* **Introduction:** Node.js is an open-source and cross-platform runtime environment for executing [JavaScript](https://www.geeksforgeeks.org/JavaScript-tutorial/) code outside a browser. You need to remember that **NodeJS is not a framework and it’s not a programming language**. Most people are confused and understand it’s a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It’s used in production by large companies such as Paypal, Uber, Netflix, Walmart, and so on.
* **Features of NodeJS:**There are other programming languages also which we can use to build back-end services so what makes Node.js different I am going to explain.

1. It’s easy to get started and can be used for prototyping and agile development
2. It provides fast and highly scalable services
3. It uses JavaScript everywhere, so it’s easy for a JavaScript programmer to build back-end services using Node.js
4. Source code cleaner and consistent.
5. Large ecosystem for open source library.
6. It has Asynchronous or Non-blocking nature.

* **Advantages of NodeJS:**Here are the benefits of using Node.js

1. **Easy Scalability:** Developers prefer to use Node.js because it is easily scaling the application in both horizontal and vertical directions. We can also add extra resources during the scalability of the application.
2. **Real-time web apps:** If you are building a web app you can also use PHP, and it will take the same amount of time when you use Node.js, But if I am talking about building chat apps or gaming apps Node.js is much more preferable because of faster synchronization. Also, the event loop avoids HTTP overloaded for Node.js development.
3. **Fast Suite:** NodeJs runs on the V8 engine developed by Google. Event loop in NodeJs handles all asynchronous operation so NodeJs acts like a fast suite and all the operations can be done quickly like reading or writing in the database, network connection, or file system
4. **Easy to learn and code:** NodeJs is easy to learn and code because it uses JavaScript. If you are a front-end developer and have a good grasp of JavaScript you can easily learn and build the application on NodeJS
5. **Advantage of Caching:**It provides the caching of a single module. Whenever there is any request for the first module, it gets cached in the application memory, so you don’t need to re-execute the code.
6. **Data Streaming:** In NodeJs HTTP request and response are considered as two separate events. They are data stream so when you process a file at the time of loading it will reduce the overall time and will make it faster when the data is presented in the form of transmissions. It also allows you to stream audio and video files at lightning speed.
7. **Hosting:** PaaS (Platform as a Service) and Heroku are the hosting platforms for NodeJS application deployment which is easy to use without facing any issue.
8. **Corporate Support:** Most of the well-known companies like Walmart, Paypal, Microsoft, Yahoo are using NodeJS for building the applications. NodeJS uses JavaScript, so most of the companies are combining front-end and backend Teams together into a single unit.

* **Application of NodeJS:**NodeJS should be preferred to build:
* Real-Time Chats,
* Complex Single-Page applications,
* Real-time collaboration tools,
* Streaming apps
* JSON APIs based application
* A **callback** is a function which is called when a task is completed, thus helps in preventing any kind of blocking and a callback function allows other code to run in the meantime. Callback is called when task get completed and is asynchronous equivalent for a function. Using Callback concept, Node.js can process a large number of requests without waiting for any function to return the result which makes Node.js highly scalable. For example: In Node.js, when a function start reading file, it returns the control to execution environment immediately so that the next instruction can be executed. Once file I/O gets completed, callback function will get called to avoid blocking or wait for File I/O.
* Event in node.js

According to the official documentation of Node.js, it is an asynchronous event-driven JavaScript runtime. Node.js has an **event-driven architecture** which can perform asynchronous tasks. Node.js has **‘events’** module which emits named events that can cause corresponding functions or callbacks to be called. Functions(Callbacks) listen or subscribe to a particular event to occur and when that event triggers, all the callbacks subscribed to that event are fired one by one in order to which they were registered. **The EventEmmitter class:** All objects that emit events are instances of the EventEmitter class. The event can be emitted or listen to an event with the help of EventEmitter. **Syntax:**

const EventEmitter=require('events');

var eventEmitter=new EventEmitter();

**Listening events:** Before emits any event, it must register functions(callbacks) to listen to the events. **Syntax:**

eventEmitter.addListener(event, listener)

eventEmitter.on(event, listener)

eventEmitter.once(event, listener)

**eventEmmitter.on(event, listener)** and **eventEmitter.addListener(event, listener)** are pretty much similar. It adds the listener at the end of the listener’s array for the specified event. Multiple calls to the same event and listener will add the listener multiple times and correspondingly fire multiple times. Both functions return emitter, so calls can be chained. **eventEmitter.once(event, listener)** fires at most once for a particular event and will be removed from listeners array after it has listened once. Returns emitter, so calls can be chained. **Emitting events:** Every event is named event in nodejs. We can trigger an event by emit(event, [arg1], [arg2], […]) function. We can pass an arbitrary set of arguments to the listener functions. **Syntax:**

eventEmitter.emit(event, [arg1], [arg2], [...])

* **MongoDB**, the most popular NoSQL database, is an open-source document-oriented database. The term ‘NoSQL’ means ‘non-relational’. It means that MongoDB isn’t based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data. This format of storage is called BSON ( similar to JSON format).

A simple MongoDB document Structure:

{

title: 'Geeksforgeeks',

by: 'Harshit Gupta',

url: 'https://www.geeksforgeeks.org',

type: 'NoSQL'

}

SQL databases store data in tabular format. This data is stored in a predefined data model which is not very much flexible for today’s real-world highly growing applications. **Modern applications are more networked, social and interactive than ever**. Applications are storing more and more data and are accessing it at higher rates.

Relational Database Management System(RDBMS) i**s not the correct choice when it comes to handling big data by the virtue of their design since they are not horizontally scalable**. If the database runs on a single server, then it will reach a scaling limit. NoSQL databases are more scalable and provide superior performance. MongoDB is such a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model.

* **RDBMS vs MongoDB:**
* RDBMS has a typical schema design that shows number of tables and the relationship between these tables whereas MongoDB is document-oriented. There is no concept of schema or relationship.
* Complex transactions are not supported in MongoDB because complex join operations are not available.
* MongoDB allows a highly flexible and scalable document structure. For example, one data document of a collection in MongoDB can have two fields whereas the other document in the same collection can have four.
* MongoDB is faster as compared to RDBMS due to efficient indexing and storage techniques.
* There are a few terms that are related in both databases. What’s called Table in RDBMS is called a Collection in MongoDB. Similarly, a Table is called a Document and A Column is called a Field. MongoDB provides a default ‘\_id’ (if not provided explicitly) which is a 12-byte hexadecimal number that assures the uniqueness of every document. It is similar to the Primary key in RDBMS.
* **Features of MongoDB:**
* **Document Oriented**: MongoDB stores the main subject in the minimal number of documents and not by breaking it up into multiple relational structures like RDBMS. For example, it stores all the information of a computer in a single document called Computer and not in distinct relational structures like CPU, RAM, Hard disk, etc.
* **Indexing**: Without indexing, a database would have to scan every document of a collection to select those that match the query which would be inefficient. So, for efficient searching Indexing is a must and MongoDB uses it to process huge volumes of data in very less time.
* **Scalability**: MongoDB scales horizontally using sharding (partitioning data across various servers). Data is partitioned into data chunks using the shard key, and these data chunks are evenly distributed across shards that reside across many physical servers. Also, new machines can be added to a running database.
* **Replication and High Availability**: MongoDB increases the data availability with multiple copies of data on different servers. By providing redundancy, it protects the database from hardware failures. If one server goes down, the data can be retrieved easily from other active servers which also had the data stored on them.
* **Aggregation**: Aggregation operations process data records and return the computed results. It is similar to the GROUPBY clause in SQL. A few aggregation expressions are sum, avg, min, max, etc
* **Where do we use MongoDB?**

MongoDB is preferred over RDBMS in the following scenarios:

* **Big Data**: If you have huge amount of data to be stored in tables, think of MongoDB before RDBMS databases. MongoDB has built-in solution for partitioning and sharding your database.
* **Unstable Schema**: Adding a new column in RDBMS is hard whereas MongoDB is schema-less. Adding a new field does not effect old documents and will be very easy.
* **Distributed data** Since multiple copies of data are stored across different servers, recovery of data is instant and safe even if there is a hardware failure.

Advantages Of MongoDB

#### **a. Flexible Database**

We know that MongoDB is a schema-less database. That means we can have any type of data in a separate document. This thing gives us flexibility and a freedom to store data of different types.

#### **b. Sharding**

We can store a large data by distributing it to several servers connected to the application. If a server cannot handle such a  big data then there will be no failure condition. The term we can use here is “auto-sharding”.

#### **High Speed**

MongoDB is a document-oriented database. It is easy to access documents by indexing. Hence, it provides fast query response. The speed of MongoDB is 100 times faster than the relational database.

#### **High Availability**

MongoDB has features like replication and gridFS. These features help to increase data availability in MongoDB. Hence the performance is very high.

#### **e. Scalability**

A great advantage of MongoDB is that it is a horizontally scalable database. When you have to handle a large data, you can distribute it to several machines.

#### **Easy Environment Setup**

It is easier to setup MongoDB then RDBMS. It also provides JavaScript client for queries.

#### **Full Technical Support**

MongoDB Inc. provides professional support to its clients. If there is any problem, you can directly reacha MongoDB client support system.