

SOMMAIRE

I - Installation	<i>page 2</i>
II - Description	<i>page 2</i>
III - Prise en main, enquête sur le référendum	<i>page 3</i>
IV - Autre exemple : système de réservations pour un spectacle	<i>page 8</i>
V - BONUS = un système de formulaires chainés	<i>page 12</i>
VI - Description des fonctionnalités :	<i>page 14</i>
VII - Formulaires multi-niveaux	<i>page 17</i>
VIII - Insertions ou suppressions multiples	<i>page 18</i>
IX - Formulaires pré-remplis	<i>page 19</i>
X - Formulaires interactifs	<i>page 22</i>
XI - Formulaires avec champ identifiant	<i>page 23</i>
XII - Variables et fonctions de contrôle	<i>page 23</i>
XIII - Exemples de scripts php	<i>page 27</i>
XIV - importer et exporter des formulaires via la base de données	<i>page 29</i>

Chrono-form est un module de chrono-site : <http://chrono.apinc.org>

Attention, pour réaliser l'installation il vous faut un hébergement et une base de données ! Vous pouvez aller dans la section hébergeurs du site <http://chrono.apinc.org> pour plus de renseignements.

I* Installation

Rien de plus simple :

* décompresser l'archive que vous venez de télécharger dans un dossier (*formulaire* par exemple).

* Complétez le fichier *variables_formulaire.php* Quelques remarques sur ce fichier :

1* Vous devez renseigner avec précision les *variables de connexion* (*\$hote* etc.)

2* La variable *\$prefixe* indique le préfixe utilisé pour les tables générées par **chrono-form**. Il n'est pas souhaitable de la changer pour raisons de compatibilité.

3* Les deux variables *identifiant* et *motdepasse* sont à changer **impérativement**, elles contiennent vos codes de connexion et doivent vous être propres !

* Expédiez alors l'ensemble de votre dossier *formulaire* sur le serveur et c'est tout ! Votre **chrono-formulaire** est prêt à fonctionner !

II* Description

Chrono-form est un module de **chrono-site** qui permet de créer des formulaires et leurs tables associées. Les fichiers sont les suivants :

* *prive_formulaire.php* qui permet la création et modifications des formulaires.

* *formulaire.php* pour faire fonctionner le formulaire.

* *listing.php* pour afficher les résultats des enregistrements faits dans la table [_1]

* *options_listing.php* pour régler les colonnes d'affichage

* *simulation.php* pour avoir un aperçu du formulaire que vous réalisez

* *faire_tables.php* qui construit les deux tables nécessaire au fonctionnement du formulaire

Les tables créées ont toujours la forme suivantes :

[prefixe][nom de table][_1]

[prefixe][nom de table][_2]

le préfixe est celui défini dans le fichier *variables_formulaire.php*, il est souhaitable de ne pas y toucher pour garder la compatibilité avec les fichiers *sql* que vous pourriez importer (ou exporter) comme exemples.

Le *nom de table* est celui que vous avez défini dans le module de création *prive_formulaire.php* (sans espaces SVP).

La table [_1] contient les enregistrements faits par les utilisateurs du formulaire.

La table [_2] contient tous les champs et les attributs de chacun d'eux. Cette table peut être modifiée une fois le formulaire déjà en action !

III * Prise en main, enquête sur le référendum

Nous allons créer un enquête sur le *referendum européen du 29 mai 2005*.

- * Nous proposons un *avis* de vote ou d'abstention.
- * Un *lien* permettra de se renseigner sur la constitution européenne.
- * Un *état* des votes obtenus sera donné.
- * Un *message* remerciera d'avoir voté !

Méthode.

- * Rendez-vous dans votre dossier formulaire sur votre site (quelque chose comme ceci :

<http://jlepine.free.fr/formulaire/>)

- * *Identifiez vous* en tapant les deux ccodes que vous avez changés dans le fichier *variables_formulaire.php*

- * Vous arrivez alors sur l'interface de création du formulaire.

- * Les étapes

1* Tapez le *nom de table* (sans espace SVP et en minuscules si possible) **referendum** puis le titre que vous désirez donner à votre enquête : *Votre avis sur le référendum du 28 mai 2005* par exemple

Création d'un formulaire chrono-site

Taper le nom d'une nouvelle table dans la base de données (sans espace svp) referendum

Ou la sélectionner Nouveau... Détruire la table Vider la table

Titre du formulaire Votre avis sur le référendum du 29 mai 2005 Fond titre #707060

Occupation (en %) 80 Décalage 10 Fond page #FFFFFF Bordure 1

Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer + -

Nous allons commencer par poser la question.

2* Cliquez sur le mot **commentaire** dans la ligne rouge (voir ci-dessus). Taper alors la question *Qu'avez-vous l'intention de faire lors du référendum du 29 Mai ? ...* On règle l'occupation à 60% et la hauteur à 40 pixels. La question occupera donc 60% de l'écran. Les 40% restants seront occupés par la proposition de vote ci-dessous.

Nous allons maintenant Créer le champ à choix multiples, *oui non blanc abstention*

Titre du formulaire Le référendum du 29 mai 2005 Fond titre #707060

Occupation (en %) 80 Décalage 10 Fond page #F0F0E0 Bordure 1

champ1 Qu'avez-vous l'intention de faire lors du référendum du 29 Mai ? Centré (1=oui) 1

Commentaire Occupation (en %) 60 hauteur 40 Numérique Fond #F0F0E0 Bordure 0

Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer + -

3* Cliquez sur le lien **choix multiples** dans la barre rouge. Tapez les trois choix dans la ligne de saisie séparés par des points virgules : `&oui;non;blanc;&abstention` (le **&** devant *oui* force ce champ à être sélectionné par défaut). Saisissez une *occupation* de 40% et une *hauteur* de 40. Vous pouvez alors cliquer sur **aperçu** dans la barre noire pour admirer le début de votre référendum.

4* Nous allons ajouter un *lien* vers le site gouvernemental de la constitution pour que les gens puissent aller se renseigner sur les textes de la constitution (avec un peu de courage c'est toujours possible). Ce lien doit être mis dans un code php car dans un code *commentaire* en html pur il ne fonctionnerait pas (il serait utilisé en relatif). Nous allons donc le mettre dans un **code php**. Cliquez sur le lien **php** de la barre rouge et tapez le code

```
echo '<a href="http://www.ladocfrancaise.gouv.fr/dossier_actualite/constitution_europeenne/" target="_blank">Documents sur la constitution Européenne</a>';
```

Ce qui donne

champ2 ex: bleu;&rouge;vert	oui;non;blanc;&abstention		0 = en colonne		1
<input type="checkbox"/> Choix multiple	Occupation (en %)	40	hauteur	40	Numérique <input type="checkbox"/> Fond #F0F0E0 Bordure 0
champ3			Centré (1=oui)	1	Ratio (%) 0
Script PHP	<pre>echo ' Documents sur la constitution Européenne';</pre>				
<input type="checkbox"/> Code PHP	Occupation (en %)	100	hauteur	60	Fond #F0F080 Bordure 0

Remarquons que la *hauteur* a été mise à 60 et la couleur a été changée. De plus le *ratio* a été mis à zéro pour donner 100% au code php et 0% au titre (puisque'il n'y a pas de titre !).

Cliquez sur **aperçu** pour voir un peu ce que cela donne !

Remarquons aussi le nom de champ de la base de données : *champ2* ou *champ3*, c'est ce nom qui doit être utilisé dans les requête **SQL** comme ci-après.

5* Nous allons maintenant construire deux champs **SQL** : un pour calculer les votes et l'autres les abstentions ! Il est possible dans un champ **SQL** de mettre toute requête sur la base de données ! Nous allons utiliser ici les deux requêtes *SUM* et *COUNT* ! Celles-ci s'appliquent sur des champs de la table (*champ2* en l'occurrence puisqu'il est le seul à contenir des informations sur le vote). Cliquons sur **SQL** pour insérer un champ **SQL**

champ4	Abstentions
<input type="checkbox"/> Requête SQL	100*SUM(champ2='abstention')/COUNT(champ2)

Tapons comme titre *abstentions* mais ce titre ne sera pas affiché ! Les champs **SQL** sont des champs *cachés*. Ils doivent être repris pour l'affichage dans des champs **php**. Comme code nous tapons `100*SUM(champ2='abstention')/COUNT(champ2)`

Cette requête va récupérer la somme des abstentions du *champ2* pour les diviser par le total des enregistrements de ce champ (*COUNT*) et convertir en pourcentage en multipliant par 100.

Redemandons une insertion SQL pour les votes oui

champ5	Votes
<input type="checkbox"/> Requête SQL	100*(SUM(champ2='oui'))/(COUNT(champ2)-SUM(champ2='abstention'))

Tapons comme titre *Votes* et comme requête :

`100*(SUM(champ2='oui'))/(COUNT(champ2)-SUM(champ2='abstention'))`

Requête un peu plus longue qui doit diviser les *oui* par le total des suffrages exprimés : tous les enregistrements *moins les absentions* !

6* Nous allons maintenant afficher ces résultats aux seins de deux champs **php**. Cliquons donc deux fois sur **php** dans la barre rouge

champ6	Abstentions	Centré (1=oui)	1	Ratio (%)	50
Script PHP	echo ROUND(\$champ4)." %";				
<input type="checkbox"/> Code PHP	Occupation (en %)	50	hauteur	40	Fond #F0F0F0 Bordure 1
champ7	Votes	Centré (1=oui)	1	Ratio (%)	50
Script PHP	echo ROUND(\$champ5)." %";				
<input type="checkbox"/> Code PHP	Occupation (en %)	50	hauteur	40	Fond #F0F0F0 Bordure 1

Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer + -

Nommons le premier champ *Abstentions* (cette fois ce terme sera affiché) et le second *OUI*.

Dans le premier champ php affichons le résultat du *champ4* en tapant

`echo ROUND($champ4)." %";`

De même dans le second tapons le résultat du *champ5* en tapant

`echo ROUND($champ5)." %";`

N'oublions pas le point-virgule final !

On fait les réglages suivants : 50% d'occupation pour chacun avec une hauteur de 40, une bordure égale à 1 et une couleur de fond #F0F0F0.

Cliquons sur **aperçu**.

6* Il nous reste a taper un petit peu de code php pour indiquer que le vote a bien été pris en compte :

Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer + -							
Code php à exécuter à la validation (peut rester vide)		echo " A voté ";					
Aperçu Construire TOUT Refaire OPTIONS Essai privé Essai public Options listing Listing privé Listing public							

Dans le cadre Code à exécuter à la validation tapons `echo "
A voté

";`

On fera attention de pas oublier le point-virgule final !

Notre formulaire est prêt il reste à construire les tables dans la base de données ! Cliquez sur le bouton **Construire TOUT** dans la bande noire !

Un message indique que les tables ont été construites (à condition que vous ayez correctement complété le fichier `variables_formulaire.php`) !

Vous pouvez commencer en faisant un **essai privé** (dans le bandeau noir). En privé vous disposez d'un *navigateur* pour saisir ou modifier autant de fiches que vous le désirez ! En public le visiteur ne disposera que des boutons **valider** et **refermer** ! A moins que vous lui autorisiez d'autres fonctions ! Pour ce faire il suffit de cocher les cases correspondantes dans la barre ci-dessous :

Aperçu	Construire TOUT	Refaire OPTIONS	Essai privé	Essai public	Options listing	Listing privé	Listing public
Cocher les champs que vous désirez rendre publics sur le formulaire				Bouton refermer		Bouton valider	
Compteur <input type="checkbox"/>	Navigation <input type="checkbox"/>	Suppression <input type="checkbox"/>	Recherche <input type="checkbox"/>	Atteindre <input type="checkbox"/>	Refermer	Enregistrer	
Script du bouton refermer					Email de retour	jean@jeanlepine.com	
Pour un formulaire privé		Identifiant	Mot(s) de passe		Droits		

Vous pouvez en profiter pour donner un nouveau nom au bouton de validation signer par exemple ! Une fois ces choix faits cliquez sur le bouton Refaire les OPTIONS qui va reconstruire la table [2] sans toucher à la table [1] qui contient les enregistrements que vous venez de faire !

En fait il n'est utile de **Construire TOUT** que lors de l'ajout ou la suppression d'un champ !

7* Il faut choisir les options pour l'affichage du listing ! Cliquez sur **Options listing** dans le bandeau noir.

Options d'affichage pour le listing									
Nb fiches par page	20								
Commentaire	champ1	Qu'avez-vous l'intention de faire lors du référendum du 29 Mai ?		Titre	Qu'avez-vous l'intention de fai				
Privé	<input type="checkbox"/>	largeur privée (%)	0	Public	<input type="checkbox"/>	largeur publique (%)	0	Fond	#FFFFFF
Choix multiple	champ2	&oui;non;blanc;abstention		Titre	Votre choix				
Privé	<input checked="" type="checkbox"/>	largeur privée (%)	60	Public	<input checked="" type="checkbox"/>	largeur publique (%)	60	Fond	#FFFFFF
Code PHP	champ3			Titre					
Privé	<input type="checkbox"/>	largeur privée (%)	0	Public	<input type="checkbox"/>	largeur publique (%)	0	Fond	#FFFFFF
Requête SQL	champ4	abstentions		Titre	abstentions				
Privé	<input type="checkbox"/>	largeur privée (%)	0	Public	<input type="checkbox"/>	largeur publique (%)	0	Fond	#FFFFFF
Requête SQL	champ5	votes		Titre	votes				
Privé	<input type="checkbox"/>	largeur privée (%)	0	Public	<input type="checkbox"/>	largeur publique (%)	0	Fond	#FFFFFF
Code PHP	champ6	Abstentions		Titre	Abstentions				
Privé	<input checked="" type="checkbox"/>	largeur privée (%)	20	Public	<input checked="" type="checkbox"/>	largeur publique (%)	20	Fond	#FFFFFF
Code PHP	champ7	OUI		Titre	OUI				
Privé	<input checked="" type="checkbox"/>	largeur privée (%)	20	Public	<input checked="" type="checkbox"/>	largeur publique (%)	20	Fond	#FFFFFF
La date									
Privé	<input type="checkbox"/>	largeur privée (%)	0	Public	<input type="checkbox"/>	largeur publique (%)	0	Fond	#FFFFFF
Refermer				Enregistrer					

Nous décidons d'afficher dans le listing les *champs de votes* (*champ2*), et des abstentions (*champ6*) et *votes* (*champ7*) avec 40 fiches par page !

Cliquons sur **enregistrer** pour valider ce choix puis sur **refermer** pour revenir à la page de création.

Après avoir rempli quelques enregistrements (**essai privé**) regardons un listing en cliquant sur le lien **listing privé**. Nous obtenons ce listing :

Page 1/12		Le référendum du 29 mai 2005		Nb fiches par page 15	
champ2	champ6	champ7			
Votre choix	Abstentions	OUI			
oui	0 %	100 %			
non	0 %	50 %			
blanc	0 %	33 %			
abstention	25 %	33 %			
blanc	20 %	25 %			
non	17 %	20 %			
oui	14 %	33 %			
oui	13 %	43 %			
oui	11 %	50 %			
oui	10 %	56 %			
oui	9 %	60 %			
oui	8 %	64 %			
non	8 %	58 %			
non	7 %	54 %			
oui	7 %	57 %			
Refermer		Imprimer		Suivante	
Requête SQL				O K ANNULE	

En cliquant sur le titre des champs on peut assurer un *tri descendant* puis en re cliquant un *tri ascendant* ! De même on peut taper une requête SQL comme **SELECT champ2='oui'** pour obtenir les fiches ayant voté **OUI** (ce qui n'a ici aucun intérêt bien entendu !)

ATTENTION, si vous utilisez **Refaire les OPTIONS** après avoir décidé des options de listing vous devrez refaire ces dernières qui seront écrasées !

8* Il s'agit maintenant de mettre notre formulaire en ligne. Rien de plus simple, il suffit de faire un *lien* appelant la page formulaire.php (elle sera publique bien entendu) ce cette façon :

formulaire/formulaire.php?nom_table=referendum

Si vous utilisez chrono-site il suffit d'affecter le lien ci-dessus à un *cadre de titre* par exemple !

De même pour appeler le listing (public aussi) il suffit d'appeler le lien

formulaire/listing.php?nom_table=referendum

Ces deux liens utilisent la table *referendum* que nous venons de créer !

IV * Autre exemple : système de réservations pour un spectacle

Taper le nom d'une nouvelle table dans la base de données (sans espace svp)		reservations	
Ou la sélectionner	Nouveau... ▼	Détruire la table	Vider la table
Titre du formulaire	Réservations pour la pièce de Feydeau "Tailleur pour dames"		Fond titre #707060
Occupation (en %)	80	Décalage 10	Fond page #F0F0E0
			Bordure 1
Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer +			

1* Le sélecteur de bases étant sur **Nouveau...** taper comme nom de table *reservations* et le titre *Réservations pour la pièce de Feydeau « Tailleur pour dames »*

2* Construire le système d'identification : insérer un **commentaire** descriptif (voir ci-dessous) puis trois fiches **saisies** (cliquer sur *saisie*) *Nom*, *prénom* et *email* et enfin un *commentaire* vide pour

équilibrer l'email de 50% (voir ci-dessous). Changer aussi la hauteur et la couleur !
 Cliquer sur aperçu pour vous rendre compte !

Titre du formulaire		Réservations pour la pièce de Feydeau				Fond titre	#707060			
Occupation (en %)		80	Décalage	10	Fond page	#F0F0E0	Bordure	1		
champ1	Identification, les champs ci-dessous sont tous obligatoires						Centré (1=oui)	1		
Commentaire	Occupation (en %)	100	hauteur	28	Numérique	<input type="checkbox"/>	Fond	#C0C0FF	Bordure	0
champ2	Nom				Max		Nb lignes	1	Ratio (%)	25
Saisie	Occupation (en %)	50	hauteur	28	Numérique	<input type="checkbox"/>	Fond	#E0E0FF	Bordure	0
champ3	Prénom				Max		Nb lignes	1	Ratio (%)	25
Saisie	Occupation (en %)	50	hauteur	28	Numérique	<input type="checkbox"/>	Fond	#E0E0FF	Bordure	0
champ4	Email				Max		Nb lignes	1	Ratio (%)	25
Saisie	Occupation (en %)	50	hauteur	28	Numérique	<input type="checkbox"/>	Fond	#E0E0FF	Bordure	0
champ5							Centré (1=oui)	1		
Commentaire	Occupation (en %)	50	hauteur	28	Numérique	<input type="checkbox"/>	Fond	#E0E0FF	Bordure	0
Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer + -										

3* *Système de réservations* adultes et enfants. Insérer un champ **commentaire** et deux champs de **saisies** comme ci-dessous

champ6	Vos réservations adultes (plus de 12 ans) et jeunes (12 ans ou moins)						Centré (1=oui)	1		
Commentaire	Occupation (en %)	100	hauteur	28	Numérique	<input type="checkbox"/>	Fond	#C0C0FF	Bordure	0
champ7	Adultes (4 €)				Max		Nb lignes	1	Ratio (%)	30
Saisie	Occupation (en %)	50	hauteur	28	Numérique	<input type="checkbox"/>	Fond	#E0E0FF	Bordure	0
champ8	Jeunes (2 €)				Max		Nb lignes	1	Ratio (%)	30
Saisie	Occupation (en %)	50	hauteur	28	Numérique	<input type="checkbox"/>	Fond	#E0E0FF	Bordure	0
Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer + -										

Respecter les *pourcentages d'occupation*, les *couleurs* les *hauteurs* et les *ratios*. Cliquer sur **aperçu**.

4* *Les états* : il s'agit ici de faire un état des réservations. *Combien de réservations* (on se fixera un maximum de 500 places) On affichera « COMPLET » en cas de dépassement. De plus on calculera la somme due par la famille qui vient de faire ses réservations. Il s'agit d'insérer un champ **commentaire**, un champ **SQL** et deux champs **PHP** et de les compléter comme ci-dessous :

champ9	Les états	Centré (1=oui)	1
<input type="checkbox"/> Commentaire	Occupation (en %) 100	hauteur 28	Numérique <input type="checkbox"/> Fond #C0C0FF Bordure 0
champ10	Réervations		
<input type="checkbox"/> Requête SQL	SUM(champ7) + SUM(champ8)		
champ11		Centré (1=oui)	1
Script PHP	if (\$champ10 < 500) {echo \$champ10." réservations sur 500";} else {echo "COMPLET !";}		
<input type="checkbox"/> Code PHP	Occupation (en %) 50	hauteur 28	Fond #C0C0FF Bordure 0
champ12		Centré (1=oui)	1
Script PHP	\$prix = \$champ7*4 + \$champ8 * 2; echo "pour une somme de ".\$prix." euros";		
<input type="checkbox"/> Code PHP	Occupation (en %) 50	hauteur 28	Fond #C0C0FF Bordure 0

Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer

Le champ **SQL** calcule nombre de places réservées (adultes + jeunes).

Le premier champ **PHP** affiche l'état des réservations ou bien affiche « COMPLET ». On remarque la nécessité de mettre un symbole \$ devant les *noms de champ* comme devant toute variable php !

Le second champ **PHP** calcule le prix puis affiche la somme due ! Attention à ne pas oublier les *points-virgules* qui marquent la fin des instructions php !

5* Il reste à écrire le *script de validation* qui est un peu long car il s'agit d'envoyer un email de confirmation à la personne qui vient de s'inscrire !

Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer

Code php à exécuter à la validation (peut rester vide)

```
if ($champ4 != "") {
    $message = "Confirmation de votre réservation au spectacle TAILLEUR POUR DAMES \n" ;
    $message = $message.$champ2." ".$champ3."\n".$champ7." adultes et ".$champ8." jeunes\n";
    $message = $message."pour un total de ".$prix." euros\n";
    mail($champ4,"Confirmation tailleur pour dames",$message);
    echo "Email de confirmation envoyé";
}
```

Aperçu Construire TOUT Refaire OPTIONS Essai privé Essai public Options listing Listing privé Listing public

Cocher les champs que vous désirez rendre publics sur le formulaire Bouton refermer Bouton val

Compteur Navigation Suppression Recherche Atteindre Refermer Enregistrer

Script du bouton refermer javascript:window.close() Email de retour jean@jeanlepine.

Pour un formulaire privé Identifiant Mot(s) de passe Droits

On remarquera les \n qui indiquent les retours à la ligne.

On utilise la fonction mail classique pour valider l'instruction mail il se peut que votre serveur n'accepte pas cette instruction ou bien qu'elle soit légèrement différente. Si c'est le cas il vous faudra adapter ! Cet email utilise la variable \$prix calculée dans le champ12 .

6 * Cliquez alors sur **Construire TOUT** pour fabriquer les deux tables dans la base de données. Vous pouvez faire quelques essais en cliquant sur **essai privé** !

7* pour obtenir le gain total il est possible de rajouter un champ **SQL**

champ13	Cumul (en euros)
Requête SQL	SUM(champ7)*4 + SUM(champ8)*2
<input type="checkbox"/> Saisie <input type="checkbox"/> Case à cocher <input type="checkbox"/> Choix multiple <input type="checkbox"/> Liste déroulante <input type="checkbox"/> Commentaire <input type="checkbox"/> Séparateur <input type="checkbox"/> SQL <input type="checkbox"/> PHP <input type="checkbox"/> Supprimer	

Ce champ est invisible au public et sera affiché au sein du *listing privé* ! Or il n'est possible d'accéder au *listing privé* qu'à partir de la page de création sur laquelle vous êtes (donc après identification).

8* Chosissons alors les **options listing** :

Options d'affichage pour le listing									
Nb fiches par page	<input type="text" value="20"/>								
Commentaire	champ1	Identification, les champs ci-dessous sont tous obligatoires			Titre	Identification, les champs ci-dessous sont tous obligatoir			
Privé	<input type="checkbox"/>	largeur privée (%)	<input type="text" value="0"/>	Public	<input type="checkbox"/>	largeur publique (%)	<input type="text" value="0"/>	Fond	<input type="text" value="#FFFFFF"/>
Saisie	champ2	Nom			Titre	Nom			
Privé	<input checked="" type="checkbox"/>	largeur privée (%)	<input type="text" value="20"/>	Public	<input checked="" type="checkbox"/>	largeur publique (%)	<input type="text" value="20"/>	Fond	<input type="text" value="#FFFFFF"/>
Saisie	champ3	Prénom			Titre	Prénom			
Privé	<input checked="" type="checkbox"/>	largeur privée (%)	<input type="text" value="20"/>	Public	<input checked="" type="checkbox"/>	largeur publique (%)	<input type="text" value="20"/>	Fond	<input type="text" value="#FFFFFF"/>
Saisie	champ4	Email			Titre	Email			
Privé	<input type="checkbox"/>	largeur privée (%)	<input type="text" value="0"/>	Public	<input type="checkbox"/>	largeur publique (%)	<input type="text" value="0"/>	Fond	<input type="text" value="#FFFFFF"/>
Commentaire	champ5				Titre				
Privé	<input type="checkbox"/>	largeur privée (%)	<input type="text" value="0"/>	Public	<input type="checkbox"/>	largeur publique (%)	<input type="text" value="0"/>	Fond	<input type="text" value="#FFFFFF"/>
Commentaire	champ6	Vos réservations adultes (plus de 12 ans) et Jeunes (12 ans ou moins)			Titre	Vos réservations adultes (plu:			
Privé	<input type="checkbox"/>	largeur privée (%)	<input type="text" value="0"/>	Public	<input type="checkbox"/>	largeur publique (%)	<input type="text" value="0"/>	Fond	<input type="text" value="#FFFFFF"/>
Saisie	champ7	Adultes (4 €)			Titre	Adultes (4 €)			
Privé	<input checked="" type="checkbox"/>	largeur privée (%)	<input type="text" value="10"/>	Public	<input checked="" type="checkbox"/>	largeur publique (%)	<input type="text" value="15"/>	Fond	<input type="text" value="#FFFFFF"/>
Saisie	champ8	Jeunes (2 €)			Titre	Jeunes (2 €)			
Privé	<input checked="" type="checkbox"/>	largeur privée (%)	<input type="text" value="10"/>	Public	<input checked="" type="checkbox"/>	largeur publique (%)	<input type="text" value="15"/>	Fond	<input type="text" value="#FFFFFF"/>
Commentaire	champ9	Les états			Titre	Les états			
Privé	<input type="checkbox"/>	largeur privée (%)	<input type="text" value="0"/>	Public	<input type="checkbox"/>	largeur publique (%)	<input type="text" value="0"/>	Fond	<input type="text" value="#FFFFFF"/>
Requête SQL	champ10	Réservations			Titre	Réservations			
Privé	<input type="checkbox"/>	largeur privée (%)	<input type="text" value="0"/>	Public	<input type="checkbox"/>	largeur publique (%)	<input type="text" value="0"/>	Fond	<input type="text" value="#FFFFFF"/>
Code PHP	champ11	Etat réservations			Titre	Etat réservations			
Privé	<input checked="" type="checkbox"/>	largeur privée (%)	<input type="text" value="30"/>	Public	<input checked="" type="checkbox"/>	largeur publique (%)	<input type="text" value="30"/>	Fond	<input type="text" value="#FFFFFF"/>
Code PHP	champ12				Titre				
Privé	<input type="checkbox"/>	largeur privée (%)	<input type="text" value="0"/>	Public	<input type="checkbox"/>	largeur publique (%)	<input type="text" value="0"/>	Fond	<input type="text" value="#FFFFFF"/>
Requête SQL	champ13	Cumul (en euros)			Titre	Cumul			
Privé	<input checked="" type="checkbox"/>	largeur privée (%)	<input type="text" value="10"/>	Public	<input type="checkbox"/>	largeur publique (%)	<input type="text" value="0"/>	Fond	<input type="text" value="#FFFFFF"/>
La date					Titre				
Privé	<input type="checkbox"/>	largeur privée (%)	<input type="text" value="0"/>	Public	<input type="checkbox"/>	largeur publique (%)	<input type="text" value="0"/>	Fond	<input type="text" value="#FFFFFF"/>
Refermer					Enregistrer				

Remarquons le *champ13* qui n'est affiché qu'en privé pour obtenir el cumul en euros !
Le titre (que vous pouvez changer) est celui des colonnes du listing.

8* Il reste à mettre le système de réservations en ligne.

le lien : formulaire/formulaire.php?nom_table=reservations (il sera public)

Si vous utilisez chrono-site il suffit d'affecter le lien ci-dessus à un cadre de titre par exemple !

De même pour appeler le listing (public aussi) il suffit d'appeler le lien :

formulaire/listing.php?nom_table=reservations

V- BONUS = un système de formulaires chaînés

Certains formulaires peuvent se montrer lassant à l'utilisateur par leur longueur. Il est intéressant dans ce cas de pouvoir découper celui-ci en plusieurs sous-parties. C'est ce que nous allons faire dans l'exemple ci-après : un formulaire **chaîne** qui demande l'identité suivi d'un formulaire **chaîne2** qui, après avoir récupéré le nom de la personne, demande son équipement informatique. Cet exemple est volontairement très court pour être plus explicatif sur les points essentiels :

1* Demandons d'abord nouveau puis tapons comme nom de table chaîne et Formulaire à étapes comme titre su formulaire ! Les questions d'identification sont simples : nom et prénom puis tranche d'âge puis Région ! Remplissons donc le formulaire comme ci-dessous :

Création d'un formulaire chrono-site

Taper le nom d'une nouvelle table dans la base de données (sans espace svp)

Ou la sélectionner

Titre du formulaire Fond titre

Occupation (en %) Décalage Fond page Bordure

champ1 Centré (1=oui)

Commentaire Occupation (en %) hauteur Numérique Fond Bordure

champ2 Max Nb lignes Ratio (%)

Saisie Occupation (en %) hauteur Numérique Fond Bordure

champ3 Centré (1=oui)

Commentaire Occupation (en %) hauteur Numérique Fond Bordure

champ4 0 = colonne

Choix multiple Occupation (en %) hauteur Numérique Fond Bordure

champ5 Centré (1=oui)

Commentaire Occupation (en %) hauteur Numérique Fond Bordure

champ6

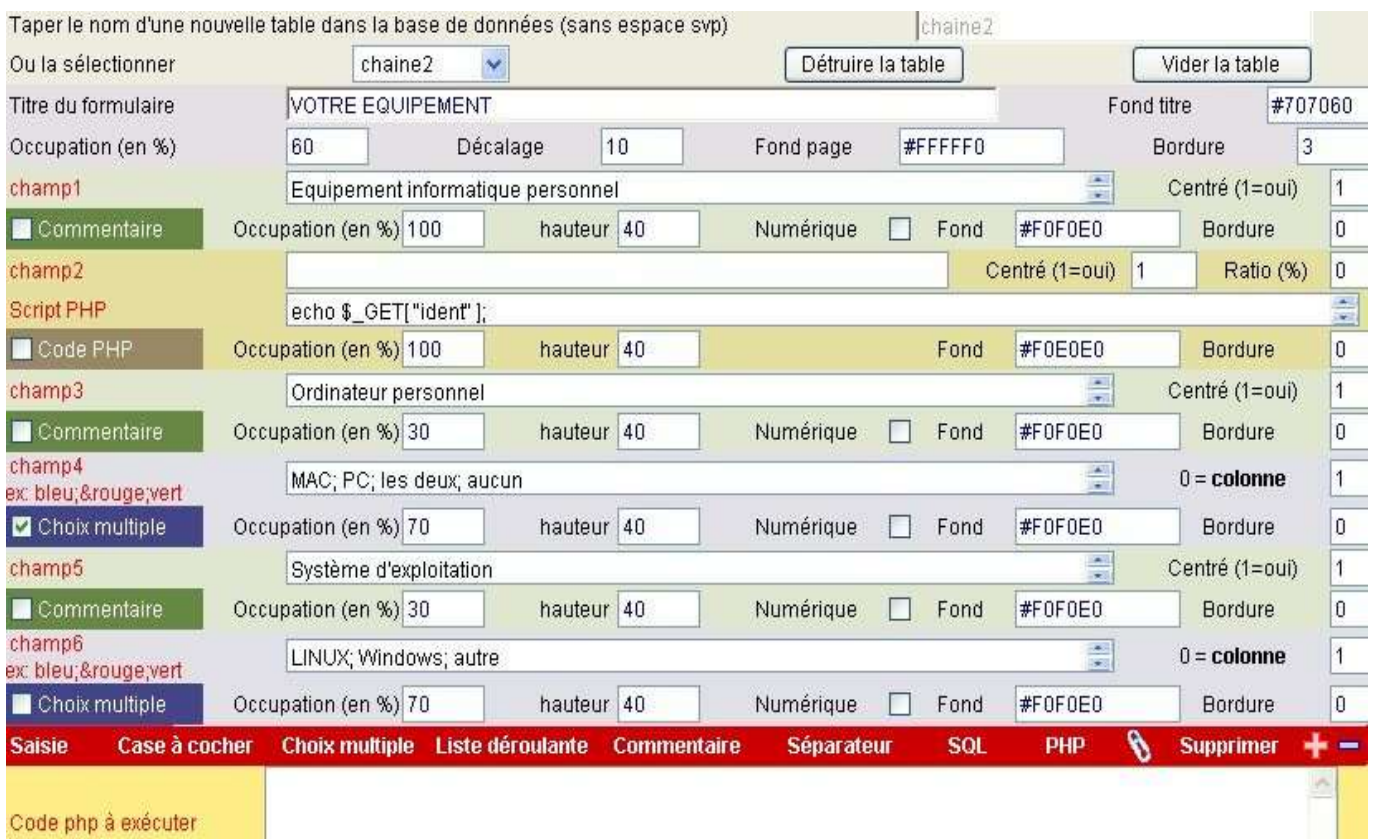
Liste déroulante Occupation (en %) hauteur Numérique Fond Bordure

Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer

2* Il reste maintenant à appeler le second formulaire en lui passant le **champ2** (nom et prénom) comme paramètre. Nous allons utiliser le lien suivant qui a deux paramètres : le nom de table (**chaîne2**) et le **champ2** (nom et prénom de l'utilisateur) : [formulaire.php?nom_table=chaîne2&ident='.\\$champ2](formulaire.php?nom_table=chaîne2&ident='.$champ2)
Ce qui va donner un le script ci-dessous dans la case de validation



3* Nous cliquons sur **Construire tout** et nous passons à la construction du formulaire *chaine2* (voir la copie d'écran ci-dessous)



Remarquons le *champ2* qui est un script **php** permettant de récupérer le paramètre passé du script précédent. Nous avons mis le contenu du *champ2* (du formulaire *chaine*) dans le paramètre nommé *ident*. Nous récupérons donc ici ce paramètre par la variable php **\$_GET['ident']** (le **\$_GET** permet une sécurité maximum).

4* Reste maintenant à **Construire TOUT** puis à revenir au formulaire *chaine* pour faire un essai privé ! Et les deux formulaires s'enchaînent merveilleusement bien ! De plus le *nom prénom* de l'utilisateur se retrouve également dans le listing de *chaine2* !!!!

5 * Le lien qu'il faudra appeler sera : *formulaire/formulaire.php?nom_table=chaine* ... il sera *public* à moins que vous n'ayez remplis les deux champs *identifiant* et *mot de passe* !

VI - Description des fonctionnalités :

- * **1** : le nom de la table dans la base de données sans espace et en minuscules si possible.
- * **2** : le sélecteur de formulaires. Celui-ci vous propose la liste de tous les formulaires que vous avez déjà créés (ou importés) ou bien *nouveau* si vous voulez en créer un autre !
- * **3** : Bouton pour détruire les deux tables associées au formulaire en cours !
- * **4** : Bouton pour vider la table 1 (donc toutes les saisies déjà faites) . Il faut utiliser ce bouton si vous désirez exporter votre formulaire pour le proposer au public (voir ci-après exporter un formulaire)
- * **5** : Titre du formulaire (sera affiché en haut de votre formulaire)

Les champs de formatage :

Occupation (en %) représente le pourcentage de la page qui doit être occupé par le formulaire.

Décalage est le nombre de pixels qui sépare le titre de la première ligne du formulaire

Fond de page et *Fond de titre* sont deux couleurs en HEXA sur le mode RVB classique ! Si besoin, certains logiciels de retouche donnent ces codages HEXA !

Manipulation des champs (barre rouge)

Insérer à la fin (ou bien avant le (ou les) champ(s) coché(s)) :

- * Un champ *saisie* : il s'agit d'un cadre de saisie qui peut éventuellement faire plusieurs lignes. Cocher l'option *numérique* si vous avez besoin de manipuler ce champ comme un nombre (exemple *l'age* avec *tri* dans les listing). ATTENTION pour que l'option numérique puisse avoir un sens il faut que les valeurs des champs **commencent pas des nombres** !!! Ce peut être par exemple pour un choix multiple : *1 lapin; 2 chèvre; 3 cheval; 4 autre*
Vous pouvez aussi lui imposer un nombre *maximum* de caractères (5 par exemple pour un code postal)
- * Une *case à cocher* : elle s'affichera à côté du titre que vous lui aurez donné

- * Un *choix multiple* : il suffit de taper les noms séparés par des points-virgules (voir premier exemple)
- * Une *liste déroulante* : même syntaxe que le choix multiple ci-dessous. Si certaines phrases de votre liste sont trop longues et font apparaître un déroulant de *très grande largeur* n'hésitez pas à y introduire des **
** (= *retour chariots* du langage *html* !)
- * Un *commentaire*
- * Un *séparateur* : peut être utile lorsque vous désirez séparer en plusieurs parties distinctes !
- * Un *champ SQL* : vous pouvez le nommer mais il ne s'affiche jamais de façon publique !

Exemples de requêtes SQL

SUM(champ5) + MAX(champ4)/COUNT(champ4)

100*MIN(champ2) + SUM(champ3 = 'oui')

SQRT(MAX(champ2))

Pour plus de renseignements sur les fonction mysql :

<http://dev.mysql.com/doc/mysql/fr/>


- * Un *champ PHP* : peut utiliser n'importe quelle fonction du langage php. Il sera fait un **eval** au sein de votre script et la syntaxe php doit donc être parfaitement respectée !

Exemples en php

```
if ($champ2 < 100) {echo 'GAGNE';} else {echo 'PERDU'; };
echo 'La somme est de '.$champ3*10 + $champ4*5.' euros';
```

Attention a bien mettre tous les points-virgules finaux !

Si vous mettez à **zéro la largeur du champ php** le code php sera exécuté mais le champ restera caché.

- * Un *document lié*  à la fiche (upload) : il faudra choisir un nom de *dossier* dans lequel le document sera stocké sur le serveur. Attention si ce *dossier* n'existe pas *chrono-form* essaiera de le créer. *Si ça ne fonctionne pas il faudra créer le dossier et lui donner les droits 777*. Vous pouvez ainsi donner un nom de dossier différent pour chaque formulaire utilisant les upload ! V

- * *Supprimer* détruit le dernier champ ou bien tous les champs cochés
- * Le signe + permet d'augmenter le nombre de lignes disponibles dans le dernier champ ou dans les champs cochés
- * Le signe - permet de diminuer le nombre de ligne de saisie.

Fonction de création (barre noire)

- * *Aperçu* : permet de voir l'aspect du formulaire sans créer les deux tables !
- * *Construire TOUT* : pour faire les deux tables du formulaire. Attention si vous avez déjà fait des enregistrements avec ce formulaire ils seront écrasés !
- * *Refaire OPTIONS* : à utiliser si **aucun champ n'a été ajouté ou supprimé** ! Ne modifie que la *table_2* sans toucher aux enregistrements ! Par contre efface le *formatage du listing*
- * *Essai privé* (ou *public*) pour essayer le formulaire en générant de véritables enregistrements en

mode *privé* (avec navigateur) ou en mode *public* !

* *Options listing* pour définir les colonnes qui seront affichés sur le listing (privé et public) et leur largeur !

* *Listing (privé et public)* les listing définis ci-dessous vous avez accès au privé d'ici alors que le public est accessible par appel direct de la page listing.php.

Imports - exports (barre rose)

Exporter sql Nom de la table reservations_bis

* **Exporter txt** vous permet de fabriquer un fichier **txt** qui pourra être ensuite réutilisé dans un tableur. Ce fichier contient tous les enregistrements de la table 1 (celle qui est exploitée dans les listings) vous pourrez donc, au sein de votre tableur habituel, supprimer ou modifier les champs à votre guise. Attention, les champs *php* ne sont pas toujours exportés convenablement et vous devrez souvent les refaire avec votre tableur ! Quand vous cliquez sur exporter vous obtenez la fenêtre suivante :



il vous faut alors récupérer le fichier (ici *inscriptions.txt*) avec le bouton droit de la souris et l'option « Enregistrer le cible du lien sous... ». Le fichier *txt* pourra alors être importé par votre tableur habituel !

* **Importer txt** pour récupérer les données d'un formulaire (et non pas sa structure comme avec le fichier *sql*) ou les données d'un tableur (voir ci-après la section *formulaires pré-remplis*).

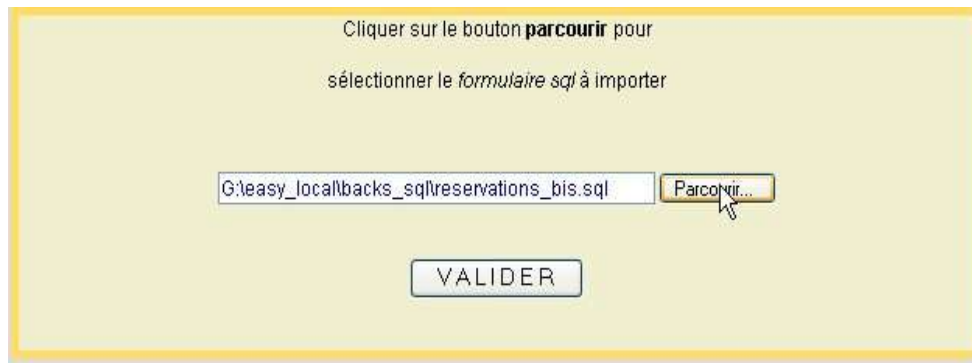
Exporter TXT Importer TXT Exporter sql Nom de la table reservations_bis Importer sql Faire BAK Récup BAK

* **Faire BAK** pour faire un fichier de sauvegarde de l'état actuel de votre formulaire. Ce bouton crée deux nouvelles tables *_3* et *_4* qui sont des copies des actuelles *_1* et *_2*.

* **Recup BAK** va chercher les deux tables *_3* et *_4* qu'il remet à la place des tables *_1* et *_2* actuelles ! Les tables *_3* et *_4* sont détruites !

* **Exporter sql** pour sauvegarder votre formulaire (le format *sql* est le format de la base de données *mysql*) . En cliquant sur ce bouton vous fabriquez un fichier *sql* avec le nom de table par défaut à moins que vous n'ayez changé ce nom. Si vous changez ce nom vous faites une *duplication* de votre formulaire ! Il vous suffit alors comme ci-dessus de récupérer le fichier *sql* avec le bouton droit de la souris ! Ce fichier peut permettre de récupérer le formulaire au sein d'une base ou plus simplement en utilisant le bouton importer ci-après !

* **Importer un formulaire** : en cliquant sur ce bouton vous ouvrez la fenêtre ci-dessous :



Il vous suffit alors de cliquer sur le bouton *parcourir* pour récupérer le fichier *sql* qui contient votre formulaire. Si le formulaire est scindé en deux fichiers il faut faire l'importation pour chacun des fichiers !

Dupliquer un formulaire : supposons que je veuille dupliquer le formulaire *réservations* pour en fabriquer un nouveau sans avoir à tout refaire et sans détruire l'ancien !

Méthode : j'ouvre le formulaire *reservations* et je renomme la table *reservations_bis*, je clique alors sur *Exporter sql*. Le fichier résultant contient mon formulaire de *réservations* mais est nommé *reservations_bis* ! Je peux donc l'importer sans crainte d'écraser le premier !

Options pour le formulaire public

- * **Compteur** : il affiche la fiche actuelle sur le nombre de fiches (25/25 par exemple)
- * **Navigation** : les touches *avant arrière* qui permettent de naviguer dans le formulaire
- * **Suppression** : cette fonction permet de supprimer un enregistrement
- * **Recherche** : en tapant un mot on sélectionne toutes les fiches contenant ce mot
- * **Atteindre** : pour se rendre directement à la fiche N° X

Souvent ces options restent réservées à l'administrateur et il n'est donc pas nécessaire de les cocher.

Nom du bouton des boutons **valider** et **refermer**

* Vous pouvez nommer comme vous le désirez les boutons de validation et de fermeture de votre formulaire : *signer* ou enregistrer ou *valider* avec *annuler*, *refermer* etc.

Script du bouton Refermer : il est par défaut *javascript:window.close()* et provoque une fermeture de la fenêtre, mais vous pouvez très bien appeler le bouton **Retour** au lieu de **Refermer** et mettre dans le

script une URL comme `http://chrono.apinc.org` à la place de `javascript:window.close()` !

Email de retour : si ce champ contient votre email un bouton « *poster* » apparaîtra pour permettre à l'utilisateur de vous envoyer les données saisies au format TXT. il vous restera alors à sauvegarder le contenu de votre email au format *txt* pour pouvoir récupérer l'ensemble des enregistrements saisis. Eviter le *copié-collé* qui risque de détenir les tabulations qui séparent les champs ! Les deux derniers champs de saisie permettent de **protéger le formulaire** par un *identifiant* et un *mot de passe* de votre choix... Cela signifie que l'utilisateur devra s'identifier pour remplir le formulaire. Si l'identifiant est vide le formulaire est public. Toute personne peut le remplir. Vous pouvez ignorer la case *droits*. Elle ne sert que dans le formulaire multi-niveaux (voir ci-dessous) !

VII - Formulaires multi-niveaux

Voici un cas d'école :

« Je souhaiterais faire un formulaire pour des dossiers de demande d'accompagnement d'auxiliaires de vie scolaire pour les enfants handicapés. Mon problème est que les informations viennent de **trois lieux différents**. Le formulaire, protégé par un mot de passe, doit être rempli par les écoles dans un premier temps. Ensuite des secrétaires de commissions de circonscription complèteraient la deuxième partie et pour finir, la CDES remplirait la troisième partie du formulaire qui la concerne. Serait-il possible de filtrer les accès pour chaque partie ? »

Voici donc la méthode pour résoudre le problème avec chrono-FORM :

Il s'agit de fabriquer un formulaire divisé en trois parties. Pour séparer chacune des parties il suffit tout simplement d'insérer un **séparateur de hauteur supérieure à 1** ! Vous disposez alors d'un formulaire découpé en trois parties « *partie A* » **séparateur 2** « *partie B* » **séparateur 2** « *partie C* » . Il s'agit maintenant d'ouvrir les parties à qui de droit ! Vous pouvez donc utiliser aussi *des séparateurs classiques* en leur laissant la hauteur 1 (celle par défaut). Pour les cloisons de séparations vous devez utiliser **des séparateurs de hauteur 2 ou plus** !

Nous allons définir des identifiants comme ci-dessous

Cocher les champs que vous désirez rendre publics sur le formulaire					Bouton refermer	Bouton valider
Compteur <input type="checkbox"/>	Navigation <input type="checkbox"/>	Suppression <input type="checkbox"/>	Recherche <input type="checkbox"/>	Atteindre <input type="checkbox"/>	Refermer	Enregistrer
Script du bouton refermer					Email de retour	
javascript:window.close()					jean@jeanlepine.com	
Pour un formulaire privé		Identifiant	Mot(s) de passe	Droits		
		profs	nivo1;nivo2;nivo3	1;2;3		

Nous avons définis trois mots de passe séparés par des points-virgules "*nivo1; nivo2; nivo3*" et un identifiant : *profs* !

Nous avons aussi donné des *droits* à chacun des mots de passe par exemple : **1** pour le *nivo1* ; **2** pour le *nivo2* et enfin **3** pour le *nivo3* ! Le sens de ces *droits* est expliqué plus bas !

Utilisation : les utilisateurs devront s'identifier pour remplir le formulaire. Suivant le mot de passe qu'ils

auront tapé il n'accéderont pas de la même façon; Voici les trois cas :

* En tapant *profs* et *nivo1* il auront accès à la *zone A* du formulaire uniquement , les autres zones étant lisibles mais *désactivées* !

* En tapant comme codes *profs* et *nivo2* ils auront accès à la *zone B* du formulaire, et pourront modifier la *zone 1* ! Ils disposeront de plus du *navigateur* pour parcourir toutes les fiches dont la *partie A* est déjà remplie ! La *partie C* ne leur sera pas accessible, elle sera désactivée !

* Les utilisateurs qui taperont comme codes *profs* et *nivo3* auront accès à la *zone C* du formulaire, et pourront modifier les zones *A* et *B* ! Ils disposeront de plus du *navigateur* pour parcourir les fiches !

Signification des droits 1 , 2 ou 3 (seules possibilités pour les droits) :

- 1 donne droit à la partie associée uniquement !
- 2 donne droit à toutes les parties précédentes !
- 3 donne droit à tout le formulaire !

Si par exemple vous tapez "1; 1; 1" comme *droits* chacun des utilisateurs *nivo1; nivo2; nivo3* ne pourra travailler que sur *sa partie* et ne pourra pas intervenir sur les autres !

Pour les *n* utilisateurs (*4 maxi*) seul le premier n'a pas la barre de navigation !

VIII - Insertions ou suppressions multiples

champ1	Commentaire d'identification						Centré (1=oui)	1			
<input type="checkbox"/> Commentaire	Occupation (en %)	100	hauteur	24	Numérique	<input type="checkbox"/> Fond	#F0F0E0	Bordure	0		
champ2	NOM						Max	Nb lignes	1	Ratio (%)	50
<input checked="" type="checkbox"/> Simple	Occupation (en %)	50	hauteur	24	Numérique	<input type="checkbox"/> Fond	#F0F0E0	Bordure	0		
champ3	Prénom						Max	Nb lignes	1	Ratio (%)	50
<input type="checkbox"/> Simple	Occupation (en %)	50	hauteur	24	Numérique	<input type="checkbox"/> Fond	#F0F0E0	Bordure	0		
champ4	Femme									Ratio (%)	50
<input checked="" type="checkbox"/> Case à cocher	Occupation (en %)	50	hauteur	24	Numérique	<input type="checkbox"/> Fond	#F0F0E0	Bordure	0		
Saisie Case à cocher Choix multiple Liste déroulante Commentaire Séparateur SQL PHP Supprimer + -											

Dans l'exemple ci-dessus où les deux champs *champ2* et *champ4* sont cochés deux cas peuvent se présenter:

* Si je clique sur **supprimer** les deux champs vont être détruits et le *champ3* sera renommé *champ2*.

* Si je clique sur **choix multiple** deux champs *choix multiple* vont être insérés, l'un au dessus du *champ2* et l'autre au-dessus du *champ4*. Seul le *champ1* ne sera pas renommé !

Si aucun des champs n'est coché l'*insertion* se fait à la fin et la *suppression* est celle du dernier !

IX - Formulaire pré-remplis

Il faut que l'utilisation des formulaires soit la plus simple possible. Il est très pénible d'avoir à ressaisir des informations standards qui sont toujours les mêmes ! Il serait pratique que le formulaire puisse remplir ces informations à notre place ! C'est possible avec chrono-form. Nous allons prendre l'exemple d'une demande d'autorisation d'absence dans le milieu de l'enseignement.

Objectif : l'enseignant qui remplit l'autorisation tape son NUMEN par exemple (il faut que le champ soit unique bien entendu, ça peut être aussi un email ou tout autre identifiant unique !) et le formulaire remplit automatiquement les cases Nom, prénom, naissance, établissement etc...

Fonctionnement :

a * Dans un tableur ouvrir le fichier contenant les informations sur les enseignants. Pour simplifier nous allons considérer que nous travaillons avec trois champs : NUMEN, Nom, prénom. Nous avons donc dans le tableur trois colonnes qui sont NUMEN, Nom, prénom. Nettoyons également les lignes inutiles comme les lignes de titres qui encombreraient inutilement le fichier TXT. Faisons alors un enregistrement au format TXT (avec tabulation comme séparateurs) ... Nous appelons ce fichier *numen.txt* !

b * Avec *chrono-form* nous allons créer un formulaire nommé *numen* qui va contenir trois champs nommés *numen*, *nom*, *prénom*. Nous faisons **construire tout** puis nous cliquons sur le bouton **importer TXT**. Choisissons alors le fichier *numen.txt* que nous venons de sauvegarder avec notre tableur et voilà ... le formulaire *numen* de *chrono-form* contient les données importées depuis le tableur. Nous allons lier notre formulaire de saisie à ce *numen* !

c * Notons qu'il peut exister une **table extérieure** (qui doit être dans la même base) contenant les numéro de NUMEN et informations ... elle pourra être liée au formulaire sans autre manipulation en utilisant la commande *liaison_ext* ! Dans la suite on appellera cette table *les_numen*.

d * Créons maintenant le véritable formulaire de saisie (autorisation d'absence) nommons ce formulaire *absences* et construisons-le de la manière suivante, les trois premiers champs seront trois champs de saisie, le premier demandera le *NUMEN*, le second le *Nom* et le troisième le *prénom*. Lors de la saisie, si le *NUMEN* (champ1) est contenu dans le formulaire *numen* que nous venons de créer (qui doit être dans la même base de données) alors les deux autres champs se rempliront en automatique avec les deux champs correspondants (*champ2* et *champ3* du formulaire *numen*). Pour que les cases se remplissent seules nous avons deux options : soit mettre un bouton à cliquer (« vérification » par exemple) soit ne rien mettre et, lors de la sortie du champ de saisie la vérification se fera ! Vous pouvez utiliser une liste déroulante si l'énumération n'est pas trop importante ! Evidemment dans ce cas il n'y aura plus aucune confidentialité puisque chacun pourra appeler n'importe quel enregistrement !

Que mettre dans chacun des champs pour lier les formulaires ?

deux syntaxes possibles : *liaison* ou *liaison_ext*

liaison pour lier le formulaire à un autre formulaire de votre base

liaison_ext pour lier à une table extérieure à chronoFORM

* avec **liaison** le *champ 1* (saisie ou liste déroulante) doit contenir ceci :

NUMEN # liaison; numen; champ1; verification

signification : *NUMEN* est la légende qui sera affichée devant le champ de saisie

est le séparateur de *script* (comme décrit plus haut)

liaison est le *mot clé* pour indiquer le *lien* avec un autre formulaire

numen est le nom de formulaire qui doit être lié à celui-ci

champ1 est le champ du formulaire *numen* qui contient le **NUMEN**

verification est le nom du bouton. Si ce paramètre est vide la vérification se fera automatiquement à la sortie du champ.

* avec **liaison_ext** le *champ 1* (saisie ou liste déroulante) doit contenir ceci :

NUMEN # liaison_ext; les_numen; numero ; verification

signification : *NUMEN* est la légende qui sera affichée devant le champ de saisie

est le séparateur de *script*

liaison_ext est le *mot clé* pour indiquer le *lien* avec une autre table (*de la même base !*)

les_numen est le nom de la table qui sera liée à ce formulaire. On peut lui adjoindre une REQUETE SQL comme par exemple :

les_numen WHERE age > 25 ... Ne seront récupérées que les fiches des personnes de plus de 25 ans !

numero est le champ de la table *les_numen* qui contient le **NUMEN**

verification est le nom du bouton. Si ce paramètre est vide la vérification se fera automatiquement à la sortie du champ.

* avec **liaison** le *champ2* (saisie) doit contenir ceci :

NOM # recup; 1; champ2

signification *NOM* est la légende du champ qui sera affichée

est le séparateur de *script* (comme décrit plus haut)

recup est le mot clé pour indiquer qu'il faut essayer de récupérer le champ ci-dessous dans le formulaire *numen* dont le *champ1* contient le *NUMEN* qu'on vient de taper

1 est un paramètre ajouté depuis la version 1.48 qui permet de récupérer sur plusieurs formulaires différents ! Il s'agit du numéro de champ où se trouve la *liaison* ! On met **7** si la formule **liaison** est dans le **champ7** !

champ2 est le nom du champ à récupérer dans *numen* !

* avec **liaison_ext** le *champ2* (saisie) contiendra ceci :

NOM # recup; 1; nom

signification **NOM** est la légende du champ qui sera affichée

est le séparateur de script (comme décrit plus haut)

recup est le mot clé pour indiquer qu'il faut essayer de récupérer le champ ci-dessous dans le formulaire *les_numen* dont le *champ numero* contient le *NUMEN* que l'on vient de taper

1 est un paramètre ajouté depuis la version 1.48 qui permet de récupérer sur plusieurs formulaires différents ! Il s'agit du numéro de champ où se trouve la liaison ! On met **5** si la formule **liaison_ext** est dans le **champ5** !

nom est le nom du champ à récupérer dans la table *les_numen* !

NOTE : si vous demandez un nombre de lignes négatif pour le champ de saisie celui-ci sera en **LECTURE SEULE** ! Si donc vous indiquez -1 comme nombre de lignes votre champ de saisie ci-dessus récupérera le nom le mettra dans le *champ2* qui **ne pourra être modifié** !

La fonction **liaison_ext** peut être utilisée **avec une table interne à chronoform**. Il suffit de connaître le véritable nom de la table dans la base de données ! Si vous avez nommé votre table **numen** elle se nomme dans la base **chrono_formulaire_numen_1** ... La requête pourrait donc être celle-ci

NUMEN # liaison_ext; chrono_formulaire_numen_1 WHERE age > 25 ; champ1; verification

On obtient ainsi les avantages de la fonction **liaison_ext** !

Remarque : à partir de la version 1.48 vous pouvez mettre dans le même formulaire autant de *liaisons* que vous le désirez ! Il suffira d'indiquer dans le champ *recup* quel est le champ de liaison concerné. Pour ce faire le paramètre après le mot clé *recup* est le numéro de champ où se trouve la liaison dans le formulaire ! Vous pouvez ainsi pré-remplir la partie *identité* avec la liaison *NUMEN* puis la partie choix du *stage* avec une liaison *STAGE* etc....

Si vous voulez utiliser une version de *chrono-form* ultérieure à la 1.47 avec un formulaire pré-rempli il vous faudra adapter vos champs *recup* et faire un « REFAIRE OPTIONS » car un « RECONSTRUIRE TOUT » écraserait tout vos enregistrements !

X- Formulaires interactifs

Cas d'école:

Je voudrais qu'un formulaire contenant *nom*, *prenom* (champs textes) et *profession* (liste déroulante) puisse être rempli par un utilisateur quelconque. J'aimerais aussi que les professions de la liste puissent elles-mêmes être mises à jour par cet utilisateur, via un autre formulaire de saisie si la profession cherchée n'existe pas encore.

Il est possible de créer ce genre de formulaire en utilisant la balise **liaison_liste**. Voici la façon de procéder :

a * Nous créons en premier un formulaire que nous nommons **professions**. Ce formulaire contient un seul *champ de saisie* (le *champ1*) nommé : *Nouvelle profession*. Nous prenons la précaution de mettre ce script sur le bouton refermer

```
javascript>window.opener.location.reload(); javascript>window.close()
```

qui va provoquer le rechargement de la page qui a appelé ! De cette façon la profession que l'on vient d'ajouter apparaîtra dans la liste déroulante au retour vers le formulaire. Revers de la médaille, le rechargement implique que les données qui auraient été saisies seront perdues !

b * Créons maintenant le formulaire principal que nous appelons **fiches**. Nous lui insérons en premier un champ commentaire dans lequel nous mettons le *script html* suivant :

```
<a href="formulaire.php?nom_table=professions" target="_blank">Cliquez ici
pour ajouter une profession</a>
```

```
<font color=red><i>à faire avant de compléter le formulaire</i></font>
```

Le lien appelle la table **professions** que nous avons créée ci-dessus qui va permettre à l'utilisateur d'insérer une profession éventuelle avant de saisir ses coordonnées. Nous insérons ensuite deux *champs de saisie* *Nom* et *prénom* puis un *champ déroulant* dans lequel nous mettons le script suivant :

```
liaison_liste;professions;champ1; order by champ1 ASC
```

explications **liaison_liste** est le mot clé pour indiquer que ce champ est liée à une autre table

professions est le nom de la table

champ1 est le champ de cette table qui va fournir la liste !

order by champ1 ASC (*optionnel*) est une instruction SQL permettant d'imposer un ordre de tri à la liste obtenue à partir du **champ1**. Vous pouvez omettre cette partie et les résultats seront triés par date. Vous pouvez utiliser un des tri ci-dessous :

order by champ1 ASC : *alphabétique ascendant*

order by champ1 DESC : *alphabétique descendant*

order by date ASC : *date de saisie ascendant (défaut)*

order by champ1 ASC : *date de saisie descendant*

c * La syntaxe ci-dessus ne peut être appliquée que sur un *champ déroulant* ou un *choix multiple*

XI - Formulaires avec champ identifiant

Il peut être très pratique de pouvoir revenir sur une fiche déjà remplie ou commencée à remplir !

Pour ce faire vous devrez utiliser un champ de saisie avec identifiant ! Ce champ permet à l'utilisateur de revenir sur une fiche commencée !

Supposons que le champ1 soit un champ de saisie avec identifiant. La syntaxe du champ sera la suivante : *Identifiant # ident; rafraichir*

signification : *Identifiant* est la légende qui sera affichée devant le champ de saisie
est le séparateur de *script* (comme décrit plus haut)
ident est le *mot clé* pour indiquer que c'est une cellule d'identification.
rafraichir est le nom du bouton. Si ce paramètre est vide la vérification se fera automatiquement à la sortie du champ.

Utilisation : au premier passage sur la fiche l'utilisateur tape un identifiant puis remplit une partie de la fiche. Il *valide* sa fiche incomplète ! Au second passage ce même utilisateur tape son identifiant et la fiche *telle qu'il la laissée* apparaît; Il peut alors finir de la compléter !

Remarque importante : ce système d'identification est très pratique pour retravailler sur sa propre fiche mais pose un problème déontologique ! En effet il faut des identifiants assez « lourds » car l'utilisateur lambda pourrait « *par hasard* » retomber sur celui d'un autre utilisateur dont la fiche s'ouvrirait !

Il est toutefois possible de verrouiller l'enregistrement : il suffit de taper un symbole # à la fin du champ identifiant comme par exemple : jean52#. Une fois enregistrée une telle fiche ne pourra plus être appelée ni modifiée !

XII - Variables et fonctions de contrôle

* la variable \$stop pour forcer le remplissage d'un champ

Il peut se faire que dans un formulaire vous désiriez « forcer » un utilisateur à remplir un champ donné, ou tout du moins à le remplir d'une façon donnée.

La variable prédéfinie **\$stop** (en minuscules svp !) est là pour empêcher l'enregistrement quand vous le désirez !

Etudions son fonctionnement sur un exemple. Ci-dessous un morceau du formulaire « enquête sur les modules de chrono-site » :

The screenshot shows a portion of a web form with four distinct fields, each with its own configuration bar:

- champ17**: Text input field with the question "Que souhaiteriez vous voir développé dans l'avenir comme nouveau module ?". Configuration: Occupation (en %) 100, hauteur 24, Numérique , Fond #F0E0A0.
- champ18**: Multiple choice field with the text "Un module QCM;Un système de blog;&autre". Configuration: Occupation (en %) 100, hauteur 24, Numérique , Fond #F0F0E0.
- champ19**: Text input field with the question "Si autre pourriez-vous préciser". Configuration: Occupation (en %) 100, hauteur 24, Numérique , Fond #F0F0C0. Includes "Max" and "Nb lign" options.
- Saisie**: A purple bar at the bottom with "Occupation (en %) 100, hauteur 24, Numérique , Fond #F0F0C0".

Nous notons que l'orque la réponse est **autre** il serait souhaitable que le le *champ19* ne soit pas vide !

Nous allons « imposer » à l'utilisateur de remplir le *champ19* quand la réponse est **autre** !

Dans le script final nous testons l'absence de *vide* dans le *champ19* dans le cas ou le *champ18* est égal à **autre**. Ce script *php* sera donc le suivant :

```
if ( ($champ18=='autre') && ($champ19=='') ) {
    $stop=1;
    echo "<font color=red><br>Veuillez compléter le champ <b>AUTRE</b> SVP !
    <br><br></font>" ;
} else {
    echo "<br>Merci d'avoir pris un peu de votre temps pour donner votre avis pour
    cet avis !<br><br>";
};
```

Le fait de forcer la variable prédéfinie **\$stop** (en minuscules SVP) à **1** va provoquer un *blocage* et l'enregistrement de la fiche en pourra pas s'effectuer ! Aucune fiche ne pourra donc avoir le champ **autre** validé sans en avoir complété le contenu !

* Autres variables prédéfinies

les variables de temps :

```
$date = la date d'aujourd'hui au format européen 04-06-2005
$annee = 2005
$mois = 6
$jour = 4
$heure = 11
$minutes = 25
$secondes = 51
```

voir ci-après l'utilisation avec une date par défaut !

Un compteur de passage :

`$passage` = le nombre d'essais successifs. Peut vous permettre de valider un enregistrement au bout de trois essais infructueux par exemple (voir ci après sur l'exemple de la fonction

`entre_dates`) !

`$kelle_fiche` = numéro de la fiche active !

* Des fonctions prédéfinies

Deux fonctions prédéfinies sont à votre disposition pour contrôler correctement les saisies.

telephone(`$champ8`) retourne TRUE si `champ8` est au format téléphonique français !

Exemple d'utilisation dans un champ `php` (le script de validation par exemple).

Contrôle du numéro de téléphone sur le `champ3` du formulaire :

```
if ( ( $champ3!='' ) && (!telephone($champ3)) )
{
    $stop = 1;
    echo 'Format téléphonique non reconnu !';
};
```

Deux mots d'explications :

la fonction `telephone` vérifie le format du `champ3` (si celui-ci *n'est pas vide* sinon la vérification se ferait dès le premier affichage !) . Si le format n'est pas correct (le *point d'exclamation* est le symbole de la *négation* en php) on met la variable `$stop` à 1 pour empêcher l'enregistrement et on affiche « *Format téléphonique non reconnu !* »

entre_dates(`$quand` , `$mini`; `$maxi`) retourne TRUE si la date `quand` est située entre les dates `mini` et `maxi` après avoir vérifié le format correct de la date `quand` !

Exemple d'utilisation à placer dans un champ `php` (le script de validation par exemple)

Contrôlons que le `champ7` est une date correcte et située entre les dates 01.01.2005 31.12.2005. Ce contrôle se fait 3 fois puis accepte la validation même si elle est incorrecte :

```
if ( ( $champ7 != '' )
    && ( !entre_dates($champ7, '01.01.2005', '31.12.2005' ) )
    && ( $passage<3 ) )
{ $stop = 1;
  echo 'ATTENTION la date n\'est pas correcte !!';
};
```

Deux mots d'explications :

la fonction `entre_dates` vérifie que le `champ7` est bien dans la plage 01.01.2005 - 31.12.2005 (si celui-ci *n'est pas vide* sinon la vérification se ferait dès le premier affichage !) . Si le format n'est pas correct (le *point d'exclamation* est le symbole de la *négation* en php) on met la variable `$stop` à 1 pour empêcher l'enregistrement et on affiche « *ATTENTION date non correcte !* ». Ceci se fait 3 fois uniquement (`$passage = 0`; `$passage = 1` et `$passage = 2`) puis accepte la date même si celle-ci n'est pas correcte.

unique(« champ1 ») retourne TRUE si le contenu du champ1 est unique ! Autrement dit cette fonction retourne FALSE si elle trouve un autre enregistrement dont le champ1 correspond au champ1 saisi !

*exemple de script avec la fonction unique() :

```
if (!unique("champ2") ) {$stop = 1; $erreur = "Cet identifiant est déjà pris ! " ; };
```

caracteres(« champ1 ») retourne le nombre de caracteres du champ.

update("table_externe" , "champ1" , \$champ1, "champ3", \$champ4) : permet de faire la mise à jour du formulaire « table_externe » dont le champ « champ1 » devra correspondre au contenu \$champ1 dans lequel le champ « champ3 » sera remplacé par le contenu \$champ4 ! Si, dans le formulaire « table_externe » aucun « champ1 » ne contient la valeur \$champ1 alors un nouvel enregistrement est créé avec cette valeur !

Exemple illustré :



A la validation du formulaire ci-dessus le formulaire « macros_suite » va être mis à jour de la façon suivante :

Supposons que j'ai tapé *jean52* dans la case *Qui donc* puis *Lépine* dans *NOM* et enfin *JEAN* dans *Prénom*.

La première ligne

```
update("macros_suite", "champ1", $champ1, "champ3", $champ2) ;
```

va tenter de chercher dans la table *macros_suite_1* l'enregistrement dont le *champ1* contient *jean52*. S'il le trouve il va mettre à jour le *champ3* avec *\$champ2* c'est à dire *Lépine*. S'il ne le trouve pas il va créer un nouvel enregistrement dont le *champ1* va contenir *jean52* et le *champ3* *Lépine* !

```
update("macros_suite", "champ1", $champ1, "champ4", $champ3) ;
```

va chercher dans la table *macros_suite_1* l'enregistrement dont le *champ1* contient *jean52* et mettre à jour le *champ4* avec *\$champ3* c'est à dire *JEAN*.

Le formulaire « macros_suite » contient donc maintenant un enregistrement *jean52, Lépine, JEAN* !

courriel(\$champ3) retourne TRUE si le *champ3* a un format correct de courriel.

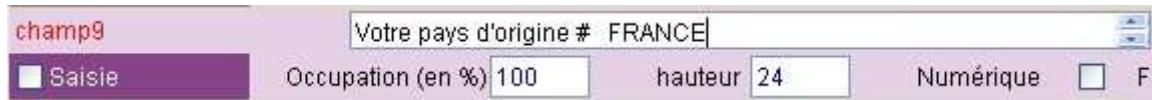
numerique(\$champ3) retourne TRUE si le *champ3* est bien un nombre.

url(\$champ3) retourne TRUE si le *champ3* a la forme d'une adresse de type *http://*!

*** Un affichage par défaut**

Il se peut que vous ayez besoin (pour accélérer la saisie) de proposer des champs pré-remplis ! Ceci est possible sur un champ de saisie en utilisant le symbole # ! La partie du champ de saisie située après le symbole # correspond à l'*affichage par défaut* ! voici deux exemples

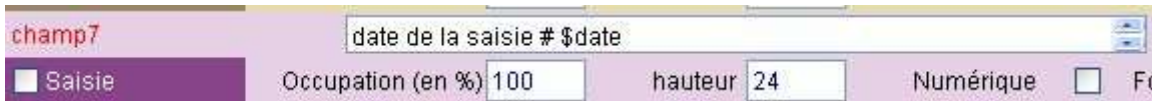
Exemple 1



Affichage résultant dans le formulaire :



Exemple 2



Affichage résultant dans le formulaire :

Bien entendu les champs restent des champs de saisie et sont donc modifiables. Si vous désirez avoir un affichage non modifiable (désactivé) il faut utiliser directement un champ *php* !

XIII - Exemples de scripts php

*a * envoi d'un email à champ4 :*

la variable \$prix est définie dans un script php. Le champ4 est un

```

if ( ($champ4 != '') && (courriel($champ4)) ) {
$message = "Comfirmation de votre réservation au spectacle TAILLEUR POUR DAMES ".chr
(13) ;
$message .= $champ2." ".$champ3."\n".$champ7." adultes et ".$champ8." jeunes".chr
(13);
$message .= "pour un total de ".$prix." euros".chr(13);
mail($champ4,"Confirmation tailleur pour dames",$message,"FROM : chrono@apinc.org");
echo "Email de confirmation envoyé";};

```

b * envoi d'un email de vérification au webmestre

```
mail(webmestre@monsite.org, "Formulaire", $champ3, "FROM : from@monsite.org") ;
```

Commentaires : le *champ3* contient le nom de la personne ou de l'école qui s'inscrit !

Vous pouvez mettre dans le *script de validation du formulaire* plusieurs lignes comme celle-ci et donc, envoyer un email de confirmation à plusieurs personnes!

c * calcul en euros

```
$prix = $champ7*4 + $champ8 * 2;
```

```
echo "pour une somme de ".$prix." euros";
```

Commentaire : le *champ7* combien le nombre d'adultes (à 4 euros) et *champ8* le nombre d'enfants (à 2 euros) !

d * une barre d'état de type histogramme

```
for ($k=1; $k<$champ34; $k++) {echo "<font color=red>X</font"; };
```

```
for ($k=$champ34; $k<60; $k++) {echo "<font color=#CCCCCC>X</font>";};
```

Commentaire : dans *champ34* le ratio (pourcentage des votes ramené à 60) !

e * un lien vers une adresse internet

```
echo '<a href="http://chrono.apinc.org/" target="_blank">Documents sur la  
constitution Européenne</a>';
```

Commentaire : le *target=_blank* pour ouvrir dans une nouvelle fenêtre sans perdre les informations déjà saisies !

XIV - importer et exporter des formulaires via la base de données

1* Pour exporter un formulaire il faut commencer par vider les tables (voir ci-dessus) de façon a ne pas laisser d'enregistrements dans le formulaire qui se retrouvera ensuite sur on ne sait quelle machine ! Ensuite il faut revenir au *bureau* de chronosite et se rendre dans la *base données* ! Sélectionnez alors les deux tables concernant le formulaire en question par exemple *chrono_formulaire_reservations_1* et *chrono_formulaire_reservations_2*. Cliquez ensuite sur le lien **Dumper les tables** et cocher comme ci-dessous avant de cliquer sur **exécuter**.

<input type="radio"/> Structure seule <input checked="" type="radio"/> Structure + données <input type="radio"/> Données seules <input type="checkbox"/> Drop Table ? <input type="checkbox"/> Insertions complètes ? Nb de requête / fichier : <input type="text" value="0"/> (Si 0, un seul fichier sera généré)	<input type="radio"/> Données CSV Séparateur CSV : <input type="text" value=";"/> Fin de ligne CSV : <input type="text" value="\n"/>
Requête SQL pour le dump : <div style="border: 1px solid gray; height: 100px;"></div>	Dumper les tables : <div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> chrono_formulaire_enquete_2 chrono_formulaire_referendum_1 chrono_formulaire_referendum_2 <li style="background-color: #e0e0e0;">chrono_formulaire_reservations_1 <li style="background-color: #e0e0e0;">chrono_formulaire_reservations_2 chrono_forum_messages chrono_forum_themes chrono_liens chrono_livredor chrono_matières </div>
<input checked="" type="checkbox"/> Transmettre ?	
<input type="button" value="Executer"/>	

Lui donner alors un nom parlant comme **chrono_form_reservations.sql** et l'enregistrer sur votre disque dur !

2* Pour importer un formulaire il faut récupérer le fichier **sql** des deux tables et l'exécuter dans votre admin sql classique. Si, par exemple vous êtes sous **Eskuel** (c'est le cas avec **chrono-site**) il faut ouvrir votre base (à gauche) puis cliquer sur le lien **insertion DUMP** (à droite) puis **sur cette page** et enfin **parcourir** pour récupérer le fichier *chrono_form_reservations.sql* ... Cliquer alors sur **OK** et les deux tables *chrono_formulaire_reservations_1* et *chrono_formulaire_reservations_2* s'insèrent dans la base de données !

Une fois revenu dans votre éditeur de formulaire (chemin *formulaire/index.php*) vous aurez un formulaire nommé *reservations* à votre disposition.