

Javascript et PDF avec Scribus

Note au lecteur : Utilisez de préférence Acrobat Reader 5.x+ pour visualiser ce document. D'autres visualisateurs comme GSView ou Xpdf par exemple ne supportent pas les fonctionnalités Javascript du format PDF décrites et utilisées dans ce document.

Une des possibilités les plus puissantes et intéressantes offertes par Scribus est la création de documents PDF javascriptés. Car oui, Javascript et PDF font très bon ménage, et la capacité offerte par Scribus d'ajouter et d'incorporer des objets et des actions en Javascript aux documents PDF fait la différence dans le monde Linux/*nix. Cette capacité ne se retrouve que dans la version complète d'Acrobat Editor, ou encore en éditant à la main le fichier PDF – plutôt risqué, à moins d'être un expert du format PDF.

Cerise sur le gâteau, il n'est même pas nécessaire d'être un expert de la programmation en Javascript pour utiliser ces petits morceaux de code. Une fois compris certaines fonctionnalités du format PDF, les outils proposés par Scribus sont simples d'emploi et disposent même de certains garde-fou vous évitant le pire.

Concept fondamental

Un concept important à comprendre pour bien utiliser le format PDF est qu'Acrobat Reader fonctionne comme un navigateur. On utilise la souris pour naviguer dans le document, des liens hypertexte, des signets, un historique, on peut effectuer des recherches et remplir des formulaires. Vous pouvez aisément ajouter des liens Internet et calculer ou faire valider des champs d'un formulaire. De même, vous pouvez ajouter des effets de présentations ou encore des boutons pour guider la navigation dans votre document. La plupart de ces fonctionnalités sont supportées sur toutes les plate-formes ayant une version 4 ou supérieur d'Acrobat Reader, ce qui représente la quasi-totalité des PC, Mac et stations de travail *nix de la planète.

Il peut être utile d'avoir parcouru la documentation concernant les boutons et boites de dialogues standards, l'utilisation d'un peu de Javascript permettant d'étendre ces fonctionnalités. Techniquement, il y a six différentes façons d'utiliser du Javascript dans un document PDF. Dans les exemples suivants, et de manière générale lorsqu'on utilise Scribus, deux seulement nous concernent : de façon globale, pour affecter le document dans son ensemble, comme le passage en plein écran décrits plus loin ou bien des messages d'informations ou d'alertes interactifs, et plus souvent en accrochant du Javascript à des champs ou des objets.

Javascript pour les débutants

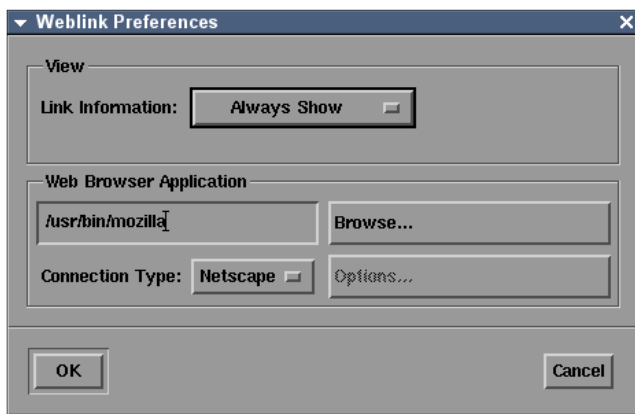
Toute la magie réside généralement dans une ou deux "commandes" prenant la forme : *commande: variable* ou *objet: variable*. La syntaxe est relativement simple, et même ceux qui s'enfuient en courant à l'approche de la moindre ligne de code source devraient s'en sortir avec les honneurs. Vous pouvez également ajouter des commentaires, qui seront ignorés par l'interpréteur Javascript, comme ceci :

```
/* Ceci est un commentaire */  
/* Les commentaire peuvent êtres  
sur plusieurs lignes */
```

Les commentaire sont utiles pour se rappeler pourquoi, six mois plus tôt, on a ajouté tel bout de code. Faites attention à bien encadrer le commentaire comme indiqué.

Pour bien commencer : Configurer Acrobat Reader

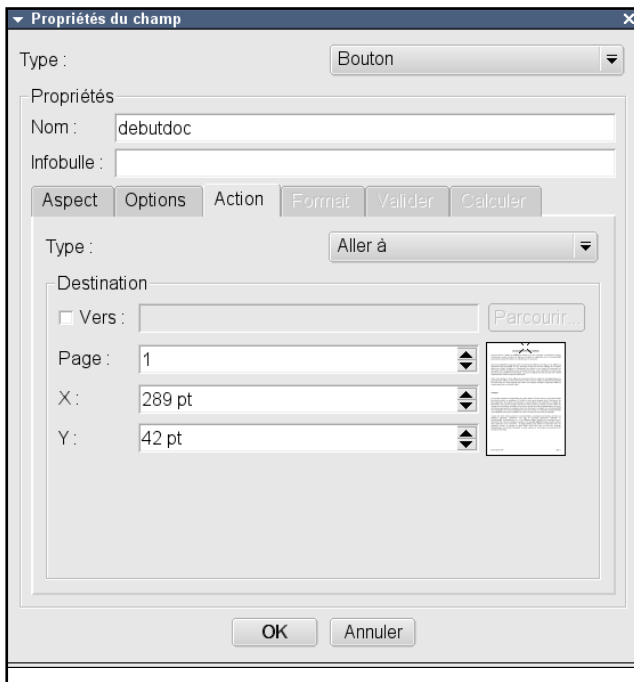
Lancez Acrobat Reader et cliquez sur Edit > Preferences > Weblink. Comme vous pouvez le voir, cette boîte de dialogue sert à configurer le navigateur Web utilisé pour gérer les liens vers internet. Dans le cas présent, j'ai sélectionné Mozilla. Si vous modifiez le choix par défaut, il faut fermer et redémarrer Acrobat Reader pour que le changement soit pris en compte. J'ai testé Opera 7.11 et Konqueror 3 et ils fonctionnent eux-aussi très bien. Sous Windows, Acrobat Reader 5.x utilise par défaut IE. Plus loin, nous verrons des exemples permettant d'ouvrir le navigateur Web sur une page Web précise. En parcourant ce document, passez doucement ou arrêtez votre souris sur les objets car ils ont souvent des bulles d'aide ou des effets visuels - encore des plus que vous pouvez inclure dans vos documents PDF.



Le plugins weblink est fourni avec Acrobat Reader 4.x+. Celui-ci prend en charge les liens vers d'autres parties du document ou vers une page HTML sur un site Web. Dans les exemples suivants, nous allons utiliser quelques commandes en Javascript pour mettre en valeur notre document PDF. Vous aurez même droit à la version Acrobat Reader du célèbre Hello world!

Exemple 1 - Un lien de navigation interne

Commençons par un lien partant d'un texte. Il peut pointer vers une autre partie du document, par exemple un lexique. Pour avoir un lien hypertexte façon "page Web", le plus simple est de créer un bouton sans contour et avec le texte en bleu comme dans la plupart des navigateurs Web. Placez-vous là où vous voulez avoir votre lien, cliquez sur le bouton dans la barre d'outils PDF, dessinez le bouton puis formatez votre texte. Ensuite, cliquez avec le bouton droit, sélectionnez Propriétés du champ > Action et choisissez Aller à dans la liste déroulante comme montré ci-dessous. Choisissez la page et la destination dans cette page puis cliquez sur OK pour sauver.

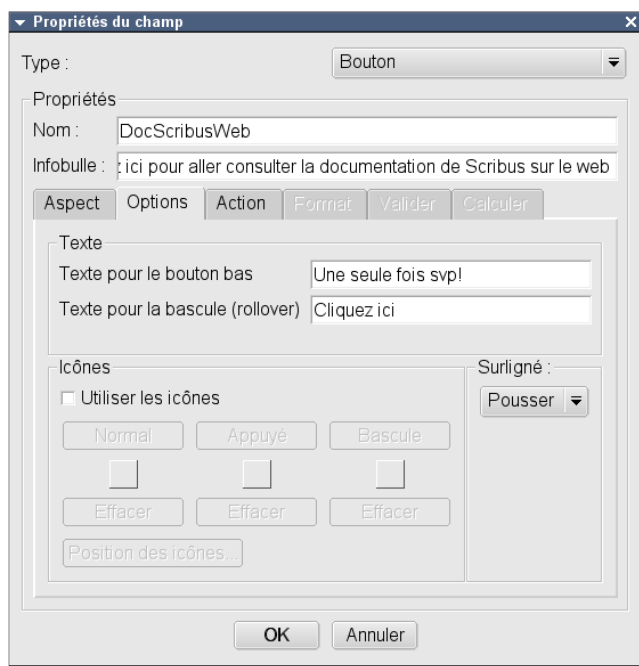


Sélectionnez la page cible puis déplacez la petite croix sur l'aperçu à droite pour positionner précisément la destination. Une fois le document exporté puis ouvert avec Acrobat Reader, le curseur de la souris se changera en une petite main en passant au dessus du lien. Pour accélérer la création de plusieurs liens de ce type, une fois créé le premier, envoyez-le dans l'album pour le réutiliser par la suite en ne modifiant que le nécessaire.

Les coordonnées de destination des liens internes au document PDF sont toujours exprimées en points. Vous pouvez aussi choisir un fichier comme destination en cochant la case Vers.

Exemple 2 - Un bouton pour aller sur une page Web

Cliquez sur le bouton dans la barre d'outils PDF, dessinez-le de la taille voulue puis entrez le texte du bouton. Ensuite, double-cliquez sur le bouton pour éditer les propriétés du champ. Là, vous pouvez décorer le bouton et saisir les différents textes affichés par le bouton suivant les événements. Au vue des qualités artistiques de l'auteur, nous nous contenterons d'ajouter du texte...



Dans la boîte de dialogue des propriétés du champ, vous choisissez toutes les options de votre bouton. Dans cet exemple, j'ai ajouté un texte pour la bascule, une infobulle et un texte différent lorsque le bouton est enfoncé. Avec du Javascript supplémentaire ou un groupe d'icônes, vous pourrez mettre en valeur vos boutons, un peu comme les menus en CSS pour le HTML.

Déplacez doucement votre souris sur le bouton ci-dessous et cliquez en laissant le bouton enfoncé pour voir les différents effets. Si votre plugin Weblink est correctement configuré, il va lancer le navigateur et ouvrir la page Web indiquée lorsque vous relâchez le bouton de la souris.

Comment faire pour lancer le navigateur et lui indiquer l'URL voulue ? Vous vous souvenez lorsque je vous ai dit que tout cela se faisait avec juste quelques petits bouts de code ? C'est aussi simple que cela. Dans l'onglet Action, cliquez sur le bouton Éditer, un petit éditeur de texte va se lancer. Saisissez le texte suivant :

```
getURL("http://atlantictechsolutions.com/scribusdocs/index.html")
```

C'est tout. Vous pouvez aussi créer ce genre de lien Web avec des champ de texte. C'est d'ailleurs parfois plus facile à utiliser lorsque vous créez une liste de lien. Pourquoi ? Parce que le texte des boutons est automatiquement centré dans un document PDF, même si le rendu à l'édition dans Scribus suis un autre alignement. Et donc, il est presque impossible d'obtenir une longue liste de lien correctement alignés, par exemple pour un sommaire en début de document. Et que perd-t-on à utiliser des champs de texte au lieu de boutons ? La principale différence réside dans les indications visuelles pour le lecteur. Les champs de textes cliquables ne changent pas le pointeur de la souris en la petite main familière lorsqu'il passe au dessus d'eux, contrairement aux boutons. C'est subtile, mais c'est à prendre en compte pour des lecteurs peu habitués à Acrobat Reader. Pour pallier à ce problème, vous pouvez utiliser des infobulles pour attirer l'oeil du lecteur.

Exemple 3 - Une présentation en PDF

Cet exemple montre comment lancer une présentation depuis le document PDF avec du Javascript.

Créez un bouton sur la première page de votre document avec "Plein écran" comme label. Ensuite, cliquez dessus avec le bouton droit, sélectionnez Propriétés du champ et choisissez Javascript dans l'onglet Action. Je vous conseille de laisser l'option Bouton souris relâché, pour éviter un cliquage malencontreux. Enfin, cliquez sur Éditer et saisissez ceci :

```
app.fs.defaultTransition = "WipeDown";  
app.fs.isFullScreen = true;
```

Enregistrez et quittez l'éditeur.

Même si vous pouvez faire en sorte que votre document PDF soit vu comme une présentation dans la boîte de dialogue d'export en PDF de Scribus, cela vous permet de lancer la présentation de façon explicite depuis une page de titre. Vous pouvez aussi cacher le bouton en assignant au texte la même couleur que l'arrière plan.

Exemple 4 - Hello World avec Acrobat

Cet exemple montre comment ajouter une boîte d'information ou d'alerte.

Une utilisation possible serait par exemple d'utiliser une fenêtre pop-up pour donner une information à l'utilisateur final lorsqu'il valide une donnée dans un champ. Cela pourrait avoir lieu au moment où la souris entre dans le champ pour donner des indications sur la nature de celui-ci ou pour demander au lecteur de bien vérifier son contenu. Pour notre part, nous allons faire la démonstration avec un bouton. Créez un bouton comme précédemment, puis, dans la boîte Propriétés du champ, sélectionnez Javascript dans l'onglet Action et cliquez sur le bouton Éditer. Ajoutez ce qui suit en une seule ligne :

```
app.alert({ cMsg: "Hello World!", cTitle: "l'exemple du Hello World" });
```

Maintenant, améliorons un peu notre boîte en permettant à l'utilisateur de donner une réponse qui pourra être utilisée pour remplir un champ ou valider une donnée.

```
app.alert({ cMsg: "Avez-vous appris quelque chose aujourd'hui ?",  
cTitle: "Question!", nIcon: 2, nType: 3});
```

Dans cet exemple, vous pourriez utiliser la réponse donnée dans un test. Le bouton Cancel ne retourne rien.

D'autres possibilités du PDF avec Scribus

Articles

Les articles sont une autre façon de naviguer à travers un document. Ils sont particulièrement utiles lorsque vous avez du texte sur plusieurs colonnes ou dans plusieurs zones chaînées. Par exemple, il est courant de découper un article de journal en mettant un ou deux paragraphes en première page et en renvoyant le reste du texte plusieurs pages plus loin. Dans Acrobat Reader, sélectionnez Window > Articles et une palette avec les articles liés va apparaître. En double-cliquant sur l'article, vous serez transporté sur le début de l'article avec un zoom. L'utilisation du Javascript n'est pas nécessaire, il suffit de cocher la case adéquate dans la boîte de dialogue d'export vers le PDF.

Signets et vignettes

Acrobat Reader a aussi un onglet de signets, tout comme les signets d'un navigateur Web. Cependant, les signets sont spécifiques au document et doivent être créés par le logiciel générant le document PDF ou ajoutés depuis Acrobat Reader. Avec Scribus, la création de signets est directe. Cliquez tout simplement avec le bouton droit sur n'importe-quel cadre de texte et sélectionnez Signet PDF. Les signets facilitent la navigation dans le document, particulièrement dans les plus longs, en vous permettant de diriger le lecteur directement vers les titres des sections. Dans ce document PDF, tous les titres de section sont des signets. Pour le voir, cliquez simplement sur l'onglet bookmarks à l'extrême gauche de la fenêtre d'Acrobat Reader. Des vignettes de chaque page peuvent aussi être utilisées pour la navigation.

FAQ Javascript et PDF

Comment créer un lien vers une page précise d'un document ?

```
getURL("http://atlantictechsolutions.com/scribusdocs/pdfs/javascript101.pdf#page=5")
```

ouvrira la page 5 du document.

Comment savoir quelles versions d'Acrobat Reader supportent certaines options et fonctions du Javascript PDF ?

Le document de référence du javascript PDF donne clairement les options supportées par les applications en fonction de leur version. Certaines fonctions Javascript ne sont présentes que dans la version complète d'Acrobat. Vous ne manquez rien, la plupart de ces fonctions concernent la création automatique de fichiers et de champs dans les documents PDF dynamiques.

Verrons-nous la version 6.0 d'Acrobat Reader pour Linux et les autres unices ?

Adobe est très réticent concernant l'annonce de nouvelles versions. Historiquement, la version pour Linux et autres unices sort après qu'ils aient ajouté des langages additionnels pour les versions Windows et Mac.

Sur quelle version de Javascript est basé l'implémentation d'Acrobat Reader pour le PDF ?

L'implémentation d'Acrobat Reader est basé sur le Javascript version 1.5 (ISO-16262), aussi connue sous le nom d'ECMAScript.

Conclusion

Scribus facilite la création de formulaires interactifs, de présentations et de bien d'autre choses encore au format PDF, avec une approche simple pour l'utilisateur. Je vous encourage à aller lire le document de référence du javascript pour Acrobat Reader. Vous y trouverez de nombreux exemples, des morceaux de code et des explications détaillées concernant ces fonctionnalités. J'espère que ce tutoriel vous amènera à explorer plus avant ce domaine.

Références pour aller voir plus loin

1.8 Mb - C'est le document de référence du Javascript pour le PDF.

10.0 Mb - 3^{ième} version du document de référence du format de fichier PDF. 978 pages.

Point d'entrée de tout ce qui concerne le Javascript sur le site pour les développeurs de Netscape.

Ce document a été réalisé à 100% avec Scribus 1.0
Première diffusion le 01:07:2003 pour Scribus 1.0 (en anglais)

Copyright (c) 2003 Peter Linnell and Franz Schmid. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, v1.0 or later with elected option number 1 (the latest version is presently available at <http://www.opencontent.org/openpub/>).

Traduction : Yann Cantin.