HTML QUESTION

Ans -1 : No it is not a tag of html. It provide information to the browser . It is used to check version of HTML.

Ans -2 : It provide meaning to the developer . It is used for lets say we are creating a webpage and we provide semantic tags and if some other developer read the code they can clearly understand which part of this code .

Ans -3 : HTML tags - It is collection of start tag and end tag
Ex : <h1>Tags</h1>
HTML elements : It is collection of self closing tag.
Ex :

Ans -4 : GitHub repo - https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/blob/master/HTML/index.html

Ans -5 : You provided Link in this question but this link is not working

Ans -6 : 1st is - Semantic tags
2nd is - Media Supports
3rd is - Introduced HTML5 input type email,number,password
4th is - Offline Support
5th is - animation and graphics support HTML5 introduced canvas elements

Ans -7 : GitHub repo - https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/blob/master/HTML/musicPlayer.html

Ans - 8 : img tag is used to embedded image and figure tag contains img tag and title of image also.

Ans -9 : HTML tags are used to define the structure and HTML attributes are used to provide additional information example in img tag we have alt attribute to provide the details

Ans -10 : You provided Link in this question but this link is not working

CSS QUESTION

Ans -1 : It describes how elements are displayed and structured on a webpage . content , margin , padding , border are the properties of box model

Ans -2 : class selector , id selector , element selector , descendant selector

Ans -3 : vw & vh are relative units of measurement of css . px is absolute units it means that it is fixed for every viewport and vw & vh are relative unit it is scale with viewport and it helps in responsive also

Ans- 4 : Inline - It does not start with a new line and it takes necessary width example Hello

Inline-block - It also does not start with a new line but it have some properties height and width

Block - It starts on a new line and occupy full width

Ans -5 : In the content box , the content inside an element and in the border- box model we gave the specific width

Ans -6 : It works on positioned element and it defines the order of overlapping html elements

Ans -7 : flex - It is 1D means one dimension either it can be row or column.

Grid - It is 2D means we can provide row and column also

Ans - 8 : fixed - It is used for any component to fixed lets take and example if we want to fix and Navabr to fixed at top so we can provided position fixed and top properties 0 so it will be fixed at top.

Sticky- It is hybrid of relative and fixed.
It allows an element to "stick" to a specific position as the user scrolls.

Absolute - It is used when nearest or parent element would be relative

Relative : It is positioned relative to its normal position

GitHub repo :
https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/blob/master/CSS/position.html

Ans -8 : GitHub repo -
https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/blob/master/CSS/periodic.html

Ans -9 : GitHub repo - https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/blob/master/CSS/layout.html

Ans -10 : https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/blob/master/CSS/responsive.html

Ans -11: GitHub repo link : https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/blob/master/CSS/ineuron.html

Ans -12 : Pseudo Classes - If we have to select pseudo classes we use this colon (:) example - :hover

Pseudo Elements - If we have to select pseudo elements we use this colon (::) example - ::after

JavaScript

Q.1 What is Hoisting in Javascript ?

Allows us to access functions and variables before they are created. 2 things that are hoisted in JavaScript: Function declarations Variable declarations sayHi(); function sayHi() {

console.log('Hi'); }
Q.2 What are different higher order functions in JS?

What is the difference between .map() and .forEach() ?

higher-order functions are functions that can take other functions as arguments or return functions as results.Some higher-order functions in JavaScript re , map(), forEach(), filter(), reduce(), and sort(). The main difference between map and forEach is their return values. Map: Map returns a new array with the same length as the original array. forEach: ForEach doesn't return an array. Q.3 What is the difference between .call() .apply() and .bind()? explain with an example? Call: The call() method calls the function with a given this value and allows passing in arguments one by one separating them with commas. let p1 = { firstName: 'Abhishek', lastName: 'Singh' }; function sayWelcome(greeting) { console.log(greeting + ' ' + this.firstName + ' ' + this.lastName); } sayWelcome.call(p1, 'Welcome') Apply: The apply() method calls the function with a given this value and allows passing in arguments as an array. Example:
let p1 = { firstName: 'John', lastName: 'Smith' }; function sayWelcome(greeting) { console.log(greeting + ' ' + this.firstName + ' ' + this.lastName); } sayWelcome.apply(p1, ['Welcome']) Bind The bind() method returns a new function and allows passing in a this array and any number of arguments. let p1 = { firstName: 'John', lastName: 'Smith' }; function sayWelcome() { console.log('Welcome ' + this.firstName + ' ' + this.lastName); } let sayWelcomeJohn = sayWelcome.bind(p1); sayWelcomeJohn()

Q.4 Explain Event bubbling and Event Capturing in JavaScript with suitable examples? Event bubbling is the most common way events propagate through the DOM. When an event occurs on an element, it first triggers the event on the target element and then bubbles up to its parent and ancestor elements.

Click In this click will triggers first than move to parent and the move to root element. Event Capturing:Event capturing is the opposite of event bubbling. When an event occurs on an element, it triggers the event on the target element's ancestors first before reaching the target element.

Q.5 What is function currying with example?

Function Curing in javascript is a technique where function with multiple arguments is transformed into a sequence of functions with each taking a single arguments.

Function add(x){ return function(y){ return a+b } } Const addfive=add(5); console.log(addfive(10)) console.log(addfive(20))

Q.6 Explain execution context diagram of following code snippets, use white board to draw execution context diagram?

console.log('First'); setTimeout(() => console.log('Second'), 0); console.log('Third'); Code Snippet 2 console.log('First'); function secondCall() { console.log('Second'); } setTimeout(secondCall, 2000); setTimeout(() => console.log('Third'), 0); console.log('Third');

In this ,first ,third ,third,second will print . In this setimout it will display after few seconds .soo in javascript it works like asynchronously in nature .soo it can not wait any function to execute .it execute another function which is immediately invoking .

Q.7 What are promises? What are the different states of a promise? Support your answer with an example where you need to create your own promise. Promise can be used to write a asynchronous code .There are three state in Promise. Pending State; Fulfilled state; Rejected state;

var promise = new Promise(function (resolve, reject) { if { resolve("Worked"); } else { reject(new Error("Failed")); } }); promise.then(function (msg) { console.log(msg); }, function (err) { console.log(err); });

Q.8 What is ‘this’ keyword in JavaScript? explain with an example & create?

This keyword refers to different objects depending on how it is used.If we are using alone than it refers to global object.And in event this keyword refers to Element that received the event. let x = this; Here this refers to object window .

Q.9 Explain event loop Call Stack Callback queue and Micro Task queue in Your Words?

Event loop:it a mechanism to handle the execution of code in an asynchronous and non-blocking manner. Call Stack:Whenever function is called It is added to the top of the stack and when a function complete its execution ,than it is removed from top of the stack.It follows last in first out principal. CallBack Queue:It is a queue that holds the callback function or tasks that are ready to be executed. Micro Task Queue:It also know as job queue.IT is a queue that hold mico tasks. Example like .then() and .catch() .

Q.10 Explain Debouncing and Create a project where you are using Debouncing?

Debouncing:Debouncing can be used to increase the performance .If the events happens multiples times within specified time period only the last occurrence will trigger the function execution.

Q.11 Explain Closures and Use cases of Closures? Closures:Closures is the combination of function bundled together with references to its surrounding state.A closures give you you access to outer function scope from inner function. 1: it can be used for data Privavcy. 2:It can used for Curring and Partial Application. 2.It can be used to perform Asynchronous operations. 3.It can be used to implement memoization.

Q -12 Github Repo - https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/tree/master/JS

React Question:

Q.1 What’s React and What are the advantages of it?

React is a javascript library for building user interfaces.It is developed by facebook and is widely used by web development. Advantages of it: 1:It allows a component based architecture. 2:It uses a virtual dom. 3:Resuability of code and compatibility.

Q.2 What's Virtual Dom in React & What are the advantages of it?

Virtual dom in react:Virtual dom is lightweight copy of the real dom. And react compares the the virtual dom with real dom to identify the minimal changes neede and updates only those parst. This approach improves performances and provides a smoother user experience.

Q.3 Explain LifeCycle of React Components? In react there are different phases during creation,updating and removal. 1.Mounting:Mounting is the phase when a component is intialized. 2:Updating:It is invoked when there is something updated.It allows updating the state based on changes in props. 3:Unmounting:This method is called just before the component is removed from the DOM.It allows performing cleanup tasks,such as cancelling timers,remving event listeners.etc

Q.4 Whats the difference between between Functional Components and Class Components?

Functional component: .Functional component are stateLess and do not have there own internal state. .They receive data and functionality throughs the props passed them as function arguments. .Functional components are easier to read and write and test due to there simplicity. .They can utilize React Hooks to manage state and perform side effects. Class Components: .Class components can have there own internal state and accessed via “this.state” and update using “this.setState()”. .They are more complex.They are used for writing complex logic. .They have life cycle like “componetDidMount”,”componetDidUpdate”, and “ComponentWillUnmount”.

Q.5 What are the hooks in React & Can we use Hooks in Class Components?

Hooks are introduced in React 16.8 that allow developers to use State and other React features in Functional components without the need of functional components. Here are some hooks in react are: 1:useState 2:useEffect 3:useContext 4:useReducer 5:useMmeo 6:useRef These all are inbuilt hooks present in React.

Q.7 What’s useState Hook & Advantages of it?

UseeStae is a inBuilt hooks present in Recat that allows functional components to manage state. The useState tools initial value as its argument and returns an array with two elements. The current state and function to update that state value. Advantage are: 1:Simplicity and Readability. 2:No class syntax require. 3:It can be use multiples times to manage multiple independent state variables. 4:It also optimised re-rendering off components.

Q8:Explain useEffect & Advantages of it?

UseFffect can be used to control side effects.Side effects are actions that happenoutside of the normal rendering process, such as fetching data from api or manipulating the dom. Advantages are: 1: Centralised side effects. 2:It handles Asynchronous operations such as data fetching or api calls by using async/await or promises. 3:Dependency Management. 4:Cleanup mechanism that runs when the component is unmountedor before re-runnig the effect.

Q.9 Explain Context Api and create a minor project on it (5 Marks)

GitHub Repo - https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/tree/master/React/theme-context-api

Q.10 Explain useReducer and Its advantages?

UseReducer can be used to manage complex state and state transitions in functional components. It takes two arguments : the reducer function and the initial state.The reducer function receives the current state and an action and it returns the new state based on the action type. Advantages: Used to manage Complex State Managements. Optimised performance.

Q-11 - Github Repo -https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/tree/master/React/todo

Q-12 - Github Repo - https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/tree/master/React/counter

Q-13 - Github repo - https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/tree/master/React/calculator

Q.15 Explain Prop Drilling & How can we avoid it?

Props driilling refers to the process of passing props from a higher -level component through intermediate componets in order to reach a lower level componet through intermediate components in order to reach a lower level component that actually needs the props. Using context Api we can avoid prop drilling. Using component composition we can avoid prop drilling .

Q-16 Github Repo -https://github.com/abhi2244s/Placement-Assignment_-Abhishek-Singh-/tree/master/React/task-manager