**Global APIs**

To scale the vue, introduced global api’s

So we creating objects for each use: 🡪objects

1. Vue.config
2. Vue.extend
3. Vue.nextTick
4. Vue.directive
5. Vue.elementDirective
6. Vue.filter
7. Vue.component: to communicate between the different objects.
8. Vue.transition
9. Vue.partial
10. Vue.use
11. Vue.mixin
12. Vue.util

Web-components are by w3c to replace activex.

**Component: reusable UI/UX**

**Vue.component**

1. Definittion:Reusable named vue instance
2. Building complex UI
3. Sharing view(either autonomous are as child)
4. Autonomous object model(mutable: work only in scope of component, immutable: work across the components)
5. Data, methods and UI.

This can be done by using @vue.cli to use components

**Vue Cli**

1. Environment for vue.js apps for server side rendering

* Node runtime for executing and rendering vuejs apps
* Manages
* All dependencies
* All objects for Vuejs
* Node.js Module Resolution for ES6
* Export default <NAME>{…}
* Import <NAME> from ‘URL of file module for <NAME>’
* Dev/Build/Prod Features for vue.js app
* Dev 🡪 Classes/functions access across module
* Build 🡪 Use Node.js module resolution for building all modules
* Prod 🡪 minify the resultant .js file for delivering/rendering to browser
* Package.json, config file for every Node.js package
* npm init –y
* Command is used to create package.json with defaults
* npm init
* Command will provide command-line wizard to create package.json
* Selection in package.json
* devDependencies
* list of all packages those are used during development
* npm install –save-dev <package>

Vue Component

1. using vue.component() object
2. using ES6

<template> html

* export default {

name: ‘<NAME-OF-THE-COMPONENT>’,

components: {

comma separated list of child component

},

Data: function() {

Return{

JSON properties representing component’s data

}

},

Props: {

JSON properties to accepts data from parent component

},

Methods: {

}

}

To pass data from parent to child component using ‘props’

Use v-bind: <props-name>=”<value from parent>”

e.g. if child component is <child> and prop is info

$event(target object and value)

$mount(component/vue instance)

$emit(‘customevent’, payload)