LES EXPRESSIONS RÉGULIÈRES (PARTIE 1/2)

Les regex sont une sorte de langage « à part » qui sert à manipuler les chaînes de caractères. Voici quelques exemples de ce que les regex sont capables de faire :

* Vérifier si une URL entrée par l'utilisateur ressemble effectivement à une URL. On peut faire pareil pour les adresses e-mail, les numéros de téléphone et toute autre syntaxe structurée ;
* Rechercher et extraire des informations hors d'une chaîne de caractères (bien plus puissant que de jouer avec indexOf() et substring()) ;
* Supprimer certains caractères, et au besoin les remplacer par d'autres ;
* Pour les forums, convertir des langages comme le BBCode en HTML lors des prévisualisations pendant la frappe ;
* Et bien d'autres choses…

Les regex en JavaScript

La syntaxe des regex en JavaScript découle de la syntaxe des regex du langage Perl. C'est un langage très utilisé pour l'analyse et le traitement de données textuelles (des chaînes de caractères, donc), en raison de la puissance de ses expressions régulières. Le JavaScript hérite donc d'une grande partie de la puissance des expressions régulières de Perl.

**Utilisation**

Les regex ne s'utilisent pas seules, et il y a deux manières de s'en servir : soit par le biais de RegExp qui est l'objet qui gère les expressions régulières, soit par le biais de certaines méthodes de l'objet String :

* match() : retourne un tableau contenant toutes les occurrences recherchées ;
* search() : retourne la position d'une portion de texte (semblable à indexOf() mais avec une regex);
* split() : la fameuse méthode split(), mais avec une regex en paramètre ;
* replace() : effectue un rechercher/remplacer.

**En résumé**

* Les regex constituent une technologie à part, utilisée au sein du JavaScript et qui permet de manipuler les chaînes de caractères. La syntaxe de ces regex se base sur celle du langage Perl.
* Plusieurs méthodes de l'objet String peuvent être utilisées avec des regex, à savoir match(),search(), split() et replace().
* L'option i indique à la regex que la casse doit être ignorée.
* Les caractères ^ et $ indiquent respectivement le début et la fin de la chaîne de caractères.
* Les classes et les intervalles de caractères, ainsi que les types génériques, servent à rechercher un caractère possible parmi plusieurs.
* Les différents méta caractères doivent absolument être échappés.
* Les quantificateurs servent à indiquer le nombre de fois qu'un caractère peut être répété.