**Dynamic Evaluation**

Node can directly evaluate code from the shell. This is useful for quickly checking a code snippet or for creating very small cross-platform commands that use JavaScript and Node core API’s.

There are two flags that can evaluate code. The **-p** or **--print** flag evaluates an expression and prints the result, the **-e** or **--eval** flag evaluates without printing the result of the expression.

The following will print *2*

**node --print "1+1"**

The following will not print anything because the expression is evaluated but not printed.

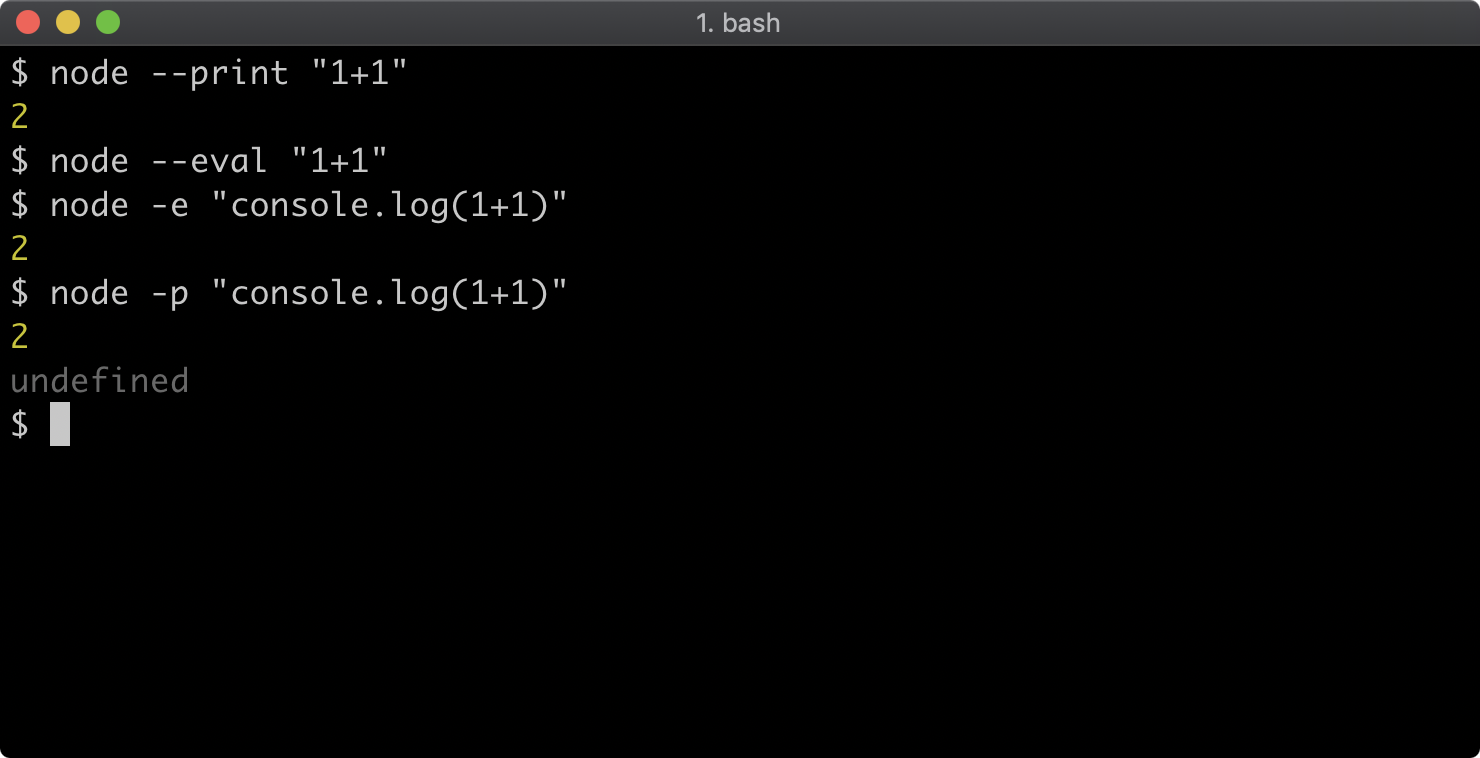
**node --eval "1+1"**

The following will print *2* because **console.log** is used to explicitly write the result of 1+1 to the terminal:

**node -e "console.log(1+1)"**

When used with print flag the same will print *2* and then print *undefined* because **console.log** returns *undefined*; so the result of the expression is **undefined**:

**node -p "console.log(1+1)"**



Usually a module would be required, like so: **require('fs')**, however all Node core modules can be accessed by their namespaces within the code evaluation context.

For example, the following would print all the files with a **.js** extension in the current working directory in which the command is run:

**node -p "fs.readdirSync('.').filter((f) => /.js$/.test(f))"**

Due to the fact that Node is cross-platform, this is a consistent command that can be used on Linux, MacOS or Windows. To achieve the same effect natively on each OS a different approach would be required for Windows vs Linux and Mac OS.