

# Firefox Extension Development Tutorial

## Table of contents

1 Overview.....	2
2 JavaScript Basics.....	2
3 Working with the DOM.....	3
4 Further Reading.....	4

## 1. Overview

All of the guts of your extension will be written in JavaScript. If you already know JavaScript than you are good to go! If you are new to JavaScript it is highly recommended that you read through the next section and the supporting documentation to get used to the syntax.

All of your JavaScript code files should go in the content directory of your extension. This is where the XUL files go as well, so they will be able to easily reference the code. Just insert the following line in an XUL file that needs to run functions from your files...

```
<script type="application/x-javascript"
src="chrome://hpsched/content/mycode.js" />
```

Make sure you use the chrome URI so that Firefox can find the file correctly.

## 2. JavaScript Basics

JavaScript is a fairly straight forward scripting language. Let's jump right into some sample code:

```
/*
 * Save the Schedules List to the
"extensions.hpsched.schedules" preference.
 * This is called by the pref's system when the GUI element is
altered.
 */
function saveSchedulesList() {
  var schedulesList =
document.getElementById("schedulesList").childNodes;
  var prefString = "";

  for (var i=0; i < schedulesList.length; i++) {
    var columns = schedulesList[i].childNodes[0].childNodes;
    var str = columns[0].getAttribute("label") + ".ITEM."
      + columns[1].getAttribute("label") + ".ITEM."
      + columns[2].getAttribute("label") + ".ITEM."
      + columns[3].getAttribute("label");
    if(prefString == "")
      prefString = str;
    else
      prefString += ".NEXT." + str;
  }

  /* return the new prefString to be stored by pref system */
  return prefString;
}
```

The first line defines a function named 'saveSchedulesList' that brings in zero arguments. The next two lines create and initialize two variables. Notice that JavaScript is a typeless language, just type var to let the parser know you are declaring a variable. The for-loop and if statement looks just like what you would expect in C or Java. You will also notice that you do not have to state that this function is (or is not) returning a value. If you would like to return a value, just use the common 'return x;' statement. If no return statement is provided then the function returns (void), which is ignored by the calling function.

Debugging JavaScript is also quite easy: just use the 'alert()' function call. Whenever Firefox is running code and sees a call such as 'alert("I am here!")', Firefox will display a graphical error box with the given message. This is a wonderful way to determine if a certain area of code is being executed, what the current value of some variable is (i.e. alert(myVar)), and more. Use alert(), it is your friend!

Please see the Further Reading section for links to some wonderful JavaScript tutorials.

### 3. Working with the DOM

JavaScript is very common in web development because it provides functionality to easily work with the DOM (Document Object Module). All HTML files on the net are rendered by the browser by first building a tree. Each node in the tree corresponds to a tag in the HTML file. Firefox's XUL files work the same way, all elements are nodes in a large tree (per file). The top level element is called the 'document'.

There is one very important JavaScript function that you will need to know to be able to work with the XUL elements in your GUI. It allows you to access the object itself, using its ID. For example, when the Home Page Scheduler wants to load the saved schedules into the tree element that displays them to the user, it need to get access to the tree object itself. The tree's ID is schedulesTree.

```
<var schedulesTree =  
document.getElementById("schedulesTree");
```

This returns the object representing the tree element. At this point we can get and set various attributes related to the tree object. For example, to get the currently selected index we just use 'schedulesTree.currentIndex'.

To see a list of attributes and methods defined for the various element types just look them up in the XUL element reference mentioned earlier: <http://xulplanet.com/references/elemref/>. You can learn all about the tree element, for example, here: [http://xulplanet.com/references/elemref/ref\\_tree.html](http://xulplanet.com/references/elemref/ref_tree.html).

## 4. Further Reading

*XUL Element Reference:* [http://xulplanet.com/references/elemref/ref\\_tree.html](http://xulplanet.com/references/elemref/ref_tree.html)

*JavaScript Tutorials and Reference:* <http://www.w3schools.com/js/default.asp>

*A great JavaScript Tutorial for the Total Non-Programmer:*

<http://www.webteacher.com/javascript/>