

# Global objects in Node.js

The Global Concept (Global) means everything has access to it. In Node.js, too, global objects are available for all modules. We do not need to declare or import them by the `require ()` method but can still use them directly.

The Global Concept (Global) means everything has access to it. In Node.js, too, global objects are available for all modules. We do not need to declare or import them by the `require ()` method but can still use them directly. These objects can be Modules, functions, strings or objects.

## `__filename` in Node.js

In Node.js, `__filename` represents the file name of the code being executed. This is the absolute path to the file containing this code.

### For example

To illustrate `__filename` in Node.js, create `main.js` with the following simple content:

```
// Print list of __filename in Node.js console . log ( __filename );
```

Run `main.js` to see the result:

```
$ node main . js
```

The result of printing is the path to `main.js` containing the above code:

```
/web/com/1427091028_21099/main.js
```

## `__dirname` in Node.js

In Node.js, `__dirname` represents the directory containing the code that is being executed.

### For example

To illustrate getting the information of `__dirname` in Node.js, create `main.js` containing the following simple code:

```
// Print list of __dirname in Node.js console . log ( __dirname );
```

Run `main.js` to see the result ::

```
$ node main . js
```

The result is the current directory containing `main.js`:

## SetTimeout function (cb, ms) in Node.js

The global **setTimeout** function (**cb**, **ms**) is used to run a callback function named **cb** after a milisecond interval.

This global function returns a value representing the Timer so that it can be deleted by the `clearTimeout ()` function.

### For example

To illustrate the usage of the `setTimeout ()` function in Node.js, create `main.js` with the following content:

```
function printHello () { console . log ( "Hello World!" ); } // Fly to print
```

Run `main.js` to see the result ::

```
$ node main . js
```

The result will appear after about 2s. This time period may be longer than 2s because the delay time is also affected by external factors such as OS Timer:

```
Hello World!
```

## The clearTimeout (t) function in Node.js

The **clearTimeout** global function (**t**) is used to stop a Timer created by the previous `setTimeout ()` function. The parameter **t** is the Timer returned from the `setTimeout ()` function.

### For example

To delete the previously set Timer, create `main.js` with the following simple content. First, we set the Timer by the `setTimeout ()` function, then use the `clearTimeout ()` function to delete this value:

```
function printHello () { console . log ( "Hello World!" ); } // Fly to print
```

Run `main.js` to see the result ::

```
$ node main . js
```

No results will be displayed on the terminal.

## The setInterval function (cb, ms) in Node.js

The global function **setInterval** (**cb**, **ms**) is used to run the callback function named **cb** repeatedly after a millisecond **ms** interval.

This global function returns a value representing the Timer so that it can be deleted by the **clearInterval (t) function**.

### For example

To see the usage of the Node.js setInterval () function, create main.js with the following simple content:

```
function printHello () { console . log ( "Hello World!" ); } // Fly, we call
```

Run main.js to see the result ::

```
$ node main . js
```

The above program will execute the printHello () function every 2 seconds.

## Some global objects in Node.js

In addition to the above global properties and functions, Node.js also includes some other global objects. Follow the following link to find out the details:

**Console object** : Used to print information on stdout and stderr.

**Process object** : Used to get information about the current process.

### According to Tutorialspoint

Previous post: Stream in Node.js

Next article: Utility Module in Node.js

You finished reading the article "**Global objects in Node.js**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.