

Personnalisation d'une zone de commentaires

CSS, Elementor, Wordpress

Web Compétences : 2

Un site web se doit d'être dynamique. Dans le cas d'un blog, chaque publication s'affiche sur une page différente dont la mise en page est identique. Pour se faciliter la vie, Elementor a mis à disposition un modèle qui sert de base à la création des publications, la publication unique. Et quand le contenu s'affiche publiquement, il est intéressant de pouvoir donner la possibilité aux visiteurs d'y laisser un commentaire.

Publié samedi 18 janvier 2020, 12h38
Modifié lundi 26 août 2024, 10h06



By Olivier Paudex

Introduction

A l'instar de la page d'archives, la page de la publication unique est également un modèle proposé par Elementor et fonctionne de la même manière. On peut y créer des sections, y placer différents widgets dont le contenu à afficher. Il est aussi possible d'y ajouter une fenêtre de commentaires.

Les deux principaux widgets

Ce chapitre ne va pas décrire en détails la réalisation de ma page de publication, mais plutôt de la création et de la personnalisation d'une zone de commentaires où le visiteur peut laisser un mot d'encouragement,

une critique, relever un point, etc ...



Le widget qui affiche le contenu de la publication



Le widget qui affiche la zone de commentaires

Les deux widgets ci-dessus qui composent la page de publication sont les deux seuls de toute la panoplie mise à disposition par Elementor à ne pas disposer de l'onglet "**style**". En effet, le contenu afficher par ces éléments se reposent sur le thème installé. Dans ce site web, il s'agit du thème **'Hello"** fournit par Elementor, et pour être plus exact, du **thème enfant**, comme parlé dans la publication sur ce sujet.

Les feuilles de styles et autres paramètres

Dans le fichier "**functions.php**", une référence au fichier PHP **"Enqueue_CSS_Styles.php"** pour charger les feuilles de styles en queue, va être ajoutée. Puis le fichier sera créé dans le dossier**"php"**.

```
// Othersinclude_once ($dir . '/php/Enqueue_CSS_Styles.php');
```

Mettre la feuille de style pour la zone de commentaires en queue

Si vous avez suivi mon blog, le chapitre concernant le thème enfant faisait mention d'une feuille de style dont le nom était "**Main_Styles.css**". Cette fois-ci, la feuille de style va être baptisé "**Comments_Styles.css** ".

A la différence du fichier "**Main_Styles.css**", celui des commentaires ne va s'exécuter que sur les pages de publications. On peut paramétrer ceci en ajoutant un test de type '**is_single()**".

De même, on désire que cette feuille de style ne s'exécute que sur les pages du site et non pas sur la console d'administration. On va ajouter un test de type "**!is_admin()**", le point d'exclamation représentant la négation. De plus, l'action à effectuer ne s'applique que sur les pages du sites. On va uniquement ajouter

l'action " **wp_enqueue_scripts**".

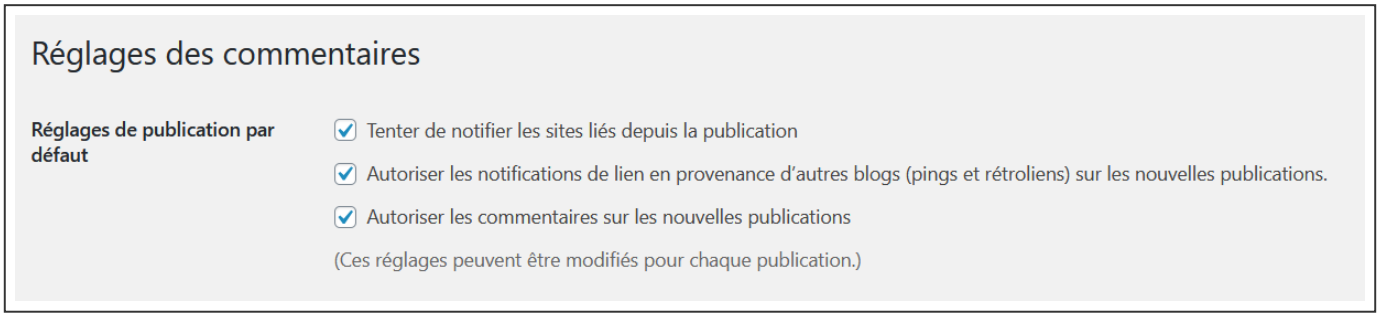
- Créez ce fichier et placez le dans le dossier "**css**", puis complétez le fichier " **Enqueue_CSS_Styles.php**".

```
<?php
  /**
  * Plugin Name: Enqueue_CSS_Styles
  * Description: Enqueue CSS Styles
  * Author: Olivier Paudex
  * Author Web Site: https://www.fuyens.ch
  */
  // Enqueue CSS Styles
  function Main_Styles() {
    wp_enqueue_style('Main_Styles', get_stylesheet_directory_uri() . '/css/Main_Styles.css', fa
  }
  add_action('wp_enqueue_scripts', 'Main_Styles');
  add_action('admin_enqueue_scripts', 'Main_Styles');
  /*****************************************************************************
  // Enqueue Comments CSS Styles
  function Comments_Styles() {
    if (!is_admin() && (is_single())) {
      wp_enqueue_style('Comments_Styles', get_stylesheet_directory_uri() . '/css/Comments_Style
    }
  }
  add_action('wp_enqueue_scripts', 'Comments_Styles');?>
```

Réglages de WordPress

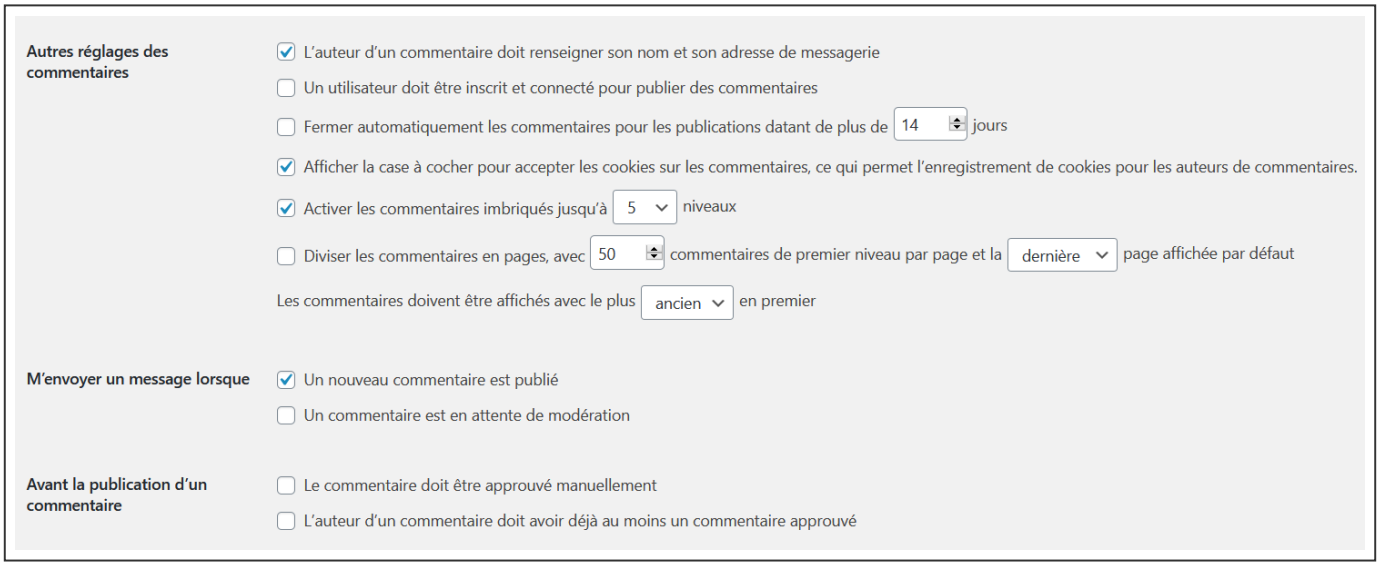
La première tâche à effectuer est situé dans les réglages de WordPress. Il faut en effet autoriser les visiteurs à pouvoir déposer un commentaire.

- Dans les **"Réglages – Commentaires"**, cochez les 3 premières options.



Les réglages de base pour autoriser les commentaires

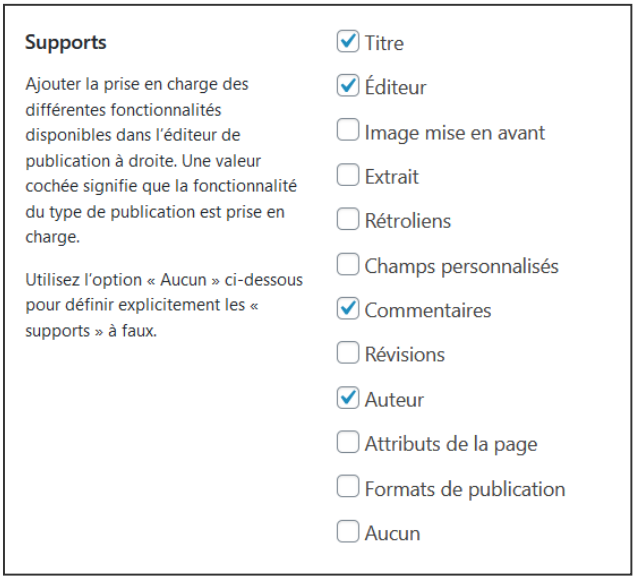
- Puis, cochez les autres options comme bon vous semble. Je vous conseille celles-ci



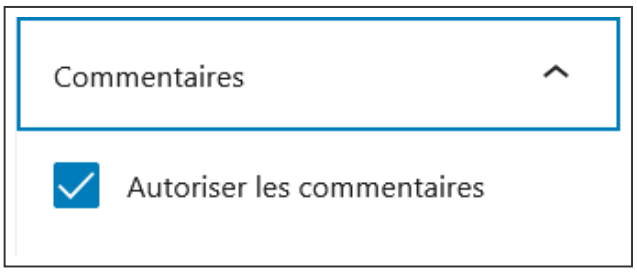
Les réglages avancés pour les commentaires

Réglages du plugin CPT UI

Dans les réglages de chaque type de publications personnalisées, cochez l'option "**Commentaires**". Ceci permet d'ouvrir ou de fermer les commentaires pour chaque publication.



Les options de support dans CPT UI



L'option commentaires dans chaque publication

Personnalisation de la zone des commentaires

Le widget Elementor des commentaires de la publication affiche par défaut, sur une page de publication, cette conception.



La zone de commentaire par défaut

Rendre une feuille de style réactive aux différents écrans

Avant de commencer à modifier l'aspect de la zone de commentaires, il est bien de garder à l'esprit que celui-ci doit également répondre aux différentes tailles d'écrans. Pour ceci, il est possible d'utiliser une fonction CSS qui se nomme **"@media only screen and ..."**.

Pour garder les spécifications d'Elementor, il faut rendre le site web réactif sur trois tailles d'écrans différentes, à savoir l'ordinateur de bureau, la tablette et le mobile. Pour chaque taille, on peut configurer les dimensions dans les réglages d'Elementor.

Ce site utilise les dimensions ci-dessous qui sont standard :

- Ordinateur de bureau : Taille minimum de 1024 pixels
- Tablette : Taille comprise entre 768 pixels et 1023 pixels
- Mobile : Taille maximum de 767 pixels

Voici un exemple standard pour rendre une feuille de styles CSS adaptive aux tailles d'écrans.

```
/* Desktop */
@media only screen and (min-width: 1024px) {
  :root {
    --font_size: 18px;
    --flex_direction: row;
  }
}
/* Tablet */
@media only screen and (min-width: 768px) and (max-width: 1023px) {
  :root {
    --font_size: 17px;
    --flex_direction: row;
  }
}
/* Smartphone */
@media only screen and (max-width: 767px) {
  :root {
    --font_size: 16px;
    --flex_direction: column;
  }}
```

Et voici comment utiliser dynamiquement une variable dans une classe ou un ID CSS. Ci dessous, les deux classes vont utiliser comme taille de police, la variable **"font_size"** et s'adapter automatiquement en fonction de la taille de l'écran.

```
/* Normal font size */
.logged-in-as a, .comment-form-comment {
  font-size: var(--font_size);}
```

Les principales classes et ID pour modifier le style de la zone de commentaires

Voici quelques unes des nombreuses classes qui permettent de personnaliser cette zone. N'oubliez pas les points ou les dièses devant le nom des classes ou des identifiants.

- **.comment-reply-title**, permet de modifier le titre "**Laissez un commentaire**".
- **.title-comments,** permet de modifier le titre du nombre de réponses.
- **#cancel-comment-reply-link**, permet de modifier le titre "**Annuler la réponse**".
- **.comment-reply-link**, permet de modifier le style du lien "**Répondre**".
- **.reply**, permet de positionner le lien "**Répondre**".

- **#submit**, permet de modifier le style du bouton "**Laissez un commentaire**" et son libellé.
- **.form-submit**, permet de positionner le bouton "**Laissez un commentaire**".
- **.comment-form**, permet de modifier la zone de saisie du formulaire dans son ensemble.
- **#author, #email, #url**, permettent de modifier le contenu des zones du formulaires.
- **.logged-in-as a**, permet de modifier le lien "**Connecté en tant que Olivier Paudex. Déconnexion ?** ", si l'utilisateur est connecté à votre site.
- **.comment-policy (comment-notes)**, permet de modifier le libellé "**Votre adresse email ne sera pas publiée. Les champs obligatoires sont marqués d'une *:**", si l'utilisateur n'est pas connecté à votre site.
- **.comment-form-author label, .comment-form-email label, .comment-form-comment label, .comment-form-url label**, permettent de modifier les différents libellés de la zone de saisie.
- **.says**, permet de modifier le mot "dit", à côté du nom de l'auteur.
- **.comment-content**, permet de modifier la zone de contenu du commentaire.
- **#comment, .comment-body, #author, #email, #url** permettent de modifier les différentes zones de saisie du commentaire.
- **.comment-meta**, permet de gérer le bloc "**auteur, la date et l'heure du commentaire et les liens de gestion**".
- **.fn et .fn a**, permettent de modifier respectivement le nom de l'auteur d'un commentaire et le nom de l'auteur connecté.
- **.comment-metadata a**, permet de modifier la date et l'heure du commentaire uniquement.
- **.edit-link a**, permet de modifier les liens "**spam, effacer, modifier**", à côté de la date et de l'heure uniquement.
- **.avatar**, permet de modifier la photo de l'auteur.
- **span.required** permet de modifier l'étoile indiquant que le champ est obligatoire.

Ci-dessous, un extrait de la feuille de style permettant de modifier le bouton.

```
/* Buttons */
#submit, .comment-reply-link {
  background-color: #2E8BC0;
  border-style: none;
  color: #FCFCFC;
  font-family: 'Open Sans', sans-serif;
  font-weight: 400;
  letter-spacing: -0.5px;
  line-height: 1.2em;
  border-radius: 5px;
  padding: 10px 20px 10px 20px;
}
/* Buttons hover */
#submit:hover, .comment-reply-link:hover {
  background-color: #145DA0;
  color: #FCFCFC;
}
```

```
/* Submit button */
.form-submit {
  display: flex;
  justify-content: center;
  margin: 20px 0px 20px 0px;}
```

Ci-dessous, la zone des commentaires modifiées selon la feuille de style.



La zone de commentaires modifiée

Le plugin “Simple Comment Editing”

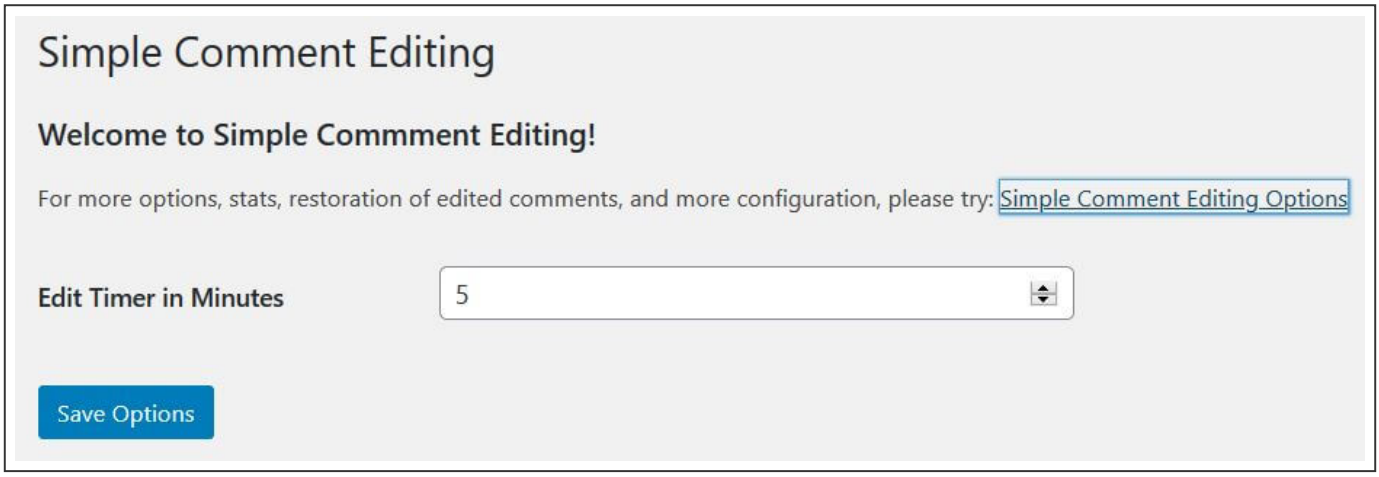
Une fois le commentaire envoyé pour publication, celui-ci ne peut-être modifié et c’est bien dommage. Ne serait-ce que pour corriger une faute d’orthographe ou de syntaxe grammaticale, l’édition d’un commentaire devrait être possible. Et pourtant, WordPress ne l’a pas prévu. Mais c’est sans compter l’énorme contribution d’auteurs indépendants, qui nous permettent d’ajouter de telles fonctionnalités grâce à des plugins, que WordPress est devenu tellement populaire.

En ce qui concerne la modification des commentaires, il existe un plugin baptisé ‘**Simple Comment Editing**“.



Le plugin “Simple Comment Editing”

Le plugin est très simple d’emploi. Il n’a qu’un seul paramètre, **le minuteur de modification (edit timer)**, qui prend une valeur en minutes.



Les paramètres du plugin

Personnaliser le plugin avec une feuille de style

Comme pour le formulaire de commentaires, il est également possible de modifier le style de ce plugin.

- **button.sce-comment-save, button.sce-comment-cancel, button.sce-comment-delete**, permettent de modifier les trois boutons de validation, annulation et suppression.
- **.sce-comment-edit-buttons**, permet de positionner les boutons.
- **.sce-edit-button a**, permet de modifier le lien "**Cliquez pour modifier**".
- **span.sce-timer, span.sce-seperator**, permettent de modifier le séparateur et la minuterie.
- **.sce-textarea**, permet de modifier la zone des boutons.
- **.sce-comment-text**, permet de modifier la zone de modification du commentaire.

Autres fonctionnalités

La zone des commentaires ne s'arrêtent pas uniquement à sa conception. Il est également possible d'ajuster d'autres éléments. Pour ceci, un autre fichier de type PHP doit être ajouté. Cette fois, il va se nommer tout simplement "**Comments.php**". Il se trouve dans le dossier "**php**".

Filtre "comment_form_defaults"

Il est possible d'ajouter une note avant ou après la zone de saisie du commentaire grâce au filtre "**comment_form_defaults**".

```
function comment_policy_text($args) {
  // Text before
  $args['comment_notes_before'] = "<p class='comment-policy'>text avant</p>";
  // Text after
  $args['comment_notes_after'] = '';
  //$args['comment_notes_after'] = "<p class='comment-policy'>text après</p>";
  // Title by default
  $args['title_reply'] = "<p class='comment-title'>Laissez un commentaire</p>";

  return $args;
}add_filter('comment_form_defaults', 'comment_policy_text');
```

Filtre "comment_form_default_fields"

Il est possible de modifier certains champs par défaut, avec le filtre "**comment_form_default_fields**". Ci-

dessous, un exemple de suppression du champ URL.

```
// Remove comment field URL
function remove_comment_url($arg) {
  $arg['url'] = '';
  return $arg;
}add_filter('comment_form_default_fields', 'remove_comment_url');
```

Filtre "edit_comment_link"

Il est possible d'ajouter, pour l'adminstrateur uniquement, des liens de gestion **"Spam, Delete, Edit"**, avec le filtre **"edit_comment_link"**.

```
// Add Spam, Delete, Update links to comments
function comment_links($link,$id) {

  // Link
  $template = '<a class="comment-edit-link" href="%1$s%2$s">%3$s</a>';
  $admin_url = admin_url("comment.php?c=$id&action=");
  // Spam.
  $link = sprintf($template, $admin_url, 'cdc&dt=spam', '[Pourriel]');

  // Delete.
  $link .= sprintf($template, $admin_url, 'cdc', '[Supprimer]');
  // Update
  $link .= sprintf($template, $admin_url, 'editcomment', '[Editer]');

  return $link;
}add_filter('edit_comment_link','comment_links',10,2);
```

Filtre "comment_reply_link"

Il est possible d'ajouter le prénom de la personne qui laisse un commentaire dans le lien de réponse, avec le filtre **"comment_reply_link"**

```
// Add first name to reply link and create a button
function reply_links($link,$args,$comment) {
  $comment = get_comment($comment);

  // If no comment author is blank, use 'Anonymous'
  if (empty($comment->comment_author)) {
      if (!empty($comment->user_id)){
          $user=get_userdata($comment->user_id);
          $author=$user->user_login;
      } else {
          $author = 'Anonymous';
      }
  } else {
      $author = $comment->comment_author;
```

```
    }

    // If the user provided more than a first name, use only first name
    if (strpos($author, ' ')) {
        $author = substr($author, 0, strpos($author, ' '));
    }
    // Add a div to the structure of the reply link
    $template = '<div class="reply-button">';
    // Replace Reply Link with "Reply to Author First Name"
    $reply_link_text = $args['reply_text'];
    $link = str_replace($reply_link_text, 'Answer to' . ' ' . $author, $link);
    // Close the div structure
    $link = $template . $link . '</div>';
    return $link;
}add_filter('comment_reply_link','reply_links',10,3);
```

Déplacer les commentaires en dessous de la zone de saisie

Par défaut, la zone de saisie se trouve en dessous des commentaires. Il est possible de déplacer celle-ci au-dessus des commentaires en suivant cette simple astuce.

Dans le thème "**Hello"**, Elementor a créé un modèle qui porte le nom de "**comments.php**". Attention à ne pas confondre avec notre fichier de configuration qui se trouve dans le dossier "**php**" du thème enfant.

- Copiez ce fichier du thème **"Hello"** vers le thème enfant, à la racine, au même endroit que se trouve le fichier "**functions.php**".
- Déplacez la portion de code au bon endroit et sauvegardez le fichier. Elle doit se trouver juste en dessous de la portion "**Comment Reply Script**".

```
<?php
/**
 * The template for displaying comments.
 *
 * This is the template that displays the area of the page that contains both the current comments
 * and the comment form.
 *
 * @package HelloElementor
 */
if ( ! defined( 'ABSPATH' ) ) {
 exit; // Exit if accessed directly.
}
// Comment Reply Script.
if ( comments_open() && get_option( 'thread_comments' ) ) {
 wp_enqueue_script( 'comment-reply' );
}
?>
<!-- Comment Form moved above comments -->
<?php
comment_form(
 array(
    'title_reply_before' => '<h2 id="reply-title" class="comment-reply-title">',
  'title_reply_after'  => '</h2>',
 )
);?>
```

Le mot de la fin

Voici le chapitre sur la zone de commentaires terminées. Tous les outils sont réunis pour créer de belles pages de publications. On peut apercevoir ici, la puissance de la feuille de style CSS qui est ici indispensable à la mise en place d'une section commentaires. Encore une très bonne raison de créer un thème enfant pour venir y ajouter son propre code. Le prochain chapitre va nous conduire vers**l'impression d'un PDF**.