What the heck is the event loop anyway? | Philip Roberts | JSConf EU

- What is javascript? A single threaded non-blocking asynchronous concurrent language.
- Javascript runtimes like V8 have a heap and a call stack.
- An addition to the Javascript runtime browsers provide Web API's like the DOM, Ajax and setTimeout, the event loop and the callback queue.
- Blocking behaviors don't work well with browsers because they stop users from doing anything until a specific request is completed.
    - Because of this blocking requests in javascript are made with asynchronous callbacks.
- In javascript multiple things can be done at once by utilizing web api's which are essentially threads (which can only be called). The runtime itself is concurrent.
- When an asynchronous call is made we are calling a web api (or c++ api on the case of the backend), which then pushes the code to be executed on the callback que. Once the stack is empty the event loop pushes code to be executed on the call stack.
    - For example: if a button has an event listener, it lives as a web api, which when clicked, pushes its code to the callback queue, which is then pushed onto the call stack by the event loop when the call stack is empty.
- The browser renderer would like to refresh every 16 ms, however if synchronous code is bing run it cannot render. The renderer however can be run in between different async function as before each call the call stack is empty.