CHAPTER AJAX

* CALL STACK

JS🡪 Nested Functions🡪 Each function added to a stack. Inner-most executed first then popped off and subsequent functions executed.

* Web API:

Js🡪 Single Threaded = One tax at a time. A lag in processing a line = Delay in response.

Web API🡪 In browsers that take over executing the next lines till JS processes.

* CALLBACK HELL

Callback🡪 Passing another function inside a function as a means to execute it one after another.

Messy callbacks and tangled codes that are hard to decipher = Callback hell.

* Promises

To avoid Callback Hell, this object provides .then and .catch to check if url was accepted or rejected. For This only 1 url to be passed as argument and not any callback functions.

The function returns Promise object with resolve and reject parameters.

Kind of like an if..else construct by chaining promises(.then-.catch construct).

* Async Functions

Async keyword allows function to automatically return a promise. Automatically promise is returned as resolved, to reject “throw error” keyword use.

* Await Keyword

Paired with async functions where when calling a promise, async is written and it waits until that promise is completed/ resolved. Then only moves to next promise. Can be used ONLY inside async block.

To implement reject in await: Usage of try…catch block to be done.