WDD330

Ribeka Nanjo

W05 Reading (L09)

HTML5 APIs

https://developer.mozilla.org/en-US/docs/WebAPI

- Web Storage
  - ➢ The Web Storage API provides a key-value store on the client's computer that is similar to using cookies but has fewer restrictions, more storage capacity, and is generally easier to use.

The Web Storage API has some crucial differences with cookies:

・ Information stored is not shared with the server on every request.

・ Information is available in multiple windows of the browser (but only if the domain is the same).

・ Storage capacity limit is much larger than the 4KB limit for cookies (there is no actual limit in the specification, but most browsers have a limit set at 5GB per domain).

・ Any data stored does not automatically expire as it does with cookies. This potentially makes cookies a better choice for something like showing a popup once a day.

- Geolocation: The Geolocation API **returns a location and accuracy radius based on information about cell towers and Wi-Fi nodes that the mobile client can detect**.

E.g.

```
function youAreHere(position) {
console.log(`Latitude: ${position.coords.latitude}, Longitude: ${}
```

- position.speed property returns the ground speed of the device in meters per second.
- position.altitude property returns an estimate of the device's altitude in meters above the WGS84 ellipsoid, which is a standard measurement for the center of the Earth.
- position.heading property returns the direction the device is moving in. This is measured as a bearing in degrees, clockwise from North.
- position.timestamp property returns the time that the position
- information was recorded.

watchPosition(): a callback function every time the position of the device is updated.

clearWatch(): used to stop the callback being called, using the ID of the watch as an argument

- Notifications

First of all, you need to get permission granted by the user. This can be achieved using the **requestPermission()** method of a Notification global object.

E.g.

```
if(window.Notification) {
Notification.requestPermission();
```

‘granted’ or ‘denied’
If it’s set to granted, you can create a notification like this:

```
if(window.Notification) {
Notification.requestPermission()
.then((permission) => {
if(Notification.permission === 'granted') {
new Notification('Hello JavaScript!');
}
});
}
```

It will produce a system notification with the title ‘Hello Javascript!’

notification.close(); - when the notification is closed

notification.click(); - when a user clicks on it

notification.show(); - when the notification appears

E.g.

```
notification.addEventListener('click', () => {
window.open('https://sitepoint.com')
}, false);
```