



---

[CODE](#) > [WORDPRESS](#)

# Créer un plugin de formulaire d'inscription WordPress personnalisé

---

by [Collins Agbonghama](#) 29 Sep 2014

Difficulty: Intermediate Length: Long Languages: Français

[WordPress](#)[PHP](#)[Plugins](#)

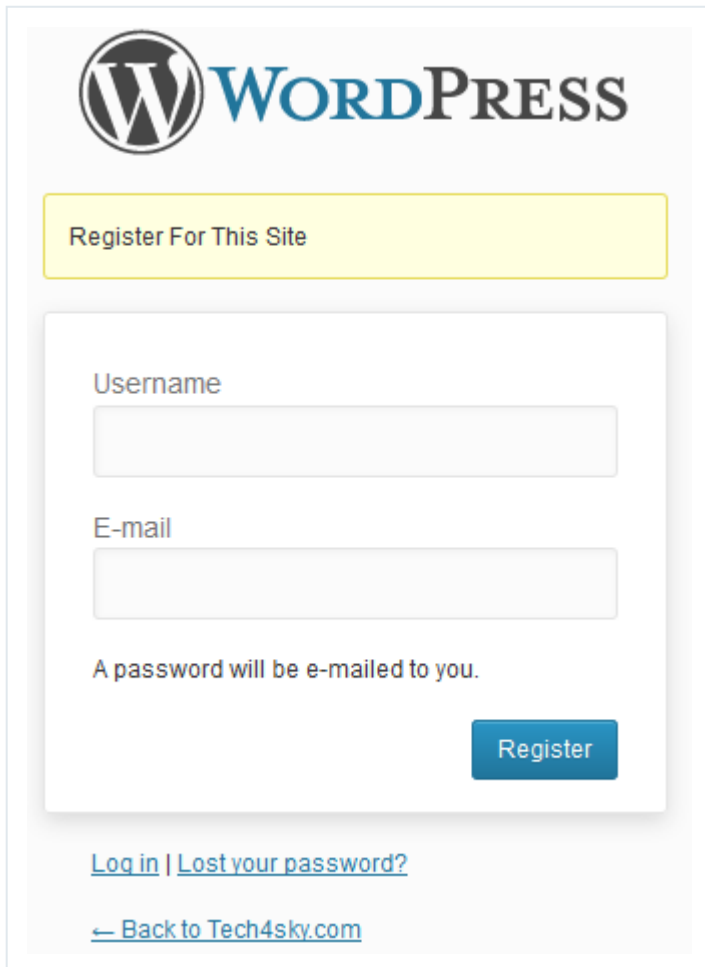
---

French (Français) translation by [Daszie Chan](#) (you can also [view the original English article](#))

Directement après l'installation, WordPress offre un formulaire d'inscription personnalisée qui peut être utilisé pour configurer un nouvel utilisateur, ou ajouter un nouvel utilisateur lorsque vous l'ajoutez à une installation existante de WordPress. Mais que se passe-t-il si vous souhaitez implémenter un formulaire d'inscription personnalisé qui ne fourni pas les options dans le tableau de bord de WordPress?

Dans ce tutoriel, nous apprendrons comment créer un formulaire d'inscription personnalisé dans WordPress en utilisant une combinaison de template tags et de shortcodes.

Le formulaire d'inscription par défaut se compose de deux champs de formulaire - nom d'utilisateur et email.

The image shows a screenshot of the default WordPress registration form. At the top, there is the WordPress logo and the word "WORDPRESS" in a large, blue, serif font. Below this, there is a yellow rectangular button with the text "Register For This Site". Underneath the button, there is a white rectangular box containing the registration form. Inside this box, there are two input fields: the first is labeled "Username" and the second is labeled "E-mail". Below the "E-mail" field, there is a line of text that says "A password will be e-mailed to you." To the right of this text is a blue rectangular button with the word "Register" in white. At the bottom of the white box, there are two links: "Log in | Lost your password?" and "← Back to Tech4sky.com".

La présence des champs nom d'utilisateur et email facilite incroyablement le processus d'inscription. Tout d'abord, vous entrez votre nom d'utilisateur et votre courriel, après quoi un mot de passe vous sera envoyé. Ensuite, vous vous connectez sur le site avec le mot de passe et ensuite complétez votre profil et changez le mot de passe pour quelque chose de plus mémorable.

Plutôt que de faire passer l'utilisateur par toutes les étapes qui précèdent juste pour s'inscrire sur votre site, pourquoi ne pas fournir un formulaire d'inscription direct composé de champs de formulaire supplémentaires importants en plus le nom d'utilisateur par défaut et l'e-mail comme un mot de passe, une URL pointant vers leur site Web, une biographie, un pseudo et leur prénom.

Ceci est particulièrement utile dans un site utilisé par plusieurs auteurs comme Tuts +.

Dans cet article, nous allons construire un plugin de formulaire d'inscription personnalisé avec les champs de formulaire suivants :

- nom d'utilisateur
- mot de passe
- e-mail
- URL du site web
- prénom
- nom de famille
- pseudo
- biographie (ou une section à propos)

Le formulaire d'inscription personnalisé peut ensuite être intégré dans WordPress en utilisant le plugin shortcode et le template tag associé.

Avec le shortcode, vous pouvez créer une page et en faire la page officielle d'inscription de votre site. Vous pouvez également utiliser le shortcode dans un post afin que l'utilisateur puisse s'inscrire sur votre site après avoir lu un de vos articles.

Si vous souhaitez ajouter le formulaire d'inscription à la barre latérale ou à un emplacement spécifique dans votre site web, vous pouvez modifier le thème WordPress mais en plaçant le template tag à l'endroit désiré.

Avant de commencer à construire le plugin du formulaire d'inscription, il est à noter que les champs nom d'utilisateur, mot de passe et e-mail sont requis.

Nous va être appliquer cette règle lors de l'écriture de notre fonction de validation.

## Option Premium

Ce tutoriel va vous apprendre à construire le plugin à partir de zéro, mais si vous cherchez une solution rapide, plug-and-play, essayez le plugin [WordPress Registration Form](#) sur Envato Market. Vous pouvez définir un large éventail de champs d'enregistrement avec un contrôle de validation. Une fois que l'enregistrement est terminé, un e-mail est envoyé au nouveau membre avec ses identifiants. Les modèles d'e-mail peuvent être modifiés pour l'inscription, le changement de mot de passe, etc..

---



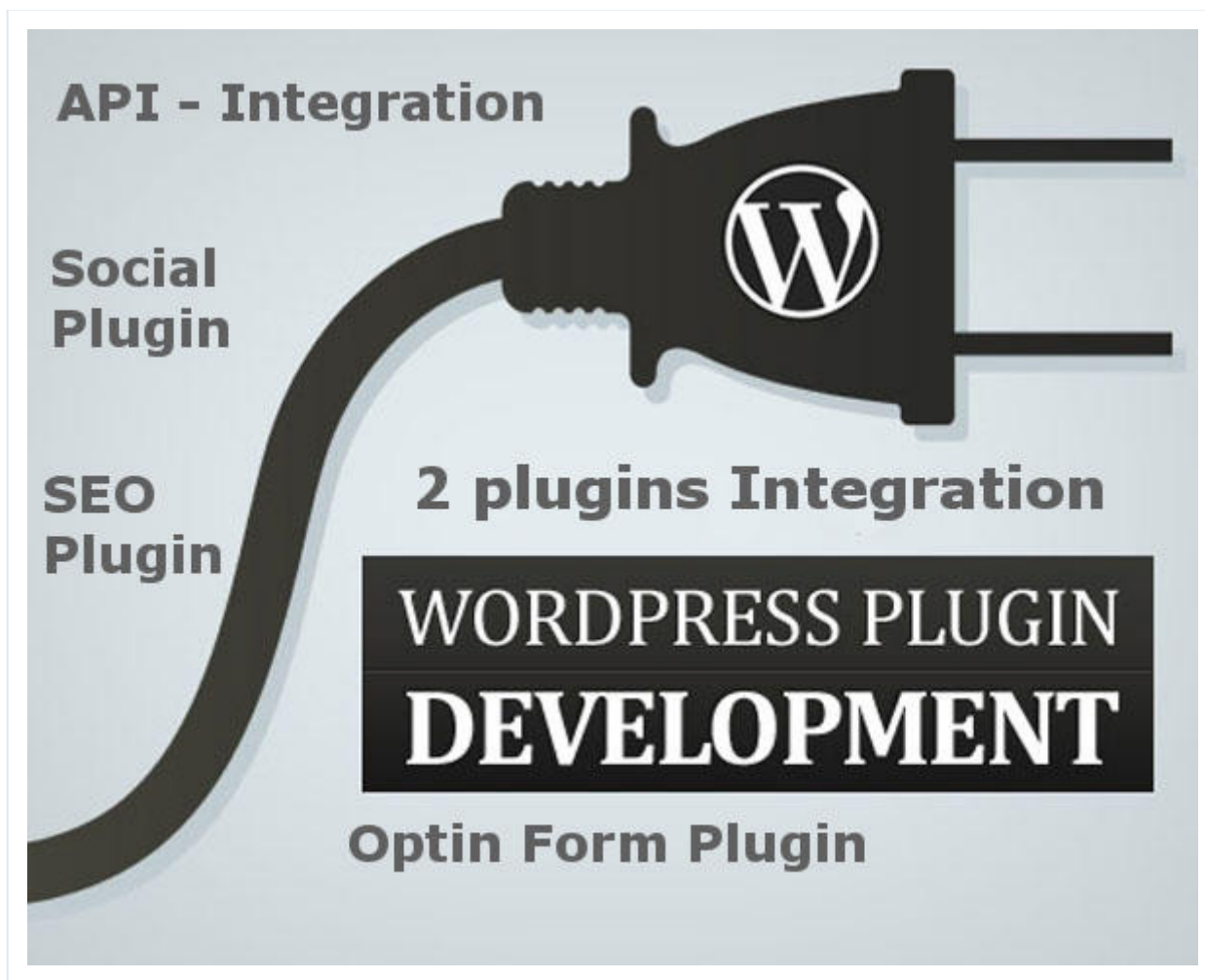
[WordPress Registration Form](#) plugin sur Envato Market

Une autre option est tout simplement pour passer une commande sur Envato Studio. Vous pouvez choisir la bonne personne parmi un large éventail de [développeurs de plugin WordPress](#) expérimentés. Puis, vous envoyez vos instructions et laissez le développeur de créer votre plugin dans les délais convenus.

Par exemple, [Alisaleem252](#) mettra au point un plugin WordPress personnalisé qui sera compatible avec la dernière version de WordPress et d'autres plugins dans le référentiel WordPress selon vos besoins.

Vous obtiendrez :

- widget personnalisé si nécessaire
- shortcode personnalisé si nécessaire
- type de message personnalisé si nécessaire
- service fiable



Développement d'un Plugin WordPress personnalisé

Le service complet coûte seulement \$300, et votre plugin sera prêt en 10 jours. Alisaleem252 a une cote d'approbation de 98 % des clients précédents. Alors pourquoi ne pas essayer son populaire service de [développement d'un Plugin WordPress personnalisé](#) !

## Construire le plugin

Cela dit, nous allons commencer avec le codage du plugin. Tout d'abord, inclure l'en-tête du plugin.

```
1 <?php
2 /*
3 Plugin Name: Custom Registration
4 Plugin URI: http://code.tutsplus.com
5 Description: Updates user rating based on number of posts.
6 Version: 1.0
7 Author: Agbonghama Collins
8 Author URI: http://tech4sky.com
9 */
```

Ensuite, nous créons une fonction PHP qui contient le code HTML du formulaire d'inscription.

```
01 function registration_form( $username, $password, $email, $website, $first_na
02     echo '
03     <style>
04     div {
05         margin-bottom:2px;
06     }
07
08     input{
09         margin-bottom:4px;
10     }
11     </style>
12     ';
13
14     echo '
15     <form action="' . $_SERVER['REQUEST_URI'] . '" method="post">
16     <div>
17     <label for="username">Username <strong>*</strong></label>
18     <input type="text" name="username" value="' . ( isset( $_POST['username']
19     </div>
20
21     <div>
22     <label for="password">Password <strong>*</strong></label>
23     <input type="password" name="password" value="' . ( isset( $_POST['passwo
24     </div>
25
26     <div>
27     <label for="email">Email <strong>*</strong></label>
28     <input type="text" name="email" value="' . ( isset( $_POST['email']) ? $e
29     </div>
30
31     <div>
32     <label for="website">Website</label>
33     <input type="text" name="website" value="' . ( isset( $_POST['website'])
34     </div>
35
36     <div>
37     <label for="firstname">First Name</label>
38     <input type="text" name="fname" value="' . ( isset( $_POST['fname']) ? $f
39     </div>
40
41     <div>
42     <label for="website">Last Name</label>
43     <input type="text" name="lname" value="' . ( isset( $_POST['lname']) ? $l
44     </div>
45
46     <div>
47     <label for="nickname">Nickname</label>
48     <input type="text" name="nickname" value="' . ( isset( $_POST['nickname']
49
```

```

50     </div>
51
52     <div>
53     <label for="bio">About / Bio</label>
54     <textarea name="bio">' . ( isset( $_POST['bio'] ) ? $bio : null ) . '</tex
55     </div>
56     <input type="submit" name="submit" value="Register"/>
57     </form>
58     ';
}

```

Notez le champ d'enregistrement étant passé à la fonction ci-dessus comme variable ? Dans le code de la fonction, vous verrez les instances du code suivant, par exemple :

```
( isset( $_POST['lname'] ) ? $last_name : null )
```

L'opérateur ternaire vérifie le contenu du tableau global `$_POST` pour voir si le formulaire contient une valeur. S'il ne contient pas une valeur, il remplit les champs du formulaire avec la valeur afin d'éviter à l'utilisateur de rentrer à nouveau dans le champ input.

Un formulaire d'inscription n'est jamais complet jusqu'à ce que vous ne validez et assainissez l'entrée d'utilisateur. En conséquence, nous allons créer une fonction de validation avec le nom `registration_validation`.

Pour rendre plus facile l'effort de validation, nous allons utiliser la classe [WordPress WP\\_Error](#). Suivez-moi dans le codage de la fonction de validation :

1. Créez la fonction et passez le champ de l'enregistrement comme l'argument de la fonction.

```
1 | function registration_validation( $username, $password, $email, $website,
```

2. Instanciez la classe `WP_Error` et rendez la variable d'instance globale pour qu'il puisse être accédé hors de la portée de la fonction.

```

1 | global $reg_errors;
2 | $reg_errors = new WP_Error;

```

3. N'oubliez pas : Nous avons dit que le nom d'utilisateur, le mot de passe et l'e-mail sont requis et ne doivent pas être laissés. Pour appliquer la règle, nous devons

vérifier si tout le champ est vide. Si il est vide, nous ajoutons le message d'erreur à la classe globale `WP_Error`.

```
1 | if ( empty( $username ) || empty( $password ) || empty( $email ) ) {  
2 |     $reg_errors->add( 'field', 'Required form field is missing' );  
3 | }
```

4. Nous vérifions également pour nous assurer que le nombre de caractères du nom d'utilisateur n'est pas inférieur à 4.

```
1 | if ( 4 > strlen( $username ) ) {  
2 |     $reg_errors->add( 'username_length', 'Username too short. At least 4 ' );  
3 | }
```

5. Vérifiez si le nom d'utilisateur est déjà inscrit.

```
1 | if ( username_exists( $username ) )  
2 |     $reg_errors->add( 'user_name', 'Sorry, that username already exists!' )
```

6. Recourez aux services de la fonction WordPress `validate_username` pour vous assurer que le nom d'utilisateur est valide.

```
1 | if ( ! validate_username( $username ) ) {  
2 |     $reg_errors->add( 'username_invalid', 'Sorry, the username you entered is invalid' );  
3 | }
```

7. Assurez-vous que le mot de passe saisi par les internautes n'est pas inférieur à 5 caractères.

```
1 | if ( 5 > strlen( $password ) ) {  
2 |     $reg_errors->add( 'password', 'Password length must be greater than 5 characters' );  
3 | }
```

8. Vérifiez si l'e-mail est une adresse e-mail valide.

```
1 | if ( !is_email( $email ) ) {  
2 |     $reg_errors->add( 'email_invalid', 'Email is not valid' );  
3 | }
```

9. Vérifier si l'e-mail est déjà enregistré.

```
1 | if ( email_exists( $email ) ) {  
2 |     $reg_errors->add( 'email', 'Email Already in use' );  
3 | }
```



```
}
```

10. Si le champ site web est complété, vérifiez si il est valide.

```
1  if ( ! empty( $website ) ) {
2      if ( ! filter_var( $website, FILTER_VALIDATE_URL ) ) {
3          $reg_errors->add( 'website', 'Website is not a valid URL' );
4      }
5  }
```

11. Enfin, effectuez une boucle dans les erreurs de notre instance `WP_Error` et affichez l'erreur individuelle.

```
01  if ( is_wp_error( $reg_errors ) ) {
02
03      foreach ( $reg_errors->get_error_messages() as $error ) {
04
05          echo '<div>';
06          echo '<strong>ERROR</strong>:';
07          echo $error . '<br/>';
08          echo '</div>';
09
10      }
11
12 }
```

Nous avons terminé de coder la fonction de validation.

La prochaine est la fonction du plugin `complete_registration()` qui gère l'enregistrement de l'utilisateur.

L'enregistrement de l'utilisateur est fait par la fonction `wp_insert_user` qui accepte un tableau de données de l'utilisateur.

```
01  function complete_registration() {
02      global $reg_errors, $username, $password, $email, $website, $first_name,
03      if ( 1 > count( $reg_errors->get_error_messages() ) ) {
04          $userdata = array(
05              'user_login'    => $username,
06              'user_email'    => $email,
07              'user_pass'     => $password,
08              'user_url'      => $website,
09              'first_name'    => $first_name,
10              'last_name'     => $last_name,
11              'nickname'     => $nickname,
12              'description'   => $bio,
13          );
14          $user = wp_insert_user( $userdata );
15          echo 'Registration complete. Goto <a href="' . get_site_url() . '/wp-
```

```

16 |     }
17 | }

```

Dans la fonction `complete_registration()` qui précède, nous fait l'instance de `WP_Error` `$reg_errors` et la variable globale de champs de formulaire afin d'accéder à la variable dans la portée globale.

Ensuite, nous vérifions si l'instance de gestion des erreurs `$reg_errors` contient une erreur. Si aucune erreur n'est trouvée, nous procédons au remplissage du tableau `$userdata` et insérons les détails d'enregistrement d'utilisateur à la base de données WordPress et affichons un message *Inscription Complète* avec un lien vers la page de connexion.

La prochaine est la super fonction `custom_registration_function()` qui met toutes les fonctions que nous avons créé précédemment en usage.

```

01 | function custom_registration_function() {
02 |     if ( isset($_POST['submit']) ) {
03 |         registration_validation(
04 |             $_POST['username'],
05 |             $_POST['password'],
06 |             $_POST['email'],
07 |             $_POST['website'],
08 |             $_POST['fname'],
09 |             $_POST['lname'],
10 |             $_POST['nickname'],
11 |             $_POST['bio']
12 |         );
13 |
14 |         // sanitize user form input
15 |         global $username, $password, $email, $website, $first_name, $last_name;
16 |         $username = sanitize_user( $_POST['username'] );
17 |         $password = esc_attr( $_POST['password'] );
18 |         $email = sanitize_email( $_POST['email'] );
19 |         $website = esc_url( $_POST['website'] );
20 |         $first_name = sanitize_text_field( $_POST['fname'] );
21 |         $last_name = sanitize_text_field( $_POST['lname'] );
22 |         $nickname = sanitize_text_field( $_POST['nickname'] );
23 |         $bio = esc_textarea( $_POST['bio'] );
24 |
25 |         // call @function complete_registration to create the user
26 |         // only when no WP_error is found
27 |         complete_registration(
28 |             $username,
29 |             $password,
30 |             $email,
31 |             $website,
32 |             $first_name,

```

```

33         $last_name,
34         $nickname,
35         $bio
36     );
37 }
38
39     registration_form(
40         $username,
41         $password,
42         $email,
43         $website,
44         $first_name,
45         $last_name,
46         $nickname,
47         $bio
48     );
49 }

```

Laissez-moi vous expliquer le code dans la fonction `custom_registration_function()`.

Tout d'abord, nous déterminons si le formulaire a été envoyé en vérifiant si le `$_POST` [`« submit »`] est défini. Si le formulaire a été envoyé, nous utilisons la fonction `registration_validation` pour valider le formulaire d'utilisateur soumis.

Nous expurgeons ensuite les données du formulaire et définissons les données d'expurgation à une variable nommée d'après le champ de formulaire. Enfin, nous appelons le `complete_registration` pour inscrire l'utilisateur.

Nous avons besoin d'appeler la fonction `registration_form` pour afficher le formulaire d'inscription.

N'oubliez pas que je l'ai mentionné que nous allons donner le shortcode support pour le plugin d'enregistrement ? Voici le code du shortcode support .

```

1  // Register a new shortcode: [cr_custom_registration]
2  add_shortcode( 'cr_custom_registration', 'custom_registration_shortcode' );
3
4  // The callback function that will replace [book]
5  function custom_registration_shortcode() {
6      ob_start();
7      custom_registration_function();
8      return ob_get_clean();
9  }

```

À ce stade, nous avons terminé le codage du plugin. Voici une image montrant à quoi le formulaire d'inscription devrait ressembler.

Notez que vous pourriez ne pas obtenir le même aspect sur votre site WordPress en raison de la variation de style CSS.

### Custom Registration Form

Username

Password

Email

Website

First Name

Last Name

Nickname

About / Bio

REGISTER

Advertisement

## Utilisation du plugin

Pour implémenter le plugin dans un post WordPress ou une page, utilisez le shortcode

```
[cr_custom_registration]
```

Pour implémenter le formulaire d'inscription dans un aspect précis de votre thème, ajoutez la balise suivante du modèle - `<?php custom_registration_function(); ?>`.

Vous pouvez obtenir le fichier du plugin dans la pièce jointe de cet article.

## Résumé

Dans cet article, nous sommes passé à travers le processus de création d'un plugin qui ajoute un formulaire d'inscription personnalisé à WordPress. Vous pouvez étendre davantage le formulaire d'inscription pour inclure des champs supplémentaires tels que le rôle d'utilisateur, le compte AOL IM, mais assurez-vous que le champ de formulaire soit une métadonnées valide pour `wp_insert_user`.

Si vous avez des questions et des suggestions, faites-le moi savoir dans les commentaires !

N'oubliez pas que si vous avez eu du mal à suivre ce tutoriel et que vous souhaitez une solution simple, vous pouvez essayer le plugin [WordPress Registration Form](#) sur Envato Market.

Advertisement



## Collins Agbonghama

Web Developer, Writer, Geek. Developer of the popular ProfilePress - Best WordPress user registration, login & profile plugin as well as MailOptin - Ultimate Lead Generation, Email Automation & Newsletter Plugin.

 [w3guy](#)

 FEED  LIKE  FOLLOW

### Weekly email summary

Subscribe below and we'll send you a weekly email summary of all new Code tutorials. Never miss out on learning about the next big thing.

Update me weekly

Advertisement

Download Attachment

### Translations

Envato Tuts+ tutorials are translated into other languages by our community members—you can be involved too!

Translate this post

Powered by



# native

0 Commentaires

Tuts+ Hub

 S'identifier ▾

 Recommander

 Tweet

 Partager

Les meilleurs ▾





Commencer la discussion...

S'IDENTIFIER AVEC

OU INSCRIVEZ-VOUS SUR DISQUS 

Nom

Soyez le premier à commenter.

 S'abonner  Ajoutez Disqus à votre site web !Ajouter DisqusAjouter



Advertisement

**QUICK LINKS** - Explore popular categories

---

ENVATO TUTORIALS



---

JOIN OUR COMMUNITY



---

HELP



tuts+

28,163  
Tutorials

1,263  
Courses

39,815  
Translations

---

[Envato.com](#) [Our products](#) [Careers](#) [Sitemap](#)

© 2019 Envato Pty Ltd. Trademarks and brands are the property of their respective owners.

Follow Envato Tuts+     
[Facebook](#) [Twitter](#) [Pinterest](#)