Javascript

The language of Wizards!

Your hosts

Shubham Akolkar

Full Stack Developer
Team Javascript

Vartika Shukla

Full Stack Developer
Team Javascript

What is Javascript?

Introduction

- It is a cross-platform, object-oriented scripting language.
- It is Interpreted language.
- It was initially designed for enhancing the interaction of a user with the web page.
- It is supported by almost all browsers e.g Chrome, Mozilla firefox, Safari etc.
- It runs at both client side as well as server side.
- It is event driven.
- It is type safe programming language.

History Of Javascript

1995 **JavaScript** is born as LiveScript

1997 **ECMAScript** standard is established

1999 **ES3** comes out and IE5 is all the rage

2000 XMLHttpRequest, a.k.a. AJAX, gains popularity

2009 **ES5** comes out and standard JSON

2015 **ES6**/ECMAScript2015 comes out

2016 **ES7**/ECMAScript2016 comes out

2017 ES.Next

History of Javascript

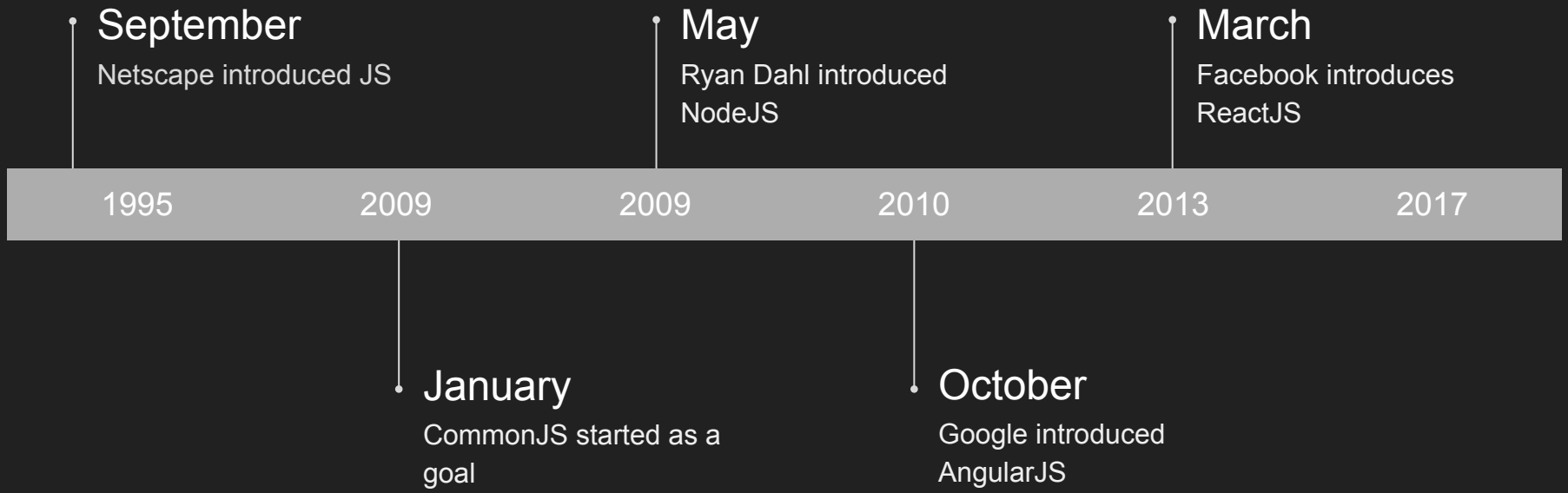
- Javascript was developed by Brendan Eich at Netscape in 1995 and has appeared in all browsers since 1996.
- It was developed within 6 months.
- The official standardization was adopted by ECMA organization in 1997.
- In 1999 ES 3 was released
- Since 2000 it started supporting XML HTTP requests (AJAX)
- Progress never stopped since then...

jQuery

What is jQuery

- It is free, open source javascript library.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- It takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- It also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

Growth of Javascript



Growth of Javascript

- Most popular technology in StackOverflow's developer survey
- Second most "wanted" language
- In the open source community, more JavaScript code is uploaded to GitHub than any other language
- JavaScript tops the charts in number of active repositories and number of commits / day.
- With tools like Johnny-Five, you can use JavaScript as a language for programming embedded systems
- You can even prototype Virtual Reality applications
- Not only has the number of platforms increased, but the number of ways you can write JavaScript has increased.
- You could write ClojureScript or Elm; if you prefer strong typing, you could write TypeScript or augment your existing codebase with Flow

EcmaScript

What is EcmaScript

- ECMAScript is a trademarked scripting-language specification standardized by Ecma International. It was created to standardize JavaScript, so as to foster multiple independent implementations
- The history is, Brendan Eich created Mocha which became LiveScript, and later JavaScript. Netscape presented JavaScript to Ecma International, which develops standards and it was renamed to ECMA-262 aka ECMAScript
- JavaScript has remained the best-known implementation of ECMAScript since the standard was first published

Difference in JS and ES

- Javascript is a language while EcmaScript is a set of standards which any language needs to follow in order to be used as Browser scripting language
- Javascript is nothing but implementation of standards defined by ECMA international

Versions of Ecmascript

- June 1997: First edition
- December 1999: Introduced Regex, try/catch, etc.
- Fourth Edition was abandoned, due to political differences concerning language complexity.
- June 2011: ES 5.1 was released
- June 2015: ES6 or EcmaScript 2015 was released, introduced classes and modules, for/of loop, promises, etc.
- June 2016: ES7 or EcmaScript 2016 was released, introduced exponentiation operator (**) and Array.prototype.includes
- June 2017: ES8 or Ecmascript 2017 was released, introduced concurrency, atomics, async/await, observable streams, etc.

Angular JS

What is Angular

- AngularJS is a JavaScript framework. It is a library written in JavaScript
- AngularJS is distributed as a JavaScript file, and can be added to a web page with a script tag
- Superheroic Javascript MV* Framework

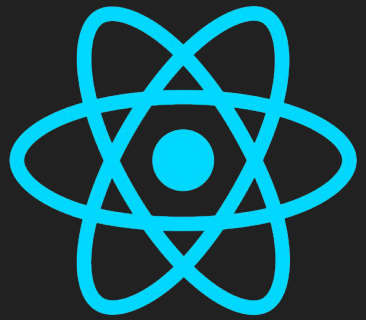
Why use Angular

- It allows you to create dynamic websites
- It is way faster
- It helps you write easy, maintainable and reusable piece of codes
- It manipulates DOM for you
- It helps you escape the headache of writing Ajax calls and the succeeding DOM manipulation writing
- Testability

Versions of Angular

- Angular 1.x: Uses services, providers, factories, $scope, directives, controller, etc
- Angular 2: Introduced Mobile support, SEO friendly, Components, code structure, etc.
- Angular 4: Introduced TypeScript, observables, AOT, View engine, reduced bundle size
- Angular 5: Introduces performance enhancements, reduced bundle size, etc.

React JS

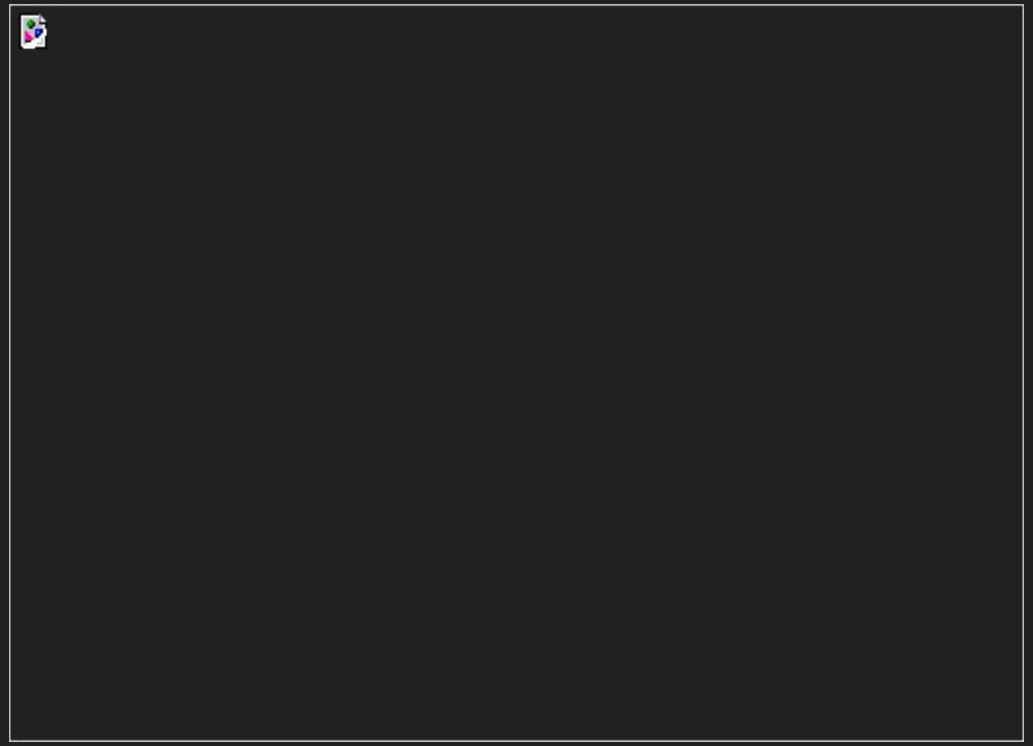


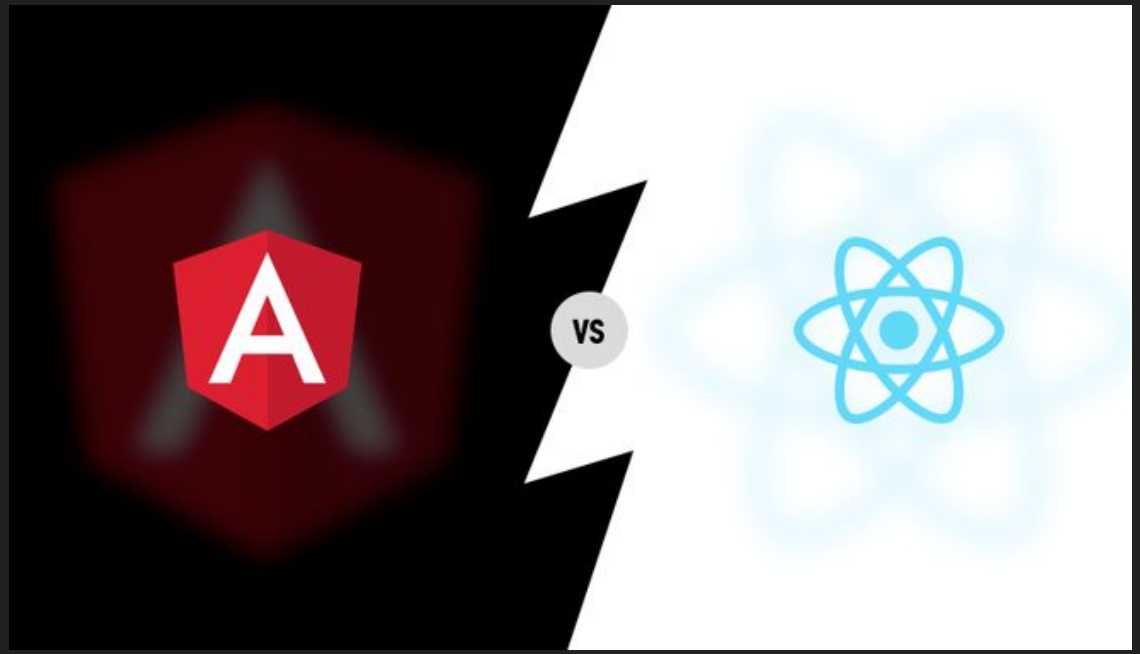
React Js

- It is developed at facebook
- It is JavaScript **library** used for building reusable UI components
- It is used as “V” in MVC
- It follows one way reactive flow
- It uses states and props to render dynamic data
- It follows JSX (Javascript Syntax Extension)
- It uses concept called Virtual Dom

Redux

- Redux is simplified implementation of Flux system
- App state is stored in single object tree i.e “Store”
- To change state emit an “Action” - describing the change
- “Reducers” receive actions and mutate “Store”
- There’s one more element “Container” which connects component with “Action” and “Reducer”





Angular vs React

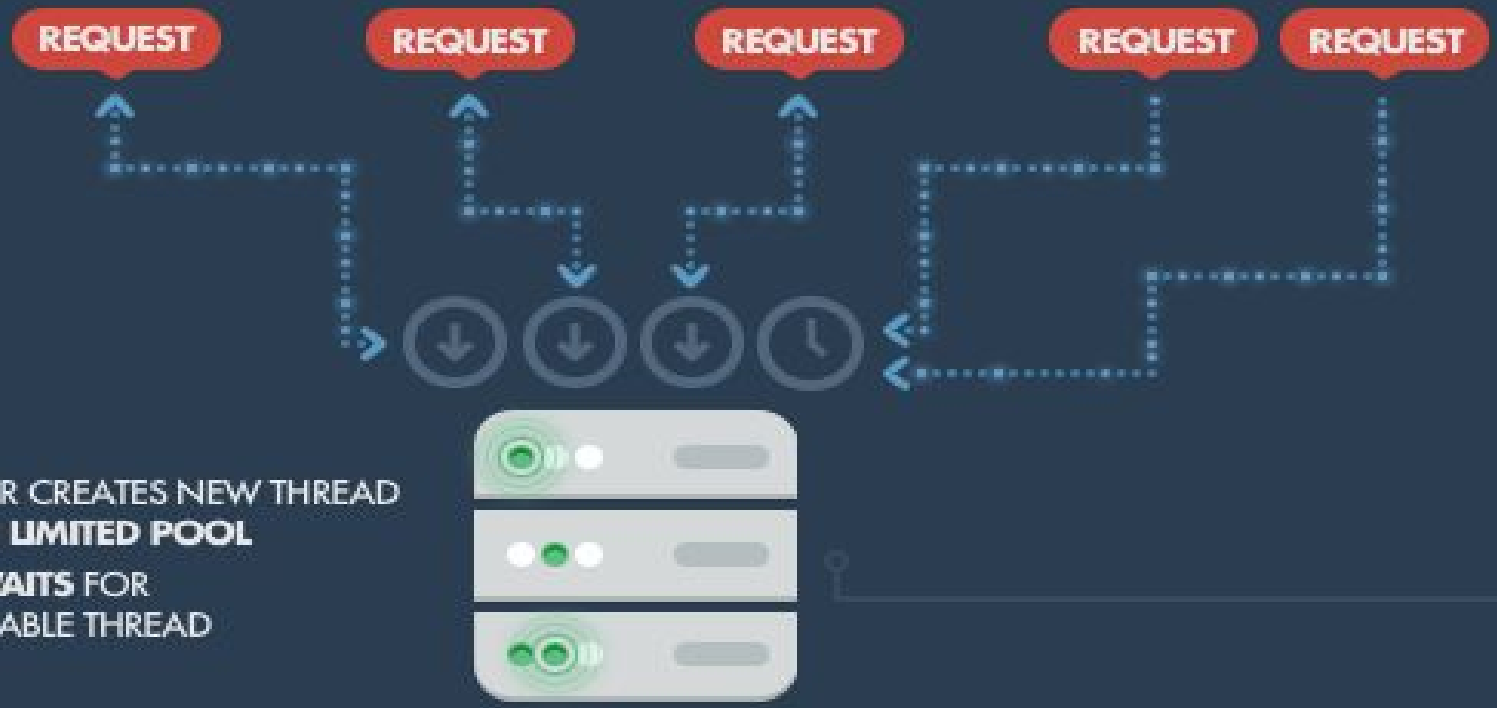
Angular vs React

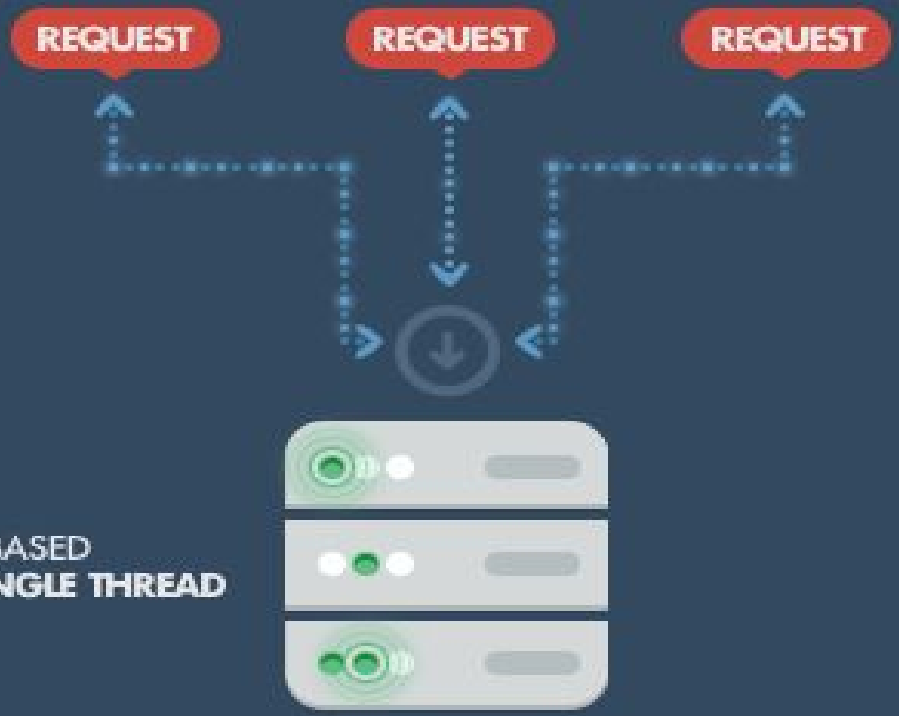
- Angular is a Framework while React is a library
- React supports large tool chain
- Angular is component based, while you have to combine React with Redux in order to attain state management
- Angularjs uses regular DOM while React uses Virtual DOM
- Angular is simple to setup and work with as it follows proper conventions while in React, there is no specific way to achieve a task, which leads us to no standardisation
- Angular maintains dependencies itself while React uses tools for management
- Angular is JS + HTML while React is JS + JSX

NodeJS

What is NodeJS

- Node.js is an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side.
- The first server-side JavaScript environment was Netscape's LiveWire Pro Web
- Dahl was inspired to create Node.js after seeing a file upload progress bar on Flickr
- In January 2010, a package manager was introduced for the Node.js environment called npm
- In June 2011, Microsoft and Joyent implemented a native Windows version of Node.js
- In December 2014, Fedor Indutny started io.js, a fork of Node.js
- By June 2015, the Node.js and io.js communities voted to work together under the Node.js Foundation


REQUEST
REQUEST
REQUEST
REQUEST
REQUEST
SERVER CREATES NEW THREAD
FROM **LIMITED POOL**
OR **WAITS** FOR
AVAILABLE THREAD
THREAD
WAITING
TRADITIONAL



HANDLES EVENT-BASED
CALLBACK ON **SINGLE THREAD**

THREAD

NODE.JS

Frameworks of Node Js

- Express Js
- Sails Js
- Meteor Js
- Koa Js
- Hapi Js

Third party Tools

TypeScript

October 2012, TypeScript is a free and open-source programming language developed and maintained by Microsoft. It is a strict syntactical superset of JavaScript, and adds optional static typing to the language.

CoffeeScript

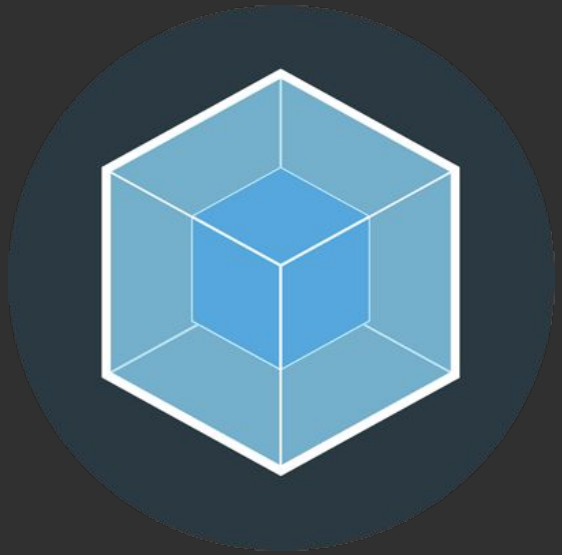
December 2009, CoffeeScript is a progaramming language that transcompiles to JavaScript. It adds syntactic sugar inspired by Ruby, Python and Haskell in an effort to enhance JavaScript's brevity and readability.

Babel

- Transpiles language
- ECMAscript and beyond
- Polyfills to support new object features
-

Webpack

March 2012, Open-source JavaScript module bundler. Webpack takes modules with dependencies and generates static assets representing those modules



Socket.IO

Socket.IO is a JavaScript library for realtime web applications. It enables realtime, bi-directional communication between web clients and servers.

Testing Tools

Testing Tools

- **Jasmine -** Jasmine is a behavior-driven development framework for testing JavaScript code. It does not depend on any other JavaScript frameworks.

- **Karma** - Karma is not a testing framework, nor an assertion library. Karma just launches an HTTP server, and generates the test runner HTML file you probably already know from your favourite testing framework.

- **Protractor -** Protractor is an end-to-end test framework for AngularJS applications. It runs tests against your application running in a real browser, interacting with it as a user would.

Process manager

- **PM2** : Forever, A simple CLI tool for ensuring that a given node script runs continuously (i.e. forever). It was authored by Charlie Robbins on Dec, 2010.

- **Forever:** Production process manager for Node.JS applications with a built-in load balancer. It was authored by Strzelewicz Alexandre on Jun, 2013.

Forever vs PM2

- Forever has been out there for longer (since 7 years ago), it also has a bigger community of maintainers, less dependencies and less open issues.
- Pm2 has more versions, more frequent updates, more daily downloads, more weekly downloads, more monthly downloads, less open pull requests, more stars on Github, more followers on Github and more forks.
- Pm2 is super easy to setup
- Pm2 gives easy to read table of each apps.

NoSQL Database

- Stands for Not Only SQL
- Non relational database
- Very compatible with distributed systems
- High Performance database
- Founded to handle huge data
- Used by Facebook, Google, wikipedia ...

Examples:

- Mongodb
- Neo4J
- Cassandra
- CouchDB
- Redis

MongoDB

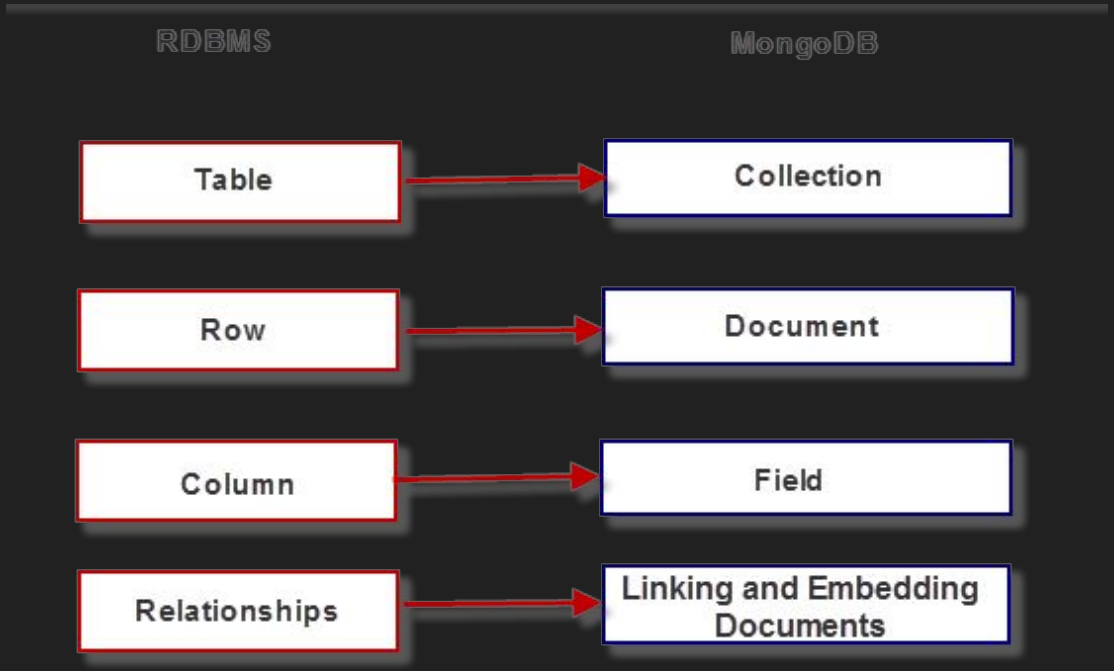
What is MongoDB?

- Scalable High performance Open source, document oriented database
- Built for speed
- Rich document based queries for easy readability
- Full index support for high performance
- Auto sharding for easy scalability
- Schema less
- Map / Reduce for Aggregation

Mongoose

- Mongoose is Object Document Mapper (ODM)
- It provides straight forward, schema based solution to modeling application data.
- It includes built-in type casting, validation, query building, business logic hooks and many more features
- It offers both strict and loose schema approach

Mapping with MySql



What is the difference?

- SQL databases are Relational while No-SQL are not
- SQL databases are Tables, No-SQL databases are documents
- No-SQL databases lacks support for complex queries
- SQL databases are vertically scalable, you can manage increasing load, No-SQL databases are horizontally scalable, you can just add few more servers easily
- Support for transactional queries is more stable for complex data in SQL databases

Any Questions ?

Thank You