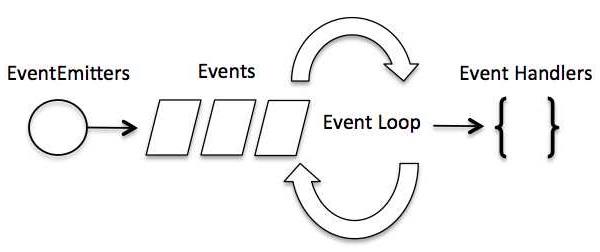
Node.js

Event-driven programming



1. Event Emitter class

All objects that emit events are instances of the EventEmitter class. These objects expose an eventEmitter.on() function that allows one or more functions to be attached to named events emitted by the object. Typically, event names are camel-cased strings but any valid JavaScript property key can be used.

When the EventEmitter object emits an event, all of the functions attached to that specific event are called *synchronously*. Any values returned by the called listeners are *ignored* and will be discarded.

The eventEmitter.on() method is used to register listeners, while the eventEmitter.emit() method is used to trigger the event.

The “on” property is used to bind a function to the event and “emit” is used to fire an event.

Note: You should define eventEmitter.on() before you emit an event to make sure that already had the event handler (listener)

As a best practice, listeners should always be added for the 'error' events.

const myEmitter = new MyEmitter();

myEmitter.on('error', (err) => {

console.error('whoops! there was an error');

});

myEmitter.emit('error', new Error('whoops!'));

// Prints: whoops! there was an error

Apply to real project: Emit events between the processes

--------------------------------------------------------------------------------------------------------------------------------

JavaScript

**Hoisting**

Hoisting is JavaScript’s default behavior of moving declarations to the top.

**JavaScript Declarations are Hoisted**

In JS, a variable can be declared after is has been used.

In other words, a variable can be used before it has been declared.

**JavaScript Initializations are NOT Hoisted**

**Declare Your Variables At the Top!**

**Closures**

JavaScript variables can belong to the **local** or **global** scope

--------------------------------------------------------------------------------------------------------------------------------

ES6

Arrow functions

Extended Parameter Handling

Default Parameter Values

Rest Parameter

Template Literals

String Interpolation