The BOM

Browser Object Model

BOM only makes sense in a browser environment.

Global objects are actually properties of the window.

Browser Info

Window.navigator.userAgent

Returns the browser info being used. Unreliable as users can masquerade their browser

Window.location.href

This returns the url as a string, if this is changed on assignment, the browser will change to the new url

Window.history can be used to get the current sessions browser history. Adding .go(#) will change the page to a previous page if # is a negative number

Window.open() is used to open a pop-up window

Window.open(url, title, ‘width=#, height=#, [other attributes]);

Screen info

Window.screen.height & .width

Cookies

Small files saved locally on the user’s computer. Allows the state of a user’s info to be saved for later use. Good for individualizing a web page.

Document.cookie = ‘hero=true’; Document.cookie = ‘city=Metropolis’;

Document.cookie – returns all cookie settings as a string

By default cookies only last as long as the browser session but they can be made to persist longer

const expiryDate = new Date();

const tomorrow = expiryDate.getTime() + 1000 \* 60 \* 60 \* 24;

expiryDate.setTime(tomorrow);

document.cookie = `name=Batman; expires=${ expiryDate.toUTCString()}`;

setTimeout

executes code after a set amount of time

setInterval

The same as setTimeout except the code is executed repeatedly

Data attribute

A custom attribute that can be used to contain other important data. Really it can be named as anything but by convention the attributes should be prefaced with data-

const superman = document.getElementById('hero');

const powers = superman.dataset.powers;

<< 'flight superSpeed'

Local storage

localStorage.setItem('name', 'Walter White');

localStorage.getItem('name');

<< "Walter White"

localStorage.removeItem('name');

localStorage.clear();

Geolocation

This is an API that can get the location of the device, however permissions must be set in order for it to work.