**Standard Input & Standard Output**

Another feature of the process object (that we discussed before) is *Standard Input & Standard Output*.

These two objects offer us a way to communicate with a process while it’s running.

**Standard Output:**

// *Sending some string to the terminal.*

*process.stdout.write*("hello");

Output in terminal:  
Hello

**A Simple Application:-**

**Code:**

const *questions* =[

"What is your name?",

"what would you rather be doing?",

"What is your preferred programming Language?"

];

const *answers* =[];

function *ask*(i=0) {

*process.stdout.write*(`\n\n ${questions[i]}`);

    // *Indicate to ourselves that we're trying to accept some sort of input.*

*process.stdout.write*(` > `);

}

*ask*();

// *It prompts the user for an answer, and then it quits. So we also leave the terminal out of whack becuase we use standard output here.*

// *So what we actually want to do is wait until the user answers the question.*

// *So what we can do is listen for a data event on this object using a function. Here, "data" is an event.*

// *So we're going to say...*

*process.stdin.on*("data", function (data) {

    // *We enter this block, if we've some new data*

    // *trim() method use to remove space from both the end of a string.*

*answers.push*(*data.toString*()*.trim*());

    if (*answers.*length < *questions.*length)

*ask*(*answers.*length);

    else

        // *to tell our process to quit.*

*process.exit*();

})

// *This(the above function) is called Data Event Listener. Using this we can now run node asynchronously. Before every other app we've run until now has run through the command synchronously and quit, leaving us back at the terminal prompt. But this time the app is still running wating for some input.*

// *"exit" event would let us to call another function*

*process.on*("exit", function () {

*process.stdout.write*("\n\n");

*process.stdout.write*(

        `Go ${answers[1]} ${answers[0]} you can finish writing ${answers[2]} later.`

    );

*process.stdout.write*("\n\n");

});

**Output:**

What is your name? > Chandan

what would you rather be doing? > Playing

What is your preferred programming Language? > JavaScript

Go Playing Chandan you can finish writing JavaScript later.