

Javascript - Expressions régulières

C'est quoi?

C'est complexe, mais aussi très puissant ! Ce n'est pas un concept lié au JavaScript, car les expressions régulières, souvent surnommées « regex », trouvent leur place dans bon nombre de langages, comme le Perl, le Python ou encore le PHP.

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.

/mot/.test()

Le sujet étant complexe, nous allons commencer par des choses simples, c'est-à-dire des recherches de mots. Ce n'est pas si anodin que ça, car il y a déjà moyen de faire beaucoup de choses. Comme nous venons de le voir, une regex s'écrit comme suit.

/mot_a_rechercher/

Si on teste, voici ce que ça donne

```
if (/raclette/.test("Je mangerais bien une raclette savoyarde !")) alert("Ça semble parler de raclette")
else alert("Pas de raclette à l'horizon")
```

On peut ajouter un "i" pour ne pas prêter attention à la casse, c'est-à-dire de rechercher indifféremment les majuscules et les minuscules.

```
if (/raclette/i.test("Je mangerais bien une Raclette Savoyarde !")) alert("Ça semble parler de raclette")
else alert("Pas de raclette à l'horizon")
```

Vérifier une adresse mail

Il y a de nombreuses façons d'écrire une expression régulière pour vérifier une adresse mail. Tout dépendra des critères pour durcir la vérifier et laisser le moins d'erreurs possibles.

Dans la majeure partie des situations, une simple recherche sur internet nous fournira plusieurs "regex", il faut donc les tester pour s'assurer qu'elles correspondent à nos attentes.

```
var email = "wellens@epse.be"
if(!/\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+\$.test(email)) alert("Email invalide")
```

Récupérer un élément dans un contenu

Pour récupérer un élément précis dans une variable. Il y a la méthode MATCH. La construction de l'expression n'est pas simple à produire, puisque l'analyse doit prendre en compte tous les cas de figure.

Par exemple, l'utilisateur donne un lien youtube, vous souhaitez ne récupérer que l'ID de la vidéo pour l'ajouter dans un code d'intégration en IFRAFME

```
const regex = /youtu(?:.*\vV|.*v\=|\.\beV)([A-Za-z0-9_\-\]{11})/
const link = "https://www.youtube.com/watch?v=dQw4w9WgXcQ"
const id = link.match(regex)[1]           // match retourne un tableau avec les différentes valeurs récupérées
```

