Jour 6 — Expressions régulières (suite)

La syntaxe `[name~"[0-9]"]` permet de trouver les noms **contenant un nombre**. On peut préciser le **nombre de chiffres** successifs avec des accolades. Ainsi la requête suivante trouve les rues dont le nom contient un nombre de 4 chiffres (généralement une année) :

```
[bbox:{{bbox}}];
way[highway][name~"[0-9]{4}"];
out geom;
```

Dans le même esprit, `["addr:housenumber"~"[a-z]"]` trouve les numéros d'adresses contenant **au moins une lettre** minuscule. Il est possible de combiner minuscules et majuscules :

```
node["addr:housenumber"~"[a-zA-Z]"]
```

Ces adresses peuvent être soit des *bis* et des *ter*, soit des lettres arbitraires accolées au numéro. Pour les distinguer on peut préciser le nombre de lettres, par exemple ne retenir que les numéros avec une lettre. Mais attention à considérer l'ensemble de la valeur. Ainsi la requête

```
node["addr:housenumber"~"[a-zA-Z]{1}$"];
```

retourne tous les numéros finissant par une lettre, mais ne dit rien des caractères précédents qui peuvent aussi être des lettres. Une possibilité est de préciser que l'adresse commence par au moins un chiffre (`^[0-9]+`), se termine par une lettre (`[a-zA-Z]$`), qui peut être **séparée ou pas par un espace** (`?`). On obtient donc la requête suivante :

```
[bbox:{{bbox}}];
node["addr:housenumber"~"^[0-9]+ ?[a-zA-Z]$"];
out;

{{style:
node {
  text: addr:housenumber;
}
}}
```

L'affichage d'une étiquette est particulièrement utile pour vérifier qu'une expression régulière a bien l'effet attendu.

Si on recherche un **caractère spécial** (notamment les caractères *+.*^$), il est nécessaire « d'échapper » chaque caractère en le faisant précéder de deux \, par exemple pour trouver les commerces dont le nom contient `...` :

```
node[shop][name~"\\.\\.\\."];
```

Pour finir, il est possible d'appliquer une expression régulière à la **clef** d'un tag, en plaçant une tilde avant la clef (qui doit alors être placée entre quotes). Cela est très utile pour les clefs utilisant un préfixe ou un suffixe. Cet exemple trouve les voies pour lesquelles un aménagement cyclable est

décrit, que ce soit avec la clef `cycleway`, `cycleway:left`, `cycleway:right` ou `cycleway:both`, quelle qu'en soit la valeur :

```
way[highway][~"cycleway"~"."];
```

Exercices

- Trouvez les rues dont le nom contient une date sous la forme « jour mois année » (ex : 8 mai 1945).
- Trouvez les adresses pour lequel le nom de ville (`addr:city`) est entièrement en majuscules.
- Trouvez les rues où le stationnement est interdit (voir https://wiki.osm.org/wiki/FR:Key:parking:lane).

© CC-by-sa Carto'Cité

From:
http://wiki.cartocite.fr/ -

Permanent link:
http://wiki.cartocite.fr/doku.php?id=tutoverpass:jour_6_expressions_regulieres_suite

Last update: **2021/01/25 17:14**