AU CSC 336 Web Programming Cheat Sheet

Spring, 2021

Bei Xiao

(this is typed in in word so make sure to use proper “quotes” works in your script.

This sheet will be updated once we learn Ajax/JSON/PHP/SQL before the finals.

# HTML

## Tags Used in the head Section

|  |  |
| --- | --- |
| Tag | Description |
| <title> text <title> | Title shown on page tab |
| <meta attribute = “value” …/> | Page metadata |
| <link href=”url” type = “text/css” rel = “stylesheet” /> | Link to a CSS Style sheet |
| <script src = “url” type =”text/javascript”/></script> | Link to JavaScript code |

## Tags Used in the body Section

|  |  |
| --- | --- |
| Tag | Description |
| <p> text <p> | paragraph |
| <h1> text </h1> |  |
| <h2> text </h2>  ….  <h6> text </h6> | (h1 for largest to h6 smallest) |
| <hr /> | Horizontal rule (line) |
| <br /> | Line break |
| <a href=”url”> text </a> | anchor(link) |
| <img src=”url” alt=”description” /> | image |
| <em> text </em> | Emphasis ((italic) |
| <strong> text </strong> | Strong emphasis (bold) |
| <ol>  <li> text </li>  </ol> | Ordered list |
| <ul></ul> | Unordered list |
| <dl></dl> | Definition list |
| <blockquote>  <p> text </p>  </blockquote> | Block-level quotation |
| <q> text </q> | Inline-level quotation |
| <code> text </code> | Computer code (monospace) |
| <pre> text </pre> | Pre-formatted text |
| <table>  <caption>text </caption>  <tr>  <th> heading 1> </th>  </tr>  …  </table> | table of data (table)  description of table (caption)  table row (tr)  table heading cell (hr)  normal table cell (td) |
| <div> ….. </div> | Block-level section of a page |
| <span> …. </span> | Inline-level section of a page |

## Content-Grouping Tags

|  |  |  |
| --- | --- | --- |
| Tag | Display | Description |
| <header> | Block | Container for a header of a document |
| <footer> | Block | Container for a footer of a document |
| <article> | Block |  |
| <section> | Block | A piece of content that is part of another (e.g. chapter section of a reading) |
| <aside> | Block | Define some content aside from the content it is placed in (e.g. a side bar in article) |
| <main> | Block | 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 (eg. Sidebars, nav links, search bars, etc.) |
|  |  |  |

## HTML Input tags

|  |  |
| --- | --- |
| **Tag** | **Description** |
| <input type=”type” name = “name”>  content </input> | form input tag  type can be text, submit, reset, checkbox, radio, file |
| <textarea rows=”num”>  initial text  </textarea> | Multi-line text input box |
| <label> text </label> | Clickable text label around a form control |
| <select>  <option> text </option>  <option>  <optground label=”text”>  <option> text </option>  <option>text </option>  </optgroup> | Drop-down selection box (select);  Each option within the box (option);  A labeled group of options (optgroup); |
| <fieldset>  <legend> text </legend> content  </fieldset> | A grouped set of form fields |

# CSS

For the following property and value table, 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 repaced with a valid color value such as 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).

## Background Styles

|  |  |
| --- | --- |
| Property | Values |
| background-attachment | Scroll | Fixed |
| background-color | *color* | transparent |
| background-image | *url* | none |
| background-origin | border-box | padding-box | content-box |
| background-position | top left | top center | top right|  center left | center center | center right|  bottom left | bottom center | bottom right  *[x-% y-%] | [x-pos y-pos]* |
| background-size | *length | % |auto | cover | certain* |
| background-repeat | repeat | repeat-x | repeat-y | no-repeat |
| Background-attachment | scroll | fixed |

## Border Styles

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

|  |  |
| --- | --- |
| border, border-\* (shortahnad) | 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*] |
| box-radius | *length* |

## Box Model

|  |  |
| --- | --- |
| **Property** | **Values** |
| float | left | right |none |
| height, width | Auto | *length* | *%* |
| min-height, max-height  min-width, max-width | none*| length |%* |
| margin, margin-\* | auto | *length | %* |
| padding, padding-\* | *length* | *%* |
| display | none | inline | block | inline-block | flex |  list-item | compact | table | inline-table |
| overflow, overflow-x, overflow-y | visible | hidden | scroll |  auto| no-display | non-content |
| Clear | left | right | both | none |

## 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] |
| letter-spacing, word-spacing | normal | *length | %* |
| text-indent | *length* | *%* |
| text-transform | none |

## 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) |
| #RRGGBB | Example: red =#FF0000 |

## 

## Selector Types

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Example** |
| Universal | Any element | \*{font: 10px Arial;} |
| Element | Any element of a given type | h1 {text-decoration: underline;} |
| Grouping | Multiple elements of different types | h1, h2, h3 {color: purple;} |
| Class | Elements with the given classname | .example {text-decoration:underline;} |
| Id | Single element with the given id | #example {text-decoration: overline;} |
| Descendent | Elements that are children at any level of another specific element | #example h1 {text-decoration: underline;} |
| Child | Elements that are direct children of another specific element | #example > p {font-weight: bold;} |
| Attribute | Elements that have the specified attribute | Input[selected] – inputs that have the selected attribute  Input[name=’test’]- inputs that have name ‘test |

# JavaScript

## DOM Methods and Properties

|  |  |
| --- | --- |
| **Method/Property** | **Description** |
| children | Returns a collection of an element’s child elements |
| parentNode | Returns the parent node of an element |
| classList | Returns the class name(s) of an element |
| className | Sets or |
| appendChild(child) | Adds a new child node, to an element as the last child node |
| [addEventListener(event, fn)](https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener) | Attaches an event handler to the specified element |
| getAttribute(attr) | Returns the specified attribute value attr of an element node |
| innerHTML | Sets or returns the content of an element |
| id | Sets or returns the value of the id attribute of an element |
| removeChild(child) | Removes a child node from an element |
| querySelector (selector) | Returns the first child node that matches a specified CSS selector(s) of an element |
| querySelectorAll(elector) | Returns all child nodes that match a specified CSS selector(s) of an element |
| getElementByClasssName(name) | Returns a NodeList containing all elements with the specified class name |
| getElementsById(id) | Returns the element that has the ID attribute with the specified value |
| getElementsByTagName (tagName) | Returns a NodeList containing all elements with the specified tag name |
| createElement(elType) | Create and returns an Element node |
| createTextNode | Create and returns a Text node |
| node.firstchild | read-only property returns the node's first child in the tree, or null if the node has no children. If the node is a Document, it returns the first node in the list of its direct children. |
| node.nextSibling | Returns the node immediately following the specified one in their parent's [childNodes](https://developer.mozilla.org/en-US/docs/Web/API/Node/childNodes" \o "childNodes), or returns null if the specified node is the last child in the parent element. |

More child node methods/properties: <https://developer.mozilla.org/en-US/docs/Web/API/ChildNode>

# Event Object Methods and Properties

|  |  |
| --- | --- |
| Method/Property | Description |
| target | Returns the element that triggers the event |
| type | Returns the name of the event |
| offsetX | Returns the horizontal coordinate of the mouse pointer, relative to the DOM element clicked |
| offset | The offsetX read-only property of the [MouseEvent](https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent) interface provides the offset in the X coordinate of the mouse pointer between that event and the padding edge of the target node. |
| clientX | Read-only property of the [MouseEvent](https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent) interface provides the horizontal coordinate within the application's [viewport](https://developer.mozilla.org/en-US/docs/Glossary/Viewport) at which the event occurred (as opposed to the coordinate within the page). |
| stopPropagation | Prevents further propagation of an event during event flow |

# See more JS mouse event: <https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent>

# Event Types

|  |  |  |  |
| --- | --- | --- | --- |
| Click | mousemove | Keydown | change |
| dbclick | douseout | Error | focus |
| mouseenter | mouseover | Success | submit |
| mouseleave | mouseup | load | select |
| mousedown | Keyup | Unload | resize |

## JavaScript Timer Functions

|  |  |
| --- | --- |
| Method | Description |
| setTimeout(fn,ms) | Executes a function after waiting a specified number of ms |
| setInteval(fn, ms) | Repeats a given function at a every given time-interval |
| clearTimeout(id) | Stops the execution of the function specified by id |
| clearInternval(id) | Stops the execution of the functions specified by id |

## JavaScript Array Methods and Properties

|  |  |
| --- | --- |
| Method/Property | Description |
| length | Sets or returns the number of elements in an array |
| push(el) | Adds new elements to the end of an array and returns the new length |
| pop() | Removes and returns the last element of an array |
| unshift(el) | Adds new elements to the beginning of an array and returns the new length |
| shift() | Removes and returns the first element in an array |
| sort() | Sorts the elements of an array |
| slice(start, end) | Selects a part of an array and returns the new array |
| join() | Joins all elements of an array into a string |
| concat(list2, …) | Joins two or more arrays and returns a copy of the joined arrays |
| toString() | Converts an array to a string and returns the result |
| indexOf(el) | Returns the index of the element in the array, or -1 if not find |

## JavaScript String Methods and Properties

|  |  |
| --- | --- |
| Method/Property | Description |
| length | Returns length of a string |
| charAt(index) | Returns the character at the specified index |
| indexof(string) | Returns the position of the first found occurrence of a specified value of a string |
| split(delimiter) | Splits a string into an array of substrings |
| substring(start, end) | Extracts the characters from a string between two specified indices |
| trim() | Removes whitespace from both ends of a string |
| toLowerCase() | Returns a lowercase version of a string |
| toUpperCase() | Returns a uppercase version of a string |
| concat(str2,….) | Joins two or more strings and returns a new joined string. |

## JavaScript Math Functions

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

## JavaScript JSON methods

|  |  |
| --- | --- |
| **Function** | **Description** |
| parse(string) | Returns the given string of JSON data as the equivalent JavaScript object |
| stringify(object) | Returns the given object as a string of JSON data |

## The Module Pattern

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

(function() {

// any module-globals (limit the use of these when possible)

window.onload = function() {

...

};

// other functions

})();

## JavaScript Ajax fetch Skeleton (not included in mid-term exam)

//you can assume checkStatus is already included

function checkStatus(response) {

if (response.status >= 200 && response.status < 300) {

return response.text();

} else {

return Promise.reject(new Error(response.status+": "+response.statusText));

}

}

function callAjax(){

let url = ..... // put url string here

fetch(url) // don't worry about cloud9 credentials

.then(checkStatus)

.then(JSON.parse) //optional line for processing json

.then(function(responseJSON) {

//success: do something with the responseJSON

})

.catch(function(error) {

//error: do something with error

});

}