# Getting Started With REPL

## REPL

If we type **$node** it will start with **REPL** mode.

REPL – Read Eval Print Loop

Node will just print the result for each line we type with output, for statement it will print undefined.

**CTRL + L** clear the REPL session

For **multiple line** command we node can automatically identify when we start with **{.**

We explicitly can open multi line command by typing **.editor** and enter. To exit we can use **CTRL + D.**

To find other command type **.help**

### Tab and underscore

**TAB** can be used for auto complete and **double tab** will list all possible starting with the provided character.

**Underscore** behaves same as **$?** In shell scripting which stores the last executed value.

>Math.random()

0.99895632

>\_

0.99895632

We can use underscore variable at any place where we can use javascript expression.

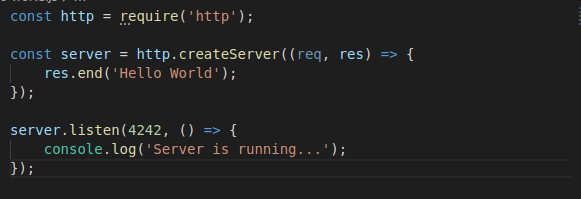
>const random = \_

### Executing script

From the location where the **.js** file is present we can execute the script as below.

$node hello-world.js

Or we can provide absolute path.



createServer() just create the server but doesn’t activate, for activating we need to use **listen()** function.

This script is continuous running and keeps the node REPL busy.

### Working with Timers

Timer functions are popular global functions and node js also has api around it.

* setTimeout()
* setInterval()

These used to delay or repeat the execution. These method returns a **TimeOut** object which can be used for clearing the time out and intervals.

To clear the timeout or interval we have methods as clearTimeout(timeOut) and clearInterval(timeOut)