

## キーワードタグを付けるか付けないか、それが問題だ

<https://magazine.joomla.org/all-issues/december-2023/keyword-or-not-keyword-tag-that-is-the-question>



2023年12月20日

最近、友人が某有名ソーシャルメディアに特異な質問を投稿した：  
「あなたはまだ記事のキーワードフィールドを使用していますか？」

何十年も前から、キーワードメタ・タグの内容は検索エンジンによってランキング要素として使われなくなっている。もちろん、コンテンツの可視性に影響がなくても、まだ使うことはできますが、他にも賢い使用例があるでしょう。

Joomla はコンテンツ管理に関しては非常に強力です。だからこそ、私たちはできるだけコアにこだわるべきなのです。私は、私たちの愛する CMS のネイティブ機能のみを使用して問題を解決するための、異なる、または新しい可能性にいつも驚かされます。

今回は、キーワードフィールドの興味深い使用例を紹介しましょう。

### コアにこだわるメリットとは

**あなたのコンテンツは com\_content によって提供されます。**

簡単に管理でき、完全にコントロールでき、さらに重要なのは、それを表示するための多くのオプションがあるということです。

### エクステンションのインストールとメンテナンスは不要

エクステンションのインストールやメンテナンスは必要ありません。学習曲線に耐える必要はありません。重たい CCK や他のページビルダーは忘れてください。ほとんどの場合、Joomla だけで望む結果を得ることができるからです。

## コアで Joomla のスキルを向上させること

先に述べたように、Joomla の ネイティブ機能のおかげで、クリーンなソリューションが見つげられることにいつも驚かされます。キーワードフィールドのこの使用例のおかげで、Joomla について何か 新しく便利なことを学ぶことができますと思います。

## 出発点は？

フロントエンドにコンテンツを表示する新しい機能を追加するとき、多くの人は 反射的にエクステンションを追加します。

もちろん、他の選択肢がない場合もありますが、これは私たちが直面しているすべての問題の体系的な答えではありません。

ご存知かもしれませんが、Joomla には非常に多くのコア機能が搭載されており、おそらく 解決策はすでにあなたのサイトにあります。

このような状況では、Joomla は非常に多くの機能を提供しているため、私の最初の 反応は常にコアソリューションに目を向けることです。

そして、私はこの健全な態度を適応することを強くお勧めします。

キーワードフィールドに戻りましょう。

このフィールドの内容は、Joomla では主に 2 つのシナリオで使用されます：

- ページのソースコードに表示される **meta キーワードタグ**に追加されます。
- 「**関連モジュール**」で使われます。

このモジュールにあまり馴染みがないなら、ここでキチンと説明をしましょう：

「このモジュールは 閲覧中の記事に関連する他の記事を表示します。これらの関係はキーワードによって確立されます。

現在の記事の全てのキーワードは、他の全ての公開された記事の全てのキーワードが検索されます。例えば、

"オウムの繁殖"に関する記事と、"黒オウムの手乗り飼育"に関する記事があるとします。もし両方の記事に「オウム」というキーワードが含まれていれば、関連項目モジュールは「黒オウムの手による飼育」を閲覧する際に「オウムの飼育」の記事をリストアップし、その逆も同様です。」

検索エンジンはこのタグの内容をインデックス要素として 使用しなくなったため、私たちの多くはこの分野を放棄しました。

私の意識では、この興味深いコア機能を葬り去るのは時期尚早です。

## また、キーワードフィールドが再び役に立つのはどうだろう？

**mod\_related\_items** モジュールの説明に戻りましょう：このモジュールは、現在読まれている記事に関連する 記事のリストを表示します。関連するとは、その記事は 同じキーワードを持つことです。

さて、あなたのウェブサイトにも複数の著者がいるとしましょう。それは、読者にとっても便利だし、多くの著作者の下で、それらの記事が表示される著作者にとっても嬉しいことでしょう。

別のエクステンションをインストールするよりも、キーワードフィールドを使うだけで どのように実現するかを紹介しましょう。

詳細は後述しますが、このトリックのおかげで、たった一つのモジュールで現在の記事と同じ著者の記事を表示することができます。

そう、すべての著者のための一つのコアモジュール。

たった1つのコアモジュールが、すべての著者へのルールなのです。

こんなに軽く、よりシンプルにすることはできません。

## モジュールのオーバーライド

この例では、各項目について、イントロ画像、タイトル、イントロテキストを含む 4つの関連項目をすべての記事の下に表示したいと思います。

デフォルトでは、**mod\_related\_items** モジュールは関連アイテムのタイトルと日付だけを表示します。そこで、希望する結果を得るためにオーバーライドする必要があります。

以下がそのコードです（ご自由にコピーしてお使いください）。

読みやすくするために、コメントを追加しました。

```
<?php
/**
 * @package Joomla.Site
 * @subpackage mod_related_items
 * @author web-eau.net
 * @copyright (C) 2006 Open Source Matters, Inc. <https://www.joomla.org>
 * @license GNU General Public License version 2 or later; see LICENSE.txt
 */

defined('_JEXEC') or die;

use Joomla\CMS\Layout\FileLayout;
use Joomla\CMS\Layout\LayoutHelper;
use Joomla\CMS\Router\Route;
use Joomla\CMS\Uri\Uri;
use Joomla\Component\Fields\Administrator\Helper\FieldsHelper;
use Joomla\Component\Content\Site\Helper\RouteHelper;
use Joomla\CMS\HTML\HTMLHelper;
use Joomla\CMS\Language\Text;

?>
```

```

<div class="pb-5 col-sm-12">

    <div class="row row-cols-1 row-cols-md-4 g-4 relateditems<?php echo $moduleclass_sfx; ?>">

        <?php foreach ($list as $item) : ?>

            <!-- calling the intro image -->
            <?php
                $article_images = json_decode($item->images);
                $article_image = "";
                $article_image_alt = "";
                if(isset($article_images->image_intro) && !empty($article_images->image_intro)) {
                    $article_image = $article_images->image_intro;
                    $article_image_alt = $article_images->image_intro_alt;
                }?>

                <div class="col">
                    <div class="h-100 bg-white">

                        <!-- displaying the intro image with a link on it -->
                        <a href="/<?php echo $item->route; ?>">
                            " >
                        </a>

                        <div class="card-body p-4">

                            <!-- article title with a link on it -->
                            <h3 class="card-title">
                                <a class="mod-articles-category-title <?php
echo $item->active; ?>" href="/<?php echo $item->route; ?>">
                                    <?php echo $item->title; ?>
                                </a>
                            </h3>

                            <!-- introtext truncated at 300 characters -->
                            <small><?php echo JHTML::_('string.truncate', $item->introtext, 300, false,
false) ; ?></small>

                        </div>

                    </div>

                </div>

            <?php endforeach; ?>

```

```

<div class="pb-5 col-sm-12">

    <div class="row row-cols-1 row-cols-md-4 g-4 relateditems<?php echo $moduleclass_sfx; ?>">

        <?php foreach ($list as $item) : ?>

            <!-- calling the intro image -->
            <?php
                $article_images = json_decode($item->images);
                $article_image = "";
                $article_image_alt = "";
                if(isset($article_images->image_intro) && !empty($article_images->image_intro)) {
                    $article_image = $article_images->image_intro;
                    $article_image_alt = $article_images->image_intro_alt;
                }?>

                <div class="col">
                    <div class="h-100 bg-white">

                        <!-- displaying the intro image with a link on it -->
                        <a href="/<?php echo $item->route; ?>">
                            " >
                            </a>

                            <div class="card-body p-4">

                                <!-- article title with a link on it -->
                                <h3 class="card-title">
                                    <a class="mod-articles-category-title
<?php echo $item->active; ?>" href="/<?php echo $item->route; ?>">
                                        <?php echo $item->title; ?>
                                    </a>
                                </h3>

                                <!-- introtext truncated at 300 characters -->
                                <small><?php echo JHTML::_('string.truncate', $item->introtext, 300, false,
false) ; ?></small>

                            </div>

                        </div>

                    </div>

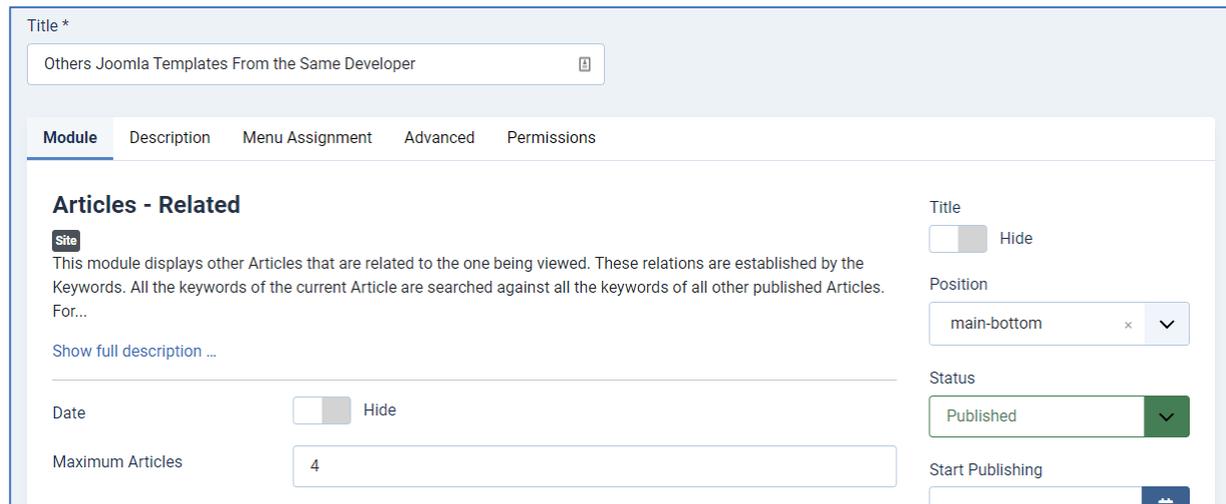
                <?php endforeach; ?>

            </div>

        </div>

```

mod\_relative\_items モジュールの設定は次のとおりです。



The screenshot shows the Joomla! administrator interface for configuring the 'Articles - Related' module. The title is 'Others Joomla Templates From the Same Developer'. The module is set to be published, with a position of 'main-bottom' and a maximum of 4 articles displayed. The status is 'Published' and the start publishing date is set to 'Now'.

### モジュール関連項目の設定

注 - この例では、モジュールは**メインコンポーネント**のすぐ下のテンプレートポジションに公開されます。

## 記事のオーバーライド

一貫性と関連性を持たせるために、これらの関連項目とともに作成者の名前を動的に表示したいと考えています。 これを実現するには、著者の記事の記事オーバーライドを作成します。

次に、次のコードをコピーして、このオーバーライドの下部に貼り付けます。

```
<h3 class="pt-5">
  Others articles from <?php echo $this->item->metakey ; ?>
</h3>
<hr class="my-4">
```

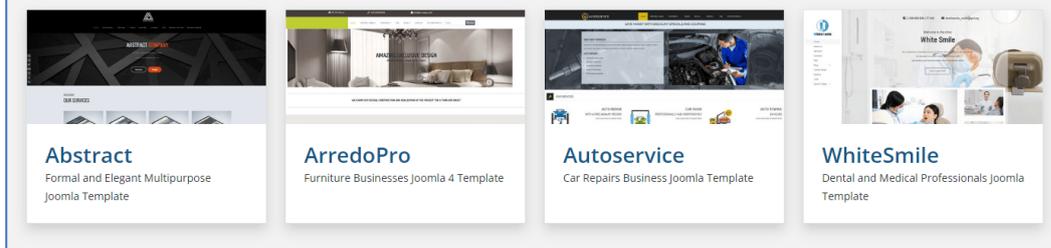
注 - この構文には、キーワード フィールドの内容が表示されます。

```
<?php echo $this->item->metakey ; ?>
```

いつものように、このトリックとこれらのオーバーライドを改善することはいつでも可能ですが、mod\_relative\_items モジュールに関連付けられたキーワード フィールドのこの使用例は、創造的で想像力豊かな人々に興味深い視点を提供すると確信しています。

ここで発生する唯一の制限は、表示される関連項目の順序です。 このモジュールはデフォルトで**最も古い項目**を最初に表示するようです。

## Others Joomla Templates created by Marlev



キーワードフィールドにより表示されるモジュール関連項目のフロントエンド表示の例。

さあ！

キーワードフィールドのおかげで、同じ著者の関連アイテムをいくつか表示できます。  
オーバーライドのおかげで、これらの関連アイテムに希望どおりのデザインを与えることができます。  
mod\_relative\_items モジュールのおかげで、拡張機能を追加する必要はありません。

この 100% Joomla ソリューションのおかげで、将来的にサイトを移行する際に問題に直面することはありません。

これらの Joomla 機能はすべてネイティブで無料です。

唯一の制限はあなたの想像力です。キーワード フィールドをどのように使用するかを下のコメント セクションでお知らせください。

以上

ABOUT THE AUTHOR

Daniel Dubois