

# CHAPITRE 19 — La grammaire des slots en détail

Référence exhaustive de tous les types de slot et de leurs attributs. Gardez ce chapitre à portée de main lors de l'annotation d'un template.

## Format général

```
{{slot:<name>|type=<type>|default=<value>|label=<text>|group=<section>|help=<text>|maxlength=<int>|options=<csv>}}
```

Tous les attributs, à l'exception de `name` et `type`, sont facultatifs. L'ordre est libre. Le séparateur est le caractère pipe.

## Règles syntaxiques

Règle	Détail
<b>Identifiant</b>	<code>[a-z0-9_]+</code> , 64 caractères maximum. Pas de tirets, pas de majuscules.
<b>Unicité</b>	Unique par page. Deux slots avec le même nom sur la même page produisent un avertissement au scan.
<b>Pas d'espaces</b>	Aucun espace de part et d'autre du signe <code> </code> . . Aucun espace dans les noms d'attributs.
<b>Échappement</b>	Le pipe à l'intérieur d'une valeur n'est pas pris en charge. Évitez-le dans les valeurs <code>default</code> . .
<b>Une seule ligne</b>	Un token de slot tient sur une seule ligne. Aucun saut de ligne ne doit y être présent.

## Les dix types de slot

## Type text — Texte court

Champ d'une seule ligne, sans mise en forme.

```
<h1>{{slot:hero_title|type=text|default=Bienvenue|label=Titre principal|maxlength=80}}</h1>
```

- Interface : champ de saisie simple.
- Idéal pour : titres, libellés, boutons d'appel à l'action, copyright.
- Compteur de caractères affiché si `maxlength` est défini.

## Type textarea — Texte long brut

Multi-lignes, sans mise en forme HTML.

```
<p class="lead">{{slot:hero_lead|type=textarea|default=Une  
accroche.|label=Accroche|maxlength=300}}</p>
```

- Interface : zone de texte multi-lignes.
- Les retours à la ligne sont automatiquement convertis en balises `<br>` au rendu.
- Idéal pour : accroches, paragraphes simples.

## Type richtext — Texte riche WYSIWYG

Mise en forme complète via CKEditor.

```
<div class="post-body">  
  {{slot:post_body|type=richtext|default=<p>Contenu de l'article.</p>|label=Corps de  
l'article}}  
</div>
```

- Interface : éditeur CKEditor (Format, gras, italique, listes, liens, couleurs, alignement, etc.).
- Idéal pour : corps d'articles, descriptions longues, conditions générales.
- La valeur stockée contient du HTML (`<p>`, `<strong>`, etc.).

## Type image — Média image

```

```

- Interface : champ texte, bouton de sélection dans la bibliothèque, bouton de suppression et vignette d'aperçu.
- Stocke soit une URL absolue, soit un shortcode `{{media:ref=X.url}}`.
- Non traduisible par défaut : la même image est servie dans toutes les langues.

## Type url — Lien

```
<a href="{{slot:hero_cta_url|type=url|default=/contact|label=URL du bouton}}">...</a>
```

- Interface : champ texte avec validation URL légère.
- Idéal pour : URL de bouton, liens vers les réseaux sociaux, URL de partage.

## Type icon — Icône FontAwesome

```
{{slot:feature_1_icon|type=icon|default=fa-solid fa-rocket|label=Icône feature 1}}
```

- Interface : champ classe, sélecteur de couleur, vingt icônes proposées en accès rapide, aperçu en direct.
- Stockage en JSON : `{"class":"fa-solid fa-star","color":"#fbbf24"}`.
- Rendu : `<i class="fa-solid fa-star" style="color:#fbbf24"></i>`.
- Sécurisé : liste blanche sur la classe (a-zA-Z0-9\_-) et expression régulière hexadécimale sur la couleur.

## Type color — Couleur

```
<section style="background-color:{{slot:section_bg|type=color|default=#19052d|label=Couleur de fond}}">
```

- Interface : sélecteur de couleur HTML5, champ hexadécimal, bouton de retour à la valeur par défaut.
- Format strict : `#RRGGBB` ou `#RRGGBBAA` (avec transparence).
- Non traduisible.

## Type number — Nombre

```
<span class="stat">{{slot:stats_clients|type=number|default=42|label=Nombre de clients}}</span>
```

## Type bool — Booléen

```
<?php if ('{{slot:hero_show_badge|type=bool|default=1|label=Afficher le badge}}'): ?>  
    <span class="badge">Nouveau</span>  
<?php endif; ?>
```

- Interface : case à cocher.
- Valeur : 0 ou 1.

## Type select — Liste déroulante

```
<section class="hero hero-  
{{slot:hero_layout|type=select|options=light,dark,gradient|default=light|label=Style du
```

```
hero}}">
```

- Interface : menu déroulant.
- Les valeurs disponibles sont définies dans `options` (CSV obligatoire).

## Valeurs par défaut spéciales

### default=@lang:KEY — Résolution via les fichiers de langue

Lorsque votre site utilise déjà des fichiers `.lang` Dolibarr (cas typique pour les sites multilingues existants), vous pouvez réutiliser les clés au lieu de les dupliquer dans les slots :

```
<h1>{{slot:hero_title|type=text|default=@lang:HeroTitle|label=Titre du hero}}</h1>
```

Au moment du rendu, le module résout via `$langs->trans('HeroTitle')` selon la langue du visiteur. Les fichiers `.lang` du site présents dans `DOL_DATA_ROOT/<entity>/website/<ref>/langs/` sont chargés automatiquement.

**Cas d'usage** — Migration progressive d'un site existant : vous conservez vos fichiers `.lang` en l'état, vous ajoutez les slots, et le client peut soit éditer dans le Studio (surcharge), soit conserver la traduction du `.lang` (canonique).

## Récapitulatif des types et de leur traduisibilité

Type	Interface	Traduisible
<code>text</code>	Champ de saisie	Oui
<code>textarea</code>	Zone de texte	Oui
<code>richtext</code>	CKEditor	Oui
<code>image</code>	Sélecteur de média	Non
<code>url</code>	Champ de saisie	Non
<code>icon</code>	Classe et sélecteur de couleur	Non
<code>color</code>	Sélecteur de couleur	Non
<code>number</code>	Champ numérique	Oui (rare)
<code>bool</code>	Case à cocher	Non

select	Menu déroulant	Non
--------	----------------	-----

## Pièges fréquents

**Slots dans des attributs JSON intégrés** — Si vous avez du JSON intégré dans une balise (par exemple `data-config="{...}"`), n'y placez pas de slot. Les doubles accolades sont aussi un délimiteur Mustache et risquent de casser les bibliothèques JavaScript qui parseraient ce JSON.

**Identifiant trop long** — La limite est de 64 caractères. Au-delà, le scanner rejette silencieusement, ce que met en évidence l'option `--lint`.

**Espaces autour des =** — Le scanner rejette `type = text`. Utilisez toujours `type=text`.

**Caractère pipe dans une valeur** — `default=A|B` casse l'analyseur. Utilisez une autre séparation, ou laissez `default` vide et saisissez la valeur via le Studio.

## Récapitulatif

### Vous savez désormais :

- La syntaxe complète d'un token de slot.
- Les dix types disponibles et leurs interfaces respectives.
- La résolution `@lang:KEY` pour réutiliser les fichiers `.lang` Dolibarr.
- Quels types sont traduisibles ou non.
- Les pièges syntaxiques fréquents.

Le chapitre suivant aborde les shortcodes, qui injectent des données Dolibarr dans le HTML.

---

Revision #2

Created 2026-05-07 11:27:38 UTC by Lucky Ranasolonirina

Updated 2026-05-08 07:35:54 UTC by Lucky Ranasolonirina