**EVENT-DRIVEN PROGRAMMING**

On a computer every action is an event. similar to the opening of a file or establishing a connection.

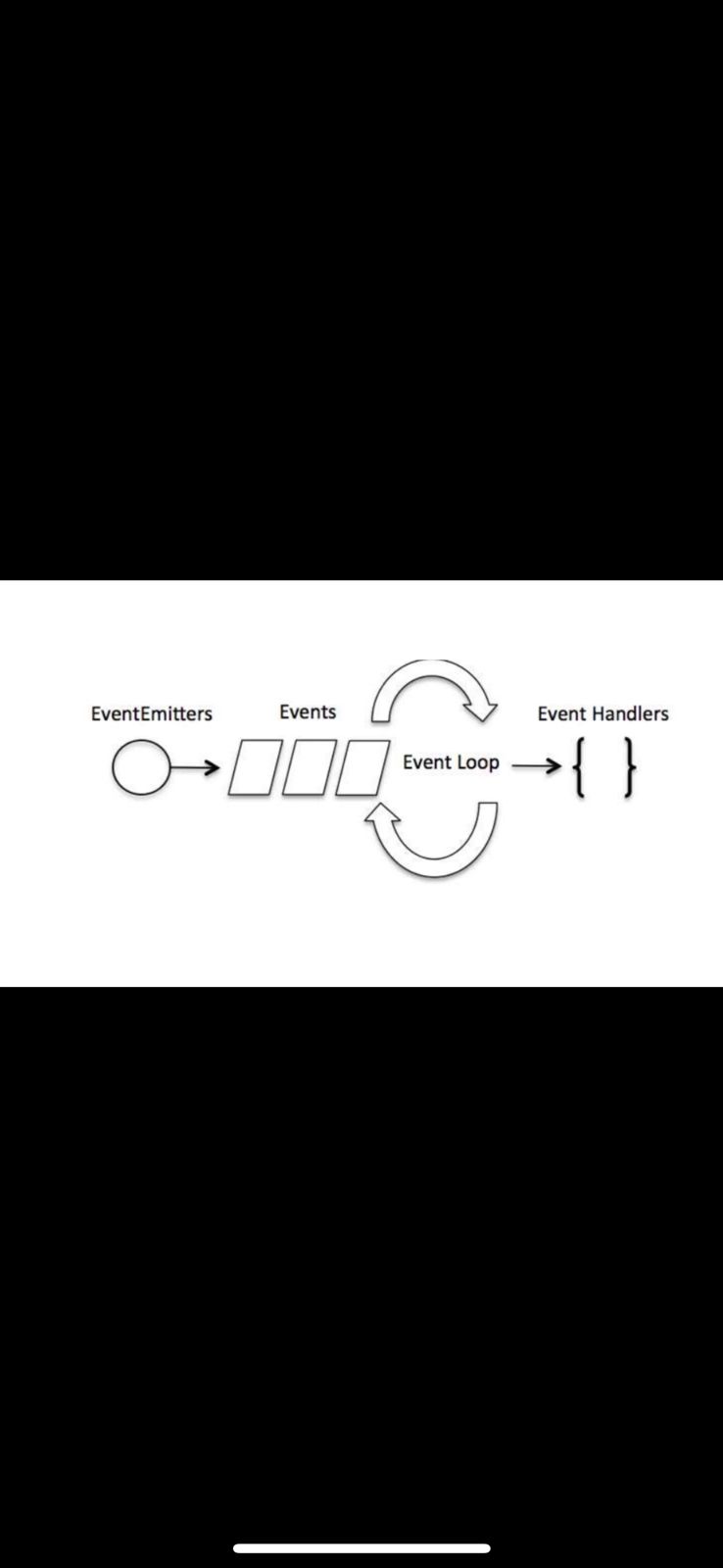
Node.js objects have the ability to fire events, such as the readStream objects, which does so when a file is opened and closed.

You may build, fire & listen for your own events using the “EVENTS” built-in modules in node.js

Additionally, an **EventEmitter** object instance is used for all event properties & methods. Make an **EventEmitter** objects to have access to these properties & methods.

Node.js makes extensive use of events, which is also one of the factors that contribute to its relative speed when compared to other related technologies. Node simply launches its variables, declares its functions & then waits for the event to happen after the server has started.

An event-driven program typically has a main loop that monitors for events & activates a callback function when one is found.



Even while callbacks & events have a similar appearance, callbacks functions are called when an asynchronous function gives its result, whereas event handling uses the observer pattern. Observers are the processes that listen to occurrences. Every time an event is fired, its listener function begins running. The events module & **EventEmitter** class in node.js provide access to a variety of built-in events which are used to bind events & event-listeners as follows.