

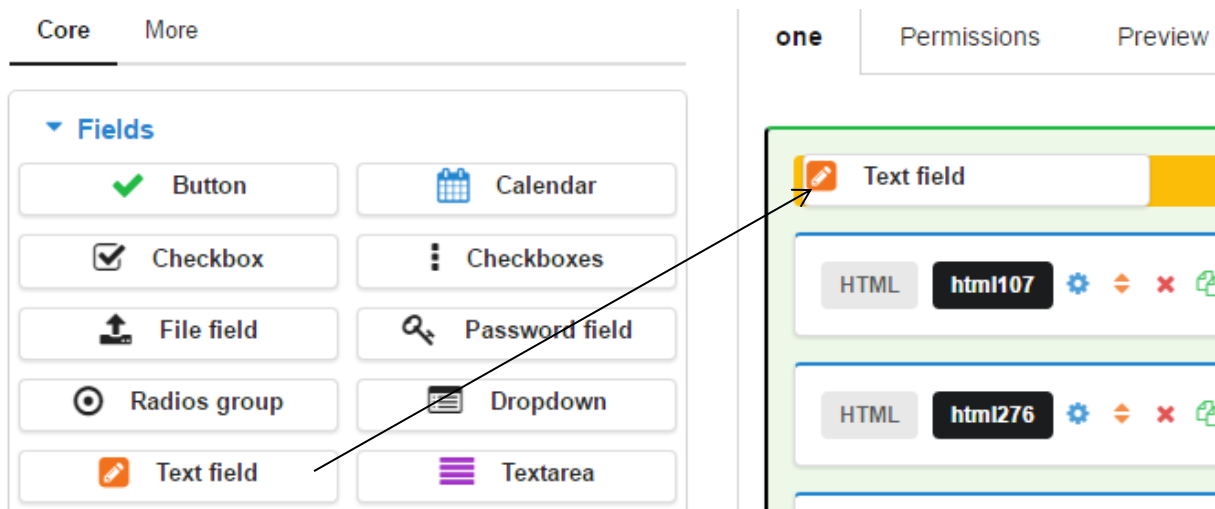
Contenu

Nouveau champ	2
Hide / show	3
Multi-fields	6
File fields (plusieurs fichiers en un)	7
Repeater Area.....	7
Check list.....	13
Checkbox / Checkboxes.....	14
Checkbox	14
Checkboxes.....	15
Email	17

Nouveau champ

Dans Chronoforms 6, il suffit de drag and drop les différents champs que l'on souhaite, comme dans Chronoforms 5.

Exemple avec un text field :



Ensuite on peut définir un nom à notre nouveau champ :



Dans l'onglet « **validation** » on peut définir des conditions à nos champs, comme par exemple si on veut un mail on peut définir le champ comme devant être un mail, ou un chiffre, ... On peut aussi dire le nombre maximal et/ou minimal de caractère que devra être notre champ.

General	Validation	Info	Advanced	Events	Permissions
<div><input type="checkbox"/> Required</div> <div><input type="checkbox"/> Optional If enabled then the validation will run if the field is NOT empty.</div>					
<div><input type="checkbox"/> Email</div> <div><input type="checkbox"/> URL</div>					
<div><input type="checkbox"/> Integer</div> <div><input type="checkbox"/> Decimal</div> <div><input type="checkbox"/> Number</div>					
<div>Regular expression</div> <div></div>		<div>Matches</div> <div>Enter the id of the matching field</div> <div></div>		<div>Different</div> <div>Enter the id of the different field</div> <div></div>	
<div>Minimum length</div> <div></div>		<div>Maximum length</div> <div></div>		<div>Contains</div> <div></div>	

Dans l'onglet « **Advanced** » on peut définir la taille de notre champ, ils sont définis en pourcentage :

The screenshot shows the 'Advanced' tab selected in a form editor. Below the tab bar, there is a text input field. Below the field, a note states: 'when another field is set to reload it.' Below this, there is a large empty rectangular area. To the right of this area, a 'Width' dropdown menu is open, showing options: 'Fluid', 'Fluid' (highlighted), '20%', '25%', '38%', '50%', and '75%'.

« **Fluid** » représentant le 100%.

ATTENTION il est important d'avoir les mêmes informations dans les champs « **Name** » et

« **ID** » :

The screenshot shows two input fields side-by-side. The left field is labeled 'Name' and contains the text 'nom_de_la_checkbox'. The right field is labeled 'ID' and also contains the text 'nom_de_la_checkbox'. Below the fields, a note states: 'No spaces or special characters should be used here.'

Hide / show

Il y a moyen d'afficher ou de cacher du contenu du formulaire en fonction de ce que l'utilisateur a besoin. Par exemple si l'utilisateur deux adresses, il pourra afficher un deuxième champ pour qu'il puisse mettre sa seconde adresse.

Adresse

☐ Deuxième adresse

Deuxième adresse

Voilà notre situation de base, on va commencer par cacher le champ deuxième adresse. Pour ce faire il faut aller dans les propriétés du champ dans l'onglet « **info** » et remplir le paramètre « **hidden** » :

Hidden

If not empty then the field will be hidden when the form is loaded.

Ce qui nous donne au chargement de la page :

Adresse

☐ Deuxième adresse

Maintenant on va mettre un événement sur la checkbox deuxième adresse pour afficher le second champ ou à nouveau le cacher si l'utilisateur s'est trompé. Pour cela il faut aller configurer les événements dans les paramètres de la checkbox, dans l'onglet « **Events** ». On va configurer un premier événement qui affichera le champ :

General Validation Info Advanced **Events**

On

Checked

Actions

Show ×

+

Element(s) identifier ⓘ

deuxime_adresse

Le paramètre « **On** » représente le type d'événement, il y a « **Checked** » quand on coche la case, « **Unchecked** » quand on décoche la case et « **change** » quand on modifie la case. Maintenant il faut définir quel tâche sera exécuté quand l'événement se déclenchera donc dans notre cas ça sera « **show** » car on veut afficher notre deuxième champ. Il reste juste à définir quel champ on va afficher il faut mettre l'id du champ dans le paramètre « **Element(s) identifier** », l'id du champ se trouve dans l'onglet « **General** ».

Adresse

☐ Deuxième adresse

→

Adresse

☒ Deuxième adresse

Deuxième adresse

Pour le désafficher il suffit de changer le paramètre « **On** » de « **Checked** » à « **Unchecked** ».

On a aussi la possibilité de afficher/cacher un area complète, on ne peut pas cacher une area au lancement de la page, pour la cacher on est obligé de faire un event.

Voici un exemple pour expliquer comme ça fonctionne, on a une « **radio group** » avec quatre valeurs, en fonction de ces deux valeurs on devra afficher tel area. On crée notre radio group avec nos valeurs, ensuite on crée nos areas par exemple une « **Message box** » et dans cette « **Message box** » on ajoute nos différents champs, ... Maintenant nos areas sont créées, il nous reste juste à faire nos events pour afficher et cacher nos areas. Pour ce faire il faut aller dans les paramètres de notre radio group, ensuite on crée un event pour quand on clique sur une valeur et quand on dé clique cette valeur. On fait ça car comme ça quand on clique sur la valeur on affiche l'area et quand on dé clique la valeur on cache notre area. Voici l'exemple des events :



Cet event affichera « **area_message_354** » quand l'utilisateur sélectionnera la valeur « **Swisscom standard** ». Pour cacher l'area il faut juste changer les deux premiers paramètres :

On	unSelecting	Actions	Hide
Value(s)	Swisscom standard	Element(s) identifier	#area_message_354

Et avec ça l'area se cachera quand on désélectionnera la valeur « **Swisscom standard** ». Pour les events avec des areas on est obligé de mettre un # devant l'id de l'area.

Voici maintenant l'exemple concret :

Etat initial :

Quel est votre abonnement actuel ? / Welches Abonnement haben Sie zurzeit?

- ☐ Swisscom standard
- ☐ Swisscom prépayé / prepaid
- ☐ M-Budget
- ☐ Autre opérateur / anderer Dienstleister

On a notre radio group avec ces 4 valeurs, ensuite on va en sélectionner une.

Quel est votre abonnement actuel ? / Welches Abonnement haben Sie zurzeit?

- ☒ Swisscom standard
☐ Swisscom prépayé / prepaid
☐ M-Budget
☐ Autre opérateur / anderer Dienstleister

Est-ce que votre abo a été renouvelé dans les 12 derniers mois ? / Wurde Ihr Abo während den letzten 12 Monaten erneuert? ☐ non ☐ oui

☐ Mon abonnement est lié à un DSL / Mein Abonnement ist zurzeit noch mit einem DSL verbunden

Et on peut constater qu'une area vient de s'afficher, il nous suffit juste de sélectionner une autre valeur pour afficher une autre area.

Multi-fields

Chronoforms 5 nous permettait de faire des multi-fields comme ceci :

Nom / Prénom *

Nom

Prénom

Il y a le champ nom/prénom à suivre, malheureusement dans Chronoforms 6 on ne peut plus faire exactement comme sur le 5. Pour pouvoir faire plus au moins la même chose il faut en premier créer « **multi field area** » qui se situe dans la catégorie « **area** ». Dès que l'on a créé notre « **multi field area** » on peut mettre nos différents champs dedans, il suffit juste de drag and drop ce que l'on souhaite mettre dans notre area.

Voici un exemple quand on met deux text fields dedans :

Nom *

Prénom *

Si on veut que les text fields prennent toutes la largeur de l'area, il faut aller modifier un paramètre, ce paramètre est :

Number of fields

Any

Divide the row evenly between fields by this number, Any will not make a division.

Ce paramètre définit combien il y a de text field dans l'area ce qui permettra d'ajuster la taille des text fields en fonction du nombre, voici le résultat quand on met qu'il y en aura 2 :

Nom *

Prénom *

On constate qu'ils prennent toute la largeur de la page.

File fields (plusieurs fichiers en un)

Le file fields est la même chose dans Chronoforms 6 que dans le 5. Il faut juste le drag and drop et il est fonctionnel. Si on veut pouvoir charger plusieurs fichiers dans une même file fields, il faut changer un paramètre :

Multi file select?

Yes

Ensuite on pourra charger plusieurs fichiers. Si on veut ensuite envoyer ces documents par mail, il faut encore changer un paramètre :

Name

cfc[]

Dans le paramètre nom de notre file fields il faut lui rajouter [] à la fin pour qu'après il envoie les différents documents. Cette option sert si par exemple un utilisateur doit mettre une image de son CFC et qu'il en a fait deux, il pourra directement mettre les deux documents dans le même « **file fields** » :

CFC relatifs aux obtentions nécessaires*

Sélect. fichiers 2 fichiers

Repeater Area

Dans Chronoforms 6, on a la possibilité de mettre un bouton qui reproduira une area prédéfini. Voici un exemple d'utilisation, dans le formulaire on demande à l'utilisateur de mettre ces expériences professionnelles, le problème c'est que l'utilisateur X en a 2 et l'utilisateur Y en a 4. On ne va pas mettre 4 fois les champs expériences professionnelles juste pour l'utilisateur Y alors que l'utilisateur X n'en utilisera que 2. Donc pour cela on va créer une « **repeater area** » qui nous permettra de copier une zone autant que l'utilisateur en aura besoin, l'utilisateur pourra en ajouter 2 et l'utilisateur Y pourra lui en ajouter 4.

Pour cela il faut ajouter une « **Repeater area** » dans notre form. Ensuite dans cette area il y a deux parties, le body et le footer. Le body représente la partie qui sera copiée et le footer est la partie qui sera toujours affichée, par exemple c'est dans le footer qu'on met le bouton pour copier le body ou par exemple si on veut mettre un commentaire sans qu'il soit copié à chaque fois on peut le mettre là.

The screenshot shows the 'Repeater area' configuration in Chronoforms. At the top, there's a tab labeled 'Repeater area' and a dropdown menu showing 'area_repeater412'. To the right of the dropdown are several icons: a gear (settings), a double-headed arrow (move), a red X (delete), a green plus (add), and a document icon. Below this, there are two main sections: 'body' and 'footer'. Each section has a colored header bar (orange for body, purple for footer) and a teal button labeled 'Minimized'.

Voilà à quoi ça ressemble.

La première étape est la configuration du repeater area, il y a plusieurs paramètres à modifier. En premier lieu il faut modifier le paramètre « **Data provider** » et ajouter « **{data : model}** » :

Data provider

Ensuite il faut changer le paramètre « **Enable content multiplier ?** » à « **yes** » :

Enable content multiplier ?

Les deux paramètres suivants représentent les classes des boutons, ils seront utilisés pour définir quel bouton permettra de copier le body et quel bouton pourra supprimer ce qu'on ne veut plus (on va utiliser ces paramètres dans la deuxième étape) :

Multiply button selector

Remove button selector

Le dernier paramètre permet de dire combien on va permettre de fois à l'utilisateur de copier le body :

Max multiplies

La deuxième étape est de créer le bouton qui copiera le body, donc il faut drag and drop un bouton dans le footer. Ensuite on peut changer le nom du bouton sous le paramètre « **Content** ». Il faut après changer la class du bouton, il suffit juste d'ajouter le mot « **multiply** », c'est le mot qu'on a défini dans la configuration de notre « **repeater area** » :

Class

Troisième étape. Maintenant que notre bouton pour copier notre body est créé, il faut ajouter ce que l'on veut copier dans le body (text field, checkbox, ...). Ensuite on a la possibilité d'ajouter un bouton pour supprimer ce que l'on vient de créer, si l'utilisateur a double cliquer sans faire exprès il

pourra supprimer celui en trop. Pour faire ça il faut juste ajouter un bouton tout simple dans le body, pour que le bouton supprime il faut juste changer la classe du bouton :

Class

remove

Pour les boutons de l'étape 2 et 3, il faut encore changer le type de bouton que c'est et mettre « **button** » :

Type

Button ▾

Pour la quatrième étape, on va modifier les différents « **fields** » du body, on va modifier 3 paramètres. L'exemple qui suit est sur un text field. Dans un premier temps il faut modifier le paramètre « **name** » et mettre les informations suivantes : `Model[{var:XX.key}][YY]`. Le prochain paramètre à modifier est l'« **id** » avec ces informations : `YY{var:XX.key}`. Le dernier à modifier est le paramètre « **value** » : `{var: XX.row.YY}`. L'`YY` représente le nom que l'on souhaite donner à la valeur, ça peut être n'importe quoi, mais si par exemple dans ce text field je demande à l'utilisateur de mettre le nom d'une entreprise, je pourrai écrire entreprise tout simplement. Le `XX` représente l'id du repeater area dans lequel il se trouve.

Voici un exemple de ces trois paramètres avec toutes les informations :

Name

`Model[{var:area_repeater412.key}][entreprise]`

ID

`entreprise{var:area_repeater412.key}`

Value

`{var:area_repeater412.row.entreprise}`

A chaque que l'on clique sur le bouton pour recrée notre body, il y a un chiffre qui s'auto-incrémente. On a la possibilité d'afficher ce chiffre, si on veut l'afficher par exemple dans un HTML il suffit de le rajouter comme suit :

`<h4>Expérience professionnelle -N-</h4>`

On peut aussi afficher ce chiffre dans un label ou autre, il faut juste rajouter « -N- ».

Voici ou on en est pour le moment :

Ajout Expérience professionnelle

De base il y a juste notre bouton pour créer nos bodys. Ensuite dès que l'on clique dessus il créera notre body autant de fois que l'on cliquera dessus jusqu'à la limite fixée :

Expérience professionnelle 1

Entreprise

Fonction

Période

A

Supprimer

Expérience professionnelle 2

Entreprise

Fonction

Période

A

Supprimer

Ajout Expérience professionnelle

On peut supprimer le bloque que l'on ne veut plus :

Expérience professionnelle 1

Entreprise

Fonction

Période

De

A

Supprimer

Ajout Expérience professionnelle

Maintenant l'utilisateur peut ajouter et supprimer autant qu'il le souhaite de bloque.

La dernière étape consiste à récupérer toutes les informations que l'utilisateur aura mises. Pour ce faire il faut aller dans l'onglet « **setup** », il faut créer dans un premier temps une « **loop** ». C'est dans cette « **loop** » qu'on ira chercher les informations. Il faut modifier le paramètre « **Data provider** » et mettre `{data :Model}` :

Data provider*

`{data:Model}`

La valeur de ce paramètre correspond à ce que l'on a mis dans la configuration de notre repeater area. Ensuite le paramètre suivant qui est « **Loop body** » nous permet de configurer comment on veut que les données seront affichées. On peut mettre du HTML tout simplement. Pour récupérer les informations il faut mettre ce code : `{var:ZZ.row.YY}`. L'YY représente les noms que l'on a mis dans la quatrième étape et le ZZ représente le nom de la loop :

Name

loop9

Voici un exemple de code HTML pour récupérer les données et les affichées correctement :

```
<tr>
<td>Entreprise</td>
<td>{var:loop9.row.entreprise}</td>
</tr>
<tr>
<td>Fonction / Domaine d'activit&eacute;</td>|
<td>{var:loop9.row.fonction}</td>
</tr>
<tr>
<td>P&eacute;riode</td>
<td>de {var:loop9.row.période} &agrave; {var:loop9.row.a}</td>
</tr>
```

Ce code sera exécuté pour chaque body que l'utilisateur aura créé. Ensuite il faut encore cocher le dernier paramètre pour qu'on puisse réutiliser le résultat de cette « **loop** » dans notre mail :



Return the result as var?

Ensuite dans le mail il suffit juste de mettre `{var:loop9}`

Voici les données qu'on rentre dans le repeater area:

Expérience professionnelle 1
Entreprise
entreprise numéro 1
Fonction
fonction numéro
Période
periode numéro 1
A
a numéro 1
Supprimer

Expérience professionnelle 3
Entreprise
entreprise numéro 3
Fonction
fonction numéro 3
Période
periode numéro 3
A
a numéro 3
Supprimer

Et le résultat dans le mail :

Entreprise	entreprise numéro 1
Fonction / Domaine d'activité	fonction numéro 1
Période	de periode numéro 1 à a numéro 1
Entreprise	entreprise numéro 3
Fonction / Domaine d'activité	fonction numéro 3
Période	de periode numéro 3 à a numéro 3

Check list

1. Configuration repeater area (data provider, les classes des boutons, ...)
2. Création du bouton dans le footer
3. Mettre les différents fields dans le body + bouton suppression
4. Modifications des paramètres des fields
5. Récupération des données + mettre les infos dans le mail

Checkbox / Checkboxes

Les checkbox permettent de mettre 1 case à cocher alors que les checkboxes permettent de mettre plusieurs cases à cocher.

Checkbox :

☐ *Ich habe die **allgemeine Bedingungen** gelesen und bin damit einverstanden

Voilà un exemple d'utilisation de la checkbox pour les conditions générales.

Checkboxes :

Ich melde mich für folgende Kurse verbindlich an:

- ☐ 1. Seltene Gemüse neu entdecken (26.09.2017)
- ☐ 2. Meine Lieblingstasche selbst genäht (03., 10.10.2017 und 07.11.2017)
- ☐ 3. Frouetränf (05.10., 02.11., 07.12.2017., 11.01., 01.02., 01.03., 19.04., 03.05. und 07.06.2018)
- ☐ 4. Dessertcremen, Flans, Panna Cotta, Crèmes brûlées (11.10.2017)
- ☐ 5. Kochen macht Spass: Anfängerkurs (24. und 31.10.2017)
- ☐ 6. Sous-vide Garen leicht gemacht (29.11.2017)
- ☐ 7. AgroTwin Cash (30.11.2017)

[Voici un autre exemple mais cette fois avec une Checkboxes, on peut avoir autant de cases à cocher que l'on souhaite.](#)

☒ Checkbox

☐ Checkboxes

Voilà comment elles sont représentées dans Chronoforms 6

Checkbox

La configuration d'une checkbox est assez simple, il y a peu de paramètre à configurer.

Label

Nom de la checkbox

Name

nom_de_la_checkbox

No spaces or special characters should be used here.

ID

nom_de_la_checkbox

Value

Valeur qui sera retourné quand la case est cochée

The value sent when the checkbox is checked.

Style

Default

Dans un premier temps il faut mettre le texte de la checkbox qui est le label.

☐ Nom de la checkbox

Ensuite on peut configurer une valeur de retour pour quand on clique sur la checkbox ça nous retourne cette valeur, c'est le champ « **value** ». Comme ça quand on prendra la valeur de la checkbox dans le mail, on recevra la valeur que l'on a configurée.

Checkboxes

La configuration d'une checkboxes n'est pas très compliquée :

The screenshot shows the configuration interface for a checkbox field in Chronoforms. It has a tabbed interface with 'General', 'Validation', 'Info', 'Advanced', and 'Events' tabs. The 'General' tab is active. It contains fields for 'Label' (containing 'nom de la liste'), 'Name' (containing 'nom_de_la_liste[]'), 'ID' (containing 'nom de la liste'), and 'Options' (containing 'option1=option1', 'option2=option2', 'option3=option3', 'option4=option4'). There is also a 'Layout' dropdown set to 'Vertical' and a 'Checked options values' field. A note below the 'Name' field states: 'No spaces or special characters should be used here. In order to submit multiple selections, the name must end with two square brackets []'.

Dans un premier temps il faut configurer le nom de la checkboxes qui sera affiché, c'est le champ label. Ensuite il faut configurer les différents options que l'utilisateur aura a coché, c'est le champ options. Il faut respecter la syntaxe suivante pour ajouter des options : XXX=XXX, il faut mettre la même chose des deux côtés du égale. Et voici le résultat :

nom de la liste

- ☐ option1
- ☐ option2
- ☐ option3
- ☐ option4

On peut modifier l'affichage des checkboxes soit que les options soient en vertical soit en horizontal, pour modifier ça il faut changer le champ « **Layout** ».

Ensuite le dernier champ à configurer est le champ « **name** ». Vous pouvez voir dans l'image de la configuration d'une checkboxes dans le champ « **name** » il y a des [] à la fin. Ces crochets permettent de dire que c'est un tableau qui peut contenir plusieurs informations donc toutes les cases que l'utilisateur aura coché.

Ich melde mich für folgende Kurse verbindlich an:

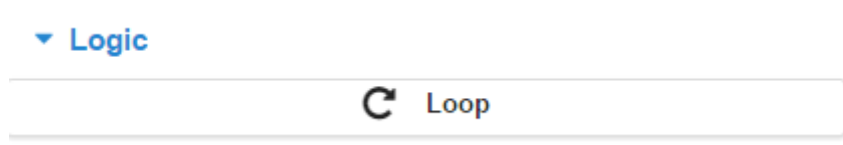
- ☐ 1. Seltene Gemüse neu entdecken (26.09.2017)
- ☐ 2. Meine Lieblingstasche selbst genäht (03., 10.10.2017 und 07.11.2017)
- ☐ 3. Froueträff (05.10., 02.11., 07.12.2017., 11.01., 01.02., 01.03., 19.04., 03.05. und 07.06.2018)

Pour une liste comme ça il faudra mettre les [] à la fin du champ « **name** ». Car si on ne met pas les [] il nous retournera uniquement la dernière valeur sélectionnée.

Pour afficher les options de la checkboxes que l'utilisateur aura coché il y a deux possibilités. La première est de juste tout simplement mettre {data :id} et voici le résultat :

```
["1.1 Hamburgers revisités","1.2 COMPLET - Pâte levée : tresses","1.2.2  
NOUVEAU - Pâte levée : tresses","1.3 Tout sur la tomate","1.4 Arrangement  
de Pâques : couronne fleurie","1.5 Desserts fruités","1.6 Le b.a.-ba du  
jardinage sur le balcon","1.7 COMPLET - One pot – en vogue !","1.7.2  
NOUVEAU - One pot – en vogue !","1.8 Donner un coup de jeune à mon  
vieux meuble préféré","1.9 Cuisine d'été à travers les continents"]
```

Le problème est que ce n'est pas très compréhensible. Pour faire que ce soit plus simple à comprendre il faut créer une loop dans l'onglet « **set up** ». La loop se trouve dans le sous onglet « **logic** » :



Ensuite il faut configurer le champ suivant :

Data provider *

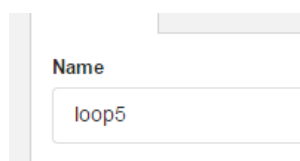
```
{data:melde1}
```

Avec l'id de la checkbox que l'on souhaite afficher le résultat ({data :id}).

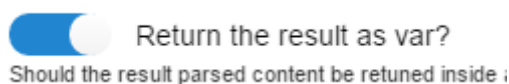
Ensuite dans le champ « **Loop body** », il faut mettre le code suivant :

```
<p>{var:loopX.row}</p>
```

loopX correspond au nom de la loop :



Finalement il faut activer l'option tout en bas qui est :



Et il suffira de mettre {var :loop5} dans le mail pour avoir nos informations mises en forme comme suit :

1. Seltene Gemüse neu entdecken (26.09.2017)
2. Meine Lieblingstasche selbst genäht (03., 10.10.2017 und 07.11.2017)
15. Weniger ist mehr – entrümpeln, entlasten, entschleunigen (12.03.2018) Ich nehme am Apéro teil

Email

L'envoi de mail sur Chronoforms6 a été simplifié. En premier lieu il faut mettre l'adresse mail de destination, qui recevra le mail. Ensuite il faut mettre le sujet du mail, puis il y a une option qui nous permet de prendre toutes les informations du formulaire et de les mettre dans le mail, ça sortira de cette façon :

Titre Monsieur

Rue impasse du test

E-mail gmail@gmail.com

Date de naissance 21.12.1997

Nationalité suisse

Pour arriver au résultat ci-dessus il faut cocher cette case :



Auto add fields data

Auto append email enabled form fields labels and values to the email body, or use {AUTO_FIELDS} where you want the data to be placed.

Si on décoche la case, on peut définir nous même une mise en page avec les informations du formulaire, pour ce faire il faut écrire du code. Voici un exemple ou on prend juste le nom et prénom du formulaire :

```
<td><strong>Coordonnées personnelles</strong></td>
<td>&nbsp;</td>
</tr>
<tr>
<td>Titre</td>
<td>{data:titre}</td>
</tr>
<tr>
<td>Nom</td>
<td>{data:nom_prenom}</td>
</tr>
<tr>
<td>Prénom</td>
<td>{data:prenom}</td>
</tr>
<tr>
<td>Rue</td>
```

```
<td>{data:rue}</td>  
</tr>
```

{Data : prenom} représente la donnée qu'a remplis l'utilisateur dans le formulaire. Voici le résultat d'un mail avec une solution formatée :

Coordonnées personnelles

Titre	Monsieur
Nom	Guillaume
Prénom	Seydoux
Rue	Impasse

Ensuite on peut décider si on veut mettre les fichiers uploadé dans le mail, il le fera automatiquement. Si par exemple un utilisateur du formulaire doit mettre un certificat ou autre, il sera inclus dans le mail. Pour ce faire il faut cocher cette case

Attachments settings



Auto Attach file fields

Auto attach attachment enabled fields to this email ?

Il nous reste plus qu'à écrire l'envoyeur du mail, voici un exemple qui fonctionne :

From email

from@grangeweb.ch

If left empty then the site from email will be used.

From name

Grangeweb|

If left empty then the site from name will be used.