HTML

**1)What is HTML**

* Hyper Text Mark-up Language
* Used for making creative, interactive webpage
* HTML give structure to the webpage or DOM
* HTML is platform independent
* Can add hypertext to the text

**2)Importance of Doctype in HTML.**

* It is a declaration statement and It stands for document type declaration
* It is not an HTML tag
* It tells the browser the type and version of HTML you are using…so browser can understand and load its properties
* Always written on the beginning.
* If we don’t use follow this, the browser opens in quirks mode

**3)Do all tags need a closing tag in HTML**

* No, ex: <br>, <img>, <hr>, <link>,<input>

**4)What do you know about metadata**

* Gives data information
* Metadata will not be displayed in the HTML page
* Used by browser for knowing how to display content and reload page,
* Ie, metadata information are read by search engines
* <meta> tags provide information about the webpage

**5)What is the purpose of <iframe> In HTML**

* Iframe tag specifies an inline frame
* Can be used to include another document, images, videos and interactive media within a webpage
* Ex: <iframe src=”http://www.abc.com”></iframe>

**6)What is the use of <pre></pre> tag in HTML**

* Defines pre-formatted text
* Displays anything given inside a pre tag will display it as it is given

**7)what is <main> tag**

* Specifies the main content of the document
* Should contain only one main tag throughout the document
* It should not be
* The content inside the main tag should be unique and should not be repeating across the document
* Eg: sidebars, navigation links, copyright information, site logos, and search forms.

**8)Use of Unicode consortium**

* Unicode consortium develops the standard, their goal is to replace the existing charset and replace it with the standard Unicode transformation format
* It is a character set
* Character set will convert the character to string

**9)UTF-8**

* It is an encoding format of Unicode charset
* Encoding is the process of converting numbers in to binary numbers
* It is a character set used to implement Unicode
* Default character set used in html is UTF-8
* A character in utf-8 can be of 1-4 bytes’ long
* It is the preferred encoding used for email and webpages

**10)what are Block level and inline element**

* Block level elements are elements which always starts from a new line and take the complete width from left to right----<h1>,<div>,<li>,<p>…etc.
* Inline elements will not start from a new line and will take necessary space only

**11)What are semantic elements in html**

* Elements which have a proper meaning to browser and the developer
* Came in html 5
* Ex: <table>, <form>, <header>…etc.
* Non-semantic will not specify a proper meaning ..Ex: <span>,<div>

**12)Difference between html tag and element**

* Html tag: If there is only a tag is present and nothing is inside it or between the opening and closing.
* Html element: if there is any text or items inside or between.

**13)What is a viewport**

* View port is the visible area in the webpage
* It differs from the larger sized displays like computer screen to mobile screens

**14)Difference between <b> & <strong> and <i> & <em> tags**

1. <strong> and <em> --will give extra importance the element, the other 2 will not.

**15) Types of list in HTML**

1.Ordered list—using <li> tag inside <ol> tag

2.Un-Ordered list—using <li> tag inside <ul> tag

3.Defenition list- used for giving term and definition

Inside <dl> tag there will be 2 tags one is <dt> tag for naming the term and inside <dt> there will be <dd> for giving the definition for <dt>

**16)span tag in html**

For providing the style for html elements we will give them inside the span tag

**17)how to create a table**

**1.<table>:** To define a table, all the other related tags have to be inside this tag only

**2.<tr> :** To define a table raw

**3.<th> :** To define table heading

**4.<td>** : To define a table data

**5.<caption>** : To define a table caption

**18)Form**

**Form elements:**

1.<input> 2.<label> 3.<select> 4. <textarea> 5. <button>

5.<fieldset> 6. <legend> 7. <datalist> 8. <output>

9. <option> 10. <optgroup>

**Form attribute**

1.action 2.target 3.method 4.autocomplete 5.non-valid

**Input types**

* <input type="button">
* <input type="checkbox">
* <input type="color">
* <input type="date">
* <input type="datetime-local">
* <input type="email">
* <input type="file">
* <input type="hidden">
* <input type="image">
* <input type="month">
* <input type="number">
* <input type="password">
* <input type="radio">
* <input type="range">
* <input type="reset">
* <input type="search">
* <input type="submit">
* <input type="tel">
* <input type="text">
* <input type="time">
* <input type="url">
* <input type="week">

**19)What is a DOM**

* DOM treat the xml or html as tree structure
* This tree structure is described by w3c i.e. it provides a standard for creating an html file with a specific order or structure
* Here DOM tree will have different nodes or tags which represent the objects

**20)**

**Cascading Style Sheet**

**1)What is CSS**

* Cascading Style Sheet is used for styling the HTML elements

**2)What is meant by Cascading**

* Cascading mean that style can fall from one style to another enabling multiple stylesheets in one HTML document
* Cascading refers to a cascading order in HTML document
* This will sort declared CSS in an order to avoid conflicts.

**3)Type of CSS**

* Internal/Embedded – Adding styling inside a style attribute …<style></style>
* Inline – Adding styles within the HTML element …

using style=background-color: “green”

* External/Linked – Adding External CSS file to the HTML document.

In head section…

<link rel=” stylesheet” href=” filename.css” />

**4)Advantage of CSS**

* Accessibility
* Consistency in design
* Lots of formatting options
* Multiple Device Compatibility
* Page will load fast
* Maintenance is easy
* Offline Browsing

**5)Components of CSS style**

* Selector: HTML element name, id name, class name
* Property: It's like an attribute such as background-color, font-size, position, text-align, color, border etc.
* Value: which defines property or values allocate for properties.

**6)Type selector in CSS**

* Type selector matches the element of specific type, ie targets a specific type only..

This is somewhat same as attribute selector

* **Syntax**

input[type=”text”]

{

color: red;

}

* There are different types available
* [type=”text”], [type=” password”], [type=” number”]

**7)Universal Selector**

* Used to select all the elements in html
* **EX:** \*{ color : green; }

**8)Descendent selector**

* Will be specific , this is for styling some element inside another element
* Syntax

.classname div ---will affect every thing inside

{

}

**9)Id selector**

* To style an element based on the id …

Syntax

#idname {}

**10)class selector**

* To style an element based on the class name denoted by dot (.)

**11)How to implement external stylesheet**

**12)Attribute selector**

* used to style an html element with a specific html attribute
* somewhat same as type selector

**13)CSS is incase sensitive**

**14)How to change the face of font in CSS(font-family)**

**15)grouping selector**

**16)Child selector**

* will affect the all the child element coming inside …will not affect elements inside another tag inside it
* **syntax**

p >. selector

{

}

**17)what is the use of float property in css**

For aligning elements to sides i.e. only horizontally.

**18)Background-position of image done using**

Background-position

**19)How to make a font oblique in CSS**

Using font-style:”oblique”;

**20)List the different types of media in CSS**

1. all: suitable for all devices
2. braille: used for devices for visually disabled peoples like braille tactical devices
3. embossed: for paged braille printers
4. handheld: for small devices which are easy portable with small screen width
5. screen: colour computer screen
6. print: for print preview screens
7. project: for projecting devices like projectors
8. speech: for speech synthesizers
9. tty: for portable devices with limited display properties
10. tv: for television type devices

**21)Font attributes in CSS**

1. Font-variant
2. Font-family
3. Font-size
4. Font-style
5. Icon
6. Caption

**22)How to remove default border appeared for image links**

By using the property image-border: none;

**23)Elements of CSS box model**

1. Content
2. Padding
3. Border
4. Margin

**24)Z-index**

Used for avoiding the overlapping elements….default value 0

**25)font-weight attribute**

**26)Attribute for cursor type:** cursor: pointer;

5 types

* Pointer
* Search
* Help
* Wait
* Hand
* Crosshair

**27)What is the syntax of CSS styling or CSS rule**

Have 2 main parts: selector and declaration part

-----Selector {property : value;}

**28)What is CSS box-model**

* All html element can be considered inside a box
* This term deals with the design and layout
* CSS box model is a box which wraps around all the html elements by default
* It will have

1. Content
2. Padding
3. border
4. Margin

**29)What are position property in CSS**

* They will specify the type of positioning property for an element
* They are of 5 types

1. Static
2. Relative
3. Absolute
4. Fixed
5. Sticky

**30)What are selectors in CSS.**

They are for selecting different html element or content for styling

1. Simple or Element selector

* Id selector
* Class selector
* Tag name selector
* Universal selector
* Grouping selector

1. Combinatory selector

* Descendant selector
* Child selector
* Adjacent sibling selector
* General sibling selector

1. Pseudo class selector

* Link
* Hover
* Active
* visited

1. Pseudo element selector

* First line
* First letter
* Before
* After
* Selection

**32)Z-index**

For placing the image or item behind(z-index=-1) or in front(z-index=1)

**31)Difference between body selector and universal selector**

**JAVA SCRIPT**

**1)EVENT BUBBLING**

-The order in which events are working

-Triggering the innermost items first then going to outer or parent when

one element is nested inside another element or tags

**2)Nan**

-Actually considered as a bug in js

-NaN-not a number, but is number datatype

-cannot be represented as meaningful number

-we can use it to check whether given is number or not

isNan(value)--Boolean result

**3)BREAK AND CONTINUE**

-

BREAK- To jump out of the current loop without executing the remaining statements

CONTINUE- Skips remaining loop statements and continues with next iteration in the loop

**4)USE OF typeof operator**

return datatype of an element as string datatype

**5)Types of error in JS**

-Syntax error-error in syntax during interpret time

-Run-time error-during execution time

-logical error-because of logical mistake, herd to find out,

-Eval

-Reference

-Type error

-URI error

-range error

**6)Use of strict mode in js**

-sometimes js display result even if there is an error, for avoiding this

give "strict"

-in this mode code will throw all type of error

-we cannot assign a variable before declaring it in strict mode

**7)Can you use cookies in js**

A **cookie** is an amount of information that persists between a server-side and a client-side.

A **cookie** contains the information as a string generally in the form of a name-value pair separated by semi-colons.

-cookies can be created, read and erased by js.

-accessed by using” document. Cookie”

**8)Difference between Null and Undefined (both means absence of value)**

-null= no value, defined a variable but not assigned a value to it

-undefined= variable is declared but not defined

**9)Data types supported by js**

= Primitive - undefined, null, Boolean, string, symbol, Number, Nan

=Object data type (Non-Primitive) – Array, Objects, functions, null

**10)Features of js**

**-**Most commonly used as client side scripting language

-js code is written in to an html page for making it interactive

-Object based scripting language

-lightweight interpreted programming language

-used for creating network centric application

-js is open and cross platform

-DYNAMICALLY TYPED

-INTERPRETED TYPE

-asynchronous on callbacks

--loosely coupled

-case sensitive

**11)Is js a case sensitive**

Otherwise it will not be considered

**12)Advantage of js**

**--**less server interaction: -we can validate user input before sending to the server

--immediate feedback to the visitor- they don’t have to wait

--interactivity – hovering, sliders, drag and drop etc.

**13)How can we create a js object**

Var objectname = { key1 : ”value1” , key2 : “value2” } **or** using a new keyword, but not recommended

**14)How to create an array in js**

**-**An array can store different types of data as a list separated by commas inside square brackets

-The array elements are accessed using array indexes

**15)What is named functions and how to define it**

**-**Parametric block of code which is once defined and can be called many times after

-set of statement that performs a task or asset of value

-Defined by

function function-name { // any task }

**16)Can you assign an anonymous function to a variable and pass it as an argument to another function**

Yes, we can declare a variable and assign an anonymous function to it and we can pass it as an argument to another function

**17)What is the scope of variable in JS**

Scope of variable defined as the area in which the variable is being declared. i.e. according to this the variable can be accessed through out the program

2 types

Global: will be visible throughout the js code

Local:wil be visible only inside the function it is defined in

**18)What is this operator in js**

The “this” **keyword** refers to the current object in a method or constructor. The most common use of the “this” **keyword** is to eliminate the confusion between class attributes and parameters with the same name

It has different values depending on where it is used:

* In a method, this refers to the **owner object**.
* Alone, this refers to the **global object**.
* In a function, this refers to the **global object**.
* In a function, in strict mode, this is undefined.
* In an event, this refers to the **element** that received the event.
* Methods like call (), and apply () can refer this to **any object**.
* With arrow functions the this keyword always represents the object that defined the arrow function.

**19)Types of validation in JS**

1. User-side: is performed by a web server, after input has been sent to the server.
2. Server-side: is performed by a web browser, before input is sent to a web server.

**20) What are closure in js.**

A closure is a function having access to the parent scope, even after the parent function has closed.

In a closure there will be a function inside another function and the inner function will have access to the outer functions variables even after the outer function has returned or done its function.

**21)what are the different types of events in javascript.**

Events are the thing happens to an html element, javascript can be used to react to these events

1.onClick 2. onMouseOver 3. onChange 4.onMouseOut 5.onKeyDown 6.onLoad

When we are using these inside event listeners we have to remove “on” prefix

**22)What are event listeners in JavaScript**

Will attach a specified event handler to an html element without overwriting the existing html event handler.

We can add as many event handler to a single html element

**23)what is event propagation in js**

It defines how the event travels through the DOM tree element which will be inside one another.

1. Event bubbling: here the event flow is from the inside element to the outer most element, the inner element will trigger the outer

Inner have higher priority

By default, all the event will be bubbling, that is the parameter value will be false

1. Event capturing: flow is from outside to inside, higher priority is for outermost element

For giving event capturing we have to give true as the parameter

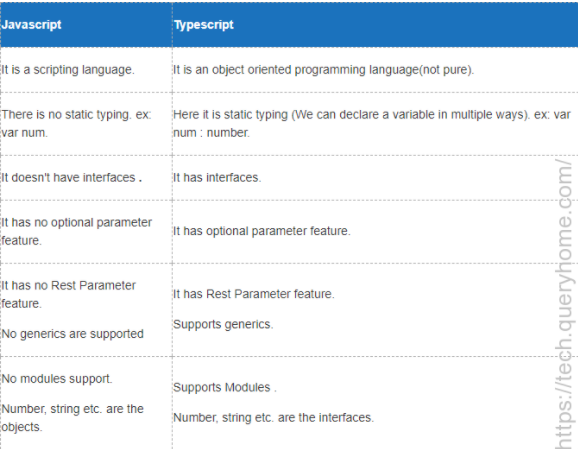
Syntax: .addEventListener(“event”, functionname, true);

**24)Can you use cookies in js**

**Cookies** are data, stored in small text files, on your computer. When a web server has sent a web page to a browser, the connection is shut down, and the server forgets everything about the user. ... When a user visits a web page, his/her name can be stored in a **cookie**.

-cookies can be created, read and erased by js

-accessed by using document.cookie

****

Typescript : takes more time for its compilation

Javascript was not recommended for big projects and we cannot use many functions in js script so introduced typescript

Cannot be used In server side but js can be

**Angular**

**(https://www.questpond.com/DemoVideos1//Angular/Angular6/angular6.php)**

**1)What is angular**

Angular is an open source client side javascript framework which is written completely on typescript language which is latter converted to javascript format, which is primarily used to create interactive Single Page Applications which is designed for different types web, mobile and desktop platform and it is maintained by google

Companies using angular: Google, Netflix, HBO, Nike etc..

Angular can be run on most of the browser.

* Angular is JavaScript binding framework used to bind HTML UI and JavaScript model
* Angular is platform or framework for making single page web application using HTML and Typescript.
* Help us to implement MVW architecture in browser
* Angular is written in typescript
* Component based architecture framework
* Features – HTTP, DI, routing, CLI
* Angular is developed and still now maintained by google
* Open-source

**2)What is the use of angular**

* Angular is a JavaScript binding framework used to bind both html UI and JavaScript model
* Angular will help you to reduce your effort in writing length codes for binding using CLI
* Angular will help us to implement MVW architecture (MVVM or MVC or MVP)
* Helps in making single page application using routing concept

**Advantage**

* Facilitate client and server side communication
* Have animation and event handlers
* Follow MVC pattern
* Supports 2-way data binding
* Supports DI

**Disadvantages**

* Sometimes complex SPA can be lag to use due to their size
* Dynamic application may not perform well
* Angular takes a descent time and effort to study
* Provides limited Search Engine Optimisation features
* It is not backward compatible
* Because of its compilation step it takes more time

**3)Difference between angular js and angular**

Angular – version 1---2010

* Language used –JavaScript
* Architecture- controller based
* Not mobile compatible
* No cli feature, so have to build thing from scratch
* No lazy loading
* Can’t make SEO friendly website
* Can’t run on server side
* For property binding Ng directives are used
* Don’t support DI

Angular- version -2,4,5,6,7,8,9,11---every year 2 versions are released on 2016 and 11 0n 2020 and 11.25 is launched on 10 march 2021 –angular 12 is planned to release on May 2021

* Language used -Typescript
* Architecture- component and directive based architecture----component as central point
* mobile compatible- cross platform app development
* Have CLI features so that there will be already built code so that we just have to modify, don’t have to code from beginning
* lazy loading (code splitting)– here we can create small bundles – so only code requested by user will be loaded
* angular can run on server side hence can create SEO friendly website
* uses () to bind events and [] for property binding
* supports DI
* every 6 months we used to get a new version of angular
* High error identification chance: allows frequent unit test using the karma and faster scenario test using protractor during compile time

Unit testing : testing one thing at time –framework used is karma jasmin

Integration testing : testing end to end at a single stretch—frmwrk is protractor

**4)What are directives**

* angular syntax which we write inside html which will change behaviour of html DOM
* markers on DOM elements which tell angular to attach a specific behaviour to that DOM element or even to the DOM elements children
* most of the directive starts with ng

**5)Types of directive**

SAC

1. Structural directive- will change the structure of the DOM by adding or removing element from dom

* ngFor
* ngIf
* ngSwitch

1. Attribute directive- change the behaviour like look and feel of html DOM

These are used as attribute of elements—hidden and shown features, color of elements

1. Component directive – self-contained control with own UI and own code, like a user control which has its own template

Thing inside the tags used to alter dom

Most common type of directive

**6)What is NPM and use of Node\_module folder**

NPM- Node Package Manager

* Package manager which make installation of different JS frameworks
* Syntax npm install ItemtObeInstalled
* Installed while installing nodejs file
* Version we are using is 6.14.11
* To check :..go to cmd … npm --v

Use of node\_module

* Is the folder to which all the packages will get installed to.

**7)Importance of Package.json file in Angular**

* File where all the reference of js frameworks which is used will be listed
* here wil show whatever package we are installed using npm and those packages we are using In our project
* so rather than installing packages one by one, they can be installed at a time in angular

**8)What is Typescript**

The typed version of javascript which help us to define types which will be easier to understand and provide a better syntax

* Superset of JavaScript compiled to JavaScript
* Downloaded by using npm--- npm install –g typescript
* Strongly typed, object oriented and compiled typed language
* Developed by the developer of c# at Microsoft
* Strongly typed language, not a dynamically typed language like js , i.e. it is a statically typed programming languages
* We write all our angular code using typescript
* But later all the ts codes will be converted to js code bcs browser will only understand js code
* ts.configure.ts: will provide the configuration of how to compile into javscript
* Object oriented programming language so development becomes easy and speeded up

**9)What is CLI**

Command Line Interface--- introduced in angular 4

* By which we can create an angular project from already existing code or already made project template, that is rather than coding from beginning we can start from already made code
* We use npm to install cli
* Syntax: -- **npm install @angular/cli** --then we will create the file – ng new filename.

**How to generate a class using CLI?**

1. Open command prompt from where we want to create our class in
2. Type -- ng generate class classname

**10)What is a component and module**

Component:

will help you to bind model and view, where we will do our binding codes in for combining both the view and model

angular allow us to create our own custom components which have several functionality and codes

Module: Will group the components which are related to each other

**11)What is a decorator in angular**

Used to say that what kind of the class it is giving by using @ to the angular framework, that is it is a component or a module,

If we don’t give decorator (@) before declaring the class the angular will consider it as just a usual typescript class.

Also called as metadata and annotation

How to create component or a module in angular?

For that we have to use decorator …. saying about what type of class we are going to create

Ex. @component means it is an angular component

@NgModule means that it is an angular module

**12)What are templates**

HTML view of angular

* Inline template
* External template: we give the template url in the component.ts file for referencing the html file i.e. component.html file…here in that we are going to create the view.

**12)Explain 4 types of Data binding**

Data binding is the process of binding both the component and the html view.

It is the way communication of components and the DOM

Here in angular it is one of the main advantage , here we don’t have to write any specific code to bind both the ui and the component.

1. Expression/ string interpolation binding {{}}: where the data is send from the component to the view and the and we can mix that data with the html
2. Property binding [] : Data flows from component to the view where the data gets attached to an user input like a text box and whenever a change in data is made from component section it will affect the view also
3. Event binding (): where data flows from view to component like a button, when action is done in view it will affect the component also
4. Two way binding [()]: The data will be sent from component to the view same as the view to the component---here we can send the data from component to view and view to component by events

**13)Explain architecture/blocks/pillars of angular**

1. Template-view

* Contains html tag that tells the angular how to render the component

1. Component

* talks with the css, model
* Control one or more views- each view is specific section of the screen
* Every angular application will have at least one component i.e root component
* Components are directive with templates

1. Module-

* Also known as ng module
* Organised block of code with specific set of capabilities
* Usually there will be at least one module i.e., the root module

1. Data binding---[],(),[()],{{}}

* The mechanism by which the parts of component and template communicate with each other

1. Directives – SAC-

* help to change DOM behaviour
* responsible for instructing angular how to transform the dom when rendering a template.
* Components are directive with templates

6.Metatdata/decorators/annotations (@)

* + For angular to know how to process a class, a metadata is attached with the class
  + For this purpose, decorators are used.

7.Services –

* anything ranging from value, function, features that is required by an angular application
* service is a class with well-defined purpose

8.Dependency injection

* Are services which some components request to angular
* these dependencies can be called by any component parameter

9.Router

* Responsible for interpreting a browser URL as an instruction to navigate to a client generated view
* The router will bound to links on a page to tell the angular when the user clicks on the link have to navigate the application view

**14)SPA**

Single page application –

Application that is loaded only once i.e. at first the main page will get loaded and when ever we reloads the whole other pages and all the files in the application don’t have to be loaded,

* Only the necessary part will get loaded this helps to increase performance.
* But for user it appears as the whole page is getting loaded.
* The common ui is loaded only once and the necessary html section will be loaded asper request of user, to implement SPA we use routing
* Here we don’t need to reload the js and css files again and again
* Ex. Gmail, google maps, GitHub, Facebook

**15)How to implement SPA & What is Routing**

Routing is collection of URL and what to bring when the url is pressed,

Routing helps to define the navigation of angular application—so for doing a SPA routing ui is necessary—

**16)How to implement Routing In angular**

Will have a {path: Home & component: from or what component we are taking it}

Path: means what is the url we are giving

Component: mean if we click on the url where the site going.

Routing from HTML

1. Define a collection- saying that for which url path which component is loaded
2. Then in root or main html page define the router-outlet-where page is loaded
3. Define the router link and say the url name in the same main html page.

OR

Routing from component

**this.route.navigate[‘/Home’]**

**17)Lazy loading**

On demand loading-i.e. loading when the user demand to…. loading what is necessary at that moment. i.e. while loading the page all the things will not be loaded only when user clicks on some page or link that page will be loaded as an SPA.

1)For this we have to divide project to separate **module**

2)use “loadChildren” instead of component in the {path: & loadChildren: } to load modules on demand…..inside app.routing.module.ts

**18)What are Angular services**

**--**we create angular services for sharing the common functionality or values to different modules.

--Singleton objects in angular are instantiated only once in the lifetime of an application

--Functions offered by a service can be invoked or called from any angular component such as controller or directive

**19)dependency injection in angular**

Dependency injection is an application pattern where rather than creating object instances from within the component, angular injects it via the constructor

**20)How to implement dependency injection in angular**

We have to use provider attribute in @NgModule decorator in app.module.ts ie if someone asks you to provide some service in @ngComponent you have to provide them using useClass

Ex.{provide: BaseLogger , useClass: httpLogger}---here what we need is baseLogger and it is inside HttpLogger, I.e. baselogger if the service we want and that services are inside httplogger

And go to component ask for the baseLogger and angular will inject it

**21)Benefit of using DI**

* Decoupling – changing value in one place it will reflect across all the places or components to which it is injected to…. i.e. it is loosely coupled coupled..
* If we are using new keyword for creating a service or object it will be tightly coupled
* That Is changing in one place it will reflect all across the project.

**22)Difference b/w ng serve and ng build**

ngserve :- builds angular application In memory or ram

ngbuild: builds angular application to the hard disk, mainly used in production,

so for production we use ngbuild –prod===this will compress the js file by removing all the spaces and comments and make your application ready for production.

**23)—prod parameter in ng build**

Actually makes the application ready for production,

Prod flag will compress the js file, removes comments, creates GUIDEs of the js file and make the application ready for production.

so for production we use ngbuild –prod===this will compress the js file by removing all the spaces and comments and make your application ready for production.

**24)Updates came in angular 11**

* Faster build
* Web pack 5 supports
* improved logging and reporting
* updated HMR support

**25)What is string interpolation- (same as expression binding)**

* {{anything}}
* type of one way binding to transfer data from .ts(component) to view
* uses template expression in double curly braces to display data from component to view
* mixing data from component with the html section.

**26)Scope Hierarchy in angular**

* Angular organises the scope with dollar scope object into a hierarchy that is typically used by views.
* It will have a root scope and will have several child roots
* A variable set inside another scope cannot be accessed or viewed by another scope

**27)What are templates in angular?**

* Templates will be written inside the HTML section which contains angular specific elements and attributes
* Templates tell the angular how to render the component in angular
* Template is associated with a component defining that components view.
* These templates contain information coming from model and controller which are then rendered to provide a dynamic view to the user

**28)Annotation and Decorator**

**Annotation:**

**Decorator:**

**28)Difference between angular and jQuery**

|  |  |
| --- | --- |
| JQuery | Angular |
| * Support DOM manipulation | * Support DOM manipulation |
| * No Restful API | * Support Restful API |
| * Has animation support | * Has animation support |
| * No routing | * Supports routing |
| * No form validation | * Support validation |
| * No two way binding | * Support two way binding |
| * Supports ajax | * Support ajax |

**29)Angular expression and JavaScript expression**

**30)CSS pre-processors**

1.SCSS 2. SASS 3. LESS 4. STYLUS

* The CSS pre-processor are used for providing more functionality, easier and efficient development
* The CSS pre-processors are used to combine multiple stylesheet in to one
* It provides a special functionality of combining multiple stylesheet into one

1. SASS-syntactically Awesome Style Sheet, will add special features such as
2. SCSS-Syntax for SASS
3. LESS- seeks to extend the default CSS syntax by means of mixins, variables and nested style rules.
4. STYLUS- [Stylus](http://learnboost.github.io/stylus/) is a dynamic and robust CSS pre-processor that comes with its own plethora of custom functions, such as saturation().

**31)two type of testing**

1. Unit testing: framework used is karma Jasmin
2. Integrated (e2e- end to end testing) testing: framework used is protractor

**32)use of nodejs**

Provides an environment for

JavaScript runtime i.e. we can run JavaScript using node

**33)What is meant MVC**

MVC is an architectural pattern which separates application layer in to Model View and Controller.

1. Model: have all data related logic
2. View: ui logic of application
3. Controller: like the brain of the setup and act as the interface between the 2

**34)What Is compiling in angular and how to compile it**

Because of angular is compiled type because we are using typescript bcs it is a compiled type language the language is compiled in to .js format from .ts format while going for execution.

Command for compiling is:--- ng build, write it in the terminal section

This will create a folder name dist/appname

Here it will have all our .ts files compiled to .js format

favicon.io: for icon displaying at tap

index.html: the root file

initial chunk files like

main.js: **main**. **js** contains all our code including components (ts, html and css codes), pipes, directives, services and all other imported modules (including third party).

polyfills.js: lines of code which make your application compatible for different browsers

the code we write mostly is in ES6

Vendor.js: This **file** contains any libraries imported into your app (app. module), including the **Angular** libraries. These libraries will not be our i.e. we are vendor

runtime.js:  is the webpack loader. This file contains webpack utilities that are needed to load other files.

style.css:  contains all styles we declare in styles section of **angular**.json file

**35)what is angular.json and package.json**

**Angular.json:**

**C#.net core**

**1)What is Static, void, public keyword**

**Static:**

* if we declare a class or method as static we cannot create an instance or object for it
* used for declaring a member of a type or for making it specific to a type
* It Is applicable for classes, variables, constructors, method
* A static typed class cannot be inherited by any other classes.

**Void:** void is used in methods where it doesn’t return any value.

**Public:** It is an access specifier in which the method of a class can be globally accessed

**2)what is C# and what are the features**

* C# is a programming language developed by Microsoft to develop different type of application including web application, windows application, games etc.
* C# is used in both front end and back end or it is used in between.

**Features**

* C# is an object oriented programming language so that it deals with classes and object
* uses .net as its framework
* Statically typed by means it strictly typed or type safe
* C# has interoperability, i.e. it can be used to write other languages like c++.
* it is a structured programming language that is it we can break the code into different section or parts—I,e. it is component oriented programming language
* modern and widely used
* Easy to grasp
* Platform independent for compilation

**Advantage**

* Scalable- i.e. it can handle high amount of load
* Supports cross platform- means we can compile it once and run it on different platforms
* Fast- because of structured or component oriented

**3)Types of comments in c#**

1. Single line comment- //
2. Multi- line comment- /star --- star/
3. XML comment- ///

**4)List the process steps of C#**

1. Compiling Source Code into Managed Module: convert our code into an Intermediate Language (IL) and make it as an .exe file
2. Combining newly created managed module into an assembly / assemblies.
3. Now the compiled language is converted in to machine instruction it done using CLR (Common Language Runtime)
4. Executing the assembly in & by CRL.

**5)List the access modifiers in C#**

1. Public: The methods can be accessed by all the classes globally i.e. from any part of the code
2. Private: The method or attributes can be only accessed only from inside the class and not by any other
3. Protected: Same like private but here the method can be used by those class who are inheriting this protected class
4. Internal access modifier: a method or an attribute can be accessed from that class from the current assembly position
5. Protected internal: its access is restricted to current project assembly

**6)What are the IDE’s for C#**

Integrated Development Environment: It is just an environment in which the developers works on

Support wide variety of IDE’s

1)Microsoft Visual Studio express 2) Mono-developer 3) dev C++

4)VSC

**7)What is break and continue statement**

Break: will jump out of a loop

Continue: jump over a certain or next iteration step and continue the process (skipping)

**8)What are the different methods for passing parameters to a method?**

1. Value parameter: here the value is just copied from the actual to formal, change in formal will not affect the actual parameter.
2. Reference parameter: here the address itself is passed, so the changes made to the formal will also affect the actual parameter.
3. Output parameter: can return more than one value

**9)What is .NET framework**

* .Net is a software development platform for creating different type of application
* It supports or can be used for C#, java, V. B.Net, pearl
* it provides some basic functionality like Classes, API and libraries
* services provided to the application: are its memory management, security, type safety, version compatibility, language interoperability.

**10)What is CLR and FCL**

CLR: Common Language Runtime

* CLR Converts managed code to machine codes and then executes it
* Manages the execution of .net programmes
* Act as a intermediate layer between OS and application

FCL:

**11)Managed an unmanaged code? Explain the two types of codes.**

**Managed code:** the execution i.e. the converting of this managed code to machine code done by Common language runtime(CLR)?

**Unmanaged code:** Code developed outside .Net framework and doesn’t run under the control of CLR is called as unmanaged code. Ex-PowerPoint, excel

**12)Call by value and call by reference**

Call by value: original value will not change only the copy of value inside a function or method will change

Call by reference: Here the original value will change, i.e. a method us referring to the address of original value

**13)Difference between Struct and class data types in C#**

Struct:

* Keyword used for creating a structure
* Helps in creating a single variable in storing related data or various data
* Value type, the values are stored on stack
* Used for storing smaller amount of data
* Will not support inheritance
* Will not support abstraction
* Can’t have destructors

Class:

* Reference type, the values are stored in heap
* Used for storing large amount of data
* Can have inheritance
* Can have abstraction
* Can have destructors

**14)Data types in C#**

Int, double, string, char, boolean,

**15)What is Boxing and Un-Boxing in C#**

Boxing: Boxing can be explained as the conversion from value type to reference type.

Un-Boxing: Unboxing is the process of converting from reference type to value type

**16)What is method Overloading and Overriding**

**Overloading:**

* There will more than one method with “same function name” but will have different parameters.
* Don’t have anything to do with inheritance and overriding
* Can have n number of overloading

**Overriding:**

Here the functionality or behaviour is changed without changing its parameters.

Overrides a function in base class by creating a function with same name in derived class by inheriting.

Here we use virtual and override function for overriding

Here different method but same goal

**14)Explain Encapsulation and how it is implemented in C#**

Encapsulation is the process of combining the data members and member function into a single unit

It is implemented by using access specifiers

1. Public
2. Private
3. Protected
4. Internal
5. Protected internal

**15)What are Pre-process Directives in c#**

* Pre-process give instruction to the compiler to pre-process the information before the actual compilation starts
* They are not statement because of that they do not end with semicolon (;)
* Only white spaces are allowed in front of pre-processors,
* They are declared by using a # in front of it

**16)Explain Polymorphism with types**

Polymorphism means different forms; i.e. same function will have different definition at different occasions.

1. Static polymorphism: Also called as compile time or early binding polymorphism

Done by method overloading

2.Dynamic polymorphism: Also called as Run-time or late binding polymorphism.

Done by using Method overriding.

**17)What is Early binding**

* Mechanism of linking an object with its functionality during the compile time…
* Also called as static binding

**18)Purpose of “Using” statements in C#**

* It is used for including a namespace in a C# programme.
* We can use n number of Using statement C# programme.
* They are usually mentioned in the beginning of a c# programme

**19)Explain Assembly Versioning**

Assembly versioning was introduced for solving the DLL (Dynamic Link Library) hell Problem used to occur in C#

Here the assembly will specify not only the library but also the version of the library

**20)Explain DLL (Dynamic Link Library)**

The DLL will have lots of function and codes which can be used by different application at the same time.

**21)Explain DLL Hell problem in C#**

* It was happened when a new DLL is installed and it will get replaced with the old one, all the programs which are using the previous DLL version will show error
* This is because the version information was not store in the system
* This problem was solved by the introduction of Assembly versioning in .NET

**22)How can you prevent a class from getting inherited**

We can prevent it by using the keyword **“sealed”**

**23)When do we declare a class as abstract**

* Explain what is abstraction
* An abstract class cannot be instantiated it can only be inherited
* An abstract class is used in large projects where we need to share common functionality to the inherited classes

**24)Difference between Abstract class and interface class**

|  |  |
| --- | --- |
| **Abstract class** | **Interface class** |
| Abstract class can have both abstracted methods and non-abstracted method | Will have abstracted method only |
| They will be define in the inherited classes | They have to be defined if they are inherited by any other child classes |
| Can have access specifiers | Cannot have access specifiers |

**25)What are generic classes**

* Classes without having specific data type for objects and classes
* These are assigned during the runtime.

**26)Explain “stringbuilder” keyword.**

The string builder keyword is used when we want a string to be edited without replacing the old one. That giving a string the mutable property

**27)Explain difference between stringbuilder and string**

**Stringbulder:**

* have mutability
* no new instance is created when an existing string is tried to modify
* used when we want to multiple string modification again and again

**String:**

* do not have mutability
* each time a strings gets modifies a new instance have to created

**30)What are jagged arrays**

* Arrays which are having arrays has its elements
* The elements can be of different dimension
* Also called as Array of Array

**31)What are custom controls and user controls**

|  |  |
| --- | --- |
| **Custom control** | **User control** |
| Compiled code components (dll files) executing on the server | Reusable controls- similar to ASP include files |
| can be added to the toolbox | cant be placed in toolbox |
| Drag and drop controls to web form | no drag and drop …  file extension will be .ascx |
| Assemblies(dll files) – can be used in multiple application | page file can be used in an SPA |

**32)Object pool**

Software creational design pattern and a container of number of objects that hold a list of other objects

Object pool will have a collection of objects which are ready to use

Whenever there is a need request for object, the pool manager will take the request and will served from the pool

**34)programming paradigm**

Classification of programming languages based on their characteristics

Main types functional and oops

**33)what is an object oriented programming language**

Object oriented programming Is a language which uses classes and objects for its functioning, it is based on real world entities like inheritance, abstraction, polymorphism etc.

* Oops work on the concept of objects which contain data in the form of fields referred to as attributed and have code in the form of procedures referred to as methods
* Mutable
* Object is primary manipulation unit
* Here a data and its associated procedures are brought together for easy understanding
* Focus on the steps i.e. how we are doing
* Divided into objects to do specific tasks
* So if we want to do any modification we can easily identify
* Not for small programs
* Highly secure because of access specifiers
* Bottom to up approach –programmers have to create modules first
* Oops is based on the 4 key principle

1.Encapsulation:

2.abstraction: abstract and non-abstract classes—abstraction over data only

3.inheritance: multiple inheritance can occur

4.polymorphism:

**34)Functional programming language**

* Primitive that they are not mutable i.e. no assignment operation once the value is assigned after it cannot be changed
* function is primary manipulation unit
* Focus on what we are doing i.e. the path
* Here the method and its definition is kept in different places
* Step by step
* Supports abstraction over data and behaviour
* Will have pure function i.e. output is completely based on the input given

**35)Procedural oriented programming**

* They mainly based on function and parameters ..
* Here program is divided in to function to do specific task
* Sequential steps for performing specific tasks
* Function created can be used anywhere in the programming
* Can use same variable in different functions
* Can have scoping
* Not much secure bcs8 no access specifiers\*963
* Can track program easily
* For large projects the code is not reusable