



Enclosing shortcodes

Nous avons vu précédemment un « **self-closing shortcode** » `[yoda]`.

L'API des shortcodes supporte également les « **enclosing shortcode** » comme par exemple `[vador]Jean-Marc[/vador]`.

Comme leur nom l'indique, les enclosing shortcodes, vont entourer un contenu et ce dans le but de pouvoir le récupérer. Ainsi le shortcodes `[vador]Jean-Marc[/vador]` retournera :

Jean-Marc, Je suis ton père !

Comment récupérer le contenu

La fonction de rappel (callback) d'un shortcode peut récupérer trois paramètres :

- `$atts` - tableau associatif contenant les attributs du shortcode ou une chaîne vide si aucun attribut passé
- `$content` - **le contenu encapsulé entre la balise d'ouverture et de fermeture**
- `$tag` - le nom du shortcode en cours, utile si on utilise la même fonction de rappel avec plusieurs shortcodes

Voici comment implémenter le shortcode `[vador]...[/vador]`

```
/**
 * Shortcode qui retourne le célèbre "Luke, Je sui ton père !" dans un
 * élément blockquote.
 * Le contenu du shortcode sera utilisé pour remplacer 'Luke'
 *
 * Exemples :
 * [vador] => <blockquote>Luke, Je sui ton père !</blockquote>
 * [vador]Serge[/vador] => <blockquote>Serge, Je sui ton père
```



```
!</blockquote>
*/

function mon_plugin_vador_shortcode($atts, $content = "") {
    //Si contenu vide
    if (empty( $content )) {
        $content = 'Luke';
    }
    return "<blockquote>" . $content . ", Je suis ton père !</blockquote>";
}

//Fonction de rappel qui retourne la célèbre citation de maître Yoda
function mon_plugin_yoda_shortcode() {
    return "<blockquote>Que la force soit avec toi jeune padawan
!</blockquote>";
}

//Enregistre les shortcodes du plugin
function mon_plugin_register_shortcode() {
    add_shortcode( 'yoda', 'mon_plugin_yoda_shortcode' );
    add_shortcode( 'vador', 'mon_plugin_vador_shortcode' );
}
add_action( 'init', 'mon_plugin_register_shortcode' );
```

Shortcodes avec attributs

Les shortcodes peuvent aussi avoir des paramètres de configurations appelés « **attributs** » .

Actuellement notre shortcode `[vador]` génère toujours une citation `<blockquote>`. Nous allons ajouter un paramètre `$tag` à ce shortcode pour que l'on aie le choix entre plusieurs éléments HTML.

```
/**
```



```
* Shortcode qui retourne le célèbre "Luke, Je sui ton père !" dans un
élément blockquote.
* Le contenu du shortcode sera utilisé pour remplacer 'Luke'
* L'attribut tag du shortcode permet de remplacer l'élément blockquote par
: p, h1, h2 ou div
*
* Exemples :
* [vador] => <blockquote>Luke, Je sui ton père !</blockquote>
* [vador]Serge[/vador] => <blockquote>Serge, Je sui ton père
!</blockquote>
* [vador tag="div"]Jean-Marc[/vador] => <div>Jean-Marc, Je sui ton père
!</div>
*/

function mon_plugin_vador_shortcode($atts, $content = "") {
    // Tag par défaut
    $tag = 'blockquote';
    // Si $tag valide on le récupère
    if(isset($atts['tag']) AND in_array($atts['tag'],
['p', 'h1', 'h2', 'div'])) {
        $tag = $atts['tag'];
    }
    // Si contenu vide
    if (empty( $content )) {
        $content = 'Luke';
    }

    return '<' . $tag . '>' . $content . ', Je suis ton père !' . '';
}
}
```

Excécuter un shortcode en PHP

Il est parfois utile d'exécuter un shortcode en PHP, pour ce faire WordPress nous propose la fonction `do_shortcode()`.



```
echo do_shortcode('Exemple de shortcode : [yoda]');  
//Exemple de shortcode <blockquote>Que la force soit avec toi jeune padawan  
!</blockquote>
```

Cela peut être très utile si un shortcode a besoin d'intégrer un autre shortcode :

```
//Fonction qui retourne le résultat d'un shortcode dans un élément strong  
function mon_plugin_strong_shortcode() {  
    return '<strong>' . do_shortcode( $content ) . '</strong>';  
}
```

Exercice

Afin de mettre en pratique vos connaissances vous pouvez essayer de réaliser l'[exercice « shortcode ico »](#).