

CSE 154: Web Programming

Exam “Cheat Sheet”

HTML

Tags Used in the <head> Section

Tag	Description
<code><title> text </title></code>	title shown on page tab
<code><meta attribute="value" ... /></code>	page metadata
<code><link href="url" rel="stylesheet" /></code>	links to a CSS style sheet
<code><script src="url"></script></code>	link to JavaScript code
<code><!-- comments --></code>	comment (can appear in head or body)

Tags Used in the <body> Section

Tag	Display	Description
<code><p>text </p></code>	Block	paragraph
<code><h1>text </h1></code> <code><h2>text </h2></code> ... <code><h6>text </h6></code>	Block	(h1 for largest to h6 for smallest)
<code><hr /></code>	Block	horizontal rule (line)
<code>
</code>	Inline	line break
<code>text </code>	Block	anchor (link)
<code></code>	Inline-block	image
<code>text</code>	Inline	emphasis (italic)
<code>text </code>	Inline	strong emphasis (bold)
<code></code> <code>text </code> <code>text </code> <code></code> <code></code> <code>nested item text</code> <code>nested item text</code> <code></code> <code></code> <code></code>	Block	ordered (<code>ol</code>) and unordered (<code>ul</code>) list; list item (<code>li</code>)

Tags Used in the <body> Section (Continued)

Tag	Display	Description
<code><dl></code> <code><dt>term 1 </dt></code> <code><dd>description 1 </dd></code> <code><dt>term 2 </dt></code> <code><dd>description 2 </dd></code> <code></dl></code>	Block	definition list (<code>dl</code>); term (<code>dt</code>), and its description (<code>dd</code>)
<code><blockquote></code> <code><p>text </p> ...</code> <code></blockquote></code>	Block	block-level quotation
<code><q>text </q></code>	Inline	inline-level quotation
<code><code>text </code></code>	Inline	computer code (monospace)
<code><pre>text </pre></code>	Inline	pre-formatted text (preserves whitespace)
<code><table></code> <code><caption>text </caption></code> <code><tr></code> <code><th>heading 1 </th></code> <code><th>heading 2 </th></code> <code></tr></code> <code><tr></code> <code><td> cell 1 </td></code> <code><td> cell 2 </td></code> <code></tr></code> ... <code></table></code>	Block	table of data (<code>table</code>) description of table (<code>caption</code>) table row (<code>tr</code>) table heading cell (<code>th</code>) normal table cell (<code>td</code>)
<code><div> ... </div></code>	Block	block-level section of a page
<code> ... </code>	Inline	inline-level section of a page

HTML5 Semantic Grouping Tags (all block elements)

Tag	Description
<code><header></code>	Container for a header of a document
<code><main></code>	Specifies the main content of a document. The content inside should be unique to the document and not contain content that is repeated across pages (e.g., sidebars, nav links, search bars, etc.)
<code><footer></code>	Container for a footer of a document
<code><article></code>	A standalone piece of content (e.g., entire blog post including title, author, etc.)
<code><section></code>	A piece of content that is part of another (e.g., a chapter section of a reading)
<code><aside></code>	Defines some content aside from the content it is placed in (e.g., a sidebar in an article)
<code><nav></code>	Defines content in a navigation bar

HTML Input Tags

Tag	Display	Description
<code><button></code> content <code></button></code>	Inline	clickable button type can be submit, reset, button
<code><input type="type" name="name"></code> content <code></input></code>	Inline	form element input tag type can be text, number, checkbox, radio, file, etc.
<code><textarea rows="num" cols="num"></code> initial text <code></textarea></code>	Inline	multi-line text input box
<code><label>text </label></code>	Inline	clickable text label around a form control
<code><select></code> <code><option>text </option></code> <code><option></code> <code><optgroup label="text"></code> <code><option> text </option></code> <code><option> text </option></code> <code></optgroup></code> ... <code></select></code>	Inline	drop-down selection box (select); each option within the box (option); a labeled group of option (optgroup);
<code><fieldset></code> <code><legend> text </legend></code> content <code></fieldset></code>	Block	a grouped set of form fields with a legend

HTML Entities Reference

Result	Description	Entity Name
	non-breaking space	<code>&nbsp;</code>
<code><</code>	less than	<code>&lt;</code>
<code>@</code>	at symbol	<code>&commat;</code>
<code>></code>	greater than	<code>&gt;</code>
<code>&</code>	ampersand	<code>&amp;</code>
<code>©</code>	copyright	<code>&copy;</code>

CSS

For the following property and value tables, anything *emphasized* represents values that should be replaced with specific units (e.g., *length* should be replaced with a px, pt, or em for many properties, and *color* should be replaced with a valid color value such as a hex or rgb code).

A use of | refers to separation of possible values (where you cannot provide two of these possible values for one property) and [value value value] refers to a grouping of possible values that can optionally be used together (e.g., [*h-shadow v-shadow blur spread color*] for box-shadow).

Selector Types

Name	Description	Example(s)
Universal	Any element	<code>.foo * { font: 10pt Arial; }</code>
Element	Any element of a given type	<code>h1 { text-decoration: underline; }</code>
Grouping	Multiple elements of different types	<code>h1, h2, h3 { color: purple; }</code>
Class	Elements with the given class name	<code>.example { text-decoration: underline; }</code>
Id	Single element with the given id	<code>#example { text-decoration: overline; }</code>
Descendant	Elements that are children at any level of another specified element	<code>#example h1 { text-decoration: underline; }</code>
Child	Elements that are direct children of another specified element	<code>#example > p { font-weight: bold; }</code>
Attribute	Elements that have the specified attribute	<code>input[selected]</code> - inputs that have the selected attribute <code>input[name='test']</code> - inputs that have a name 'test'

Background Styles

Property	Values
<code>background-color</code>	<i>color</i> transparent
<code>background-image</code>	<i>url</i> none
<code>background-origin</code>	border-box padding-box content-box
<code>background-position</code>	top left top center top right center left center center center right bottom left bottom center bottom right [<i>x-% y-%</i>] [<i>x-pos y-pos</i>]
<code>background-size</code>	<i>length</i> % auto cover contain
<code>background-repeat</code>	repeat repeat-x repeat-y no-repeat
<code>background-attachment</code>	scroll fixed

Border Styles

Note: Replace '*' with any side of the border (top, right, left, bottom) for the desired effect.

Example style: 'border: 2px solid red' applies a solid red border with a width of 2px to all four sides of the element, while 'border-left: 2px solid red' only applies that border to the left border'.

Property	Values
border, border-* (shorthand)	border-width, border-*-width border-style, border-*-style border-color, border-*-color
border-width, border-*-width	thin medium thick length
border-style, border-*-style	none hidden dotted dashed solid double groove rigid inset outset
border-color, border-*-color	color
box-shadow	none inset [h-shadow v-shadow blur spread color]
border-radius	length

Font and Text Styles

Property	Values
font-style	normal italic oblique inherit
font-family	fontname
font-size	length %
font-weight	normal bold inherit
text-align	left right center justify
text-decoration	none [underline overline line-through blink]
text-shadow	none [color length]
text-indent	length %
text-transform	none capitalize uppercase lowercase
list-style-type	none asterisks box check diamond disc hyphen square decimal lower-roman upper-roman lower-alpha upper-alpha lower-greek upper-greek lower-latin upper-latin footnotes

Color Values

Value	Description
colorname	Standard name of color, such as red, blue, purple, etc.
rgb(redvalue, greenvalue, bluevalue)	Example: red = rgb(255, 0, 0) or red = rgb(100%, 0, 0)
#RRGGBB	Example: red = #FF0000

Box Model


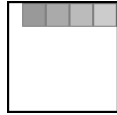

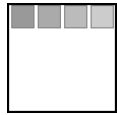
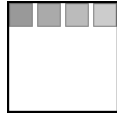
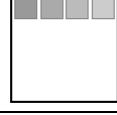
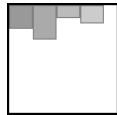
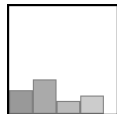
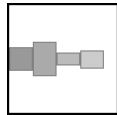
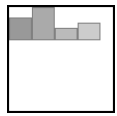
Property	Values
float	left right none
height, width	auto <i>length</i> %
min-height, max-height min-width, max-width	none <i>length</i> %
margin, margin-*	auto <i>length</i> %
padding, padding-*	<i>length</i> %
display	none inline block inline-block flex list-item compact table inline-table
overflow, overflow-x, overflow-y	visible hidden scroll auto no-display no-content
clear	left right both none

Flex Box

Property	Values	Element Type	Description
display	flex	Flex container	Sets all children to become 'flex-items'
flex-direction	row row-reverse column column-reverse	Flex container	Indicates if the container flows horizontally (row) or vertically (column)
flex-wrap	nowrap (default) wrap wrap-reverse	Flex container	Indicates whether flex items are forced onto one line (nowrap) or can wrap onto multiple lines (and which direction)

(Flex Box continued on next page)

(Flex Box continued from previous page)

Property	Values	Element Type	Description
justify-content	flex-start flex-end center space-around space-between space-evenly	Flex container	Indicates how to position the flex-items in the parent container along the main axis.      
align-items	stretch (default) flex-start flex-end center baseline	Flex container	Indicates how to space the items inside the container along the cross axis    
order	number	Flex item	Specifies the order in which the element appears in the flex container (by default, flex items are laid out in the source order)
align-self	flex-end flex-start center baseline stretch (default)	Flex item	Indicates where to place this specific item along the cross axis

JavaScript

window Methods and Properties

Method/Property	Description
<code>document</code>	Returns a reference to the document contained in the window
<code>getComputedStyle(element)</code>	Returns an object that reports the values of all CSS properties of an element after applying active stylesheets and resolving any basic computation those values may contain

document Methods and Properties

Method/Property	Description
<code>getElementById(id)</code>	Returns a DOM object whose id property matches the specified string. If no matches are found, null is returned.
<code>getElementsByName(name)</code>	Returns a collection of all elements which have all of the given name. If no matches are found, null is returned.
<code>querySelector(sel)</code>	Returns the first DOM element that matches the specified selector, or group of selectors. If no matches are found, null is returned.
<code>querySelectorAll(sel)</code>	Returns a list of the document's elements that match the specified group of selectors. If no matches are found, null is returned.
<code>createElement(elType)</code>	Creates and returns an Element node
<code>createTextNode(data)</code>	Creates and returns a Text node with the given data

DOM Element Methods and Properties

Method/Property	Description
<code>el.children</code>	Returns a collection of the child elements of <code>el</code>
<code>el.parentNode</code>	Returns the parent node of <code>el</code>
<code>el.appendChild(child)</code>	Adds a new child node to <code>el</code> as the last child node
<code>el.insertBefore(newNode, refNode)</code>	Adds <code>newNode</code> to parent <code>el</code> before <code>el</code> 's child <code>refNode</code> position
<code>el.addEventListener(event, fn)</code>	Attaches an event handler function <code>fn</code> to the specified element <code>el</code> to listen to <code>event</code>
<code>el.removeEventListener(event, fn)</code>	Removes the event handler <code>fn</code> to the specified <code>el</code> listening to <code>event</code>
<code>el.getAttribute(attr)</code>	Returns the specified attribute value <code>attr</code> of <code>el</code>
<code>el.removeChild(child)</code>	Removes a child node from an element
<code>el.innerHTML</code>	Sets or returns the HTML content of an element
<code>el.textContent</code>	Sets or returns the text content of the specified node

Other DOM Element Attributes

Recall that if you have an HTML element on your page that has attributes, you can set those attributes through JavaScript as well. For instance if your

```

```

You could do the following in your JavaScript code (using the `id` alias for `document.getElementById`):

```
id("dogtag").alt = "My really cute dog";
```

Example DOM Element attributes include are:

Attribute	Description
<code>id</code>	The value of the <code>id</code> attribute of an element
<code>value</code>	The value attribute of form elements (<code>input</code> , <code>textarea</code> , <code>checkbox</code> , <code>radio</code> , <code>select</code> , etc.)
<code>name</code>	The value of the <code>name</code> attribute of a form element
<code>classList</code>	Returns the class name(s) of <code>el</code>
<code>className</code>	Sets or returns the value of the <code>class</code> attribute of <code>el</code>
<code>href</code>	The <code>href</code> for a <code>link</code> or a <code>tag</code>
<code>src</code>	The value of the <code>src</code> attribute of an image.
<code>alt</code>	The value of the <code>alt</code> attribute of an image
<code>disabled</code>	Whether or not this DOM element is disabled on the page

DOM Element `.classList` Methods

Method/Property	Description
<code>add(class)</code>	Adds specified class values. These values are ignored if they already exist in the list
<code>remove(class)</code>	Removes the specified class value
<code>toggle(class)</code>	Toggles the listed class value. If the class exists, then removes it and returns false, if it did not exist in the list add it and return true
<code>contains(class)</code>	Returns true if the specified class value is exists in the <code>classList</code> for this element

Event Object Methods and Properties

Method/Property	Description
<code>target</code>	Returns the element that triggered the event
<code>type</code>	Returns the name of the event
<code>offsetX</code>	Returns the horizontal coordinate of the mouse pointer, relative to the DOM element clicked
<code>offsetY</code>	Returns the vertical coordinate of the mouse pointer, relative to the DOM element clicked

Event Types

click	mousemove	keydown	change
dblclick	mouseout	error	focus
mouseenter	mouseover	success	submit
mouseleave	mouseup	load	select
mousedown	keyup	unload	resize

JavaScript JSON Methods

Function	Description
<code>parse(string)</code>	Returns the given string of JSON data as the equivalent JavaScript object
<code>stringify(object)</code>	Returns the given object as a string of JSON data

Other handy JavaScript Methods

Function	Description
<code>parseInt(string, radix)</code>	function parses a string argument and returns an integer of the specified radix (the base in mathematical numeral systems). If no radix is passed, returns the integer as base-10.
<code>console.log(data)</code>	Writes the <code>data</code> to the JavaScript console

JavaScript Array Methods and Properties

Method/Property	Description
<code>length</code>	Sets or returns the number of elements in an array
<code>push(el)</code>	Adds new elements to the end of an array and returns the new length
<code>pop()</code>	Removes and returns the last element of an array
<code>unshift(el)</code>	Adds new elements to the beginning of an array and returns the new length
<code>shift()</code>	Removes and returns the first element in an array
<code>sort()</code>	Sorts the elements of an array
<code>slice(start, end)</code>	Returns a new array containing the sequence of elements of the original array from start index (inclusive) to end index (exclusive)
<code>join()</code>	Returns a string concatenating all elements of an array (maintaining order)
<code>concat(list2, ...)</code>	Joins two or more arrays and returns a copy of the joined arrays
<code>toString()</code>	Returns the string representation of an array
<code>indexOf(el)</code>	Returns the index of the element in the array, or -1 if not found

JavaScript string Methods and Properties

Method/Property	Description
<code>length</code>	Returns the length of a string
<code>charAt(index)</code>	Returns the character at the specified index
<code>indexOf(string)</code>	Returns the position of the first found occurrence of a specified value in a string
<code>split(delimiter)</code>	Splits a string into an array of substrings
<code>substring(start, end)</code>	Extracts the characters from a string between two specified indices
<code>trim()</code>	Removes whitespace from both ends of a string
<code>toLowerCase()</code>	Returns a lowercase version of a string
<code>toUpperCase()</code>	Returns an uppercase version of a string

JavaScript Timer Functions

Method	Description
<code>setTimeout(fn, ms)</code>	Executes a function <code>fn</code> after a delay of <code>ms</code> milliseconds. Returns a value representing the ID of the timeout being set.
<code>setInterval(fn, ms)</code>	Executes a function <code>fn</code> at every given time-interval (in milliseconds). Returns a value representing the ID of the interval being set.
<code>clearTimeout(id)</code>	Stops the execution of the delay timer specified by <code>id</code>
<code>clearInterval(id)</code>	Stops the execution of the interval timer specified by <code>id</code>

JavaScript Math Functions

Method	Description
<code>Math.random()</code>	Returns a double between 0 (inclusive) and 1 (exclusive)
<code>Math.abs(n)</code>	Returns the absolute value of <code>n</code>
<code>Math.min(a, b, ...)</code>	Returns the smallest of 0 or more numbers
<code>Math.max(a, b, ...)</code>	Returns the largest of 0 or more numbers
<code>Math.round(n)</code>	Returns the value of <code>n</code> rounded to the nearest integer
<code>Math.ceil(n)</code>	Returns the smallest integer greater than or equal to <code>n</code>
<code>Math.floor(n)</code>	Returns the largest integer less than or equal to <code>n</code>
<code>Math.pow(n, e)</code>	Returns the base <code>n</code> to the exponent <code>e</code> power, that is, n^e
<code>Math.sqrt(n)</code>	Returns the square root of <code>n</code> (NaN if <code>n</code> is negative)

The Module Pattern

Whenever writing JavaScript, you should use the module pattern, wrapping the content of the code (`window` `load` event handler and other functions) in an anonymous function. Below is a template for reference:

```
"use strict";
(function() {

    // any module-globals (limit the use of these when possible)
    window.addEventListener("load", init);

    function init() {
        ...
    }

    // other functions
})();
```

Helper Alias Functions

You may use any of the following alias functions in your exam without defining them:

```
function gen(tagName) {
    return document.createElement(tagName);
}

function id(idName) {
    return document.getElementById(idName);
}

function qs(selector) {
    return document.querySelector(selector);
}

function qsa(selector) {
    return document.querySelectorAll(selector);
}
```