



# Adapt Oscar, not your workflow:

Use Greasemonkey to extend Oscar to fit your  
workflow!  
Here's how...

# Greasemonkey & Oscar

- What is Greasemonkey?

Greasemonkey is a software tool, a Firefox extension that allows Firefox to extend Oscar.

- What is a Firefox extension?

Firefox extensions are what eForms are to Oscar. They add extra functionality to Firefox.

# Installing Greasemonkey

- Firefox addons/extensions are curated by the Mozilla foundation at [addons.mozilla.org](https://addons.mozilla.org)

<https://addons.mozilla.org/en-US/firefox/>

- Greasemonkey is available on that site:

<https://addons.mozilla.org/en-US/firefox/addon/greasemonkey/>



*Greasemonkey 1.8*

by [Anthony Lieuallen](#), [Aaron Boodman](#), [Johan Sundström](#)

Customize the way a web page displays or behaves, by using scripts

[Download Now](#)

[Privacy Policy](#)

# Greasemonkey Scripts

- What are they and where do we get them?

Scripts are sets of instructions to Firefox, by following which, Firefox is able to extend Oscar to provide extra functionality

- Oscar-in-Tabs specific scripts

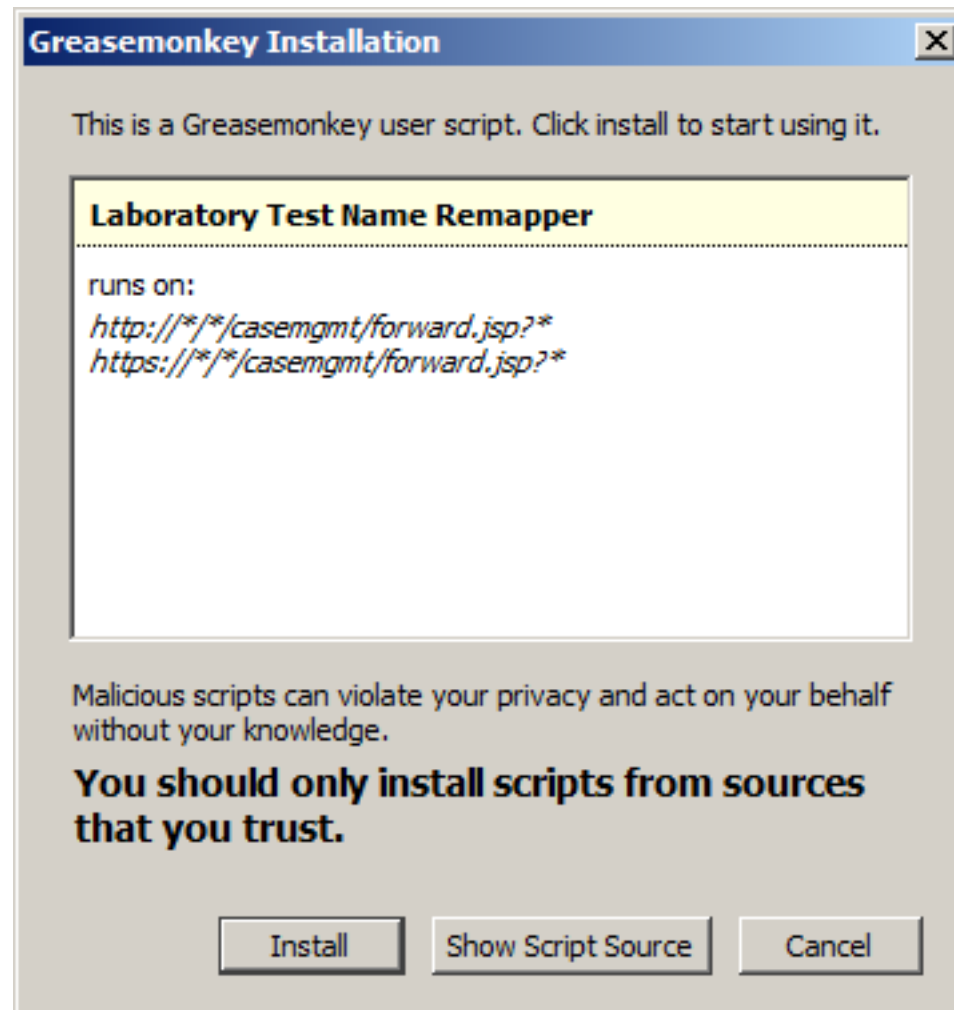
<http://www.oscarcanada.org/oscar-users/emr-resource/running-oscar-in-browser-tabs>

- Main repository for Oscar scripts

<http://www.oscarcanada.org/oscar-users/emr-resource/eform/eforms-in-development-beta-testing/useful-code/greasemonkey-scripts>

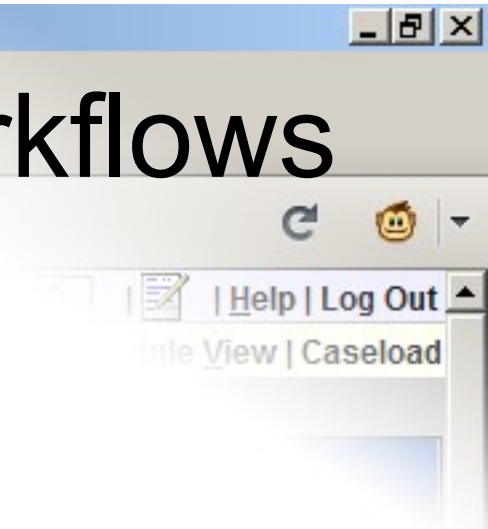
# Installing a Greasemonkey script

- Open the script file or drag and drop into Firefox



# Existing script enhanced workflows & interfaces

- Oscar-in-Tabs
- Express Billing
- Send and Archive for Oscar Messenger
- Signing Rx using Smartphone/Tablet
- Printing Rx (Oscar LocalPrint)
- Keyboard Shortcuts for Echart & Schedule



# Express Billing

- This script reduces the number of clicks required per billing.

## oscarBC Billing

Service Code:  Diagnostic Code:

- It brings the 'Save Bill' and 'Another Bill' buttons to the first billing page.
- It adds a button, 'Save Bill & Open Next' to save the bill and open the next unbilled chart from the Schedule.

# Send and Archive



This script adds a 'Send and Archive' button to Oscar Messenger when replying to a message.

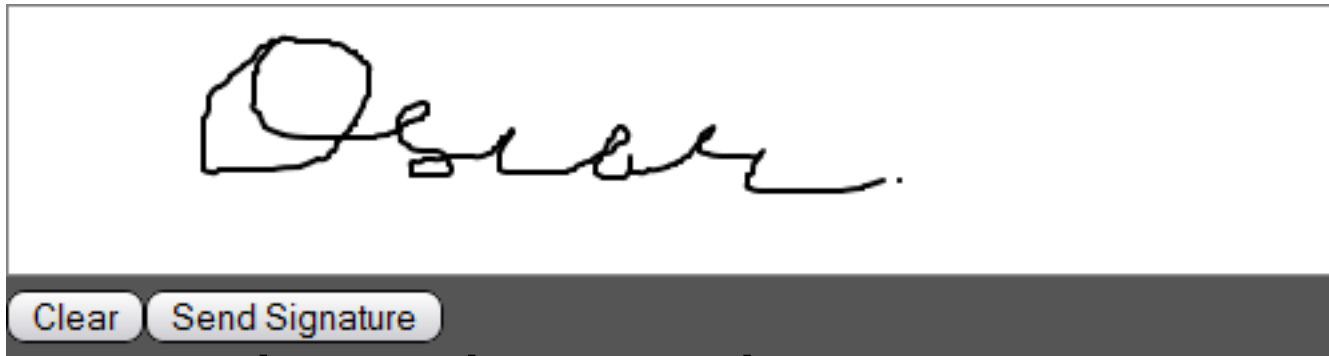
The button allows you to reply to a message and archive the message with a click.



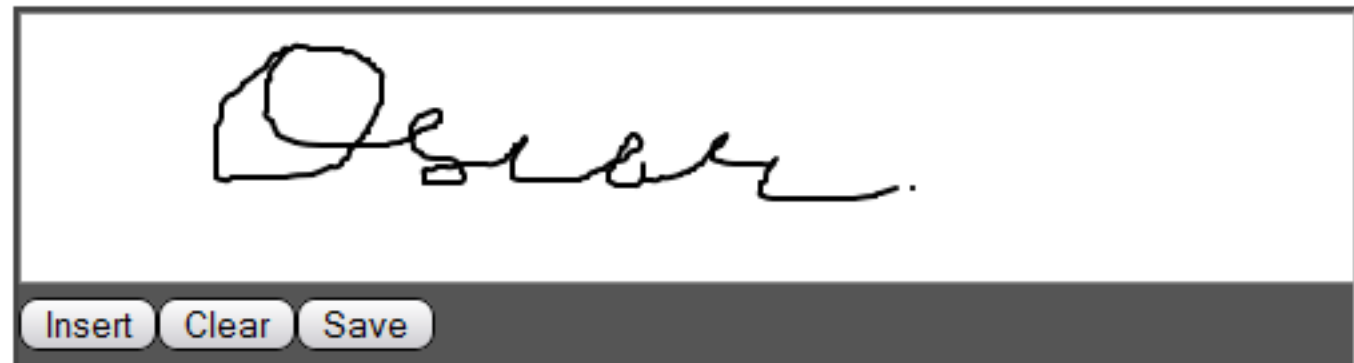
# Rx Signing Enhancement

- Rx Signing

This script along with SignaturePad.htm for smartphones/tablets allows you to sign on your smartphone/tablet...



... and send your signature to your Oscar workstation.



# Rx Printing

- Rx Printing

This script and the extension for Firefox on Android allows you to print securely from Oscar to locally shared printers.

## Actions

Size of Print PDF :

Print PDF

Print Server:

Printers:

Printer:

Send To Printer

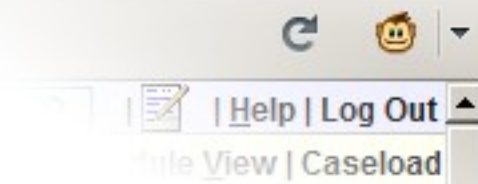
Status: Idle

Print

Print & Paste into EMR

Create New Prescription

Close Window



# Keyboard Shortcuts

- Keyboard Shortcuts for the EChart

This script adds a keyboard shortcut to every EChart. Pressing the Alt key and 4 will 'Save, Sign & Bill' the EChart.

- Keyboard Shortcuts for the Schedule

This script adds a keyboard shortcut to the Schedule. Pressing the Alt key and U will open the next unbilled chart from the Schedule.

- Both scripts contain instructions on how to add more keyboard shortcuts.

# How can I extend Oscar with a Greasemonkey script?

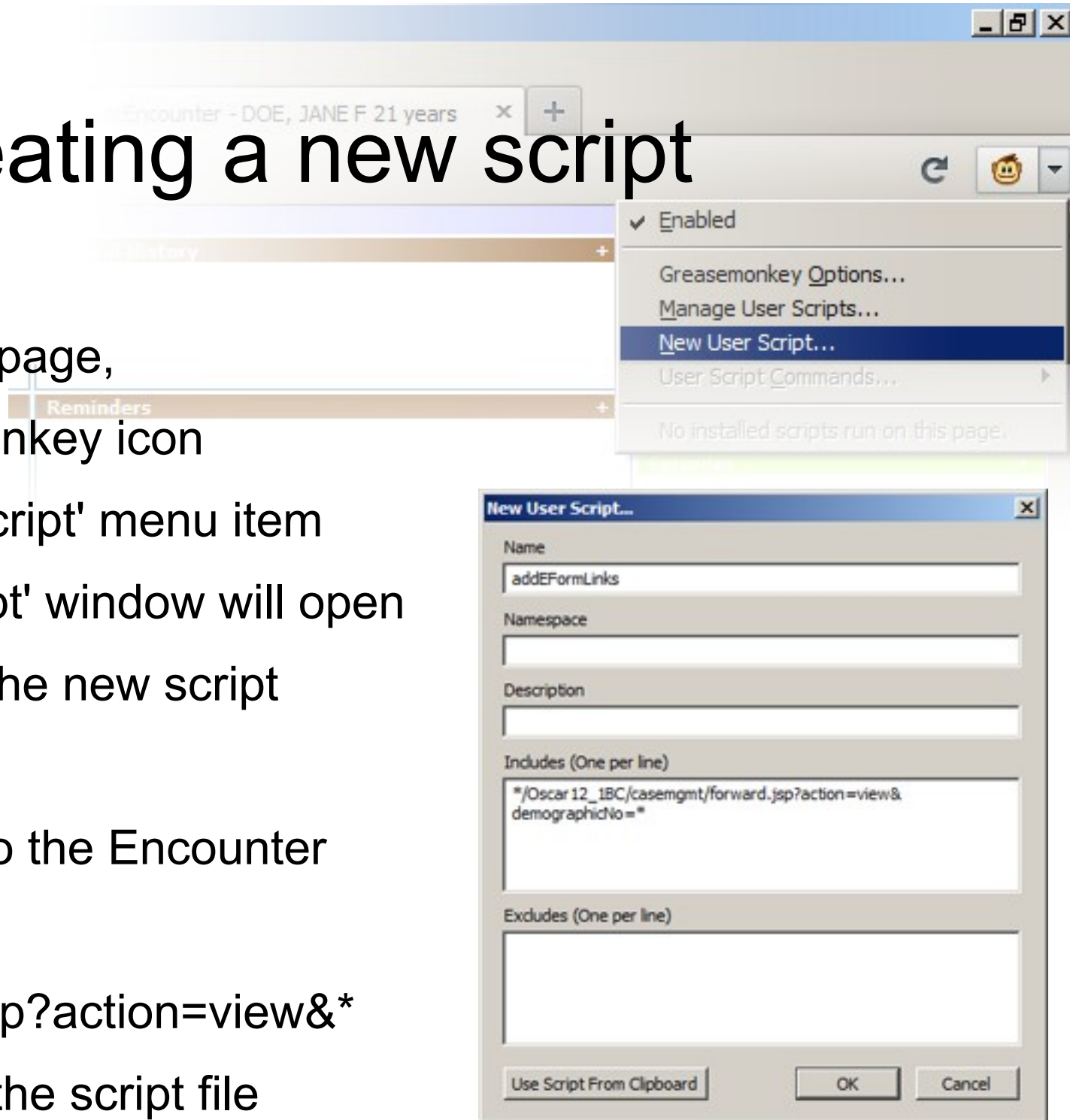
- If you have written eForms for Oscar, you are already on your way to writing Greasemonkey scripts for Oscar.
- The key insight is to realize that Oscar itself is a series of eForms.
- Greasemonkey allows you to modify that series of eForms to suit your needs.

# The Script Tutorial

- For this tutorial, let's add a link to the Lab Req eForm at the bottom of the left column on the Encounter pages in Oscar.
- Let's begin by letting Greasemonkey create a script file, ready for us to start writing the script...

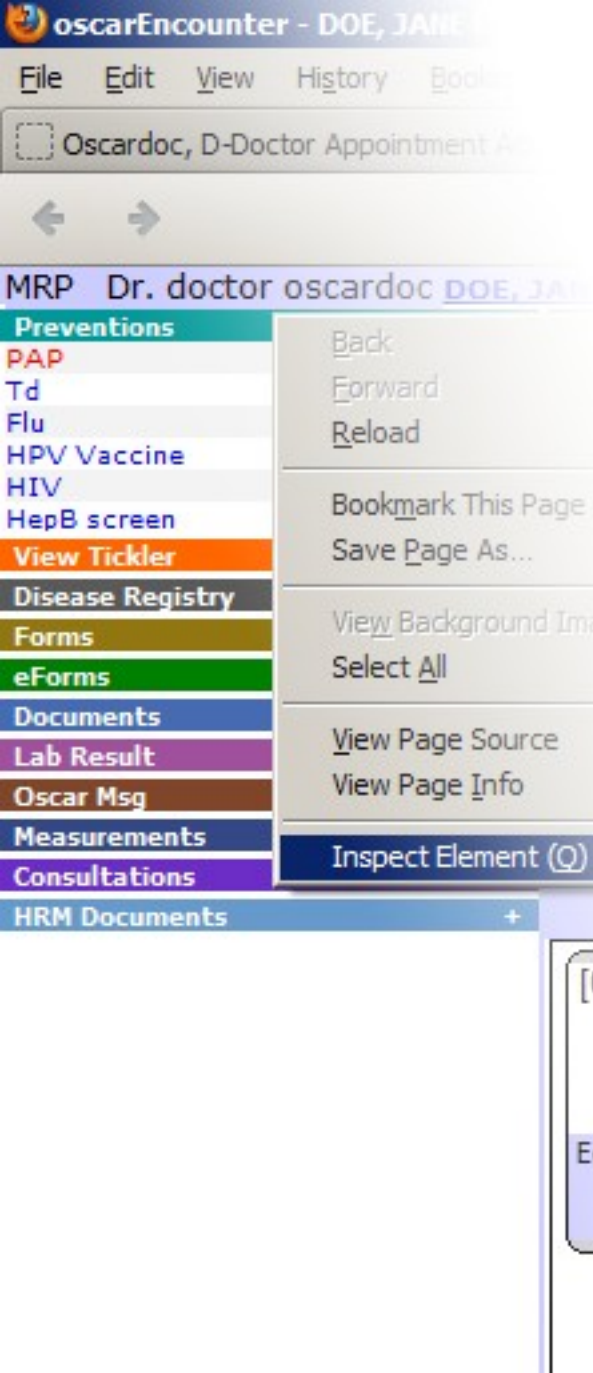
# Creating a new script

- From an Encounter page, click on the Greasemonkey icon
- Click on the 'New Script' menu item
- The 'New User Script' window will open
- Enter the name for the new script
- In the 'Includes', you will find the URL to the Encounter
- Edit it to as follows:  
`*/casemgmt/forward.jsp?action=view&*`
- Click 'OK' to create the script file



# Two Questions

- Before we write the first line of our script, there are two questions that need answers...
- How do we tell Greasemonkey where we want to add the eForm link on the Encounter page?
- How do we tell Greasemonkey which eForm we want to link to?



# Under the hood:

- To answer the first question, we have to look under the hood of the Encounter page...
- Building blocks of Oscar are HTML elements: Links, buttons, boxes, ...
- Every Oscar page is made up of these elements
- Firefox allows us to inspect these elements
- Right click above the Preventions bar
- Click 'Inspect Element'





- Flu
- HPV Vaccine
- HIV
- HepB screen
- View Tickler
- Disease Registry
- Forms
- eForms
- Documents
- Lab Result
- Oscar Msg
- Measurements
- Consultations
- HRM Documents

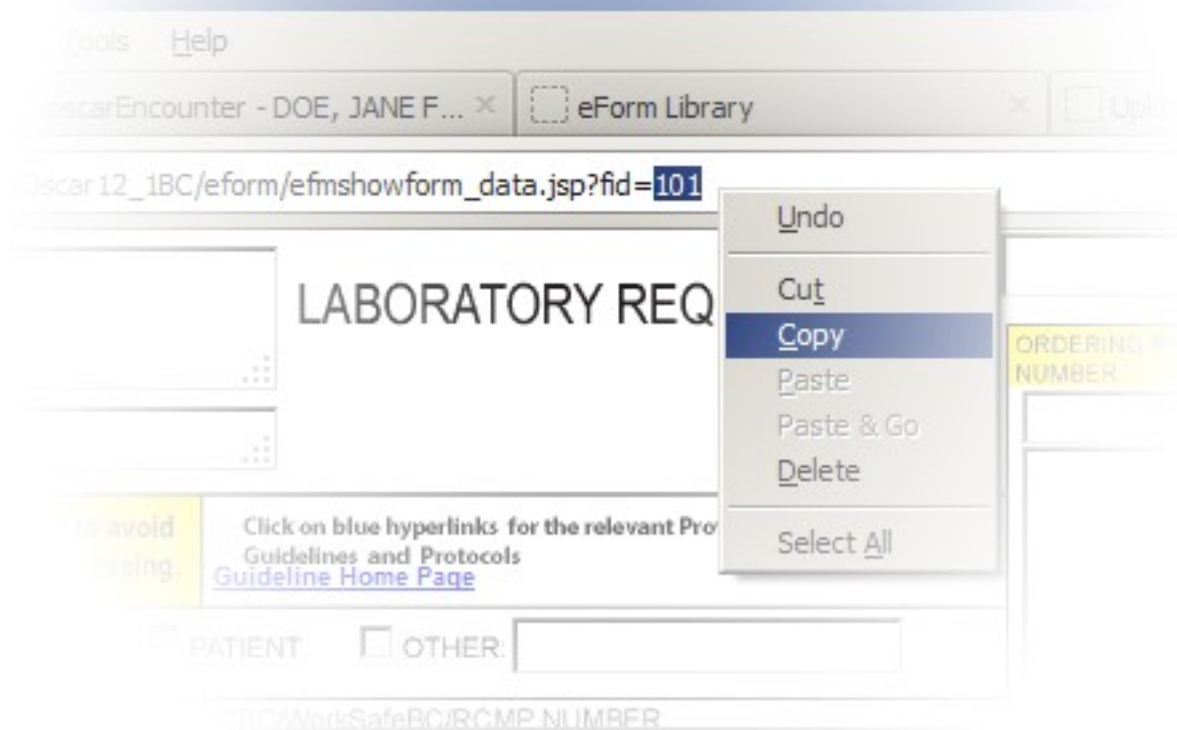
# Inspected

- Important Oscar elements have id's
- We tell Greasemonkey where to add the link by referring to the closest Oscar element by id
- When we inspect an element in Firefox, that will tell us the id of that element.
- The id of the left column is...
- id="leftNavBar"



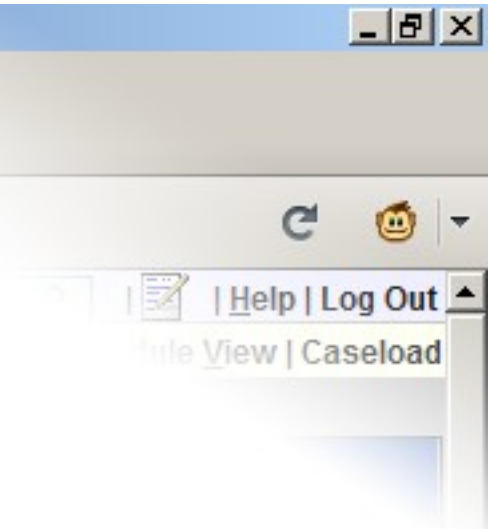
# An eForms's fid

- Every eForm has a fid
- To open an eForm from a script, the eForm's fid is required
- An eForm's fid is part of the link to/ address of the eForm. Easiest option is to open an eForm and copy the fid from Firefox's address bar.



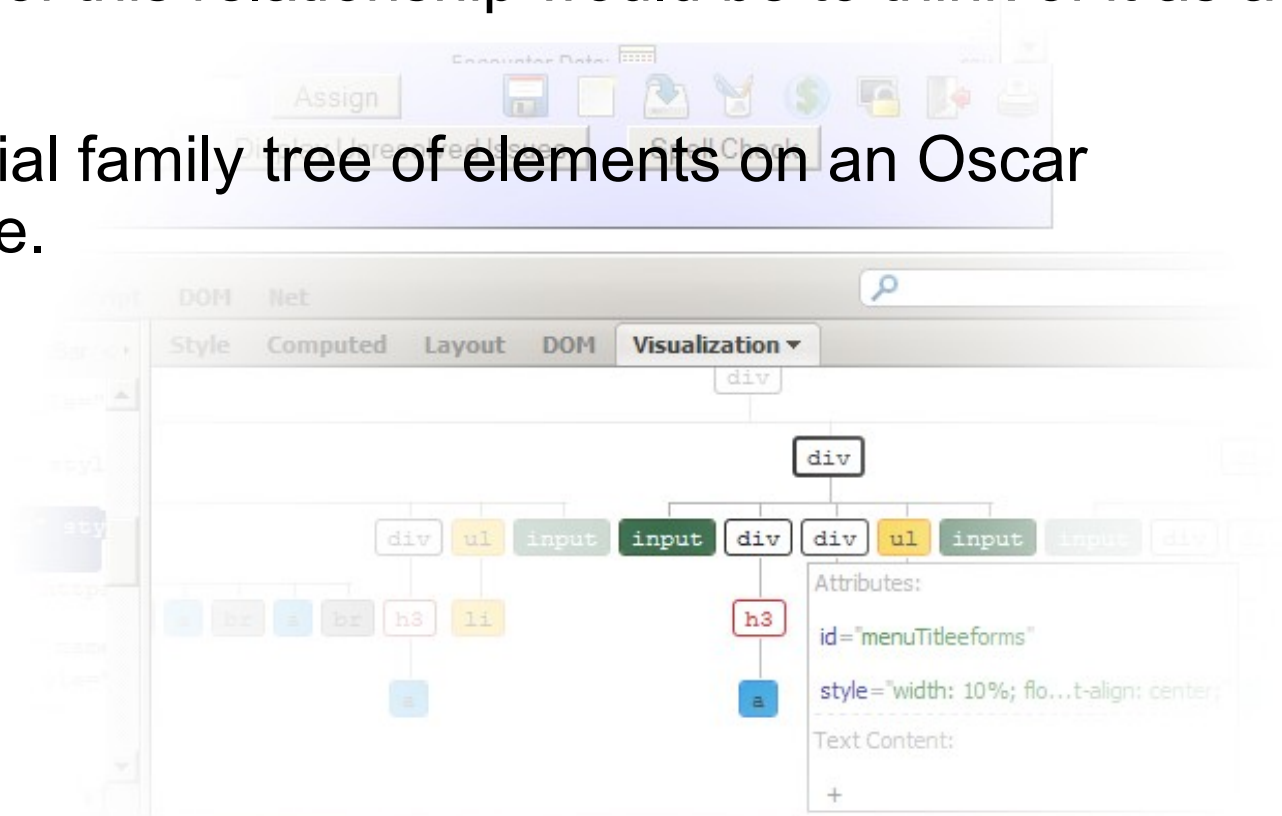
# Putting it all together

- We know where we want to add:  
Under the left column, the element with  
`id="leftNavBar"`
- We know what we want to add:  
A link to eForm with `fid="101"`
- Now we need to put it all together with  
Javascript



# Elemental Hierarchy

- The elements that make up every Oscar page have a hierarchical relationship with each other. This relationship becomes important when adding a new element to an Oscar page.
- A visualization of this relationship would be to think of it as a family tree
- Below is a partial family tree of elements on an Oscar Encounter page.



# Oscar Scripting Primer

- Finding an element:

```
var theTarget = document.getElementById("leftNavBar");
```

- Creating a link or an anchor element:

```
var theLink = document.createElement('a');
```

- Inserting the link: (Importance of elemental hierarchy)

```
theTarget.appendChild(theLink);
```

# Oscar Scripting Primer

- Final pieces of the puzzle...
- In our script, we tell Firefox to load our building block onto the Encounter page after all the other blocks for the Encounter page have loaded. This is important as all the building blocks have to be in place in order for us to build ours.

i.e. We can't build the chimney before the roof is built.

Javascript for that is:

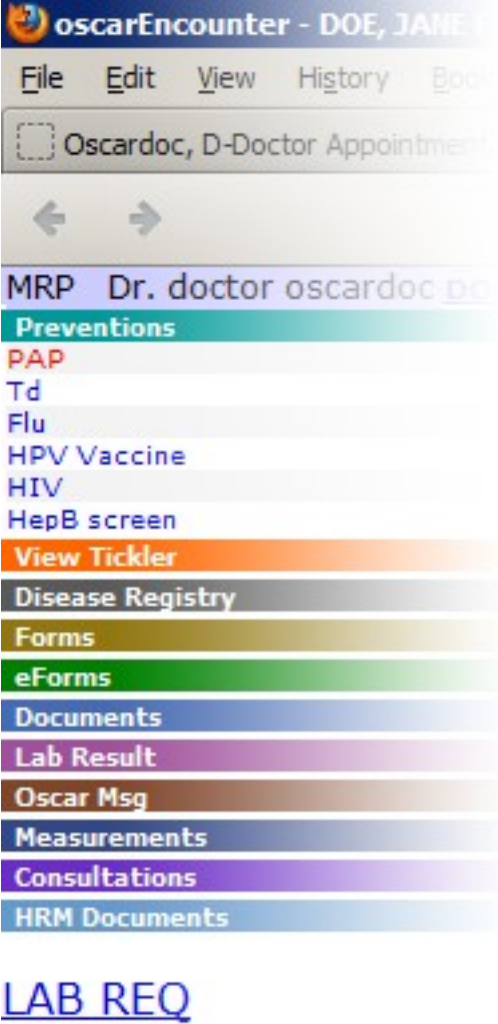
```
window.addEventListener("load", function() {},false);
```

- When opening an eForm, we tell Oscar which patient that the eForm applies to by specifying the patient's demographic number. In an Encounter page, Oscar helps us by assigning that value to demographicNo which we use to create the link to the eForm.

```
var theLabReqForm = "../eform/efmformadd_data.jsp?fid=101";  
theLabReqForm+="&demographic_no="+demographicNo;
```

# The Script

```
window.addEventListener("load", function() {  
    var theTarget = document.getElementById("leftNavBar");  
    var theLink = document.createElement("a");  
    /*Replace 101 with the Lab Req eForm's fid from your Oscar  
    server*/  
    var theLabReqForm = "../eform/efmformadd_data.jsp?fid=101";  
    theLabReqForm+="&demographic_no="+demographicNo;  
    theLink.href="javascript:void(open(""+theLabReqForm+"));"  
    theLink.innerHTML="LAB REQ";  
    theTarget.appendChild(theLink);  
}, false);
```



# The Script in Action

- Once the script is installed, all Encounter pages will have the 'LAB REQ' link, clicking which will open the Lab Req eForm.
- Using this script as template, quick links to other eForms are easily added to the Encounter pages



# The Final Word

We hope that you found this presentation and tutorial useful and that it has demonstrated the potential and power of the Greasemonkey extension to extend Oscar to fit individual workflows.

I, Darius, would like to thank Dr Robertson for reviewing and co-presenting this presentation.