**setTimeout & setInterval**

Another way that we can work with Node.js asynchronously is through timing functions.

So, the time functions like set timeout, clear timeout, set interval, and clear interval work the same way that they do in the browser, and are available to us globally.

Ex:

// *setTimeout*

const *waitTime* =3000;

*console.log*(`setting a ${waitTime / 1000} second delay`);

const *timerFinished* =()=>{

*clearInterval*(interval);

*console.log*("done")

};

*setTimeout*(timerFinished, waitTime);

// *setInterval*

const *waitInterval* =500;

letcurrentTime=0;

const *incTime* =()=>{

currentTime+=waitInterval;

*console.log*(`*waiting* ${currentTime/1000} *seconds*`);

}

const *interval* = *setInterval*(incTime,waitInterval);

Output:

setting a 3 second delay

waiting 0.5 seconds

waiting 1 seconds

waiting 1.5 seconds

waiting 2 seconds

waiting 2.5 seconds

done

**Reporting progress with setInterval**

**Code:**

// *setTimeout*

const *waitTime* =3000;

*console.log*(`setting a ${waitTime / 1000} second delay`);

const *timerFinished* =()=>{

*clearInterval*(interval);

*console.log*("\n done")

};

*setTimeout*(timerFinished, waitTime);

// *setInterval*

const *waitInterval* =500;

letcurrentTime=0;

const *incTime* =()=>{

currentTime+=waitInterval;

const *p* = *Math.floor*((currentTime/waitTime)\*100);

*process.stdout.clearLine*();

*process.stdout.cursorTo*(0);

*process.stdout.write*(`*waiting...* ${p}`);

}

const *interval* = *setInterval*(incTime,waitInterval);

**Output:**

setting a 3 second delay

waiting... 83

done