Lo 187 - Styles, Attributes Classes

Async functions run in background, with one or more awaits, until finished.

Async await - don't have to chain .then methods, don't have to return anything, don't have to use a callback function.

Async await – synthetic sugar over consuming promises (like classes).

Async await - can't attach .catch method so a try catch block is used.

Try-catch used in regular Javascript also (not specific to async await). Probably been in JS since beginning.

DOM API: Contains many methods and properties provided by the browser such as:

Event Target: addEventListener, removeEventListener

Node Object: parentNode, childNodes, cloneNode, textContent

Element Object: querySelector, addEventListener, createElement, innerHTML, append, closest, remove, insertAdjacentHTML, children, parentElement, classList, setAttribute, matches, scrollIntoView, etc

Number.parseFloat()

Number.isFinite() – Infinity and string numbers (‘23’) are false

Advantage of promises is allows for then sequential processing.

Async function immediately returns a promise and runs in the background.

. then chaining - Can ‘return’ a promise string or a promisified function. If returning a string then the promise is resolved immediately.

.catch – Can be located anywhere in the promise chain and will catch errors located before it. Can have multiple .catch methods in a chain and .then statements located below a .catch will be processed (promise chain will continue). If termination of chain is desired then locate .catch at the end of the .then chain. All errors will cause subsequent .then's in chain to be bypassed until the next closest .catch is executed. Promise is reset to ‘pending’ when next .then is initiatially processed? (Not a new promise?)

Always use try catch with async functions

forEach - Continue and break do not work.

Function side effect – performing work without returning anything – such as with a forEach.

Form button click event is also triggered by the enter key within a input field.

All numbers are represented as floating point (even integers). i.e. 23 == 23.0

Bankist Changes

Add other accessories

Error for bad login, invalid input for loan, transfer, delete.