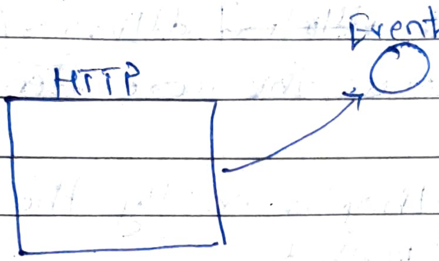
The naming of the first letter of every word should be uppercase, this is the convention that indicates that the event emitter is a $\boxed{\text{Class}}$, it's not a fn., not simple value.

e.g: Const EventEmitter = require('Events');

* Events Module:

A Event is basically a signal that indicates that something has happened in our application.

e.g: A class called ~~No~~ HTTP, which is used to build a web server so we listen on a given port & every time we receive a request on that port that HTTP class raises an event.



Our job is to respond to that event which basically involves reading that request & returning the right response.

Several classes in node raises different kinds of events.

→ Class: Event Emitter: This class is one of the core building blocks of node & a lot of classes are based on this event emitter.

When we call the require fn. with event we get the event emitter.

In class EventEmitter we have many methods, only few/2 we use:
(i) emit().

(i) emit(): It's used to raise an event. (emit means making a noise, produce, here it means we're signaling that an event has happened

Date
Page

~~There should be listener which will help to raise~~
If there's no Listener ~~there~~ in application then nothing will happen cuz, ~~i.e.~~ listener only will be interested in the event.

Listener is an event that will be called when that event is raised.

→ We have an method ~~addListener~~ [addListener], but we've alias for this ~~this~~ i.e. 'on' we use this more often. (jQuery used in).

→ 'on' / 'addListner' they're exactly the same ~~too~~ but quite often we use 'on' Method.
This method takes 2 arguments (i) is the name of the event (ii) is the callback fn / actual listener.

→ The order is Imp if we register ~~the~~ the listener after calling the emit method, nothing would happen because when we call the emit ~~ter~~ method, the emitter iterates over all the registered listeners & calls them synchronously

→ So, this is the basic of raising events & handling them using the ~~event~~ Event Emitter class.