

tutorialspoint
SIMPLY EASY LEARNING

www.tutorialspoint.com



<https://www.facebook.com/tutorialspointindia>



<https://twitter.com/tutorialspoint>

About the Tutorial

Node.js is a very powerful JavaScript-based framework/platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications. Node.js is open source, completely free, and used by thousands of developers around the world.

Audience

This tutorial is designed for software programmers who want to learn the basics of Node.js and its architectural concepts. This tutorial will give you enough understanding on all the necessary components of Node.js with suitable examples.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of JavaScript. As we are going to develop web-based applications using Node.js, it will be good if you have some understanding of other web technologies such as HTML, CSS, AJAX, etc.

Execute Node.js Online

For most of the examples given in this tutorial, you will find a **Try it** option, so just make use of this option to execute your Node.js programs on the spot and enjoy your learning.

Try the following example using the Try it option available at the top right corner of the below sample code box (on our website):

```
/* Hello World! program in Node.js */  
console.log("Hello World!");
```

Copyright & Disclaimer

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Execute Node.js Online.....	i
Copyright & Disclaimer	i
Table of Contents	ii
1. Introduction.....	1
What is Node.js?.....	1
Features of Node.js	1
Who Uses Node.js?.....	2
Concepts.....	2
Where to Use Node.js?.....	2
Where Not to Use Node.js?.....	2
2. Environment Setup	3
Try it Option Online	3
Local Environment Setup.....	3
Text Editor	3
The Node.js Runtime	3
Download Node.js Archive	4
Installation on UNIX/Linux/Mac OS X and SunOS.....	4
Installation on Windows.....	4
Verify Installation: Executing a File.....	5
3. First Application.....	6
Creating Node.js Application	6
4. REPL Terminal	9
Online REPL Terminal	9
REPL Commands	11
Stopping REPL.....	11
5. NPM.....	12
Installing Modules using NPM	12
Global vs Local Installation	12
Using package.json	14
Attributes of Package.json.....	19
Uninstalling a Module	19
Updating a Module.....	19
Search a Module.....	19
Create a Module.....	19
6. Callback Concept.....	21
What is Callback?.....	21
Blocking Code Example.....	21
Non-Blocking Code Example.....	22

7. Event Loop	23
Event-Driven Programming	23
Example	24
How Node Applications Work?	25
8. Event Emitter	26
EventEmitter Class	26
Methods	26
Class Methods	27
Events	28
Example	28
9. Buffers	30
Creating Buffers	30
Writing to Buffers	30
Reading from Buffers	31
Convert Buffer to JSON	32
Concatenate Buffers	33
Compare Buffers	33
Copy Buffer	34
Slice Buffer	35
Buffer Length	36
Methods Reference	36
Class Methods	41
10. Streams	43
What are Streams?	43
Reading from a Stream	43
Writing to a Stream	44
Piping the Streams	45
Chaining the Streams	46
11. File System	48
Synchronous vs Asynchronous	48
Open a File	49
Get File Information	51
Writing a File	53
Reading a File	54
Closing a File	56
Truncate a File	57
Delete a File	59
Create a Directory	60
Read a Directory	61
Remove a Directory	62
Methods Reference	63
12. Global Objects	69
__filename	69
__dirname	69
setTimeout(cb, ms)	70
clearTimeout(t)	70
setInterval(cb, ms)	71
Global Objects	71

Console Object	72
Process Object	74
13. Utility Modules	81
OS Module	81
Path Module	83
Net Module.....	85
DNS Module.....	92
Domain Module.....	95
14. Web Module	99
What is a Web Server?	99
Web Application Architecture	99
Creating a Web Server using Node.....	100
Make a request to Node.js server	102
Creating a Web client using Node	102
15. Express Framework.....	104
Express Overview	104
Installing Express	104
Hello world Example.....	104
Request & Response.....	106
Request Object	106
Response Object.....	109
Basic Routing	115
Serving Static Files	118
GET Method.....	119
POST Method.....	121
File Upload.....	123
Cookies Management.....	125
16. RESTful API	126
What is REST Architecture?	126
HTTP methods	126
RESTful Web Services	126
Creating RESTful for a Library	126
List Users	128
Add Users	129
Show Detail.....	130
Delete a User	131
17. Scaling an Application	133
The exec() method.....	133
The spawn() Method	135
The fork() Method	137
18. Packaging.....	139
JXcore Installation	139
Packaging the Code	140
Launching JX File.....	140

About the Tutorial

Node.js is a very powerful JavaScript-based framework/platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications. Node.js is open source, completely free, and used by thousands of developers around the world.

Audience

This tutorial is designed for software programmers who want to learn the basics of Node.js and its architectural concepts. This tutorial will give you enough understanding on all the necessary components of Node.js with suitable examples.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of JavaScript. As we are going to develop web-based applications using Node.js, it will be good if you have some understanding of other web technologies such as HTML, CSS, AJAX, etc.

Execute Node.js Online

For most of the examples given in this tutorial, you will find a **Try it** option, so just make use of this option to execute your Node.js programs on the spot and enjoy your learning.

Try the following example using the Try it option available at the top right corner of the below sample code box (on our website):

```
/* Hello World! program in Node.js */  
console.log("Hello World!");
```

Copyright & Disclaimer

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial	i
Audience.....	i
Prerequisites.....	i
Execute Node.js Online.....	i
Copyright & Disclaimer	i
Table of Contents	ii
1. Introduction.....	1
What is Node.js?.....	1
Features of Node.js	1
Who Uses Node.js?.....	2
Concepts.....	2
Where to Use Node.js?.....	2
Where Not to Use Node.js?.....	2
2. Environment Setup	3
Try it Option Online	3
Local Environment Setup.....	3
Text Editor	3
The Node.js Runtime	3
Download Node.js Archive	4
Installation on UNIX/Linux/Mac OS X and SunOS.....	4
Installation on Windows.....	4
Verify Installation: Executing a File.....	5
3. First Application.....	6
Creating Node.js Application	6
4. REPL Terminal	9
Online REPL Terminal	9
REPL Commands	11
Stopping REPL.....	11
5. NPM.....	12
Installing Modules using NPM	12
Global vs Local Installation	12
Using package.json	14
Attributes of Package.json.....	19
Uninstalling a Module	19
Updating a Module.....	19
Search a Module.....	19
Create a Module.....	19
6. Callback Concept.....	21
What is Callback?.....	21
Blocking Code Example.....	21
Non-Blocking Code Example.....	22

7. Event Loop	23
Event-Driven Programming	23
Example	24
How Node Applications Work?	25
8. Event Emitter	26
EventEmitter Class	26
Methods	26
Class Methods	27
Events	28
Example	28
9. Buffers	30
Creating Buffers	30
Writing to Buffers	30
Reading from Buffers	31
Convert Buffer to JSON	32
Concatenate Buffers	33
Compare Buffers	33
Copy Buffer	34
Slice Buffer	35
Buffer Length	36
Methods Reference	36
Class Methods	41
10. Streams	43
What are Streams?	43
Reading from a Stream	43
Writing to a Stream	44
Piping the Streams	45
Chaining the Streams	46
11. File System	48
Synchronous vs Asynchronous	48
Open a File	49
Get File Information	51
Writing a File	53
Reading a File	54
Closing a File	56
Truncate a File	57
Delete a File	59
Create a Directory	60
Read a Directory	61
Remove a Directory	62
Methods Reference	63
12. Global Objects	69
__filename	69
__dirname	69
setTimeout(cb, ms)	70
clearTimeout(t)	70
setInterval(cb, ms)	71
Global Objects	71

Console Object	72
Process Object	74
13. Utility Modules	81
OS Module	81
Path Module	83
Net Module.....	85
DNS Module.....	92
Domain Module.....	95
14. Web Module	99
What is a Web Server?	99
Web Application Architecture	99
Creating a Web Server using Node.....	100
Make a request to Node.js server	102
Creating a Web client using Node	102
15. Express Framework.....	104
Express Overview	104
Installing Express	104
Hello world Example.....	104
Request & Response.....	106
Request Object	106
Response Object.....	109
Basic Routing	115
Serving Static Files	118
GET Method.....	119
POST Method.....	121
File Upload.....	123
Cookies Management.....	125
16. RESTful API	126
What is REST Architecture?	126
HTTP methods	126
RESTful Web Services	126
Creating RESTful for a Library	126
List Users	128
Add Users	129
Show Detail.....	130
Delete a User	131
17. Scaling an Application	133
The exec() method.....	133
The spawn() Method	135
The fork() Method	137
18. Packaging.....	139
JXcore Installation	139
Packaging the Code	140
Launching JX File.....	140

1. Introduction

What is Node.js?

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36. The definition of Node.js as supplied by its [official documentation](#) is as follows:

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Node.js = Runtime Environment + JavaScript Library

Features of Node.js

Following are some of the important features that make Node.js the first choice of software architects.

- **Asynchronous and Event Driven** – All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- **Very Fast** – Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- **Single Threaded but Highly Scalable** – Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- **No Buffering** – Node.js applications never buffer any data. These applications simply output the data in chunks.
- **License** – Node.js is released under the [MIT license](#).

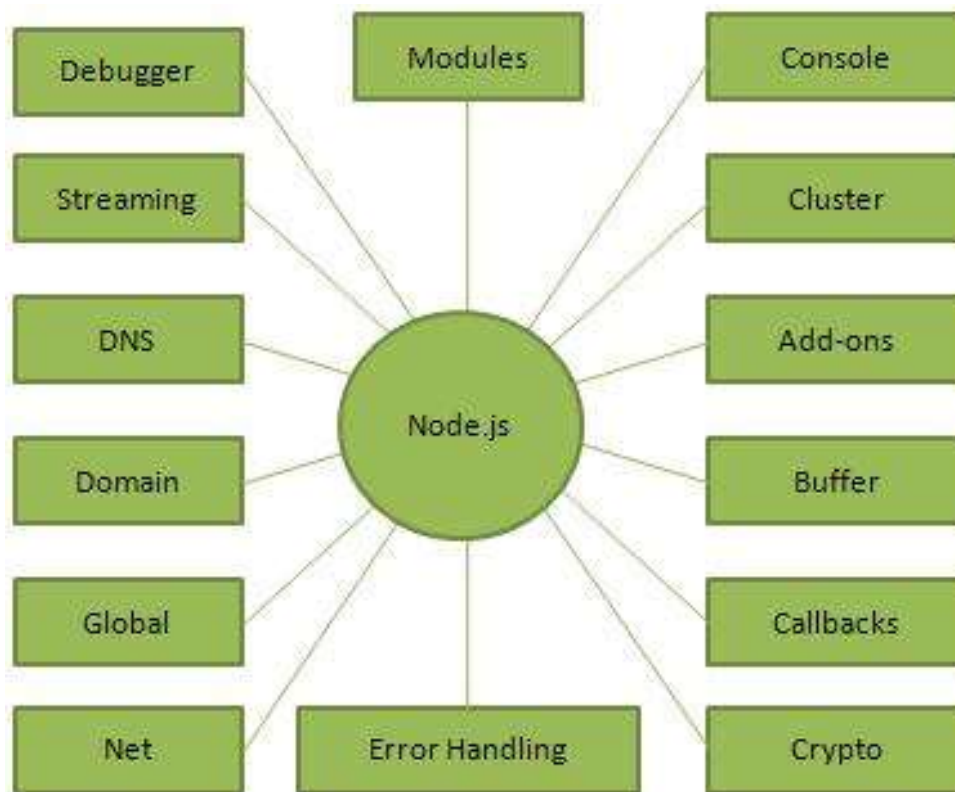
Who Uses Node.js?

Following is the link on github wiki containing an exhaustive list of projects, application and companies which are using Node.js. This list includes eBay, General Electric, GoDaddy, Microsoft, PayPal, Uber, Wikipins, Yahoo!, and Yammer to name a few.

- [Projects, Applications, and Companies Using Node](#)

Concepts

The following diagram depicts some important parts of Node.js which we will discuss in detail in the subsequent chapters.



Where to Use Node.js?

Following are the areas where Node.js is proving itself as a perfect technology partner.

- I/O bound Applications
- Data Streaming Applications
- Data Intensive Real-time Applications (DIRT)
- JSON APIs based Applications
- Single Page Applications

Where Not to Use Node.js?

It is not advisable to use Node.js for CPU intensive applications.