==================================================================

**What are local, global, private scope in Java Script?**

<http://www.w3schools.com/js/js_function_closures.asp>

1. JavaScript variables can belong to the **local** or **global** scope.
2. In a web page, global variables belong to the window object.
3. Global variables can be used (and changed) by all scripts in the page (and in the window).
4. A local variable can only be used inside the function where it is defined.
5. Variables created **without** the keyword **var**, are always global, even if they are created inside a function.
6. Global variables live as long as your application (your window / your web page) lives.
7. Local variables have short lives. They are created when the function is invoked, and deleted when the function is finished.
8. JavaScript **closure** makes it possible for a function to have "**private**" variables.

==================================================================

**How JavaScript Event Delegation Works?**

<https://davidwalsh.name/event-delegate>

DOM event delegation is a mechanism of responding to ui-events via a single common parent rather than each child, through the magic of event "bubbling" (aka event propagation).

<http://stackoverflow.com/questions/1687296/what-is-dom-event-delegation>

==================================================================

**What is Variable Hoisting?**

[**http://www.w3schools.com/js/js\_hoisting.asp**](http://www.w3schools.com/js/js_hoisting.asp)

In JavaScript, a variable can be declared after it has been used.  
Hoisting is JavaScript's default behavior of moving all declarations to the top of the current scope

==================================================================