

w.c.s. - Bug #29183

lors de la transformation d'un texte en HTML, M. devient "une ligne qui commence par 13"

18 décembre 2018 17:06 - Thomas Noël

Statut:	Fermé	Début:	18 décembre 2018
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
Parce que quand mail_body est			
<code>\nM. Foobar\n</code>			
on a génération de			
<code>\n<ol class="upperalpha simple" start="13">\nFoobar\n\n</code>			

Révisions associées

Révision bb926723 - 19 décembre 2018 11:25 - Frédéric Péters

misc: only let rst consider arabic sequences (#29183)

Historique

#2 - 18 décembre 2018 17:17 - Thomas Noël

C'est certainement lié à la gestion des <http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html#enumerated-lists> qu'il faudrait sans doute désactiver... mais comment ?

#3 - 18 décembre 2018 17:21 - Frédéric Péters

En monkeypatchant l'affaire, pour modifier

```
enum.sequencepats = {'arabic': '[0-9]+',  
                    'loweralpha': '[a-z]',  
                    'upperalpha': '[A-Z]',  
                    'lowerroman': '[ivxlcdm]+',  
                    'upperroman': '[IVXLCDM]+', }
```

Pour y avoir plutôt [A-LN-Z], genre.

(mais il y a quantité de niveaux d'indirection dans docutils pour compliquer l'affaire)

#4 - 18 décembre 2018 17:36 - Thomas Noël

Je cherche pour l'instant à désactiver complètement le mécanisme qui, à mon sens, n'a aucun intérêt pour nous. Mais je ne trouve pas.

#5 - 18 décembre 2018 17:40 - Benjamin Dauvergne

`enum.sequencepats = {'dsfsdjf': 'oijewroijeoirjewoirj'}` ?

#6 - 18 décembre 2018 17:52 - Frédéric Péters

Je cherche pour l'instant à désactiver complètement le mécanisme qui, à mon sens, n'a aucun intérêt pour nous. Mais je ne trouve pas.

&

`enum.sequencepats = {'dsfsdjf': 'oijewroijeoirjewoirj'}` ?

Sauf qu'il y a bien quelque part un workflow qui attend que des listes soient correctement créées.

(et le truc compliqué c'est trouver comment arriver à patcher cette variable au milieu des indirections de docutils).

#7 - 18 décembre 2018 19:03 - Benjamin Dauvergne

Pour aller au plus simple on pourrait juste dire à Fondettes de mettre un backslash devant le nom de la personne comme indiqué dans la doc¹.

¹<http://docutils.sourceforge.net/docs/ref/rst/restructuredtext.html#enumerated-lists>

#8 - 19 décembre 2018 09:05 - Frédéric Péters

- Fichier 0001-misc-only-let-rst-consider-arabic-sequences-29183.patch ajouté
- Statut changé de Nouveau à Solution proposée
- Patch proposed changé de Non à Oui

Patch pour considérer uniquement les séquences utilisant des chiffres arabes.

#9 - 19 décembre 2018 09:40 - Thomas Noël

Un test rapide et ça semble pas fonctionner :

```
@pytest.mark.skipif('docutils is None')
def test_email_with_enumeration(emails):
    pub = create_temporary_pub()
    pub.cfg['emails'] = {'footer': 'Footer\nText'}
    mail_body = '''
A. FooAlpha1
B. FooAlpha2

1. Num1
2. Num2

M. Francis Kuntz

#. One
#. Two
'''
    send_email('test', mail_body=mail_body, email_rcpt='test@localhost')
    assert emails.count() == 1
    assert emails.emails['test']['msg'].is_multipart()
    assert emails.emails['test']['msg'].get_content_subtype() == 'alternative'
    assert emails.emails['test']['msg'].get_payload()[0].get_content_type() == 'text/plain'
    assert emails.emails['test']['msg'].get_payload()[1].get_content_type() == 'text/html'
    text = emails.emails['test']['msg'].get_payload()[0].get_payload()
    html = emails.emails['test']['msg'].get_payload()[1].get_payload()
    assert '\n  A. FooAlpha1\n  B. FooAlpha2\n' in text
    assert '\n  M. Francis Kuntz\n' in text
    assert '"upperalpha simple">\n<li>FooAlpha1</li>\n<li>FooAlpha2</li>' in html
    assert 'M. Francis Kuntz' in html
```

donne

```
> assert 'M. Francis Kuntz' in html
E       assert 'M. Francis Kuntz' in '<!DOCTYPE html>\n<html xmlns=3D"http://www.w3.org/1999/xhtml">\n<head>\n<meta charset=3D"utf-8">\n</head>\n<body>\n...r: #888; margin-top: 2em;">\n      <p>Footer<br />Text</p>\n</div>\n      =\n\n      =\n\n      </div>\n</body>\n</html>\n'
```

parce html est :

```
<blockquote>
<ol class=3D"upperalpha simple">
<li>FooAlpha1</li>
<li>FooAlpha2</li>
</ol>
<ol class=3D"arabic simple">
<li>Num1</li>
<li>Num2</li>
</ol>
<ol class=3D"upperalpha simple" start=3D"13">
<li>Francis Kuntz</li>
<li>One</li>
<li>Two</li>
</ol>
```

```
</blockquote>
</div>
```

#10 - 19 décembre 2018 09:45 - Frédéric Péters

- Statut changé de *Solution proposée* à *En cours*
- Patch *proposed* changé de *Oui* à *Non*

Chez moi ça marchait (évidemment) dans un rapide test, j'ai peut-être zappé des trucs, je vais ajouter ton test (merci) pour assurer ça.

#11 - 19 décembre 2018 09:51 - Frédéric Péters

Mais en fait le test sort de la zone attendue,

```
mail_body = '''
M. Francis Kuntz
'''
send_email('test', mail_body=mail_body, email_rcpt='test@localhost')
assert emails.count() == 1
assert emails.emails['test']['msg'].is_multipart()
assert emails.emails['test']['msg'].get_content_subtype() == 'alternative'
assert emails.emails['test']['msg'].get_payload()[0].get_content_type() == 'text/plain'
assert emails.emails['test']['msg'].get_payload()[1].get_content_type() == 'text/html'
text = emails.emails['test']['msg'].get_payload()[0].get_payload()
html = emails.emails['test']['msg'].get_payload()[1].get_payload()
assert 'M. Francis Kuntz\n' in text
assert 'M. Francis Kuntz' in html
```

passee.

Là, avec les espaces posés en début de ligne, ça bascule dans le mode "blockquote", avec sans doute d'autres trucs à attraper :/

#12 - 19 décembre 2018 09:56 - Frédéric Péters

```
assert "'upperalpha simple">\n<li>FooAlpha1</li>\n<li>FooAlpha2</li>' in html
```

Le patch, dans son intention, doit totalement désactiver ce type de liste.

Et #. One c'est du markdown qui vient s'ajouter dans l'histoire, quel résultat attendu ?

Bref je serais pour que le test vérifie que 1./2. produit bien une liste , et que le reste n'en produit pas.

#13 - 19 décembre 2018 10:06 - Frédéric Péters

- Fichier *0001-misc-only-let-rst-consider-arabic-sequences-29183.patch* ajouté
- Statut changé de *En cours* à *Solution proposée*
- Patch *proposed* changé de *Non* à *Oui*

Patch qui reprend le test, modifié pour correspondre à ce que j'attends.

Le comportement à l'intérieur des blockquotes n'est pas altéré parce que trop intrusif dans docutils d'aller chercher les NestedStateMachine et traffiquer etc. (si quelqu'un trouve intéressant d'y passer ses vacances, je prendrai bien sûr un patch).

#14 - 19 décembre 2018 10:08 - Thomas Noël

Frédéric Péters a écrit :

```
assert "'upperalpha simple">\n<li>FooAlpha1</li>\n<li>FooAlpha2</li>' in html
```

Le patch, dans son intention, doit totalement désactiver ce type de liste.

Ouais, j'avoue que c'est un test que j'ai écrit hier soir en cherchant à gérer juste le "M."

Et #. One c'est du markdown qui vient s'ajouter dans l'histoire, quel résultat attendu ?

Ca marche en fait, dans docutils, de faire un début de liste puis la suite avec « #. ».

Bref je serais pour que le test vérifie que 1./2. produit bien une liste , et que le reste n'en produit pas.

Yep yep. Mon test était pas exactement pour ton code, sorry.

#15 - 19 décembre 2018 10:09 - Thomas Noël

- *Statut changé de Solution proposée à Solution validée*

Impec, bravo.

#16 - 19 décembre 2018 11:26 - Frédéric Péters

- *Statut changé de Solution validée à Résolu (à déployer)*

```
commit bb926723c9fd0731416d5ed58f0d7ecb275e2ad5
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Wed Dec 19 09:03:15 2018 +0100
```

```
misc: only let rst consider arabic sequences (#29183)
```

#17 - 23 décembre 2018 14:31 - Frédéric Péters

- *Statut changé de Résolu (à déployer) à Solution déployée*

Fichiers

0001-misc-only-let-rst-consider-arabic-sequences-29183.patch	2,26 ko	19 décembre 2018	Frédéric Péters
0001-misc-only-let-rst-consider-arabic-sequences-29183.patch	3,72 ko	19 décembre 2018	Frédéric Péters