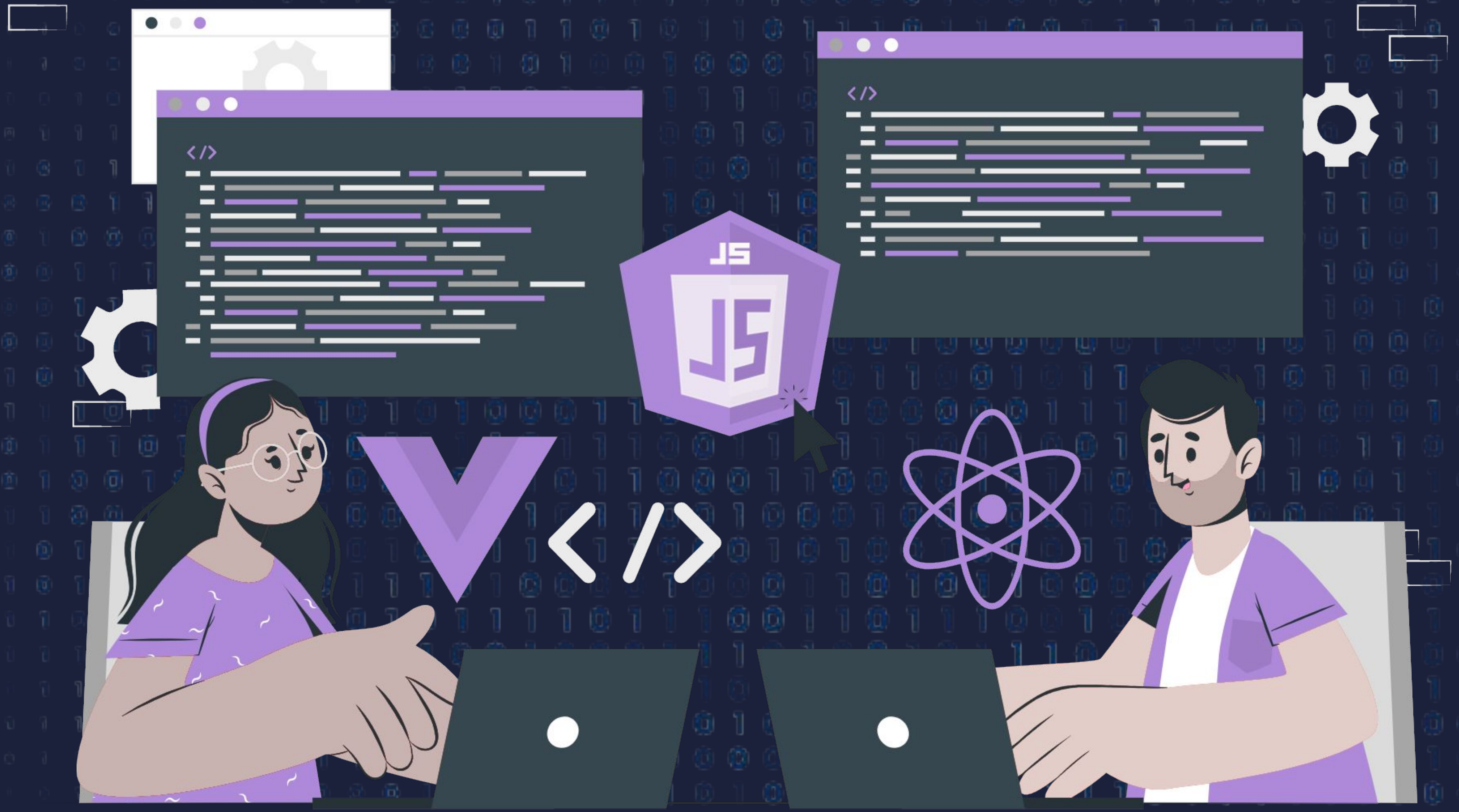
PW SKILLS

Event Module



List of content:

1. Listening Events
2. Removing Listener
3. Special Event
4. Asynchronous Event

Event module

- Node.js is a JavaScript runtime that operates asynchronously and in an event-driven manner, as per its official documentation.
- Its architecture enables it to handle asynchronous tasks. The 'events' module in Node.js allows it to emit named events that can result in corresponding functions or callbacks being invoked.
- These functions or callbacks can subscribe to a specific event, and when the event is triggered, all the callbacks that were registered in the order of their registration are executed.
- The EventEmitter class is responsible for emitting events, and all objects that emit events are instances of this class. To listen or emit an event, one can make use of the EventEmitter.

Listening Events

- Before emitting any event, it must register functions(callbacks) to listen to the events.
- The functions eventEmitter.on(event, listener) and eventEmitter.addListener(event, listener) have similar functionality.
- They append the listener at the end of the listener's array for the specified event. If multiple calls are made to the same event and listener, the listener will be added multiple times and subsequently fired multiple times.
- Both functions return the emitter, so they can be chained. In contrast, eventEmitter.once(event, listener) fires only once for a specific event and is then removed from the listeners array.
- It also returns the emitter, allowing for chained calls. In Node.js, each event is identified by a name, and we can trigger an event using the emit(event, [arg1], [arg2], [...]) function. We can pass any number of arguments to the listener functions.

Removing Listeners

- The function eventEmitter.removeListener() accepts two arguments, namely, event and listener.
- It removes the specified listener from the listeners array that is subscribed to the given event.
- On the other hand, eventEmitter.removeAllListeners() removes all listeners that are subscribed to the specified event from the listeners array.
- Removing the listener from the array will change the sequence of the listener's array, hence it must be carefully used.
- The eventEmitter.removeListener() will remove at most one instance of the listener which is in front of the queue.
- By default, a maximum of 10 listeners can be registered for any single event. To modify this default value for all instances of EventEmitter, we can use the EventEmitter.defaultMaxListeners property.
- We can use the **eventEmitter.getMaxListeners()** function to retrieve the maximum number of listeners allowed, which can be set using setMaxListeners() or is set to the default value of 10 if not explicitly set.
- It is important to note that although the maximum number of listeners is not a hard limit, if a new instance is added, EventEmitter will allow it but will issue a warning message indicating a possible memory leak.

Special Events

- Whenever new listeners are added, EventEmitter instances emit the 'newListener' event, and whenever existing listeners are removed, they emit the 'removeListener' event.

Asynchronous event

- The listeners that were registered are called by the EventEmitter synchronously, in the order of their registration. Nevertheless, it is possible to perform asynchronous calls using either setImmediate() or process.nextTick().

SKILLS

THANK YOU