Node.js forEach – Examples

Node.js forEach

Node.js forEach is used to execute the provided function for each element.

Syntax – forEach

The syntax of forEach is;

```
let arr = [element1,
```

```
let arr = [element1, element2, elementN];

arr.forEach(myFunction(element, index, array, this){ function body });
```

`myFunction` function is executed for each `element` in `arr`. The `element` of array is passed as argument to the function during each iteration.

Example 1: forEach on Array of elements

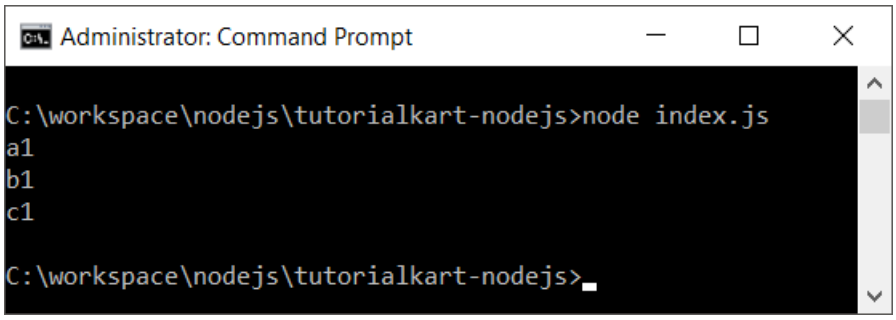
In this example, we will use forEach to apply on each element of array.

```
let array1 = ['a1', 'b1',
```

```
let array1 = ['a1', 'b1', 'c1'];

array1.forEach(function(element) {
  console.log(element);
});
```

Output



Example 2: forEach on Array of elements with external function passed as argument

In this example, we will use forEach to apply on each element of array. And we define the function separately and pass as argument to forEach.

```
let array1 = ['a1', 'b1',
```

```
let array1 = ['a1', 'b1', 'c1']

let myFunc = function(element) {
  console.log(element)
}

array1.forEach(myFunc)
```

Example 3: forEach on Array with access to Element, Index and Array

In this example, we will access index and array along with element in each iteration.

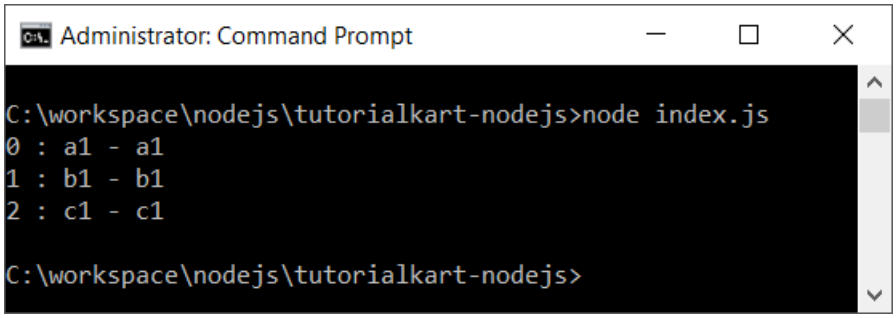
```
let array1 = ['a1', 'b1',
```

```
let array1 = ['a1', 'b1', 'c1']

let myFunc = function(element, index, array) {
  console.log(index + ' : ' + element + ' - ' + array[index])
}

array1.forEach(myFunc)
```

Output



Get Started With Node.js

- Install Node.js Ubuntu Linux
- Install Node.js Windows
- Node.js - Basic Example
- Node.js - Command Line Arguments
- Node.js - Modules
- Node.js - Create a module
- Node.js - Add new functions to Module
- Node.js - Override functions of Module
- Node.js - Callback Function
- Node.js - forEach

Express.js

- Express.js Tutorial
- What is Express.js?
- Express.js Application Example
- Install Express.js
- Express.js Routes
- Express.js Middleware
- Express.js Router

Node.js Buffers

- Node.js Buffer - Create, Write, Read
- Node.js Buffer - Length
- Node.js - Convert JSON to Buffer
- Node.js - Array to Buffer

Node.js HTTP

- Node.js - Create HTTP Web Server
- Node.js - Redirect URL

Node.js MySQL

- Node.js MySQL
- Node.js MySQL - Connect to MySQL Database
- Node.js MySQL - SELECT FROM
- Node.js MySQL - SELECT WHERE
- Node.js MySQL - ORDER BY
- Node.js MySQL - INSERT INTO

Node.js MySQL - INSERT INTO
Node.js MySQL - UPDATE
Node.js MySQL - DELETE
Node.js MySQL - Result Object

Node.js MongoDB

Node.js MongoDB
Node.js - Connect to MongoDB
Node.js - Create Database in MongoDB
Node.js - Drop Database in MongoDB
Node.js - Create Collection in MongoDB
Node.js - Delete Collection in MongoDB
Node.js - Insert Documents to MongoDB Collection
MongoError: failed to connect to server

Node.js Mongoose

Node.js Mongoose Tutorial
Node.js Mongoose - Installation
Node.js Mongoose - Connect to MongoDB
Node.js Mongoose - Define a Model
Node.js Mongoose - Insert Single Document to MongoDB
Node.js Mongoose - Insert Multiple Documents to MongoDB

Node.js URL

Node.js - Parse URL parameters

Node.js FS (File System)

Node FS
Node FS - Read a File
Node FS - Create a File
Node FS - Write to a File
Node FS - Append to a File
Node FS - Rename a File
Node FS - Delete a File
Node FS Extra - Copy a Folder

Node.js JSON

Node.js Parse JSON
Node.js Write JSON Object to File

Node.js Error Handling

- Node.js Try Catch

Node.js Examples

- Node.js Examples
- Node.js - Handle Get Requests
- Node.js Example - Upload files to Node.js server

Useful Resources

- Node.js Interview Questions
- How to Learn Programming