

Intro to Greasemonkey

Having it Your Way on the Web Since 2005

Eric Promislow

ActiveState Software Inc.

What is it?

What is it?

- A Firefox extension that changes the look and behavior

What is it?

- A Firefox extension that changes the look and behavior
- A creativity enhancement tool that builds personal growth

How?

- Gets you to constantly question web sites
 - What's wrong?
 - What am I missing?
 - What do I want this site to do?
 - How can I improve it?

Cost/Time Tradeoff

- The more scripts you write, the lower the marginal cost of future scripts

Today's Agenda

- Installing GM
- Installing scripts
- Editing scripts
- JS & CSS in a nutshell
- Cautions

Installing Greasemonkey

- google
- "extension developer's extension"
- Ted's Mozilla page
- Gives you a JS command-line

userscripts.org

- Useful for installing commonly-used scripts
- Always read the script first
 - Understanding focus of this talk

Convenience

- Scripts are stored in `<profile>/gm_scripts`
- Edit live
- Demo Komodo Edit:
 - Open... Shortcut
 - Code-Completion, JS-Aware

JavaScript

- What you need to know after you've learned the basics

JS Context

- Each web page runs in its own JS **context**.
- Each context has its own set of globals, state, cookies
- Create a library of shared code with an extension
- Each GM script lives in the page's context

Anonymous Functions

```
(function() {  
  var a = 3;  
  alert("Inside: a=" + a);  
})();  
alert("Outside: a=" + a)
```

Closures

```
var f = function(form, name, value) {  
    alert("field " + name + "= value" + value);  
}  
setTimeout(f, 1000, this, 'foo', this.foo.value);
```

Wrong:

```
setTimeout("f(" + form + "," +  
    name + "," + value + ")", 1000);
```

Loops

For arrays, loop by number. Either this

```
for (var i = 0; i < coll.length; i++) {  
    var item = coll[i]; ...  
}
```

or:

```
for (var item, i = 0; item = coll[i]; i++) {...
```

not:

```
for(var idx in coll) { var item = coll[idx]; ... }
```

XPath Expressions

```
snaps = document.evaluate(expn,  
    node,  
    null, // namespace resolver  
    XPathResult.  
    UNORDERED_NODE_SNAPSHOT_TYPE,  
    null); // prev xpath result  
  
for(var i = 0; i < snaps.snapshotLength; i++) {  
    var snap = snaps.snapshotItem(i)
```

XPath

```
function xpath(expn, node) {  
  if (typeof(node) == "untitled")  
    node=document;  
  return document.evaluate(expn,  
    node, null,  
    XPathResult.  
    UNORDERED_NODE_SNAPSHOT_TYPE,  
    null);  
} // define every time or write an extension
```

XPath in a Nutshell

`//form -- document.forms`

`//input[@type='password']`

`//a/@href -- document.links`

XPathResult

- Curious?
- Search lxr.mozilla.org (TextSearch)
- Find `nsIDOMXPathResult.idl` for other values
- Also try JS shell: `XPathResult.<tab>`

DOM

```
newNode = doc.createElement('name');  
parent.appendChild(newNode);
```

```
node.parentNode.removeChild(node);
```

```
newNode = currNode.parentNode.insertBefore(  
    newNode, currNode.nextSibling);
```

```
oldNode.parentNode.replaceChild(  
    newNode, oldNode);
```

DOM Cheat Sheets

- [DOM site:w3.org](http://www.w3.org)
- Find a link that says something like “Appending B: IDL Definitions”
- Start with level 1
- Try them out in the JS Shell

Debugging

- You can't (anymore)
- Test parts in isolation with Javascript Shell
- GM_log and Error Console

Anatomy of a Script

```
// ==UserScript==  
// @name      Script Title  
// @namespace http://me.org/name  
// @description does this...  
// @include  http://*  
// @include  https://*  
// @exclude  http://gmail.google.*/*  
// ==/UserScript==
```

```
(function(){ ... })();
```

Patterns

Changing Styles

```
// This adds a global <style> tag to the HTML head
function addGlobalStyle(css) {
  var head = document.getElementsByTagName('head')[0];
  if (! head) return;
  var style = document.createElement('style');
  style.type = 'text/css';
  style.innerHTML = css;
  head.appendChild(style);
}
```

```
addGlobalStyle(('body {'
  + 'background-color: white; \n'
  + 'color: black; \n}'));
```

More Styles

- `node.style.cssAttrName = value;`
 - `div.style.backgroundColor = 'white';`
- Wrong:

```
node.setAttribute("style",  
    "background-color: white;");
```

Adding Buttons

- Example: Freerice cheat
- Create the button
- Style it
- `addEventListener('click', function, false)`
- Press it

Events

- No more onclick handlers, etc.
- `var func = function(event) {
 ...
};`
- `node.addEventListener('click', func, false);`

Use whitelists

- Linkify
 - Use to check for styles and scripts
 - Now checks for allowed parents

Security

- GM runs in a sandbox, albeit a powerful one
- API adds capability to user scripts:
 - GM_setValue, GM_getValue
 - GM_xmlHttpRequest - not limited

Sandbox

- window, document, etc. are “wrapped” objects
- If it isn't in the IDL file, you can't do it
- document.evaluate - ok
- div.onclick - no, use
div.addEventListener('click', ...)

Warning signs

- `unsafeWindow`
- `node.wrappedJSObject`
- Don't run these with arbitrary web pages,
- Only web pages you trust

References

- <http://diveintogreasemonkey.org/>
- Greasemonkey hacks - O'Reilly
- userscripts.org
- <http://blogs.activestate.com/ericp>
- ericp@activestate.com