A callback function is a function which is:

* passed as an argument to another function, and,
* is invoked after some kind of event.

Once its parent function completes, the function passed as an argument is then called.

Ex:

<script>

function mySandwich(param1, param2, callback) {

alert('Started eating my sandwich.\n\nIt has: ' + param1 + ', ' + param2);

callback();

}

mySandwich('ham', 'cheese', function() {

alert('Finished eating my sandwich.');

});

</script>

**Pseudocode:**

// The callback method

function meaningOfLife()

{

log("The meaning of life is: 42");

}

// A method which accepts a callback method as an argument

// takes a function reference to be executed when printANumber completes

function printANumber(int number, function callbackFunction) {

print("The number you provided is: " + number);

}

// Driver method

function event() {

printANumber(6, meaningOfLife);

}

**Result if you called event():**

The number you provided is: 6

The meaning of life is: 42

Callbacks are so-called due to their usage with pointer languages. If you don't use one of those, don't labour over the name 'callback'. Just understand that it is just a name to describe a method that's supplied as an argument to another method, such that when the parent method is called (whatever condition, such as a button click, a timer tick etc) and its method body completes, the callback method is then invoked, or in other words "called at the back" of the other function.