Html questions:

* <!DOCTYPE html> is not a html tag.

It is used to specify the version and type of HTML being used(the newest version is HTML5). It is placed at the very beginning of HTML file. It ensures compatibility and proper rendering across different websites.

* Semantic tags in HTML are specific HTML elements that provide meaning and structure to the content of the web pages. The help search engines, assertive technologies, and developers understand the purpose and significance of the content.

Examples of some semantic tags in HTML:

* <header> represents container for introductory element.
* <nav> contains navbar.
* <section> contains a specific section of the whole html body.
* Html elements consist of tags and contents inside it.

Tag is a markup used to differentiate between different types of Html elements.

Inside tags, contents are added.

E.g. <p>hello my name is Barunabha Poddar</p>

Here <p> and </p> are tags.

The whole paragraph including those mentioned tags is element.;

6. Advantages of HTML5 over it’s previous versions are”

i. Semantic elements- In HTML5, we have the benefit of using semantic elements such as <section>, <nav>, <aside>, <header> which can improve readability of the document.

Ii. HTML5 has improved multimedia support. It can have usage of <audio> and <video> tag without having to use adobe flash player(which has to be used for previous versions of html>

Iii. HTML5 has features for mobile device compatibility. It makes easier to create web applications and websites responsive and mobile-friendly.

Apart from that, HTML5 provides graphics and canvas support, improved fonts and controls, enhanced accessibility, cross-browser compatibility.

8. <figure> tag is used to encapsulate and represent self-contained content, typically accompanied by a caption. It generally contains images, diagrams, videos, code snippets etc. It represents a standalone container. It can have a caption.

<img> tag is used for showing images and it contains src attribute which contains the source of the file. It does not include any caption. A <figure> element can have <img> element to provide a meaningful addition in an HTML document along with a caption.

9. HTML tags represent element and defines the structure and content of an HTML document. HTML attributes, on the other hand, are used within the opening tags of HTML elements to provide additional information or modify the behaviour of those elements.

Some of the most popular global attributes are class, id, style, title. Others are aria, accesskey, contenteditable, data, hidden, draggable etc.

Class: It specifies one or more CSS class names to apply to an element allowing for styling and javascript manipulation.

Id: It specifies one element with a single id to apply to an element allowing it for styling and javascript manipulation.

Style: It allows the element(s) to use inline CSS.

Title: It specifies some additional information about an element.

Css:

* Css box model is a concept which describes the layout of a webpage. It consists of border, margin, padding, content.

Content is the actual material of an element.

Padding is the space between the content and border.

Border is the outline around content.

Margin is the space between border of the element and neighbouring elements.

* Different type of selectors are class selectors, id selectors, element selectors, pseudo-selectors, pseudo-elements, child selectors, descendant selectors, attribute selectors.

Elements selectors are used to target elements.

Class, Id, attribute selectors are used to target element(s) with specific ids, classes and attributes respectively.

Pseudo classes and pseudo element selectors are used for a specific state of the element and position of the element respectively.

Child and descendant selectors are used to select direct children and every possible children respectively.

* The ‘vw’ means 1% unit of the viewport(i.e. the visible portion of the browser window) width. The ‘vh’ means 1% unit of the viewport(i.e. the visible portion of the browser window) hieght.

They are used to give elements to give the size according to the viewport.

Vw/vh is the relative width/hieght but the px value of an element is absolute.

The first is dependent of the device and other is independent of the device.

* Inline elements do not have line break and does not have height and width properties.

Block elements have line breaks before and after them and have height and width properties.

Inline block elements have line break before them and can have height and width properties.

* In content box, the height and width property defines the size of content only. The margin and padding will apply over the content of the defined height and width.

In border box, the height and width property defined the size of content including margin and padding.

* Z-index controls the stacking order of positioned elements on the z-axis, which determines how elements are visually layered and displayed on top of each other.

Z-index is used for positioning the HTML elements in correct order using an integer value. It is generally used with css positions.

* Css grid is a two dimensional layout system that allows us to create complex layouts with both rows and columns. Grid can expand in row and column directions. Ideal for creating magazine style layouts, image galleries and overall page structure.Provides powerful features like grid areas and template areas for easier visualization and management of layout.

Css flexbox is a one dimensional layout system that allows us to create layout in either vertically or horizontally. It is perfect for building dynamic layouts within a single row or column. Easier to use than grid. It can be used to create navigation menu, flexible card layouts, and vertical alignment content.

* Css positions:

Static: It is the default positioning properties of HTML elements.

Relative: Relative position refers to a position relative to its original position.

Absolute: It is a position relative to its parent element.

Fixed: It is a fixed position relative to the viewport.

Sticky: Elements are positioned based on the user’s scroll position.

12. Pseudo classes target elements based on a specific state or condition, such as user interaction or element positon. They are represented by a colon(:) followed by the pseudo-class name. Examples: ‘:hover’, ‘:active’, ‘:focus’, ‘:visited’.

Pseudo elements target specific parts of an element, such as the first line of a paragraph or the content before or after an element. They are reperesented by two colons followed by the pseudo element name. Examples of pseudo-elements include ‘::first-line’, ‘’::before’, and ‘::after’.

Javascript Assignments:

* Hoisting is a behavior in Javascript where variable and function declarations are moved to the top of their containing scope during the compilation phase. This means that we can use variables and functions before they are actually declared in our code.

Hoisting is of two types

* Variable Hoisting
* Function Hoisting

2. Different types of higher order functions

* Function as an argument,
* Function as a return value.
* Array higher-order functions

Map is a higher order function used to iterate over a collection or sequence of elements and it transforms each element into a new element.

Foreach is a higher order function used to iterate over a collection or sequence of elements to perform specific operation(s) or execute a function to each element.

3. Call:

The call method is used to invoke a function and explicitly specify the value of ‘this’ in the function within its first argument. The subsequent arguments are passed using a comma.

Apply:

This method is similar to call and it uses array or array like objects as as second arguments which is going to be spread as an individual arguments to the function being called.

Bind:

This method returns a new function with a specified value for ‘this’ with pre filled arguments(optional)

4. Event bubbling and event capturing:

These are the two mechanism in the DOM event handling process that describe the order in which the events are triggered and propagated through the DOM tree.

Event capturing:

I. It is the first phase of the event propagation process.

II. The event is triggered on the root element and then propagates down from the root element to the target element. During the event capturing, the event handlers attached to the ancestor elements are executed before the event reaches the target element.

Event bubbling:

I. It is the second phase of the event propagation process.

II. The event is triggered on the target element and then propagates up to the root element. During the event bubbling, the event handlers attached to the target element and the ancestor elements are executed in order, propagating up to the root element.

5. Function currying is a technique in Javascript to transform a function that takes multiple arguments into a sequence of functions, each taking a single argument. The result is a more flexible and reusable function.

6. First part:

In console.

1.First is printed.

2. Second is printed

3. Third is printed

Second part:

In console

1. First is printed

2. After 2 seconds, Second is printed.

3. Third is printed.

4. Third is printed.

7. Promises are a fundamental concept in Javascript for handling asynchronous operations. They provide a way to handle the result(fulfilled) or failure(rejected) of an asynchronous operation in a more structured and manageable manner.

A promise has three states.

I. Pending: The initial state when the promise is neither fulfilled nor rejected.

II. Fulfilled: The promise is fulfilled with a value.

III. Rejected: The promise is rejected with a reason(usually an error object).

8. In Javascript, the ‘this’ keyword is a special keyword that refers to the context in which a function is executed. It allows you to access and manipulate properties and methods within an object or determine the context of a function call.

9. The event queue is a key component of JavaScript’s event-driven, single-threaded model.It manages the execution of events and callbacks in a sequential and non-blocking manner.

A call stack is a data structure used by computer programs to track of function calls during program execution. When a program calls a function, the current state of execution, including the memory allocation and next instructions to be executed, is pushed onto the call stack.

The callback queue is a data structure used in event-driven programming to manage asynchronous function calls or events. It is closely associated with the event loop, which is a central component of many programming languages like JavaScript.

The microtask queue is a concept related to event loop in Javasript. It is a separate queue that exists alongside the callback queue and is responsible for handling microtasks

.

10. Debouncing:

It is a technique used in Javascript to control the frequency of certain events or function calls. It ensures that a function is not invoked repeatedly within a short period but rather delays its execution until a certain amount of time has passed without the event triggering again. It is commonly used in scenarios where an event may fire rapidly, such as window resizing, scrolling, or keyboard input, and you want to perform an action only after the event has settled.

11. Closure:

It allows functios to retain access to variables from the outer scope even after the outer function has finished executing. In other words, a closure is a combination of a function and the lexical environment within which is declared.

React Questions.

* React is a open-source Javascript library developed and maintained by Facebook. It is widely used for building user interfaces in web applications. React follows a component-based approach, where the UI is broken down into reusable and independent components, each responsible for rendering a specific part of the user interface.

Advantages of react:

* Components:

React allows developers to break UI in small components and merge them in specific way to create complex UI.

Virtual Dom:

React allows developers to make changes in the virtual dom which is a copy of the Dom By using a virtual Dom, React can efficiently update and render components in response to changes, optimizing performance and minimizing direct manipulations to the actual Dom.

One way data flow:

React allows unidirectional data flow, where data flows from parent components to child components through properties(props). That simplifies data management and helps maintain predictable and understandable data flow within the application.

2. In react, virtual dom is a lightweight in-memory representation of the actual browser Document Object Model(DOM). It serves as a intermediary layer between the application’s components and the browser DOM, enabling efficient rendering and updating of the user interface.

Advantages:

When a React component is rendered, it generates a Virtual Dom representation.

When a component’s state or prop change, React re-renders the component and generates a new Virtual Dom representation. It then performs a process called “diffing” by comparing the new Virtual Dom with the previous one.

3. React components’ life cycle consists of three phases. They are Mounting(putting elements into the dom), Updating(updating a component), Unmounting(removing a component from Dom).

4. Functional components vs class components:

* Functional components accept props as their arguments and return JSX to describe components UI.

Class components are defined using Javascript classes that extends ‘React.Component’ base class. They have a render method that returns JSX to define components structure and content.

* Functional components have useState() hook to manage states within the component.

Class components have their own state object which can be accessed by this.state and changed by this.setState().

* Functional components use props and states to render UI which makes it appropriate for small projects.

Class components encapsulate both state and behavior within the component instance which makes it appropriate for complex projects.

5. Hooks are introduced in React 16.8 to use state and other react features in functional components. They provide a more flexible and composable way to reuse the stateful logic without the need of class components. Popular hooks in React are useState, useEffect, useContext, useRef, useCallback etc.

Hooks are specifically designed for the functional components. They cannot be used directly in class components.

6. Lifecycle methods in React are special methods in React that are called at different stages of components’ life cycle to provide hooks into specific moments during a components creation, update and removal. They offer control over initialization, rendering, state changes, and cleanup processes.   
Advantages of life cycle methods:

* Lifecycle methods allow us to initialize and set up the initial state, default props, and other necessary configurations before the component is rendered.
* They provide hooks into components “mounting” and “unmounting” phase.
* They give us control over components updates and re-rendering processes.
* They are useful to handle errors and fallback UIs.

7. UseState hook:

The useState hook is a feature provided by React, a popular JavaScript library for building user interfaces. It is used to add stateful behavior to functional components in React. This hook allows to declare state variables and retrieve their current values and update their values.

Advantages of useState hook:

* The useState hook simplifies the process of managing state in functional components. It allows us to declare and update state variables without the need of class components and the complexities associated with them.
* The useState hook provides the same functionality as the ‘this.state’ syntax in class components.
* This hook can be used multiple times in a single component to manage multiple independent state variables making different aspects of the components’ state easy to manage.

9. UseEffect hook:

This hook allows us to perform side effects in functional components, such as fetching data from an API, subscribing to events, manipulating the DOM, or updating the document title. This hook takes two parameters: a function that represents the side effects, an optional array of dependencies.

Advantages:

* With useEffect hook, we can handle side effects in a declarative and centralized manner within functional components. This makes it easier to reason about and manage side effects, leading to a cleaner and more maintainable hook.
* It covers different aspects of the component’s lifecycle.
* It is well suited for handling asynchronus operations, such as data fetching or API calls.

9. Context API:

It is a feature provided by React that allows us to manage global state and share data between components without the need to pass props through multiple levels of the component tree. It consists of two main components ‘provider’ and ‘consumer’.

10. UseReducer hook:

This is a feature provided by React that allows us to manage state in functional components using a reducer pattern. It takes a reducer function and an initial state as parameters and returns an array with two elements: the current state value and a dispatch function. The reducer function is responsible for specifying how the state should be updated based on the dispatched options.

Advantages:

* It is useful when dealing with complex state logic that involves multiple actions or where the state transitions are not straightforward.
* By using a reducer function, we have full control over how the state is updated.
* It works well for state management in larger applications.

15. Prop drilling is the process of passing the props through multiple levels of nested components in order to deliver the props to a deeply nested component that needs access to the data or functionality contained in the props.

The most popular way of avoiding prop drilling is to use context API which is to manage a global state and share data between components without following the component hierarchy.

Express:

2. A middleware is a software component or a layer that sits between two other components or systems that facilitates communication, data processing, or functionality integration between them. It acts as a bridge or intermediary, enhancing the capabilities of the components it connects.

* Authentication:

It is the process of verifying the identity of a user or entity, ensuring that they are who claim it to be. It confirms the validity of user credentials, such as username and password, and establishes user’s identity within a system.

Authorization:

It is the process of granting or denying access to specific resources and functionalities within a system. It determines what actions a authenticated user is allowed to perform based on their assignment permissions.

* Js module was initially designed for server side only. Ejs module is designed for both client side and server side rendering.

Js module follows synchronous programming. Ejs follows asynchronous approach.

Js module does not support dynamic importing. Ejs supports dynamic importing.

* Jwt stands for Json web token. It is an open-standard that defines a compact and self-contained way of securely transmitting information between parties as a JSON object. JWTs are commonly used for authentication and authorization purposes in web applications and APIs.
* Instead of storing password in a plain text,
* The password should be hashed using a strong hashing algorithm.
* Use a random salt value before hashing.
* Encourage users to create strong password policy.
* Ensure that the transmission of the password from the client to server is done securely via HTTPS.

8. Event loop in Js

In node js, event loop is a functional part of its architecture and is responsible for handling I/O operations, callbacks, and event-driven programming. Understanding the event loop is crucial for writing efficient and non-blocking code in Node.js.

In node js, event loop follows a single-threaded, asynchronous, and non-blocking model. It allows Node.js to handle concurrent operations efficiently without blocking the execution of other code.