React Training

Trainer
Kavita Gupta

Agenda - Day 1

ES6

NPM

Nodejs

Npm scripts

Npm usage

Nodejs

- Node.js is an open source server environment
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- Node.js uses JavaScript on the server

NPM

NPM stands for Node Package Manager.
npm is a package manager for the JavaScript programming language. npm, Inc. is a subsidiary of GitHub, that provides hosting for software development and version control with the usage of Git. npm is the default package manager for the JavaScript runtime environment Node.js

What is JavaScript

JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript.
JavaScript is also being used widely in game development and Mobile application development.
It is an interpreted programming language with object-oriented capabilities.

History of JavaScript

JavaScript was developed by Brendan Eich in 1995, which appeared in Netscape, a popular browser of that time.

The language was initially called LiveScript and was later renamed JavaScript. There are many programmers who think that JavaScript and Java are the same. In fact, JavaScript and Java are very much unrelated. Java is a very complex programming language whereas JavaScript is only a scripting language. The syntax of JavaScript is mostly influenced by the programming language C.

Run JavaScript

Being a scripting language, JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it. The main advantage of JavaScript is that all modern web browsers support JavaScript

Simple JavaScript Program

```
<html>
<head>
        <title>My First JavaScript code!!!</title>
        <script type="text/javascript">
                alert("Hello World!");
        </script>
</head>
<body>
</body>
</html
```

ES6/ES2015(EcmaScript)

ES6 has come up with new features like classes, arrow functions,scope,rest,spread.
ES6 stands for EcmaScript

Classes

JavaScript Classes are templates for JavaScript Objects.
Use the keyword class to create a class.
A class in terms of OOP is a blueprint for creating objects. A class encapsulates data for the object.

E.g. Car is a class, which can have different models called objects

Default, Rest and Spread

Default function parameters allow formal parameters to be initialized with default values if no value or undefined is passed

The rest parameter syntax allows representing an indefinite number of arguments as an array.

Spread allows you to make a shallow copy of an array or object, meaning that any top level properties will be cloned, but nested objects will still be passed by reference. For simple arrays or objects, a shallow copy may be all you need.

Destructing

Destructing assignment is a syntax that allows you to assign object properties or array items as variables. This can greatly reduce the lines of code necessary to manipulate data in these structures. There are two types of destructuring: Object destructuring and Array destructuring.

Object Destructing

Object destructuring allows you to create new variables using an object property as the value.

Note : Destructuring an object does not modify the original object. You could still call the original note with all its entries intact.

Array destructing

Array destructuring allows you to create new variables using an array item as a value. Nested arrays can also be destructured.

Object.assign()

The Object.assign() method copies all enumerable own properties from one or more source objects to a target object. It returns the target object.

Object Initializer

Objects can be initialized using **new Object(), Object.create(), or using the literal notation** (initializer notation). An object initializer is a comma-delimited list of zero or more pairs of property names and associated values of an object, enclosed in curly braces ({}).

Template Literals

Template literals are string literals allowing embedded expressions. You can use multi-line strings and string interpolation features with them.

- Expression interpolation
- Multi-line strings

Modules

A module organizes a related set of JavaScript code. A module can contain variables and functions. A module is nothing more than a chunk of JavaScript code written in a file. By default, variables and functions of a module are not available for use

Exporting a Module

The export keyword can be used to export components in a module. Exports in a module can be classified as follows −

- Named Exports
- Default Exports

Named Export

Named exports are distinguished by their names. There can be several named exports in a module.

```
//using multiple export keyword
export component1
export component2
...
...
export componentN


//using single export keyword

 export {component1,component2,....,componentN}
```

Default Export

Modules that need to export only a single value can use default exports. There can be only one default export per module.

```
export default component_name
```

Importing a module

To be able to consume a module, use the import keyword. A module can have multiple import statements

```
import {component1,component2..componentN} from
module_name

 import {original_component_name as
 new_component_name }

//default import
 import any_variable_name from module_name
```

Best Practices (Code Organization & Conventions)

Follow the DRY Principle

DRY stands for “Don’t Repeat Yourself.
The same piece of code should not be repeated over and over again.

Best Practices
(Code Organization & Conventions)

Limit line length

Long lines are hard to read. It is a good practice to avoid writing horizontally long lines of code.

Best Practices
(Code Organization & Conventions)

Naming conventions

Use of proper naming conventions is a well known best practice. Is a very common issue where developers use variables like X1, Y1 and forget to replace them with meaningful ones, causing confusion and making the code less readable.

Best Practices
(Code Organization & Conventions)

Keep the code simple

The code should always be simple. Complicated logic for achieving simple tasks is something you want to avoid as the logic one programmer implemented a requirement may not make perfect sense to another.
So, always keep the code as simple as possible.

Best Practices
(Code Organization & Conventions)

Write comments and documentation

commenting your code and providing proper documentation will guide the other developers through the algorithm and logic that you implemented.

Thank You