1. **[package.json](https://nodesource.com/blog/the-basics-of-package-json-in-node-js-and-npm/)** - versioning and dependencies file
   * Using the ***npm init***command to initialize a new package.json file
   * devdependencies vs dependencies
   * using the ***--save*** while installing a package, we can automatically create a dependence in the package.json file
2. [**semantic versioning**](https://medium.com/@hossam.hilal0/package-json-vs-package-lock-json-vs-npm-shrinkwrap-json-33fcddc1521a) - (patch, minor, major)
   * ^1.2.3 – The '^' means that it will support any higher version that follow the current version and of the same major version (1.2.4, 1.2.5, 1.4.3, ..., 1.9.9)
3. [**package-lock**](https://medium.com/@hossam.hilal0/package-json-vs-package-lock-json-vs-npm-shrinkwrap-json-33fcddc1521a).**json** – Using that file the dependencies will be install with the same version as written in the package-lock file. That way anyone no matter the environment in which the app will run, it will always run with the same versions that you run it in development mode.
4. [**What is require?**](https://nodejs.org/en/knowledge/getting-started/what-is-require/)- reads a JavaScript file, executes the file, and then proceeds to return the exports object
5. [**NodeJs expresss**](https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction) – the most popular *Node* web framework, and is the underlying library for a number of other popular [Node web frameworks](https://expressjs.com/en/resources/frameworks.html). It provides mechanisms to:
   * Write handlers for requests with different HTTP verbs at different URL paths (routes)
   * Integrate with "view" rendering engines in order to generate responses by inserting data into templates
   * Set common web application settings like the port to use for connecting, and the location of templates that are used for rendering the response
   * Add additional request processing "middleware" at any point within the request handling pipeline
6. [**CORS**](https://stackabuse.com/handling-cors-with-node-js/) - It is a mechanism to allow or restrict requested resources on a web server depend on where the HTTP request was initiated.
7. [**Backticks**](https://betterprogramming.pub/javascript-how-backticks-work-de269e0fb8ba?gi=82a4f605ed1f) **(look like this: `)**
8. [**app.listen**](https://stackoverflow.com/questions/33222074/about-app-listen-callback)
9. [**export in NodeJs**](https://www.sitepoint.com/understanding-module-exports-exports-node-js/) &[**module.export vs export**](https://www.hacksparrow.com/nodejs/exports-vs-module-exports.html)
10. [**req.params & req.query & req.body**](https://dev.to/gathoni/express-req-params-req-query-and-req-body-4lpc)
11. [**Mongoose API**](https://mongoosejs.com/docs/api.html#model_Model.findByIdAndUpdate) – It will allow us to see all the functions which is available when using the mongoose model. **With that we can see the functions that are used inside the tutorial.controller.js**