Code Fellows

Class 15 Event-Driven Applications

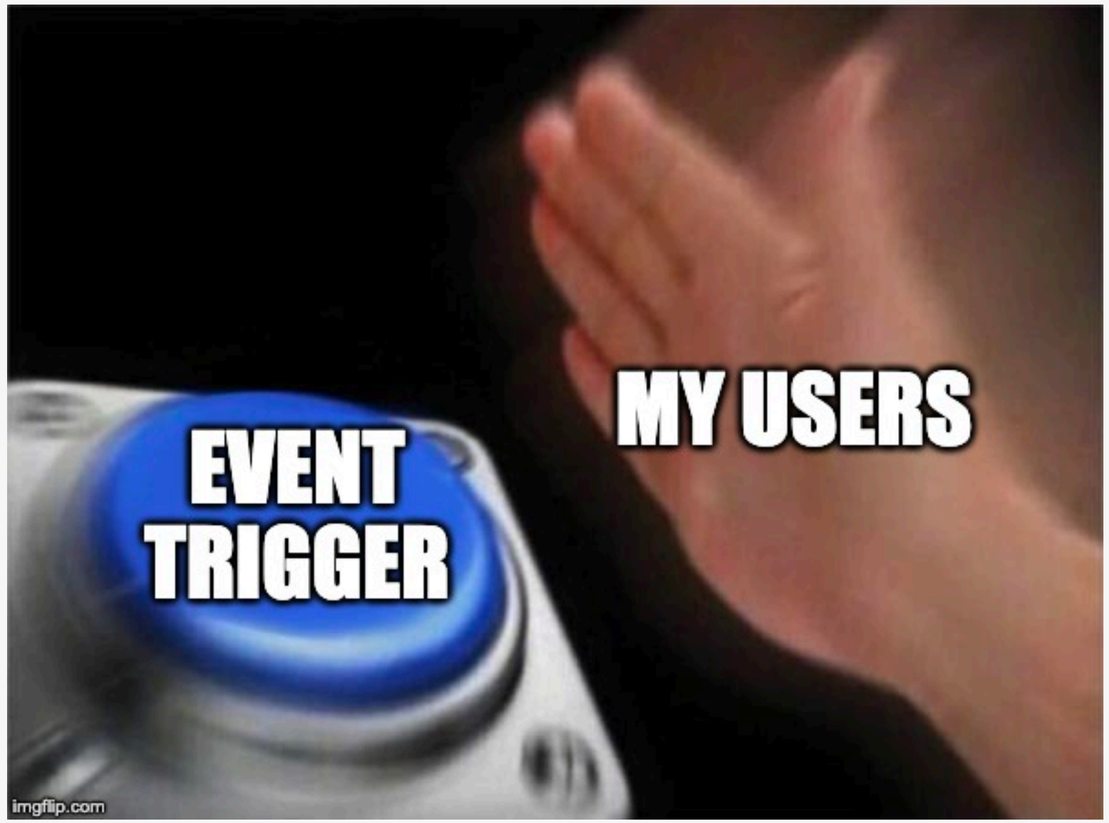
seattle-javascript-401n14

Lab 14 Review

Code Challenge 14 Review

What is an Event?

- A signal that something happened
- Someone clicked a button, pressed a key
- Events are system-wide or application-wide
- Events are “raised”, “triggered” or “**emitted**”



Listeners and Handlers

- Events by themselves don’t do anything
    - It’s like a person shouting in a forest. If no one is there to hear it…
- If we care about an event, we need to create a **listener** for it
- The listener is set to run a **handler function** whenever it “hears” an event
- There can be multiple listeners per event

You might be familiar with…

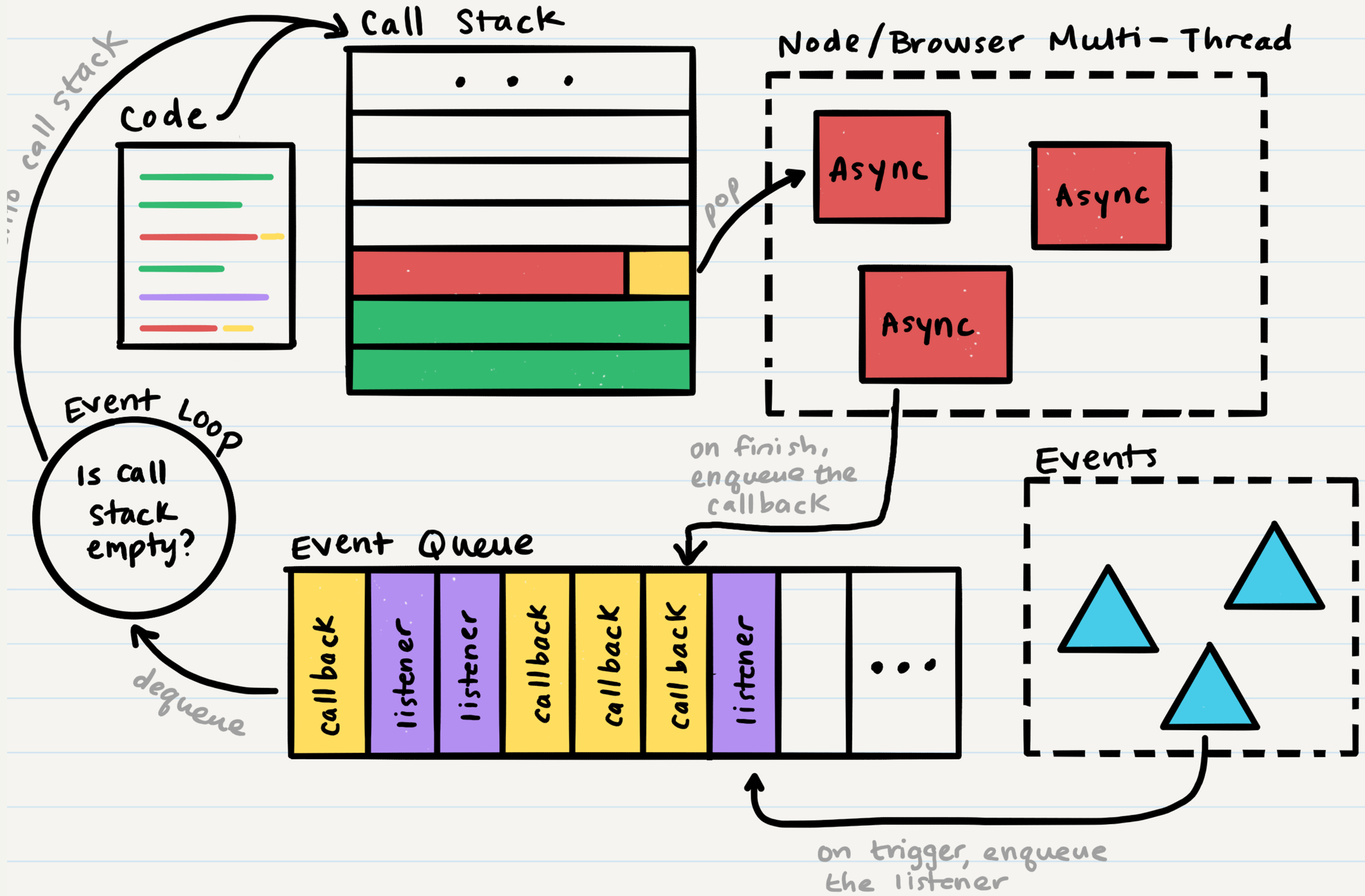
```
<form onsubmit="handleSubmit(event)"></form>

<button onclick="handleClick(event)"></button>


window.addEventListener("resize", function() {} );

$("#my-btn").click(function() {});
```

These are listeners and handlers!


Call Stack
Code
Node/Browser Multi-Thread
Async
Async
Async
pop
Event Loop
Is call stack empty?
on finish, enqueue the callback
Events
Event Queue
callback
listener
listener
callback
callback
callback
listener
dequeue
on trigger, enqueue the listener

Events in Node

- We can create our own events and listeners using the Node `events` module
- This module exposes a class `EventEmitter`
- We make one object from this class and use the same object throughout our application
- Multiple `EventEmitter` objects in one application won't be able to share events with one another
- Events should be system-wide or **application-wide**

Demo

demo/events

Let’s get some hands on experience with making our own events and event listeners using the Node `events` module

What's Next:

- Due by Midnight Tonight:
  - **Learning Journal 15**
- Due by Midnight Sunday:
  - **Career Coaching Assignments**
  - **Feedback Week 08**
- Due by Midnight Monday
  - **Code Challenge 15**

- Due by 6:30pm Tuesday
  - **Reading Class 16**
  - **Lab 15**
- Next Class: **Class 16 - TCP Protocol**

Questions?