1. Why is it important to write clean code?

Writing clean code is important because it makes code easier to read and understand, which can help prevent bugs and helps to make the code more efficient. It also helps other developers who may need to work on the code in the future.

1. What is the difference between good comments and bad comments?

Good comments provide helpful information about the code, such as what it does, why it does it, and how it works. Bad comments are vague or irrelevant, such as comments that simply repeat the code or are not related to the code.

1. What is an array?

An array is a collection of related data items that are stored in a specific order.

1. What are arrays useful for?

Arrays are useful for storing and manipulating related data items, such as lists of numbers, strings, or objects. They can also be used for iterating over a set of data, such as in a for loop.

1. How do you access an array element?

You can access an array element by using its index, which is the position of the element in the array.

1. How do you change an array element?

You can change an array element by using its index and assigning a new value.

1. What are some useful array properties?

Some useful array properties include length, which returns the number of elements in the array, and sort, which sorts the elements in the array.

1. What are some useful array methods?

Some useful array methods include push, which adds an element to the end of the array, and pop, which removes the last element in the array.

1. What are loops useful for?

Loops are useful for repeating an action multiple times. This can be used to iterate through an array or object, or to execute a set of instructions multiple times.

1. What is the break statement?

The break statement breaks out of a loop, so that the loop is no longer executed.

1. What is the continue statement?

The continue statement skips the current iteration of the loop and moves on to the next iteration.

1. What is the DOM?

The DOM (Document Object Model) is a representation of an HTML document in the form of a tree structure. It is used to access and manipulate elements in an HTML document.

1. How do you target the nodes you want to work with?

You can target nodes you want to work with by using query selectors, such as getElementById or querySelector, or by traversing the DOM tree.

1. How do you create an element in the DOM?

You can create an element in the DOM by using the createElement method.

1. How do you add an element to the DOM?

You can add an element to the DOM by using the appendChild or insertBefore methods.

1. How do you remove an element from the DOM?

You can remove an element from the DOM by using the removeChild method.

1. How can you alter an element in the DOM?

You can alter an element in the DOM by using methods such as setAttribute, removeAttribute, or style.

1. When adding text to a DOM element, should you use textContent or innerHTML?

When adding text to a DOM element, you should use textContent, as it is more secure and faster than using innerHTML.

1. Where should you include your JavaScript tag in your HTML file when working with DOM nodes?

You should include your JavaScript tag at the end of the HTML file, just before the closing body tag.

1. How do “events” and “listeners” work?

Events are actions that a user can perform, such as clicking a button or hovering over an element. Listeners are functions that are triggered when an event occurs.

1. What are three ways to use events in your code?

Three ways to use events in your code are to use inline event handlers, addEventListener, and use an event listener on a group of nodes.

1. Why are event listeners the preferred way to handle events?

Event listeners are the preferred way to handle events because they are more efficient and more flexible than inline event handlers.

1. What are the benefits of using named functions in your listeners?

The benefits of using named functions in your listeners are that they are easier to debug and can be reused in other parts of your code.

1. How do you attach listeners to groups of nodes?

You can attach listeners to groups of nodes by using querySelectorAll to select all the nodes and then adding an event listener to each one.

1. What is the difference between the return values of querySelector and querySelectorAll?

The return value of querySelector is a single node, while the return value of querySelectorAll is a nodelist, which contains all the nodes that match the selector.

1. What does a “nodelist” contain?

A nodelist contains all the nodes that match a given selector.

1. Explain the difference between “capture” and “bubbling”.

Capture is the process of capturing an event when it is first triggered, and bubbling is the process of capturing an event when it is bubbled up the DOM tree.

1. What is the difference between objects and arrays?

The difference between objects and arrays is that objects are collections of key-value pairs, while arrays are collections of related data items stored in a specific order.

1. How do you access object properties?

You can access object properties by using dot notation or bracket notation.