

# Taking Input in JavaScript

---

## Overview

This module covers methods using which we can take command line inputs from users after the execution of the program.

Taking Command-line inputs in JavaScript is not possible, but using the NodeJS, i.e. JavaScript run-time environment.

## Process.stdin

The process.stdin property is an inbuilt application programming interface of the process module which listens for the user input. The stdin property of the process object is a Readable Stream. It uses on( ) function to listen for the event.

**Syntax:** process.stdin.on( );

**Return Value:** It doesn't return any value.

**Parameters:** This property takes Input from the user.

## readline( )

Readline Module in Node.js allows the **reading of input stream line by line**. This module wraps up the process standard output and process standard input objects. The readline module makes it easier to input and reading the output given by the user.

**NOTE:** It is used to take inputs in different lines.

## main( ) function

In many programming languages, the main function is where **a program starts its execution**. But this is not the case with JavaScript. It doesn't require the main function to execute.

But to maintain consistency, we will also make a main( ) function and take all inputs and outputs into that function.

## Simple program to take a number input

```
1. process.stdin.resume();
2.
3. var input_stdin = "";
4. var input_stdin_array = "";
5. var input_currentline = 0;
6.
7. process.stdin.on('data', function (data) {
    a. input_stdin += data;
8. });
9.
10. process.stdin.on('end', function () {
    a. input_stdin_array = input_stdin.split("\n");
    b. main();
11. });
12.
13. function readLine() {
14.     return input_stdin_array[input_currentline++];
15. }
16.
17. function main() {
18.     let num = Number(readLine()) ;
19.     console.log(num);
20. }
```

Line number 18 is the main part where we need to focus

## Taking String input

It will be the same as the above program, just line number 18 will changes as -

```
function main() {
    let num = String(readLine()) ;
    console.log(num);
}
```

## Taking array input/inputs in the same line

Changing the main() function in above program

```
function main() {  
    const arr = readline().replace(/\s+$/g, '').split(' ');  
    console.log(arr);  
}
```

### Input

```
1 4 5 3
```

### Output

```
[ '1', '4', '5', '3' ]
```

## Printing Output

**console.log()** is most commonly used to print the output on the console.  
console.log() prints and takes to the next line, which means two consecutive console.log() statements written will be printed in different lines