Monday

Tuesday

Higher order functions - functions that are passed as parameters to other functions.

For of loop - ES6 syntax - New syntax for working with arrays and strings. Example:

for (var grade of grades) { // assuming grades is an array or string

if (grade >= 70) {

result.push(grade);

}

}

... - ES6 syntax - The spread operator is used to break an array into its elements to pass as arguments. Works like Math.max.apply() .apply takes an array as an argument and breaks/passes to max.

Object.assign({}, obj1, obj2); // Merges all objects together into an object passed as first argument.

To update apps with brew

brew update - to get latest brews

brew outdated - to see outdated programs

brew upgrade appName - to upgrade a version of appName

brew cleanup

Surge: Make the working directory the directory to surge

add a CNAME file with the domain name you want to use -

typically chadlatham-assignmentName.surge.sh

add a link to the folder with https:// prepended to the domain name in the readme.md file

also add the domain name in the readme.md file

do a git commit and git push

surge and accept defaults

go to git and copy link. Place link into workbook exercises to give Ryan the link to the surge page.

HTML

<small> commonly used for legal notices.

CSS

width & height do not work on inline level elements.

http://www.fillmurray.com/300/400

DOM

document

document.documentElement - returns the HTML

document.childNodes - returns an array like object. Position 0 is doctype, 1 is HTML (childNodes are present at each level of the DOM)

childNodes.length - get the length of childNodes

node.firstChild or node.lastChild - Get the first or last child under a node (not a property of childNodes)

node.firstElementChild - to get first element rather than first node.

node.nodeName - the tag or node name ("#text") in the case of the text.

node.tagName - gets the tag name or undefined in the case of text node.

use .textContent to get the text from a tag. NOT .innerText - deprecated.

element.getAttribute & element.setAttribute to get and set attributes.

elem.className - a property to assign classname - only accepts a string with multiple classes separated by a space.

.getElementById - gets an element by its id attribute. (#1 tool - fastest)

.getElementsByTagName

.getElementsByClassName

.parentElement - gets the parent element from any current node

#ID's are used very commonly to select with JavaScript - not for CSS

QuerySelectors are a little slower than getElement(s)By methods

.querySelector('same as CSS') - Used to get the first element by css selector

.querySelectorAll('') - Gets more than 1 element by css selector

node.style to get the css styles applied - console.log it to see them all!!!!

.createElement('tagName') - to create an orphan element

element.appendChild(element) - to append object to the end of the element (on the 2nd call to appendChild, it moves it from original location)

element.parentElement.removeChild(element) - to remove an element.

element.remove() - to remove an element directly.

node.replaceChild(newElement, oldElement);

parentNode.insertBefore(newNode, referenceNode);

The word Element refers to an HTML tag. A node refers to either a tag or its text.

Thursday

DOM Events

monitorEvents(window) - to monitor all window level events

mouse events

keyboard events

focus/blur events

Use function expressions (var x = function() {};) because it is more explicit

addEventListener('eventName', eventHandlerFn) - Every DOM element has its own - Use non-anonymous functions

removeEventListener('eventName', eventHandlerFn) - to remove

Event handler functions take a parameter of the event object. Event.target gets the object that triggered the event - also keyword this is similar. See below.

this keyword in an event handler evaluates to the object where the event handler was placed. The difference between this and event.target is that the this keyword will always point at the object that has the event listener. The event.target will point at the specific child inside the monitored element that triggered the event or the element itself in the case that there are no children.

debugger; - a breakpoint in JavaScript code.

document.addEventListener("DOMContentLoaded", callBack) - this event is what is fired once the DOM is built.

window.onload (simply assign to the property) event fires after DOMContentLoaded and relates to all asynchronous calls completed.

Friday

http://www.king5.com/news/local/shortened-sentence-gives-man-early-second-chance/185701325

Review

Mon

Semantic HTML: Giving your document meaning through structure tag names.

display: block - takes up entire horizontal width of container, inline - only takes up space of the content of the element; inline will not accept height or width, inline-block - a mixture of the two that accepts width and height, only takes up content space, and doesn't force a new line.

table: table, thead, tbody, tfoot, tr, th, and td tags - for displaying tabular data.

html boilerplate: !doctype, html, head, meta, and body tags.

Tue

CSS: Cascading Style Sheets - Separation of concerns. Never use inline styles (except for formatting emails which also use tables for layout - lame!) Uses rule declaration to define styling rules to apply. Called a declarative language.

Specificity: lowest to highest: tag, class, id. They are additive, so a selector with 2 classes is more specific than with one. Selectors have specificity.

Selector: It's what selects the element in the HTML. Typically, don't use id as a selector, but consider the option for nested selection because it's the fastest way to select inside a group. ID the container then use a tag inside. The space operator lets you look at all children, "." operator is for class, "#" is for ID, ":" for pseudo selectors, ">" for direct children, "\*" is for all and tends to be slow.

Pseudo-selectors: Pseudo selectors are like functions that allow criteria to be applied to standard selectors. :hover, :before, :after, :first-child, :nth-child, :nth-of-type, :visited, :active, etc...

Positioning: top, right, bottom, left, static, relative, absolute, fixed.

Static - In normal document flow. Used to reset non-static positioning.

Relative - Allows position to be adjusted relative to the normal position in document flow (enables top, bottom, left, and right).

Absolute - Positioned relative to it's nearest non-static parent

Fixed - Fixed to the viewport - even with scroll.

Box-model: width and height apply to content, then padding, then border, then margin. Box-sizing: border-box, content-box, padding-box - resets or normalize may affect box-sizing.

Reset vs. Normalizer: Reset - Adjusts all settings to be the same (usually 0). Normalizer sets defaults to be same on all browsers.

Colors: hex, rgb(r,g,b), rgba(r,g,b,a), named colors, HSLA

Fonts: Used to keep user attention and for good looking pages.

Media Queries: Use to make styling changes based on viewport dimensions.

Units: pixels, percents, em, rem, vw & vh (viewport width and height).

Wed

DOM - Document Object Model - Built at refresh of each page. Consists of nodes.

Node properties: .style, .childNodes, .textContent, .value, .tagName, .className, .parentElement, .parentNode, .firstChild, .nextSibling, .lastChild, .

Node methods: getAttribute(), setAttribute(), getElementByID(), getElementsByTagName(), getElementsByClassName(), querySelector(), querySelectorAll(), appendChild(), remove(), removeChild(), replaceChild(), insertBefore(), createElement()

Browser - Contains DOM and rendering engine which repaints screen 60xP/S

Arrays (or nodeLists) are live - Don't modify array in a loop. Inspect first and track changes.

Thu

Events: Triggered by interpreter when a user interaction or significant point in the application execution is reached.

Mouse events: mouseup, mousedown, click, dblclick, mousemove, mouseleave, mouseover

Keyboard events: keydown, keyup, keypress

Scroll events: scrollup, scrolldown

Blur/Focus events:

Add an event by using node.addEventListener('eventName', callBackFn)

Remove an event by using node.removeEventListener('eventName', originalFnRef)

Default tab index is based on the DOM tree. node.tabindex is to override.

Callback: a function used to pass into another function. It's code to run later. A callback is never invoked right away.

Higher-order function: A function that takes another function as an argument.

this: For events, this is a pointer to the object with the event listener

event.Target: a pointer to the descendent of this that was actually clicked on.

capturing vs. bubbling: capturing happens first, bubbling next. Capturing comes from the top down in the DOM tree. Bubbling goes up from the leaf to trunk.

event.currentTarget: a pointer to the currently bubbling object

Event object: .target, .currentTarget,

document.addEventListener('DOMContentLoaded', fn) to executes after DOM is created.