# **Difference between window, screen, and document in JavaScript**

**Window**

Each browser tab has its own top-level ***window*** object. Each ***<iframe>***(and deprecated ***<frame>***) element has its own ***window*** object too, nested within a parent window. Each of these windows gets its own separate global object. ***window.window*** always refers to ***window***, but ***window.parent*** and ***window.top*** might refer to enclosing windows, giving access to other execution contexts. In addition to ***document*** and ***screen*** described below, ***window*** properties include

* ***setTimeout()*** and ***setInterval()*** binding event handlers to a timer.
* ***location*** giving the current URL.
* ***history*** with methods ***back()*** and ***forward***() giving the tab's mutable history.
* ***navigator*** describing the browser software.

**Screen**

Each ***window*** object has a ***document*** object to be rendered. These objects get confused in part because HTML elements are added to the global object when assigned a unique id. E.g., in the HTML snippet

*<body>*  
*<p id=”hello”>This is the first paragraph.</p>*  
*</body>*

the paragraph element can be referenced by any of the following:

* ***window.hello*** or ***window[“hello”]***
* ***document.getElementById(“hello”)***
* ***document.body.firstChild***
* ***document.body.children[0]***

**document**

The ***window*** object also has a ***screen*** object with properties describing the physical display:

* screen properties **width** and ***height*** are the full-screen.
* screen properties ***availWidth*** and ***availHeight*** omit the toolbar.

The portion of a screen displaying the rendered document is the **viewport** in JavaScript, which is potentially confusing because we call an application’s portion of the screen a window when talking about interactions with the operating system. The ***getBoundingClientRect()*** method of any ***document*** element will return an object with ***top, left, bottom,*** and ***right*** properties describing the location of the element in the viewport.