Exemple de parsing d'un fichier avec SED (avec des sous chaines multiples)

Imaginons un fichier html avec des entrées

- référence 1 : 6 lettres en capitales
- étagère : “ETAGERE” + 2 chiffres
- référence 2 : suite de 6 chiffres
- référence de la boite : BOX + 2 chiffres
- référence de l'unité de stockage : UNITE + 1 lettre en capitale

```
<html>
<body>Exemple avec SED</body>
<p>VOici un exemple</p>
<h1>Stockage BATIMENT 1</h1>
<h2>ABCDEF</h2><label class="tel"
title="ETAGERE34">12345678</label><p><br>BOX06<br>UNITEA</p>
<h3>
<h2>ZERDEF</h2><label class="tel"
title="ETAGERE04">88888888</label><p><br>BOX12<br>UNITER</p>
<h3>Description d'autre chose</h3>
<p>Une description et du blablabla</p>
<p><b><i>Juste pour remplir</i></b><p>
.....
<h2>UYRDEF</h2><label class="tel"
title="ETAGERE12">87654321</label><p><br>BOX65<br>UNITEZ</p>
</body>
</html>
```

Imaginez des centaines de pages de ce type et votre mission, récupérer les valeurs des 5 champs pour les mettre dans un tableur. Un travail titanesque à la main...

Pour cela utiliser, sed et les sous-chaines.

- ^<h2> : Ne travailler qu'avec les lignes commençant par la balises <h2>
- [A-Z]{6} : pour le 1er motif de recherche 6 lettres en capitales
- .* : n'importe quoi jusqu'au prochain motif
- ETAGERE[0-9]{2} : le terme ETAGERE suivi de 2 chiffres
- .* : n'importe quoi jusqu'au prochain motif
- [0-9]{8} : une suite de 8 chiffres
- .* : n'importe quoi jusqu'au prochain motif
- BOX[0-9]{2} : le terme BOX suivi de 2 chiffres
- .* : n'importe quoi jusqu'au prochain motif
- [A-Z]{6} : une suite de 6 lettres en capitales
- .* : n'importe quoi jusqu'à la fin de la ligne

Voilà pour les expressions régulières (**regex**). Pour les rendre compréhensibles par sed, il faudra “échapper” (\) les caractères {, }, (et) afin qu'ils ne soient pas considéré comme un élément d'une chaine de caractère mais comme un élément d'une expression régulière..

Ce qui donnera comme regex

```
 ^<h2>\([A-
Z]\{6\}\).*\(ETAGERE[0-9]\{2\}\).*\([0-9]\{8\}\).*\(BOX[0-9]\{2\}\).*\([A-
Z]\{6\}\).*
```

Tout est bien beau, mais comment virer le code HTML et le blablabla. En entourant chaque motif de recherche par des parenthèses tout simplement et en rappelant ces extraction par \1 pour la première, \2 pour la seconde et ainsi de suite

- le premier motif \
- le second : \2
- le 3e : \3
- Et ainsi de suite

Comme c'est pour un tableur, nous passerons par un fichier CSV. Il suffit donc de récupérer les motifs et de les séparer par un point virgule

```
\1;\2;\3;\4;\5;
```

Et comme il s'agit d'une substitution, on utilisera l'option -s de sed

Donc au final :

```
 sed -n 's/<h2>\([A-
Z]\{6\}\).*\(ETAGERE[0-9]\{2\}\).*\([0-9]\{8\}\).*\(BOX[0-9]\{2\}\).*\([A-
Z]\{6\}\).*/\1;\2;\3;\4;\5;/p' mon-fichier-entree.html
```

Si vous voulez créer un fichier csv, simplement rediriger la sortie du terminal > sortie.csv Si le fichier doit être incrémenté par la lecture de plusieurs sources, utiliser » à la place de > afin de ne pas écraser le fichier existant mais le compléter.

Enjoy