

Commonly asked JavaScript interview questions

vikasvrao

February 7,
2010

1. Difference between window.onload and onDocumentReady?

The onload event does not fire until every last piece of the page is loaded, this includes css and images, which means there's a huge delay before any code is executed.

That isn't what we want. We just want to wait until the DOM is loaded and is able to be manipulated.

onDocumentReady allows the programmer to do that.

2. What is the difference between == and === ?

The == checks for value equality, but === checks for both type and value.

3. What does "1"+2+4 evaluate to? What about 5 + 4 + "3"?

Since 1 is a string, everything is a string, so the result is 124. In the second case, it's 93.

4. What is the difference between undefined value and null value?

undefined means a variable has been declared but has not yet been assigned a value. On the other hand, null is an assignment value. It can be assigned to a variable as a representation of no value.

Also, undefined and null are two distinct types: undefined is a type itself (undefined) while null is an object.

Unassigned variables are initialized by JavaScript with a default value of undefined. JavaScript never sets a value to null. That must be done programmatically.

5. How do you change the style/class on any element?

```
document.getElementById("myText").style.fontSize = "20";
```

-or-

```
document.getElementById("myText").className = "anyclass";
```

6. What are Javascript closures? When would you use them?

Two one sentence summaries:

- * a closure is the local variables for a function – kept alive after the function has returned, or
- * a closure is a stack-frame which is not deallocated when the function returns.

A closure takes place when a function creates an environment that binds local variables to it in such a way that they are kept alive after the function has returned. A closure is a special kind of object that combines two things: a function, and any local variables that were in-scope at the time that the closure was created.

The following code returns a reference to a function:

```
function sayHello2(name) {  
  var text = 'Hello ' + name; // local variable  
  var sayAlert = function() { alert(text); }  
  return sayAlert;  
}
```

Closures reduce the need to pass state around the application. The inner function has access to the variables in the outer function so there is no need to store the information somewhere that the inner function can get it.

This is important when the inner function will be called after the outer function has exited. The most common example of this is when the inner function is being used to handle an event. In this case you get no control over the arguments that are passed to the function so using a closure to keep track of state can be very convenient.

7. What is unobtrusive javascript? How to add behavior to an element using javascript?

Unobtrusive Javascript refers to the argument that the purpose of markup is to describe a document's structure, not its programmatic behavior and that combining the two negatively impacts a site's maintainability. Inline event handlers are harder to use and maintain, when one needs to set several events on a single element or when one is using event delegation.

```
1 <input type="text" name="date"
  />
```

Say an input field with the name "date" had to be validated at runtime:

```
1 document.getElementsByName("date")[0].
2     addEventListener("change", validateDate,
3 false);
4
5 function validateDate() {
6 }
```

Although there are some browser inconsistencies with the above code, so programmers usually go with a javascript library such as JQuery or YUI to attach behavior to an element like above.

8. What is Javascript namespacing? How and where is it used?

Using global variables in Javascript is evil and a bad practice. That being said, namespacing is used to bundle up all your functionality using a unique name. In JavaScript, a namespace is really just an object that you've attached all further methods, properties and objects. It promotes modularity and code reuse in the application.

9. What datatypes are supported in Javascript?

Number, String, Undefined, null, Boolean

10. What is the difference between innerHTML and append() in JavaScript?

InnerHTML is not standard, and its a String. The DOM is not, and although innerHTML is faster and less verbose, its better to use the DOM methods like appendChild(), firstChild.nodeValue, etc to alter innerHTML content.