

What is the DOM?

DOM is a programming interface for HTML and XML documents and stands for “Document Object Model”. It represents the page so that we can change the style, document structure and also content. JavaScript is a language that the browser reads and does interactions with, and the DOM is where it all happens. A lot of what seems to be a "JavaScript Thing" is more accurately to be a "DOM API".

List three methods to manipulate the DOM.

1. Accessing the elements
2. Accessing a single element
3. Adding and Removing HTML content

What is an asynchronous call?

An asynchronous function is defined by an `async` function declaration, that returns an `AsyncFunction` object. It operates asynchronously via an event loop, using a “Promise” to return it's result(s).

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

Sources tab is mainly used for debugging. You can set up different breakpoints within code lines and load the page again, so that it's possible to check line by line where the problem is at.

What is a JavaScript promise?

A promise is an object that is capable of producing a single value, some time in the future of time: either it can be a resolved value, or a reason that it's not resolved (for instans errors in network etc). A promise may be in one of 3 possible states: pending, fulfilled or rejected.

What is an API?

An API stands for application-programming interface. It is a set of instructions and standards that are given so accessing a software application or Web tool is possible. For instance: a software company releases different API's to the public so that other software developers can use it to power their products by designing them after their needs.

What is a JavaScript Event?

JavaScript interacts with HTML handled through something called "events". These occur when a user or the browser manipulates a page. When the page loads, the process is called an event. When the user clicks a button, that click is also an event.

What is event bubbling?

Event bubbling is a term that relates to the order of how the event handlers are called when one element is nested inside a second element, and both elements have registered a listener for the same event.