Interactive Data Visualisation

Web Languages Quick Reference Guide

HTML

A full list of HTML tags can be found here:

https://developer.mozilla.org/en-US/docs/Web/HTML/Element

Basic File Structure

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document Title</title>
    <link rel="stylesheet" href="path/to/stylesheet.css">
    <script src="path/to/script.js"></script>
</head>
<body>
    <div>
        This is a section.
        <h1>This is a heading</h1>
        <p>This is a paragraph.</p>
        <!-- This is a comment -->
    </div>
</body>
</html>
```

Input Elements

Texts and Search

```
<input type="text" name="text" id="textId" placeholder="type here">
<input type="search" name="search" id="searchId" placeholder="search here">
<textarea name="area" id="areaId" cols="30" rows="10" placeholder="type
here"></textarea>
```

Buttons

```
<input type="button" value="Button Title">
<button>Button <b>title</b></button>
```

Range Sliders and Numbers

```
<input type="range" name="range" id="rangeId" min="2" max="6" step="0.5">
```

```
<input type="number" name="number" id="numberId" min="2" max="6" step="0.5">
```

Radio and Checkboxes

```
<input type="radio" name="radio" id="radioId1" value="1">
<label for="radioId1">1</label>
<input type="radio" name="radio" id="radioId2" value="2">
<label for="radioId2">2</label>
<input type="checkbox" name="check" id="checkId1">
<label for="checkId1">1</label>
<input type="checkbox" name="check" id="checkId2">
<label for="checkId2">2</label>
```

Dropdown Selection

```
<select name="select" id="selectId">
    <option selected value="Option1">Option 1</option>
    <option value="Option2">Option 2</option>
    <optgroup>
        <option value="Option3">Option 3</option>
        <option value="Option4">Option 4</option>
    </optgroup>
</select>
```

Date and Time

```
<input type="date" name="date" id="dateId">
<input type="time" name="time" id="timeId">
```

CSS

Selectors

```
div#myid { } /* div with id myId*/
div.myClass { } /* div with class myClass*/

div > p { }  /* p direct child of div*/
div p { }  /* p descendant of div */

div, h1, h2 { } /* div, h1 and h2 */
```

Layouts

Flexbox

A nice guide to flexbox can be found here:

https://css-tricks.com/snippets/css/a-guide-to-flexbox/

Grid

From the same website, here is a guide to grids:

https://css-tricks.com/snippets/css/complete-guide-grid/

Variables and Functions

You can find a complete guide here:

https://css-tricks.com/complete-guide-to-css-functions/

Variables

```
div{ /* applies inside div only */
    --my-color: #e5233f;
}

:root{ /* global declaration: applies to whole DOM */
    --my-global-color: #e5233f;
}

div > p.special {
    color: var(--my-color);
}

h2.special {
    color: var(--my-global-color);
}
```

Calc Function

```
div {
    width: calc(100% - 50px);
    font-size: calc(1.3em * 1.4);
}
```

Comparison functions

```
div {
    /* preferably 200px, but definitely between 50px and 40% */
    width: clamp(50px, 40%, 200px);
    font-size: max(1.3em, 12px); /* min also works */
}
```

JavaScript

Variables

Declaration

```
// variable declaration
let variable;
// constant declaration / immutable value
const constant;
```

Primitive Types

```
// string
let str = 'value';
// number (integer and float)
let num = 1.2;
// boolean
let bool = true;

// dynamic typing
let a; // type undefined
a = 'string'; // type string
a = 2.2; // type number
a = false; // type boolean
a = null; // type null
```

Data Structures

Arrays

```
// declaration
let arr = [1, 'string', false];
let arr2 = new Array();
// accessing/setting values
arr[1];
arr[6] = 2.5;
```

You can find a full list of array methods here:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array

Objects

```
// declaration
let obj = {'key1': 2,
           'key2': 'str'};
let obj2 = new Object();
// accessing/setting values
obj.key1;
```

```
obj['key2'] = 'value';
```

Maps

```
// declaration
let m = new Map();
// accessing / setting values
m.set('key', 'value');
m.get('key');
```

You can find a full list of map methods here:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Map

Sets

```
// declaration
let s = new Set();
// adding values
s.add(2);
```

You can find a full list of set methods here:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Set

Functions

```
// declaration
function myFunction(a, b){
    let c;
    if(a < b){
        c = b - a;
    } else {
        c = a - b;
    }
    return c;
}
// or
let myFunc = function(){
    // ...
}

// call
myFunction(3,4);

// using arrow functions (compact, but no scope) with ternary operator
let myFunc = (a,b)=>a < b ? b-a : a-b;

// using default parameters
function func(a=3,b=true){
```

```
    //...
}
```

Classes

```
// declaration
class MyClass extends SuperClass{
    publicField ;
    #privateField ;
    static field = 'value';

    constructor(a, b){
        this.publicField = a;
        this.#privateField = b;
    }
    publicMethod(){
        return this.#privateField;
    }
    #privateMethod(){
        console.log(this.publicField);
    }
    static staticMethod(){
        return MyClass.field;
    }
}
// instantiation
let myObj = new MyClass(3,6);
```

Modules

Individual Named Exports

```
export function sum(a,b){  }
export function prod(a,b){  }
export let pi = 3.1415
```

List (Re)Named Exports

```
export {sum, prod, pi as approxPi}
```

(Re)Named Imports

```
import {sum as mySum, approxPi} from 'myMath.js';
```

Full Module Import

```
import * as myMath from 'myMath.js';
```

Default Export/Import

```
export default class MyClass{  }
```

```
import ClassA from 'myClass.js'
```

Loading Modules in HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document Title</title>
    <!-- using JS script that imports modules -->
    <script type="module" src="myModule.js"></script>
</head>
<body>
    <!-- ... -->
</body>
</html>
```