

Module assignment 4, level 2

1. What is the DOM?

The DOM is short for Document Object Model. It defines the standard for accessing HTML documents. It is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure and style of a document.

`Document.getElementById`.

2. List three methods to manipulate the DOM.

1. `getElementById()` method:

This method returns the element that has the ID attribute with the specific value.

2. `addEventListener()` method:

This method attaches an event handler to the document. Events refer to what happens to an HTML element, such as clicking, focusing or loading, to which we can react with JavaScript. We can assign JavaScript functions to listen for these events in elements and do something when the event has occurred.

3. `removeEventListener()`:

This method detaches an event listener previously added with the `addEventListener()` method.

3. What is an asynchronous call?

An asynchronous method call is a method that returns to the caller immediately before the completion of its processing and without blocking the calling thread. When the application calls a asynchronous method, it can simultaneously execute along with the execution of the asynchronous method that performs its task. An asynchronous method runs in a thread separate from the main application thread. The processing results are fetched through another call on another thread. It does not block or wait for the API call to return from the server. Execution continues on in your program, and when the call returns from the server, a “callback” function is executed.

4. Briefly explain what the sources tab in Chromes Developers Tools does.

The sources tab in Chromes developer tool allows you to set breakpoints and evaluate expressions in JavaScript whether that JavaScript was loaded from a separate file or was part of a page.

5. What is a JavaScript promise?

A promise represents the eventual result of an asynchronous operation. It is a placeholder into which the successful result value or reason for failure will materialize. Promises provide a simpler alternative for executing, composing and managing asynchronous operations. They also allow you to handle asynchronous errors using approaches similar to synchronous try/catch.

It can be in 1 of 3 states: pending, fulfilled and rejected. When a promise is pending, it can transition to the fulfilled or rejected state. Once a promise is fulfilled or rejected, it will never transition to any other state, and its value or failure reason will not change.

6. What is an API?

API is short for application programming interface. It is a way to serve data. It allows developers to create complex functionality more easily. They abstract more complex code away from you, providing some easier syntax to use in its place.

7. What is a JavaScript Event?

JavaScript's interaction with HTML is handled through events. They occur when the user or the browser manipulates a page. When the page loads, it is called an event. It's also an event when the user clicks a button, close a window, resize a window or press any key. Developers can use these events to execute JavaScript code responses, which cause buttons to close windows, messages being displayed to users, data to be validated and virtually any other type of response. Events are a part of the document object model and every HTML element contains a set of events which can trigger JavaScript code.

8. What is event bubbling?

The bubbles event property returns a Boolean value that indicates whether or not an event is a bubbling event. It directs an event to its intended target. If a button is clicked and the event is directed to the button and an event handler is set for that object, the event is triggered. If an event handler is set for that object, the event is triggered. But if no event handler is set for that object, the event bubbles up to the object's parents. So, if an event happens on an element, it first runs the handlers on it, then on its parent, then all the way up to other ancestors.

For example, on a form with <div> and <p>, a click on the inner <p> first runs onclick on that <p>, then the outer <div> and then the outer <form>. Then it

moves upwards till the document object. So, if we click on <p>, we'll see three alerts: p

form.

This process is called bubbling because events “bubble” from the inner element and up through parents. It is also called event propagation.