with a secret key to ensure the data in a JWT is unaltered.

Base64 encoding

Base64 encoding is a technique that converts data into 64 printable characters.

Animation

12.13.1 Token based authentication process.

Question set

12.13.2 Steps in token based authentication.

 **Animation**

12.13.3 Creating a JWT.

 **Question set**

12.13.4 JSON Web Tokens.

 **Figure**

12.13.1 Encoding and decoding a JWT with the jwt-simple module.

 **Figure**

12.13.2 Using jwt-simple in an Express server.

 **Question set**

12.13.5 JWTs in an Express server.

 **Aside**

Storing the secret key.

 **Figure**

12.13.3 Node.js project uses token based authentication with user data in a MongoDB database.

 **Question set**

12.13.6 Token based authentication with a MongoDB database.

 **Aside**

Saving passwords in the database

 **Animation**

12.13.7 Authenticating with Ajax and storing the JWT in localStorage.

 **Question set**

12.13.8 Authenticating with Ajax.

 **Learning tool**

12.13.9 User authentication with Ajax.

 **Aside**

Authentication protocols with no passwords

12.14 Password hashing (Node)

cryptographic hash function

A cryptographic hash function is a mathematical algorithm that converts text of any length into a fixed-length sequence of characters called the hash digest or "hash".

hash digest

A cryptographic hash function is a mathematical algorithm that converts text of any length into a fixed-length sequence of characters called the hash digest or "hash".

Password cracking

Password cracking is the process of recovering passwords from data, like the database of a compromised website.

dictionary attack

A dictionary attack is a popular password-cracking strategy where the attacker feeds a number of possible passwords, such as words from a dictionary, into a hash function and compares the stolen hashes to the generated hashes. Passwords are revealed for any matching hashes.

salt

A salt is a random string that is combined with a password so two identical passwords produce different hashes.

strong password

A strong password is a password that is difficult to guess.

Figure

12.14.1 Hashing various passwords using the MD5 cryptographic hash function.

Try

12.14.1 Try generating your own MD5 hashes.

Question set

12.14.1 Cryptographic hash functions.

Animation

12.14.2 Verifying passwords by comparing password hashes.

Figure

12.14.2 Using salt to protect against precomputed table attacks.

Figure

12.14.3 Secure password check indicates 'qwerty123!' is a poor password.

Question set

12.14.3 Password hashes.

Aside

2012 LinkedIn data breach

Figure

12.14.4 Generating a hash and comparing hashes with the bcrypt-nodejs module.

Question set

12.14.4 Password hashing with bcrypt.

Figure

12.14.5 Node.js project uses token based authentication and password hashing with bcrypt-nodejs.

Question set

12.14.5 Securing users' passwords.

13. IN-PROGRESS Wk 10 Prep: Relational Databases and SQL (56)

13.1 Relational databases

relational database

A relational database stores data about entities.

entity

An entity is any object that an application wants to store information about.

table

A table is a collection of related entities that is organized into columns and rows, similar to a spreadsheet.

column

A column is a set of values of the same type.

row

A row is a set of values, one for each column.

primary key

Every table must have a primary key, a column or combination of columns that uniquely identifies each row.

foreign key

Two tables that have a relationship are linked together using a foreign key, a primary key from one table that is shared in another table.

relationship

A relationship describes an association between two entities.

one-to-one relationship

A one-to-one relationship (1:1) means that entity A is associated with only one entity B and vice versa.

1:1

A one-to-one relationship (1:1) means that entity A is associated with only one entity B and vice versa.

one-to-many relationship

A one-to-many relationship (1:M) means that entity A is associated with one or more B entities, and one B entity is associated with only one A entity.

1:M

A one-to-many relationship (1:M) means that entity A is associated with one or more B entities, and one B entity is associated with only one A entity.

many-to-many relationship

A many-to-many relationship (M:N) means that one A entity is associated with many B entities, and one B entity is associated with many A entities.

M:N

A many-to-many relationship (M:N) means that one A entity is associated with many B entities, and one B entity is associated with many A entities.

linking table

A linking table links two tables together by creating a primary key that is composed of the foreign keys from the two linked entities.

Animation

13.1.1 Students table in a University database.

Question set

13.1.2 Relational database.

Question set

13.1.3 Column data types.

Animation

13.1.4 course and class tables linked with course_id.

Question set

13.1.5 Primary and foreign keys.

Aside

Entity relationship diagrams

 **Animation**

13.1.6 Creating a linking table.

 **Question set**

13.1.7 Relationships.

13.2 Structured Query Language (SQL)

relational database management system

A relational database management system (RDBMS) is software that implements a relational database.

RDBMS

A relational database management system (RDBMS) is software that implements a relational database.

Structured Query Language

Structured Query Language (SQL) is a language used by an RDBMS to create, query, modify, and delete databases, tables, and table data.

SQL

Structured Query Language (SQL) is a language used by an RDBMS to create, query, modify, and delete databases, tables, and table data.

statement

A SQL statement is a complete command composed of one or more clauses.

clause

A clause groups SQL keywords with table names, column names, conditions, etc.

CRUD

The acronym CRUD names the four common operations that are performed on data: Create, Retrieve, Update, and Delete.

CREATE DATABASE

A database is created with the CREATE DATABASE statement.

DROP DATABASE

The DROP DATABASE statement deletes a database.

 **Animation**

13.2.1 SQL interaction with a RDBMS.

 **Question set**

13.2.2 RDBMS and SQL.

 **Animation**

13.2.3 Three clauses in a SELECT statement.

 **Table**

13.2.1 Summary of SQL syntax features.

 **Question set**

13.2.4 SQL syntax.

 **Question set**

13.2.5 SQL statements for tables.

 **Question set**

13.2.6 SQL statements for table data.

 **Aside**

SQL Fiddle

13.3 Creating, altering, and deleting tables

CREATE TABLE

The CREATE TABLE statement creates a new table by specifying the table name, column names, column data types, and the primary key.

constraint

A constraint is a rule that applies to table data.

PRIMARY KEY

The PRIMARY KEY constraint is used in the CREATE TABLE statement to name the column(s) that uniquely identify each row.

NULL

A NULL value is a special value that can mean "unknown" or "nothing".

NOT NULL

The NOT NULL constraint, listed by a column name in a CREATE TABLE statement, prevents a column from having a NULL value.

FOREIGN KEY

A table is created with a foreign key using the FOREIGN KEY constraint.

referential integrity

Foreign keys help a database maintain referential integrity, which means that if a foreign key contains a value, the value is guaranteed to exist in another table.

auto-increment column

An auto-increment column is a column that is assigned an automatically incrementing value.

AUTO_INCREMENT

MySQL uses the AUTO_INCREMENT keyword to define an auto-increment column.

ALTER TABLE

After creating a table, the table can be altered by adding or removing columns or changing a column's data type by using the ALTER TABLE statement with the appropriate clause.

DROP TABLE

The DROP TABLE statement deletes a table along with all the table's rows.

Figure

13.3.1 CREATE TABLE syntax.

Table

13.3.1 Common data types in MySQL.

Animation

13.3.1 Creating a student table.

Question set

13.3.2 Create a faculty table.

Table

13.3.2 NULL values in the student table.

Figure

13.3.2 NOT NULL constraint: First name cannot be NULL.

Question set

13.3.3 NULL value.

Figure

13.3.3 faculty table with foreign key.

Question set

13.3.4 Foreign keys.

Figure

13.3.4 stu_id is an auto-incrementing column.

Question set

13.3.5 Auto-increment column.

 **Figure**

13.3.5 ALTER TABLE syntax.

 **Figure**

13.3.6 DROP TABLE syntax.

 **Question set**

13.3.6 Altering and deleting tables.

13.4 Inserting rows

INSERT

An INSERT statement inserts one or more rows into a table.

 **Figure**

13.4.1 INSERT syntax.

 **Animation**

13.4.1 Inserting 3 rows.

 **Question set**

13.4.2 INSERT statement.

 **Question set**

13.4.3 Identify the syntax errors.

 **Figure**

13.4.2 No value for auto-increment column stu_id.

 **Question set**

13.4.4 Insert faculty members.

13.5 Selecting rows

SELECT

The SELECT statement retrieves data from a table.

index

An index speeds-up a RDBMS's ability to retrieve data.

WHERE

The WHERE clause works like an if statement in a programming language, specifying conditions that must be true for a row to be selected.

LIKE

The LIKE operator matches text against a pattern using the two wildcard characters % and _.

BINARY

The LIKE operator performs case-insensitive pattern matching by default or case-sensitive pattern matching if followed by the BINARY keyword.

ORDER BY

The ORDER BY clause sorts the selected rows in ascending order. The DESC keyword with the ORDER BY clause sorts the rows in descending order.

DESC

The ORDER BY clause sorts the selected rows in ascending order. The DESC keyword with the ORDER BY clause sorts the rows in descending order.

Figure

13.5.1 SELECT syntax.

Animation

13.5.1 Selecting from the student table.

Aside

Indexes

Question set

13.5.2 SELECT statement.

Table

13.5.1 WHERE condition operators.

Animation

13.5.3 Using the WHERE clause.

Question set

13.5.4 WHERE clause.

Aside

Regular expressions

Question set

13.5.5 LIKE operator.

Figure

13.5.2 Sorting selected results by columns.

Question set

13.5.6 ORDER BY clause.

13.6 SQL functions

aggregate function

An aggregate function is a function that works on a group of values.

COUNT()

COUNT() - Counts the number of rows retrieved by a SELECT statement.

MIN()

MIN() - Finds the minimum value in a group.

MAX()

MAX() - Finds the maximum value in a group.

SUM()

SUM() - Sums all the values in a group.

AVG()

AVG() - Finds the arithmetic mean (average) of all the values in a group.

GROUP BY

Aggregate functions may be used with the GROUP BY clause, which groups results using one or more columns.

HAVING

The HAVING clause restricts the selected criteria from a GROUP BY clause.

Animation

13.6.1 Using aggregate functions in a SELECT statement.

Animation

13.6.2 Using GROUP BY with aggregate functions.

Aside

HAVING clause

Question set

13.6.3 Aggregate functions and the GROUP BY clause.

Table

13.6.1 Arithmetic operators.

Table

13.6.2 Common numeric functions.

Question set

13.6.4 Numeric functions.

Table

13.6.3 Common string functions.

Question set

13.6.5 String functions.

Table

13.6.4 Common date and time functions.

Question set

13.6.6 Date and time functions.

13.7 Joining tables

JOIN

A SQL JOIN joins information from two or more tables in a SELECT statement.

INNER JOIN

An INNER JOIN clause performs an inner join, joining all rows from the left and right tables where the join condition is met.

LEFT OUTER JOIN

LEFT OUTER JOIN clause - Joins all rows from the left table along with rows from the right table where the join condition is met.

RIGHT OUTER JOIN

RIGHT OUTER JOIN clause - Joins all rows from the right table along with rows from the left table where the join condition is met.

FULL OUTER JOIN

FULL OUTER JOIN clause - Joins all rows from the right table regardless if the join condition is met.

full join

An outer join is therefore sometimes called a full join or full outer join.

full outer join

An outer join is therefore sometimes called a full join or full outer join.

Figure

13.7.1 Venn diagrams showing which rows are joined from the left table A and right table B in each join type.

 **Question set**

13.7.1 Join types.

 **Figure**

13.7.2 JOIN syntax.

 **Animation**

13.7.2 Inner join of the student and grade tables.

 **Question set**

13.7.3 Inner join.

 **Aside**

RDBMS support for outer join

 **Animation**

13.7.4 Left vs. right join.

 **Figure**

13.7.3 Results of a full outer join.

 **Question set**

13.7.5 Joining student and grade tables.

 **Question set**

13.7.6 Joining department and faculty tables.

13.8 Updating and deleting rows

UPDATE

An UPDATE statement changes the values of existing rows in a table.

DELETE

The DELETE statement removes rows from a table.

 **Figure**

13.8.1 UPDATE syntax.

 **Animation**

13.8.1 UPDATE examples.

 **Question set**

13.8.2 UPDATE statement.

 **Figure**

13.8.2 Updating multiple rows.

Question set

13.8.3 UPDATE multiple rows.

Figure

13.8.3 DELETE syntax.

Animation

13.8.4 DELETE examples.

Question set

13.8.5 DELETE statement.

14. IN-PROGRESS Optional: jQuery (65)

14.1 Getting started

library

A library is a collection of functions that focus on a related set of tasks.

jQuery

JQuery is a JavaScript library that focuses on a broad range of tasks, many of them associated with the visual elements of a web page.

framework

A framework is a suite of libraries designed to offer a more comprehensive platform in which to program.

Content Delivery Network (CDN)

A Content Delivery Network (CDN) hosts popular web files around the globe and automatically routes requests to the closest server, thus speeding up the delivery of the files.

Subresource Integrity

The integrity and crossorigin attributes are used for Subresource Integrity (STI) checking, which allows web browsers to verify resources hosted on third-party servers have not been altered.

STI

The integrity and crossorigin attributes are used for Subresource Integrity (STI) checking, which allows web browsers to verify resources hosted on third-party servers have not been altered.

jQuery()

The jQuery library defines a primary function called `jQuery()`.

\$()

The `$()` function is the same as the `jQuery()` function, which developers often use to type less code.

▀▀▀ Aside

Library vs. framework

▀▀▀ Table

14.1.1 Common tasks performed by jQuery.

███ Question set

14.1.1 jQuery tasks.

▀▀▀ Figure

14.1.1 Downloading jQuery library from the local web server.

▀▀▀ Figure

14.1.2 Downloading jQuery library from the code.jquery.com CDN.

▀▀▀ Aside

Minification

▀▀▀ Figure

14.1.3 Single server distribution vs. CDN distribution.

███ Question set

14.1.2 Loading the jQuery library.

███ Animation

14.1.3 Creating jQuery objects from DOM nodes.

▀▀▀ Figure

14.1.4 Using the `$()` function.

███ Question set

14.1.4 The `jQuery()` and `$()` functions.

14.2 Selectors

selector

A selector is a string that is crafted to match specific DOM elements.

addClass()

The jQuery method `addClass()` is used to add a CSS class to selected elements.

removeClass()

The jQuery method `removeClass()`, which removes a class from a selected element.

Attribute selector

Attribute selector - Selects elements based on an element attribute.

Basic filter selector

Basic filter selector - Selects elements based on a variety of properties.

Child filter selector

Child filter selector - Selects child elements based on location or other properties.

Content filter selector

Content filter selector - Selects elements based on an element's contents.

Hierarchy selector

Hierarchy selector - Selects elements based on an element's location within the DOM hierarchy.

Animation

14.2.1 Adding classes to all paragraphs.

Example

14.2.1 Using `$()` and `removeClass()` on a single line.

Table

14.2.1 Basic jQuery selectors.

Question set

14.2.2 Using basic selectors.

Table

14.2.2 Additional jQuery selectors.

Question set

14.2.3 Using additional selectors.

Learning tool

14.2.4 Selector practice.

Progression

14.2.1 jQuery selectors.

14.3 Events

callback function

A callback function is a function that is executed when an event occurs. A callback function is also called an event listener or event handler.

event listener

A callback function is a function that is executed when an event occurs. A callback function is also called an event listener or event handler.

event handler

A callback function is a function that is executed when an event occurs. A callback function is also called an event listener or event handler.

Figure

14.3.1 Registering callback functions in JavaScript and jQuery.

Table

14.3.1 Mouse events and shortcut methods.

Question set

14.3.1 Mouse events.

Figure

14.3.2 Waiting for the ready event.

Figure

14.3.3 Registering a ready event callback function with \$().

Question set

14.3.2 The ready event.

Learning tool

14.3.3 Mouse event practice.

Table

14.3.2 Keyboard events and shortcuts.

Table

14.3.3 Form events and shortcuts.

Question set

14.3.4 Keyboard and form events.

Learning tool

14.3.5 Keyboard event practice.

 **Progression**

14.3.1 jQuery events.

14.4 Styles and animation

css()

CSS properties can be added to selected elements with the css() method.

jQuery effects

JQuery animation methods are also called jQuery effects.

queue()

JQuery provides a queue() method to aid in queuing code that should be executed after the previous animations complete.

animate()

JQuery provides the animate() method that can animate any CSS property with a numeric value. Ex: Width, position, and opacity.

easing function

An easing function is a function that determines the speed at which an animation progresses.

 **Animation**

14.4.1 Using the .css() method.

 **Question set**

14.4.2 Using the .css() method.

 **Table**

14.4.1 Animation methods.

 **Table**

14.4.2 The speed argument for animation methods.

 **Learning tool**

14.4.3 Animation methods.

 **Question set**

14.4.4 jQuery effects.

 **Animation**

14.4.5 Waiting for an animation to complete.

 **Figure**

14.4.1 Queuing animations on the same element.

Figure

14.4.2 Only animation methods are queued.

Figure

14.4.3 Using the .queue() method.

Question set

14.4.6 Animation order.

Animation

14.4.7 The .animate() method.

Question set

14.4.8 The .animate() method.

Learning tool

14.4.9 Practice with animate() and queue().

14.5 DOM manipulation

attr()

The attr() method gets and sets attribute values of a DOM element.

remove()

The jQuery methods remove() and detach() remove DOM nodes.

detach()

The jQuery methods remove() and detach() remove DOM nodes.

html()

JQuery has two methods for getting and setting the HTML or textual content in a web page: html() and text().

text()

JQuery has two methods for getting and setting the HTML or textual content in a web page: html() and text().

Figure

14.5.1 Changing image attributes with attr().

Question set

14.5.1 jQuery attr() method.

Table

14.5.1 Methods for adding DOM nodes.

 **Question set**

14.5.2 Adding to the DOM.

 **Table**

14.5.2 Methods for removing DOM nodes.

 **Table**

14.5.3 Methods for modifying DOM text.

 **Question set**

14.5.3 Altering the DOM.

 **Learning tool**

14.5.4 DOM manipulation practice.

 **Progression**

14.5.1 jQuery DOM manipulation.

14.6 Ajax

load()

The `load()` method is used for asynchronously loading a short snippet of HTML that is inserted into a web page.

\$.get()

The jQuery methods `$.get()` and `$.post()` are general-purpose functions for making asynchronous HTTP GET and POST requests to the web server.

\$.post()

The jQuery methods `$.get()` and `$.post()` are general-purpose functions for making asynchronous HTTP GET and POST requests to the web server.

\$.ajax()

The `$.ajax()` method is a general purpose method for making Ajax requests that can perform the same actions as `$.get()` and `$.post()`.

jqXHR

The `$.ajax()`, `$.get()`, and `$.post()` methods all return an `jqXHR` object, which is an abbreviation for jQuery XMLHttpRequest object. The `jqXHR` object adds additional capabilities to the XMLHttpRequest object the browser uses to make Ajax requests.

done()

The `jqXHR` object has a callback method `done()` that is called when a successful HTTP response is received.

cross-origin HTTP request

A cross-origin HTTP request is a request made to another domain.

fail()

The jqXHR object has a callback method fail() that is called when a 40x or 500 response code is received from the Ajax request.

Animation

14.6.1 Using .load() to load content asynchronously.

Question set

14.6.2 The .load() method.

Figure

14.6.1 Server-side program interacting with web browser and database.

Aside

XML vs. JSON

Animation

14.6.3 Using .get() to load JSON asynchronously.

Figure

14.6.2 Example call to \$.ajax().

Aside

Promises

Question set

14.6.4 Sending Ajax requests.

Aside

Cross-origin HTTP request

Question set

14.6.5 Robust Ajax code.

Learning tool

14.6.6 Ajax practice.

Aside

The user interface and Ajax

Progression

14.6.1 jQuery Ajax.

14.7 Plugins

Question set

14.7.1 Popular jQuery plugins.

Learning tool

14.7.2 Cycle2 plugin.

Learning tool

14.7.3 jQuery UI interactions.

Learning tool

14.7.4 jQuery UI animations.

Learning tool

14.7.5 jQuery UI widgets.

Question set

14.7.6 More about jQuery UI.

14.8 Example: Weather Comparison with jQuery

Animation

14.8.1 Using the OpenWeatherMap web API to retrieve a weather forecast.

Question set

14.8.2 Weather Comparison HTML and CSS.

Figure

14.8.1 Event callback functions.

Question set

14.8.3 Handling events.

Figure

14.8.2 getWeatherForecast() function.

Question set

14.8.4 Requesting the forecast.

14.9 LAB: Currency Conversion (jQuery)

Lab activity

14.9.1 LAB: Currency Conversion (jQuery)

14.10 LAB: Quote web API (jQuery)

Lab activity

14.10.1 LAB: Quote web API (jQuery)

14.11 LAB: Memory game animations (jQuery)

 **Lab activity**

14.11.1 LAB: Memory game animations (jQuery)

14.12 LAB: To-do list (jQuery)

 **Lab activity**

14.12.1 LAB: To-do list (jQuery)

15. Scratch Space

15.1 Sandbox

 **Figure**

15.1.1 select None as the "Code block" type and then paste embedded video