Steps for Progamming Apps with in JS

Using Fetch—

Make request
--fetch url, then put response in json format, then put in variable
Collect Data and Format (sometimes using a forEach loop)
Put in HTML

Not Using an API

--Identify elements from HTML Document and put in varaibles using QuerySelector, ElementByID, etc.
--

Chapter 9—The Window Object—

Global object. Variables that have a global scope…are properties of the this object.

THE BROWSER OBJECT MODEL
Collection of properites and method that contain information about the browser and the computer screen.
--there is no official standard but major browsers follow a de facto.
--BOM

Going global

Created with out using var, let, or const…they are properties of the window object.
Use them without the window object to keep code clean and simple.
Functions are actually methods of the window object.

Dialogs
-alert(), confirm(), and prompt()—not recommended as best practice unless you are wanting to make sure a user doesn't delete something important.—they stop execution until the prompt is resolved.

BROWSER INFORMATION

Properties and methods can provide information about the user's browser.

Which Browser

Navigator object contains information about the browser and user agent property will return information about the oberating system—can be disguised so don't rely on them.

Location, Location, Location
--Herf property to return full URL..can also assign. Window.location.href
--protocol—returns a string descinign the protocol

--host returns a string describing the domain of the current URL
--hostname returns string describing the domain of the current UFL
--port describes the port number
--pathname the path that follows the domain
--search—string that starts with ? followed by query parameters.
--hash—returns a string that starts with #
--origin string that shows the protocol and domain where page originated from.
--reload() method to force a reload of current page.
--assign()
--toSTring()

BROWSER HISTORY
Window.history.length()—how many pages were visited before this one.
Window.history.go()—"0" is the current page..-1, 1 are the beore and after.
…forward() and …back()

CONTROLING WINDOWS
--open(url, window title, attributes)
--close()
--move(X, Y)
--window.resizeTo();

SCREEN INFORMATION—mostly used on mobile devices
Window.screen
--.height
--.width
--.availHeight, .availWidth---how big the screen can get without menus.
--.colorDepth

THE DOCUMENT OBJECT
Additional methods not in chapter 6.

Document.write—replaces the whole window.
--document.write with tags creates a node.--- document.write('<h1>Hello, world!</h1>');
--can be placed in script tags---

```
<h1>
   <script>document.write("Hello, world!")</script>
</h1>
```

Not recommended to use the "document.write method".

Cookies—browsers do not remember anything from one request to the another…cookies store information that can be reused…they can only be read by the a page from the same domain.

localStorage API can be used now too.

Cookies take the form of text file that contain a list of name/value pairs.

Creating Cookies.
document.cookie = 'name=Superman';
<< "name=Superman"---the name
---adding another name just appends it.
--to reassign, just the same name with a different value.
--to read---document.cookie will list all the cookies.

Cookie Expiry Dates—by default they expire once the browser is finished.
--to extend add a different expiration date.
const expiryDate = new Date();
const tomorrow = expiryDate.getTime() + 1000 * 60 * 60 * 24;
expiryDate.setTime(tomorrow);

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

max-age property will set an expiration date value in seconds. (86400 is a day)

The Path and Domain of Cookies
--can change within the root directory
document.cookie = 'name=Batman; path=/'
document.cookie = 'name=Batman; domain=sitepoint.com';
; secure will only transmit over secure HTTPS networks.

Deleting Cookies—to remove, set the expire date to a time in the past.

Timing Functions

--window.setTimeout(function, milliseconds)—scedules the executing of a function at intervals.
--widnow.clearTime()—to clear timeout.

-setInterval()
Works the same as .setTimout() expect that it will continue to callback the function . Can be an anonymous function or a named function.
Example:
const summon = window.setInterval(chant,1000);
window.clearInterval(summon);

“this” is binding globally and can be confusing.

ANIMATION

setTimeOut() and setInterval() methods can be used to animate elements.

requestAnimationFrame()—takes the place of the rotation code. It is an argument that calls the function recursively…about 60 frames per second.

To stop—cancelAnimationFrame(id);

Chapter 14-HTML5 APIs

HTML5—will allow features and modules to be released more frequently as features are completed instead of waiting for other features.

--A useful site called “Can I Use” to check on specific features

THE DATA-ATTRIBUTE
--The data- attribute is a way of embedding data in a web page using custom attributes that are ignored by the browser.

--start with “data”
--contain only lowercase letters, numbers, hyphens, dots, colons, or underscores
--include optional string value.

Examples:
data-powers = 'flight superSpeed'
data-rating = '5'
data-dropdown
data-user = 'DAZ'
data-max-length = '32'

to access drop the data prefix…example…const powers = superman.dataset.powers.
--use camel case
--can code as a JSON string and convert later.

HTML5-APIs
Hardware evolves quickly, and APIs are frequently introduced to give developers access, and control new features that appear in the latest devices.

HTML5 Web Storage
--web storage api can…
Stored is not shared
Available in multiple windows
Storage capacity is limit is larger than for cookies.

Data does not automatically expire.

Window object has a property called….localStorage…key-value pairs and strings for pairs only
--.setItem(key, pair)
localStorage.setItem('name', 'Walter White');

localStorage.getItem(keyName);

localStorage.keyName = ….

localStorage.removeItem(keyName);
localStorage.clear();

Events are only fired in other windows or tabs from the same domain and only if the value of the idtem being saved changes.

Strings can be saved is not a restrictions because JSON can store the objects in the local storage.

Geolocation
Can be used to find the users exact location.
It is the navigator object with properties
--getCurrentPosition()…uses longitude and latitude.
Position object has properties too.
--position.speed
Position.altitude, .heading, .timstamp

WatchPosition() method calls a callback function every time the position of the device is updated.
clearWatch() to stop callbacks

Web Workers
Allow processes to be run in the background, adding support for concurrency in JS
--Worker()---to create new worker
--postMessage()—to send messages and start the worker working

A message is an event fired so they can be dealt with using an event listener.
--terminate() to stop when task is completed…or the close() method from inside the script.

Shared Web Workers
--create web workers that allow lots of different scripts on the same domain to access the same worker object.

Service Workers

--The Service Worker API allows a worker script to run in the background with the added benefit of being able to intercept network requests.—if th network is offline.

Websockets
--a new protocol that allows a two-way communication with a server also known as push messaging…for like in chat apps.
--send()—sends message to URL that the websockets is connected to.

Notifications
These APIs allows you to show messages using the system's notifications….sometimes popup in the corner
--requestPermission()
--notifications close after a short period of time on most browers, but some require you to close them.—click(), close(), show() methods are used.

Multimedia
Media API for controlling vidoes and audios on webpages
--<audio> and <video>--In HTML

In JavaScript—video.play(), .pause(), .volume(), muted, .currentTime, playbackRate, loop, duration.

OTHER APIS
APIs are constantly growing, check Mozilla for updates.

DRAWING with CANVAS
Canvas element was introduced to allow graphics to be drawn ontl a web page. In HTML <canvas> add height and width.

<canvas id='canvas' width='400' height='400'>Sorry, but your browser does not support the canvas element</canvas>

const canvasElement = document.getElementById('canvas');

const context = canvasElement.getContext ('2d');
--context.fillStyle, .strokeStyle, linewidth, .fillRect(10, 10, 100, 50)
.strokeRect(x, y, z, u) etc.

SHIMS AND POLYFILLS

Modern browsers are quick to up-date and so are the APIS. There are libraries of code that allow you to use APIs.

Shim is a piece of code that ads some missing functionality to a browser.

Polyfill is a shim that achieves the same functionality while using the API commands that would be used if the feature was supported.