

SphinxOnline

Manuel d'utilisation

Module " Programmation HTML "

Le Sphinx Developpement - Parc Altais - 74650 CHAVANOD France - Tel : +33 (0)4 50 69 82 98 -
Fax : +33 (0)4 50 69 82 78 - www.lesphinx-developpement.fr - contact@lesphinx-developpement.fr

Ce document décrit comment enrichir vos formulaires, tableaux de bord, etc... à l'aide d'éléments de programmation. Les formulaires, les tables d'observations, les tableaux de bord d'une enquête sont affichés par le serveur d'enquêtes sous forme de pages HTML générées dynamiquement.

Il est possible de rajouter des éléments de programmation (html, javascript, css) pour enrichir ces pages :

- Enrichir la typographie une légende
- Insertion d'une vidéo d'un site de partage (YouTube, Dailymotion, ...)
- Insérer un contenu (page, site, ...) externe dans la page html
- Insérer des résultats (table de données ou tableaux/graphique) dans un formulaire
- Saisie d'un code à partir de la liste de codes associée
- Saisie d'une adresse à partir d'une carte Google
- Insertion d'une page d'upload de documents
- Applications de saisie interactives en Flash
- Créer un bouton permettant de recopier une observation

Ces éléments de programmation peuvent faire référence à des images ou à des fichiers de données qu'il est donc nécessaire d'uploader.

Trois possibilités :

- Depuis le gestionnaire du compte, utiliser le bouton Compléments pour uploader un fichier
- Dans un formulaire de travail mettre une légende xxx en y associant l'action "Montrer une page". On indiquera le nom de l'image ou du fichier. Tous ces fichiers seront alors publiés en même temps que l'enquête
- Il est également possible de faire référence directement à des images disponibles sur un autre serveur en donnant leur url absolue.

Une enquête exemple présentant toutes ces possibilités est disponible à la consultation et au téléchargement. Pour y accéder, cliquez ici

Enrichir une légende

Enrichir la typographie

Il est possible d'indiquer des enrichissements typographiques pour mettre une partie du texte en gras, en couleur ou en taille différente.

Il n'y a pas (dans la version courante de Sphinx) d'éditeur intégré et il faut écrire les "balises". La syntaxe est la suivante :

```
<red>le texte est en rouge</red>
<blue>le texte est en bleu</blue>
<green><b>le texte est en vert et en gras</b></green>
<i>le texte est en italique</i>
<u>le texte est souligné</u>
<font color="#A000A0">le texte est dans une couleur personnalisée</font>
<font size="+3">la taille du texte est supérieure de 3 points </font>
<font size="-3">la taille du texte est inférieure de 3 points</font>
<u><gray><font size="18">le texte est en gris, souligné, taille 18</font></gray></u>
```

Les mots clés des couleurs sont : **red, blue, green, yellow, magenta, violet, white, pink, orange** Les balises html interprétées sont : **u,i,b,font** (attributs **color, size**)

Insérer des liens hypertexte

Il est possible d'insérer un lien hypertexte (ou un mailto) dans une légende d'un formulaire ou d'un tableau de bord mais uniquement lorsque celui-ci est proposé en html (et non pas sous forme d'image).

Pour cela on indique que la légende est une légende "html" en encadrant le texte par la balise ****. Ensuite on indique le lien hypertexte avec la syntaxe html.

Exemples :

```
<span><a href="http://www.domain.ext/" target="_blank">LIEN</a></span>
<span><a href="mailto:toto@domain.ext">Ecrire un mail à M. Toto</a></span>
```

Insertion d'une vidéo d'un site de partage (YouTube, Dailymotion, ...)

Sur les sites de partage (Daily Motion, You Tube) il est possible d'exporter le code html à insérer dans une page html pour jouer la vidéo. Ce peut être dans un formulaire ou dans un tableau de bord.

Le code ci-dessous (copié depuis Daily motion) est à copier tel quel dans une légende.

```
<div><object width="420" height="339"><param name="movie" value="http://www.dailymotion.com/swf/x9tjb7_le-loup-
reapprenons-a-le-connaître_animals" /><param name="allowFullScreen" value="true" /><param name="allowScriptAccess"
value="always" /><embed src="http://www.dailymotion.com/swf/x9tjb7_le-loup-reapprenons-a-le-connaître_animals" type="application/
x-shockwave-flash" width="420" height="339" allowFullScreen="true" allowScriptAccess="always"></embed></object><br /><b><a
href="http://www.dailymotion.com/swf/x9tjb7_le-loup-reapprenons-a-le-connaître_animals">le loup réapprenons à le connaître</a></
b><br /><i>par <a href="http://www.dailymotion.com/jymets">jymets</a></i></div>
```

Insérer un contenu (page, site, ...) externe dans la page html

Ces contenus externes sont insérés sous forme de code HTML dans une légende. Attention à bien commencer et terminer le texte par une balise, par exemple `<span>`.

La fonction n'est pas wysiwyg et dans le formulaire sous Windows, on ne voit que le code html.

Ce n'est qu'une fois l'enquête publiée que le code html sera correctement interprété par le navigateur.

Incrusté dans la page

```
<script>openUrl("http://www.....",500,200)</script>
```

- http://www... : l'url du fichier à afficher
- 500,200 : la largeur et la hauteur de la zone d'affichage

Fenêtre "popup"

```
<script>openPopup("cliquez ici", "http://www.....",500,200)</script>
```

- cliquez ici : le texte du lien
- http://www... : l'url du fichier à afficher
- 500,200 : la largeur et la hauteur de la fenêtre pop-up

Insérer des résultats dans un formulaire

Insertion d'une table de données

Dans un formulaire ou un tableau de bord (en html), on peut insérer une table des données de l'enquête. On utilise la fonction précédente `openUrl` ou `openPopup` avec comme url une requête (ou un raccourci), par exemple

```
<script>openUrl("tables.htm",500,200)</script>
```

ou

```
<script>openUrl('tables.htm?sample=csp=[csp.nombre]',500,200,1)</script>
```

montrera les réponses des observations ayant la même csp que l'observation courante.

ou

```
<script>openUrl('enquete.hyp?query=table&table=1&frames=0&variables=1;5&sample=csp=[csp.nombre]',500,200)</script>
```

montrera les réponses des observations ayant la même csp que l'observation courante.

Saisie d'un code à partir de la liste de codes associée

Permet de saisir une variable ouverte code (ou une variable texte en mode avancé) en sélectionnant la valeur dans un fichier au format texte (il n'y a pas de limite sur la taille de ce fichier de valeurs).

Le fichier au format texte doit être uploadé sur le serveur comme indiqué ci-dessus.

La fonction permet aussi de remplir plusieurs champs en une seule fois si ces champs sont présents dans le fichier (le séparateur est alors tabulation ou point-virgule). Le nom de ces champs doit correspondre aux noms des variables que vous souhaitez renseigner à l'aide de cette liste de codes.

Si l'on tape les premiers caractères dans le champ de la variable, la liste se limite automatiquement aux résultats disponibles.

Liste automatique

A partir du moment où vous associez une liste de code à une **variable de type code**, que votre fichier de liste de code est présent sur le serveur et que votre variable est présente sur votre formulaire web, la liste sera affichée automatiquement une fois votre enquête en ligne.

De même, si votre fichier comporte plusieurs colonnes et que des variables codes portant le même nom que ces colonnes sont affichées sur la même page du formulaire, celles-ci seront automatiquement remplies avec les valeurs correspondantes lorsque vous sélectionnerez un élément de la liste.

Liste avancée

Les listes de code peuvent être définies "à la main" dans une légende "html". Cela permet d'avoir une liste de valeurs aussi pour une **variable de type texte** et/ou de choisir des paramètres avancés d'affichage de la liste.

Exemple :

```
<script>openCodesListEx({varName: "v19", codeFile: "sphinxpeople.txt", nbcols: 2, filter: true, forcefilter: true, blurmode: "selectandhide", searchmode: "startwith", showicons: true})</script>
```

Paramètres obligatoires :

- varName : le nom de la variable (sous la forme Vxx, où xx est le numéro de la variable) qui déclenchera l'affichage de la liste de code lors d'un clic dans son champ éditable
- codeFile : le fichier contenant la liste (Attention, le fichier doit être enregistré avec le codage utf-8)

Paramètres optionnels :

- nbcols : le nombre de colonnes à afficher (pertinent uniquement si le fichier comporte plusieurs colonnes)
- filter : affiche les filtres par défaut
- forcefilter : le filtrage est forcé (la liste est vide si l'on ne sélectionne pas de valeur dans la liste déroulante de filtre)
- blurmode : détermine le mode d'affichage ("selectandhide" est le mode par défaut, il cache la liste en cas de sélection d'un élément ou bien d'un clic à l'extérieur de la liste; "hide" ne cache la liste qu'en cas de clic à l'extérieur de la liste; "none" ne cache la liste qu'en cas de clic sur l'icone de fermeture de la liste)
- searchmode : détermine le mode de recherche ("startwith" est le mode par défaut, il permet d'effectuer une recherche en tapant au clavier les premières lettres de l'élément recherché; "contains" permet d'effectuer une recherche en tapant une partie de l'élément recherché)
- showicons : affiche l'icone des filtres

Saisie d'une adresse à partir d'une carte Google

```
<script>openGMap("V1","V2","v3","Grenoble",500,500,13,"V4",1)</script>
```

- V1,V2,V3 : les noms (sous la forme Vxx) des variables qui contiendront respectivement l'adresse, le code postal et la ville
- Grenoble : une adresse, une ville ou un pays sur lequel la carte sera centrée par défaut
- 500,500 : la largeur et la hauteur de la carte
- 13 : le niveau de zoom affiché par défaut (entre 1 et 16)
- V4 : optionnellement une variable qui contiendra les coordonnées du point choisi
- 1 : 1 ou 0 pour afficher au démarrage la carte (1) ou un bouton "ouvrir" (0)

Upload de documents

Insertion dans le formulaire

Il est possible, dans un formulaire de proposer l'upload d'un ou plusieurs documents. On insère la légende "html" suivante :

```
<script>showUpload("[identificateur]",600,100,"name=photo&ext=jpg;png;gif&mode=mini")</script>
```

- [identificateur] : ce premier paramètre définit le nom du répertoire dans lequel seront enregistrés les fichiers de cette observation, vous pouvez utiliser n'importe quelle variable de l'enquête en la notant entre crochets : [nomvariable] (attention : bien choisir une variable dont les valeurs seront uniques...)
- 600,100 : la largeur et la hauteur de l'interface d'upload (en mode normal : largeur min. = 600 et hauteur min. = 370, en mode mini : largeur min. = 550 et hauteur min. = 75)
- name=photo (le fichier prendra le nom indiqué) ou prefix=cv- (le fichier commencera par cv-)
- ext=jpg;png;gif les extensions de fichiers autorisées
- mode=mini : (optionnel) ce paramètre permet d'utiliser un mode d'affichage minimal
- nodel=1 : (optionnel) ce paramètre retire la possibilité de supprimer les fichiers uploadés

Consultation par le répondant

Pour afficher au répondant les fichiers qu'il a uploadés, on insère la légende "html" suivante :

```
<script>showUpload("[identificateur]",700,300,"consult=1")</script>
```

- [identificateur] : ce paramètre correspond à celui qui a été utilisé pour l'upload des fichiers que l'on souhaite consulter
- 700,300 : la largeur et la hauteur de l'interface d'upload (largeur min. = 600 et hauteur min. = 270)
- consult=1 : affiche la consultation des fichiers uploadés

Consultation par le chargé d'étude

Vous pouvez consulter les fichiers uploadés lorsque vous accédez à l'enquête depuis votre compte soit via le bouton "Docs uploadés", soit via le lien "uploads.htm" dans la partie "Résultats".

Remarque : si l'accès aux données est protégé par un mot de passe, la page "uploads.htm" ne fonctionne pas directement, pour qu'elle fonctionne il faut ajouter un paramètre dans l'url comme ceci : "uploads.htm?pwd=XXX" (où XXX est le mot de passe).

Gestion du temps de saisie et de la progression

Limiter le temps total de saisie

Vous pouvez insérer dans votre formulaire un timer permettant d'indiquer le temps de saisie et / ou de limiter ce temps de saisie (fonctionnalité intéressante pour un questionnaire de type Quiz). Pour ajouter un tel timer dans le formulaire, cocher la case "Indiquer le temps écoulé" dans les propriétés du formulaire, onglet "Internet". Le timer apparaîtra à côté de la barre de commande de ce questionnaire.

Pour limiter le temps de saisie, cocher la case "Limiter le temps de saisie" et indiquer le temps limite en secondes. Lors de l'accès en saisie au questionnaire, au terme de la valeur limite du timer, le questionnaire est sauvegardé. Le répondant est alors redirigé vers la page de fin de saisie ou vers une autre URL que l'on définit au préalable.

Gérer le temps de saisie pour une page spécifique du formulaire

Il est possible de limiter le temps de saisie d'un écran particulier, à l'aide d'une légende "html" dont la syntaxe est :

```
<script>submitTimerPage({'mintime': 5, "maxtime": 20, "timeShow": false, "position": "navbar"})</script>
```

Chacun des paramètres est optionnel. Voici à quoi il correspond :

"mintime": 5

Le boutons de navigation ne s'affichent qu'après 5 secondes.

"maxtime": 20

Le temps maximum d'affichage de la page est de 20 secondes, passé ce délai on passe automatiquement à la page suivante.

"timeShow": false

Option qui définit si le temps doit être affiché ou non (par défaut cette option est à **true**).

"position": "navbar"

Option qui définit où doit être affiché le temps : "navbar" affiche le temps au niveau de la barre de navigation, "inline" (valeur par défaut) l'affiche là où est placé la légende html.

Afficher un élément au bout de X secondes

Vous avez la possibilité de n'afficher certains éléments d'une page qu'au bout d'un certain temps. Pour cela il faut utiliser :

- Une variable spécifique (que nous nommerons ici vX) qui contiendra le temps écoulé depuis le chargement de la page
- Une légende html de type `<script>setTimerVariable("vX");</script>` qui permettra de cacher la variable vX et d'incrémenter sa valeur à chaque seconde écoulée
- Une condition d'affichage de type `vX >= n` (où n est le nombre de secondes) sur chacun des éléments à cacher

Il faudra que la variable **vX** soit présente dans le formulaire sur la page concernée. Elle apparaîtra dans l'édition du formulaire mais sera cachée dynamiquement par le script **setTimerVariable** en ligne.

Afficher un chronomètre

Pour afficher un chronomètre, on utilise également une légende "html".

Avec remise à zéro à chaque changement de page :

```
<script>submitTimerPage({'chrono': true})</script>
```

Sans remise à zéro (durée totale depuis le début de la saisie) :

```
<script>submitTimerPage({'chrono': true, 'totaltime': true})</script>
```

Afficher une barre de progression spécifique

Pour afficher une barre de progression informant le répondant de l'avancement dans le questionnaire, on utilise une légende "html" avec la syntaxe :

```
<script>displayProgressBar();</script>
```

Pour que cette barre de progression apparaisse dans toutes les pages, on la mettra dans le masque.

On peut changer la couleur (BLUE, PURPLE, GREEN et GREY) et la forme (CB01, CB02, CB03, CB04 et CB05) de la barre de progression avec la syntaxe :

```
<script>displayProgressBar("BLUE_CB01");</script>
```

Applications de saisie interactives en Flash

Insertion d'une application flash "tourne-page"

```
<script>openPageFlip('v1','flip.xml',250,240)</script>
```

- V1 : le nom (sous la forme Vxx) de la variable texte qui contiendra les numéros des images choisies
- flip.xml : le fichier qui contient la description des images(ou animations) qui apparaîtront sur le tourne-pages
- 250 : la largeur du tourne-page
- 240 : la hauteur du tourne-page qui doit correspondre à la hauteur de la plus grande image (ou animation)

Syntaxe du fichier xml

```
<content hcover="false" transparency="true">  
<page src="pf1.jpg"/>  
<page src="pf2.jpg"/>  
<page src="pf3.jpg"/>  
</content>
```

Insertion d'une application flash "Étagère défilante"

```
<script>openBookShelf('v22','shelf.xml',3,610,260)</script>
```

- V22 : le nom (sous la forme Vxx) de la variable texte qui contiendra les numéros des images choisies
- shelf.xml : le fichier qui contient la description des images qui seront placées sur l'étagère
- 3 : le nombre maximum d'éléments que l'on peut sélectionner
- 610,360 : la largeur et la hauteur de l'application flash

Syntaxe du fichier xml

```
<content nbmax="3" coche="coche.png">  
<page src="img1.jpg"/>  
<page src="img2.jpg"/>  
<page src="img3.jpg"/>  
<page src="img4.jpg"/>  
</content>
```

Insertion d'une application flash "Clics image"

```
<script>openImgClick("V23","clic.xml",554,400)</script>
```

- V23 : le nom (sous la forme Vxx) de la variable texte qui contiendra les coordonnées (x/y) des points cliqués
- clic.xml : le fichier décrit les éléments de l'application (fond, logo, nombre maximum de clic possible)
- 554,400 : la largeur et la hauteur de l'application flash

Syntaxe du fichier xml

```
<content nbmax="10" >
<background src="fond.jpg"/>
<logo src="coche.png"/>
</content>
```

Insertion d'une application flash "Zones sensibles"

```
<script>openShapes("V24","shapes.xml",554,400)</script>
```

- V24 : le nom (sous la forme Vxx) de la variable texte qui contiendra les numéros des zones cliquées
- shapes.xml: dans ce fichier xml est indiqué l'image de fond et les polygones cliquables (en spécifiant pour chacun une couleur)
- 554,400 : la largeur et la hauteur de l'application flash

Syntaxe du fichier xml

```
<content bgimage="test.jpg" nbmax="3">
<shape color="00AAFF" borderColor="00FF00" name="test1" alpha="0.2">10,10,50,10,50,50,10,50</shape>
<shape color="00AAFF" borderColor="00FF00" name="test2" selectIni="false" alpha="0.2">100,100,100,200,200,200,100</
shape>
<shape color="00AAFF" borderColor="00FF00" name="test3" selectIni="false" alpha="0.2">300,300,250,350,300,350</shape>
.....
</content>
```

Construire un polygone

Il est difficile de construire ces polygones directement en éditant le fichier xml.

Sphinx propose un petit outil permettant de dessiner ces formes puis de copier/coller le xml correspondant : http://www.ergole.fr/_templates/drawshapes.htm

Insertion d'une application flash "Drag 'n Drop"

Cette application permet de faire réaliser des associations, par exemple dans le cadre de quiz.

```
<script>openDragNDrop("v25",'dragndrop.xml',610,360)</script>
```

- V25 : le nom (sous la forme Vxx) de la variable texte qui contiendra le résultat du drag 'n drop.
- dragndrop.xml : le fichier qui contient la description de l'application : les éléments à disposer (image et/ou texte), l'image de fond et les coordonnées des "cibles".
- 610,360 : la largeur et la hauteur de l'application flash

Syntaxe du fichier xml

```
<content img="fondpubs.jpg">
<token src="j.jpg" name="Johnny" label="Johnny Halliday"/>
<token src="l.jpg" name="Laporte" label="Bernard Laporte"/>
<token src="z.jpg" name="Zidane" label="Zinédine Zidane"/>
<token src="c.jpg" name="Clooney" label="George Clooney"/>
<target x="617" y="273" name="Optic2000"/>
<target x="339" y="54" name="Madrangé"/>
<target x="711" y="47" name="Danone"/>
<target x="119" y="239" name="Nespresso"/>
</content>
```

Affecter des variables Sphinx depuis une fonction JavaScript

Pour identifier une variable Sphinx dans un formulaire en JavaScript, il suffit de connaître le numéro de la variable, et de récupérer sa valeur en utilisant la fonction **getInput(champ)**, où "champ" vaut "v23" par exemple si la variable interrogée est la 23.

Si la variable n'existe pas au moment de l'appel de cette fonction, celle-ci retourne -1.

Pour affecter une valeur à une variable Sphinx, vous pouvez utiliser la fonction **setInput(champ, value)**.

Exemple d'affectation de variable :

Prenons une variable Sphinx (v1) devant contenir le Referrer, c'est à dire l'URL à partir de laquelle a été appelé le formulaire Sphinx.

Pour affecter cette variable en JavaScript, il vous suffira d'insérer une légende dans votre formulaire contenant le texte suivant : **<script>setInput("v1",document.referrer);</script>**.

Cette balise script sera directement interprétée par le serveur d'enquête, et affectera donc la variable "v1".

Utilisation des cookies

Vous avez la possibilité dans un formulaire Sphinx de manipuler des cookies. Pour cela, deux fonctions principales sont à votre disposition. Etudions l'utilisation de ces fonctions dans le cadre d'un exemple.

Ex: Nous voulons écrire un cookie contenant la date de dernière saisie afin de le relire ultérieurement lorsque le client accédera à un nouveau questionnaire Sphinx.

Dans le premier questionnaire, nous créons un nouveau formulaire de retour après enregistrement de l'observation.

Dans ce formulaire, nous écrivons le code suivant dans une légende : **<script>setCookie ("dernieresaisie", getInput("v25"));</script>** où "dernieresaisie" est le nom donné au cookie et "v25" correspond à la variable contenant la date de saisie.

La fonction setCookie s'écrit donc sous la forme **setCookie (name, value)**.

Dans un nouveau questionnaire, nous voulons récupérer la date de dernière saisie du répondant en vérifiant et en stockant la valeur du cookie "dernieresaisie" (s'il existe) dans une variable spécifique de l'enquête.

Pour cela, dans le formulaire de saisie de ce nouveau questionnaire, nous écrivons le code suivant dans une légende : **<script>setInput("v1", getCookie("dernieresaisie"));</script>** où "dernieresaisie" est le nom du cookie que l'on recherche.

Si le cookie existe, alors ce code affectera la variable "v1" de la date de dernière saisie.

La fonction getCookie prend un paramètre, qui est le nom d'un cookie recherché, et s'écrit sous la forme **getCookie (name)**.

Faire appel à un script externe

Il est possible, au cours de la saisie, de faire appel à une application externe afin de modifier dynamiquement des variables.

Sur un bouton d'action (aller à la page, page suivante, ...), on peut déclencher un "appel à un script externe". Ce script recevra les valeurs présentes dans le formulaire et renverra des nouvelles valeurs. La syntaxe est la suivante :

L'url est de la forme : `http://appli/script.aspx?var1=[var1]&var2=[var2]`

Les portions entre crochets seront remplacées par les valeurs de l'observation avant l'appel au script.

Le retour doit être de la forme : **OK &var3=XXX**.

La variable var3 sera donc mise à jour dans le formulaire.

Si l'application souhaite retourner un message d'erreur ou d'avertissement, le retour devra être le message qui sera affiché par le serveur d'enquêtes dans une alerte.

Recopier une observation

Il est possible de reprendre des valeurs depuis une autre observation en faisant appel à un script externe particulier.

On indique la variable identifiant les observations et les variables à reprendre.

Si aucune observation ne correspond un message d'avertissement est affiché.

La syntaxe de ce script externe est la suivante :

`http://www.sphinxonline.com/comptes/retrievedata.ashx?`

`&a=compte&s=enquete&login=xxx&var=identifiant&val=[identifiant]&retvars=2;3;8-20;25&ind=0&lang=[enquete.langue]`

- a=compte : nom du compte
- s=enquete : nom de l'enquête (ou plusieurs enquêtes séparées par des points-virgules)
- login=xxx : mot de passe d'accès aux données
- var=identifiant : nom de la variable identifiant les observations
- val=[identifiant] : valeur de la variable identifiant les observations
- retvars=2;3;8-20;25 liste des numéros des variables à reprendre avec la syntaxe 2;3;8-20;25
- ind=0 (optionnel) si plusieurs observations correspondent on retiendra la dernière (0) ou la première (1)
- lang=[enquete.langue] (optionnel) si l'enquête est multilangue