


Préambule

 Les expressions ici utilisent la syntaxe reconnue par python qui suit le standard POSIX (à quelques exceptions près ¹⁾).

Capture

Les parenthèses () de groupement permettent de délimiter un ensemble d'alternatives et d'opérer des captures. Le plus souvent si nous réalisons une capture nous voudrions pouvoir y faire référence via un nom que nous aurons choisi.

En python la syntaxe qui permet cela est la suivante

```
(?P<nom_du_groupe_de_capture>...)
```

La sous-expression régulière (ici symbolisée par ...) qui aura été trouvée sera accessible par le nom **nom_du_groupe_de_capture**

Dès lors nous pouvons y faire référence dans la suite de l'expression régulière elle-même :

```
(?P=nom_du_groupe_de_capture)
```

Mais dans le cas de EzGED nous y ferons le plus souvent référence lors d'un post formatage grâce à la syntaxe @<groupname>.

Tout de suite un petit exemple.

Numéro de facture

Soit le texte extrait suivant

```
RueDuCommerce SA
Facture 44-50, avenue du Capitaine Glarner
93585 Saint-Ouen Cedex
EzDEV
```

2

```
Rue Adolphe Pégoud
90130 PETIT-CROIX
```

```
Numéro de facture : 56497
```

```
Référence client : 100132
```

```
Date de la facture : 04/11/2015
```

Cet exemple illustre bien l'un des cas que vous retrouvez le plus souvent. Il s'agit de récupérer un numéro (ici de facture).

Nous pourrions être tentés de faire une recherche avec le filtre de champ

```
Numero de facture : .*
```

puis un post formatage

```
@keepdigits(@self)
```

Ce qui pourrait fonctionner si nous spécifions correctement la longueur du champ.

Soit le filtre de champ suivant

```
Numero de facture : (?P<numero>\d+)
```

Pour ne garder que le numero on fera ceci dans le champ post de formatage :

```
@caststr(@<numero>)
```

Recherche multi-ligne

Prenons cet exemple de texte extrait

The screenshot shows a configuration window for a search field. On the left, the configuration includes:

- Id:** 1304
- Type:** Position Variable
- Description:** FACTURE_NUMERO
- Champ de destination:** FACTURE_LIVRAISONADR
- Position de démarrage X:** 0
- Position de démarrage Y:** 5
- Position de fin X:** 40
- Position de fin Y:** 0
- Longueur du champ:** 0
- Filtre de champ:** Facture N.([\d])\n)*s(?P<num>\d+)s
- Option de filtre de champ : ignorer la casse:** checked
- Post Formatage:** @caststr(@<num>)
- Suppression des blancs:** Aucun

On the right, a preview of the document text is shown with a red box highlighting the search area. The text is as follows:

```

0 1 2 3 4 5 6
--123456789|0123456789|0123456789|0123456789|0123456789|0123456789|0123456789
1Tricatel
2120, Avenue Foch
394000
4Créteil
5
6
7
8
9 Facture N° Date de la facture
10 681476 04/11/2015
11
12
13 Numéro de client 456
14 Paiement |d0 04/12/2015
15 Modalité de paiement 30 jours
16 Emis par Jacques Tricatel
17 Contact client Noël Groboulchon
18 Date de la vente/prestation 20/10/2015
19 Adresse de livraison Gastro Discount Lyon, ZAC De la
20 Date de livraison Du 05/11/2015 au 10/11/2015
21
22
23
24
25 Infos additionnelles
26
27
28 Merci d'avoir choisi Ma Compagnie pour nos services.
29
30
31
32
33 Description
34 Cassoulet de synthèse
35 Véritable salade en polymère
  
```

Si l'on souhaite récupérer le numéro de facture on pourrait le faire assez simplement en utilisant la macro @relativeto. Ce serait une bonne solution mais il existe une alternative qui ne nous obligera pas à créer deux champs d'analyse.

Comme le montre la capture nous utilisons un champ de type variable et nous ciblons à l'aide des paramètres de position la zone de recherche. La zone de recherche est encadrée en rouge (l'image est tronquée, elle s'étend jusqu'au bas du document).

Et l'expression régulière que nous utilisons est la suivante :

```
Facture N. ([^\d]|\n)*(?P<num>\d+)
```

Décomposons l'expression.

Expression	Signification
Facture N.	Ce que l'on recherche commence par cette chaîne. Le point remplace le signe ° qui n'est pas bien géré.
([^\d] \n)*	Ce qui suit doit être un caractère (excluant les chiffres) ou un retour à la ligne. Ce motif peut se répéter 0 ou plusieurs fois
(?P<num>\d+)	On doit trouver au moins un chiffre. Cette suite de chiffre est capturée dans la variable num

En résumé : Nous cherchons une chaîne qui **commence par "Facture N"**, qui est suivit **éventuellement** (mais pas nécessairement) par **tout type de caractère sauf les chiffres** mais en **incluant les sauts de ligne** et cet enchaînement est suivi par **au moins un chiffre**.

L'exemple dans la capture ci-dessus nous renvoi comme valeur

```
681476
```

1)

Article wikipedia sur les expression rationnelles

https://fr.wikipedia.org/wiki/Expression_rationnelle#Python

From:

<http://wiki.ezdev.fr/> - **EzGED Wiki**

Permanent link:

<http://wiki.ezdev.fr/doku.php?id=cold:tutoriaux:regex&rev=1447946761>

Last update: **2023/03/17 09:56**

