

Chapitre 4 : Expressions régulières

- écrire une fonction qui indique si une chaîne passée en paramètre est constituée uniquement de chiffres

→ parcourir la chaîne, caractère par caractère, et tester si c'est un caractère numérique (i.e. compris entre '0' et '9')

Code de la fonction

```
function nombre(ch) {  
  let nb=true;  
  let i=0;  
  while( nb && i<ch.length)  
    if (ch.charAt(i)<"0" || ch.charAt(i)>"9")  
      nb=false;  
    else i++;  
  
  return nb;  
}
```

Définition

- Une expression régulière décrit un modèle de caractères :
par exemple un nombre, une adresse mail...
- La classe RegExp représente les expressions régulières et définit des méthodes pour :
 - la recherche de modèles,
 - la recherche et le remplacement de texte.
- Définition d'un objet de la classe Regexp :
// chaîne qui finit par un s
let pattern=new RegExp(" s\$ ");
équivalent à let pattern=/s\$/;

- Décrire un ensemble de caractères
 - . n'importe quel caractère (sauf « fin de ligne »)
- [abc] → une des lettres a, b ou c
- [^abc] → un caractère autre que a, b et c
- NB : . perd sa signification particulière
- [a-zA-Z] → ?

Expression régulière de classes de caractères

`\w` [a-zA-Z0-9_]

`\W` [^a-zA-Z0-9_]

`\s` un caractère « whitespace » (espace, tabulation, ...)

`\S` un caractère qui n'est pas un « whitespace »

`\d` [0-9]

`\D` [^0-9]

Caractères particuliers

- comment rechercher un caractère qui a une signification particulière

`^ $. * + ? = ! : | \ / () [] { }`

→ `\` pour annuler la signification particulière

Répétition

$\{n,m\}$	entre n et m fois
$\{n,\}$	au moins n fois
$\{n\}$	exactement n fois
$?$	$\{0,1\}$
$+$	$\{1,\}$
$*$	$\{0,\}$

Exemple : $/[a-z]\{2,3\}[0-9]\{1\}/$

2 ou 3 lettres minuscules suivies d'un chiffre

Quizz

$\wedge d\{2,4\}/$

$\wedge w\{3\} \backslash d? /$

$\wedge s^+ \text{java} \backslash s^+ /$

$/[^\wedge()^* /$

Préciser la position

^ la correspondance est en début de la chaîne
\$ fin de la chaîne

Exemple : /^javascript\$/"

Alternative et groupement

| alternative - matche avec la partie gauche ou la partie droite

(...) groupement – unité à utiliser avec *, +, ?, ...

Exemple : $/9\{1\}|[0-3]\{2\}/$

flags

- i case-insensitive
- g recherche toutes les correspondances
(sinon stoppe à la première)

Exemple : `/^javascript$/i`

Méthodes de la classe String

- Recherche

`texte.search(pattern)` → retourne la position du caractère de début de la première correspondance rencontrée (-1 sinon)

`texte.match(pattern)` → retourne un tableau contenant les correspondances

Exemple : `"javascript".search(/script/i);`

`"23 sept 2017".match(/\d+/g);`

- Remplacement

`texte.replace(pattern, chaine)` → remplace toutes les correspondances par chaine

Méthodes de la classe RegExp

- `pattern.test(texte)` retourne `true` si le texte contient une correspondance
- `pattern.exec(texte)` retourne un tableau avec les informations sur la correspondance

- Exemple :

```
let re = /<(.)>/ig;  
let result = re.exec('C"est cool <javascript> yeh');  
//result[0]  
// result[1] → la sélection entre ()  
//result.index
```

Exercices

Donner les expressions régulières pour que le modèle soit :

- un nombre
- une chaîne exclusivement alphanumérique
- un nombre réel

À quel modèle correspond cette expression :

- $^[a-zA-Z0-9._-]^+@[a-zA-Z0-9._-]{2,}\.[a-zA-Z]{2,4}\$$

Donner l'instruction pour :

- Récupérer l'extension d'une URL
- Enlever les espaces en début et en fin de chaîne
- Extraire chaque phrase d'un texte.
- Amélioration : Mettre en majuscule la première lettre ainsi que celles qui suivent un point. Exemple : "bonjour. ceci est un exemple. mais il y en a d'autres..."