Opening PDFs in a new window with JavaScript

Opening documents such as PDFs in a new window should be **automated using JavaScript** for the following reasons:

- Users will often close the web browser when a PDF is opened, mistakenly believing the document has been opened in Adobe Reader
- The attribute historically used to open a new window, target, has been removed from the HTML 4.01 Strict specification (it’s now deprecated)
- Opening a new window is a behaviour and should be moved to the behavioural layer.

Using JavaScript can also be particularly useful when a website is content managed. Rather than having to rely on site editors to remember to open a link to a PDF in a new window the process is handled by a simple JavaScript function.

Check out this fully functioning example to see what we’ll create. Check out Advanced Layer Popup if you want to open your PDF in a layer that pops up or Advanced Open Window if you want to create a new window with the press of a button.

Registering an event

The first task is to create an event that occurs when the page has loaded. Rather than trying to execute a function call using <body onload=“callfunction()”> within the HTML page we’re going to use Simon Willison’s addLoadEvent(func). This will allow us to add function calls once the page has loaded.
We’ll type the following JavaScript into the .js file (all subsequent code should be entered before this function so the addLoadEvent is last):

```
function addLoadEvent(func)
{
var oldonload = window.onload;

if (typeof window.onload != ‘function’)
{
window.onload = func;
} else {
window.onload = function()
{
oldonload();
func();
}
}
}
addLoadEvent(fNewPDFWindows);
```

The function fNewPDFWindows()

We'll create an empty function in the JavaScript file called fNewPDFWindows. To avoid any JavaScript errors with the script, we also check to see if the command getElementsByTagName is available:

```
function fNewPDFWindows ()
{
if (!document.getElementsByTagName) return false;
}
The second task is to create an HTML object collection of any links within the page.
The following line gets all links within the page:
var links = document.getElementsByTagName(“a”);
We insert this line after the check to see if getElementsByTagName object method exists
as follows:
function fNewPDFWindows ()
{
if (!document.getElementsByTagName) return false;
var links = document.getElementsByTagName(“a”);
}
```

The next task is to loop through all of the links and check to see if we want to open any of the links in a new window. If the link is to a PDF document then we'll open it in a new window.

Looping through the links

First we create the loop that goes through all the links in the page, as follows:

```

function fNewPDFWindows ()
{
if (!document.getElementsByTagName) return false;
var links = document.getElementsByTagName(“a”);
for (var eleLink=0; eleLink < links.length; eleLink ++) {
}
}
```

The indexOf method

Next we need to determine as we loop through the links whether the link is to a PDF document or not. The indexOf method is ideal for this by returning the index of the search value (the position of the search value in the string). The indexOf method requires a search value but you can also specify where to start the search from within the string (in this example we don't need to pass the method this parameter):
So pass the indexOf method the string ".pdf" to find out if the file being linked to is a PDF document. The indexOf method returns either -1 if the string isn't found or the index of the matching text.

```
function fNewPDFWindows ()
{
if (!document.getElementsByTagName) return false;
var links = document.getElementsByTagName("a");
for (var eleLink=0; eleLink < links.length; eleLink ++) {
if (links[eleLink].href.indexOf('.pdf') !== -1) {
}
}
}
```

onclick function

Next we need to apply an onclick event to each of the links to PDF documents, so when they're clicked the new window is opened:

```
links[eleLink].onclick =
function() {
}
```

First, let's open the new window, and give the window some parameters. The parameters we're going to pass are as follows.

- URL - the document we want to display in the new window
- Specs - a comma separated list of window properties such as scrollbars etc

For a comprehensive list of windows properties take a look W3C schools DOM open method page.

The completed function in this example is as follows. This opens the link address in a new, re sizable window with scrollbars:

```
links[eleLink].onclick =
function() {
window.open(this.href,'resizable,scrollbars');
return false;
}
```

We insert the onclick event within the if statement that detects whether a PDF document has been found:

```
function fNewPDFWindows ()
{
if (!document.getElementsByTagName) return false;
var links = document.getElementsByTagName(“a”);
for (var eleLink=0; eleLink < links.length; eleLink ++) {
if (links[eleLink].href.indexOf(‘.pdf’) !== -1) {
links[eleLink].onclick =
function() {
window.open(this.href,’resizable,scrollbars’);
return false;
}
}
}
}
```

Warning users the document opens in a new window

The final task is to ensure users are aware that the link will open in a new window. We need to be as clear as we can with this to minimise confusion.

We’re going to do this by firstly amending the title text of the link and secondly inserting an image with alternate text of “(opens in a new window)”. The link title can be set as follows:

```
links[eleLink].title += “\n(opens in a new window)”;
```

Next we will create an image element and set its src and alt attributes. Finally we’ll append the image to the hyperlink using the appendChild method.

```
var img = document.createElement(“img”);
img.setAttribute(“src”, “i/new-win-icon.gif”);
img.setAttribute(“alt”, “(opens in a new window)”);
links[eleLink].appendChild(img);
```

The final function is as follows:

```
function fNewPDFWindows ()
{
if (!document.getElementsByTagName) return false;
var links = document.getElementsByTagName(“a”);
for (var eleLink=0; eleLink < links.length; eleLink ++) {
if (links[eleLink].href.indexOf(‘.pdf’) !== -1) {
links[eleLink].onclick =
function() {
window.open(this.href,’resizable,scrollbars’);
return false;
}
var img = document.createElement(“img”);
img.setAttribute(“src”, “i/new-win-icon.gif”);
img.setAttribute(“alt”, “(opens in a new window)”);
links[eleLink].appendChild(img);
}
}
}
```

Conclusion

This very short function demonstrates how easy it is to open links in new windows automatically. This means your site will be more likely to validate, more likely to work correctly in future browsers and will be more usable for its users. Check out this fully functioning example.

This article was written by Paul McCarthy. Paul’s crazy about CSS and using JavaScript for good (not evil) - so crazy that he works for Webcredible helping to make the Internet a better place for everyone. He can often be found carrying out an accessibility audit and spends much of his time working on the world’s most accessible CMS.