Les expressions régulières

Introduction sur les expressions régulières

- Un propre langage en soi!
- De nombreux langages (JS, python, ..) supportent les regex et fournissent des outils pour les utiliser

- Regex = motifs (ou patterns ou masques)
  - Effectuer des recherches
  - Remplacements dans une chaine de caractère

Utilisation

- 2 façons d'écrire une regex
  - Déclaration de manière littérale

```
1   //manière littérale
2   let regex1 = /Becode/;
3
```

  - Déclaration par le constructeur `new RegExp()`

```
4   //constructeur
5   let regex2 = new RegExp('Becode');
6
```

```
24      // Manière littérale dans un constructeur
25      let regex2 = new RegExp(/Becode/)
```

Puissance de la regex??

Les classes de caractères

- Classes de caractères = permet de fournir différents choix de correspondances pour un caractère en spécifiant un ensemble de caractères. Se fait via `[ ]`

  - Exemple 1 : Pattern d'une voyelle

```
let regexVoyelle = /[aeiouy]/

let regexVoyelleBis = /[aeioyu] [aeiouy]/
```

  - Exemple 2 : Pattern de minuscules ou majuscules

```
let regexMinus = /[a-z]/

let regexMaj = /[A-Z]/
```

  - Exemple 3 : Pattern de chiffres

```
let regexNum = /[0-9]/
```

Les classes de caractères

Recherche globale!!

```
let regex = /*insérer votre regex ici*/g
```

```
<body>
    <p></p>
</body>

    <script>
        let string = "J'aime Pas Les Majuscules"
        /**==================================
         *  Expression non globale
         * ==================================
         */
        let regex = /[A-Z]/

        document.querySelector("p").innerHTML = string.replace(regex, "")

    </script>
</html>
```

'aime Pas Les Majuscules

```
<body>
    <p></p>
</body>

    <script>
        let string = "J'aime Pas Les Majuscules"
        /**==================================
         *  Expression globale
         * ==================================
         */
        let regex = /[A-Z]/g

        document.querySelector("p").innerHTML = string.replace(regex, "")

    </script>
</html>
```

'aime as es ajuscules

Les classes de caractères abrégées

- Ce sont des plages de valeurs appelées via le backslash qui « résume » les classes de caractères :

Classes abrégées	Description
\w	Tout les caractères de type 'mot'. Equivaut à [a-zA-Z0-9_]
\d	Représente un chiffre. Equivaut à [0-9]
\s	Représente un caractère blanc (espace, retour à la ligne)
\S	Représente tout ce qui n'est pas un caractère blanc
\t	Représente un espace
\n	Représente un saut de ligne

Les ancres

- Ancres = métacaractères permettant d'ancrer les patterns
- Ancre de début : `^`

```
let regexVoyelleDebut = /^[aeiouy]/g
let regexVoyelleDebut2 = /^[^aeiouy]/g
```

- Ancre de fin : `$`

```
let regexConsonneFin = /[^aeiouy]$/g
```

https://regexr.com/

Les quantificateurs

- Quantificateurs = métacaractères représentant une certaine quantité d'un caractère ou d'une séquence de caractères

Quantificateur	Description	Exemple
a{x}	On veut une séquence de X 'a'	`let regexB = /ir{2}/g`
a{x,y}	On veut une séquence de X à Y fois 'a'	`let regexB = /ir{2,5}/g`
a{x,}	On veut une séquence d'au moins X fois 'a' sans limite supérieure	`let regexB = /ir{2,}/g`
a?	On veut 0 ou 1 'a'	`let regexB = /ir?/g`
a+	On veut au moins une fois 'a'	`let regexB = /ir+/g`
a*	On veut 0, 1 ou plusieurs 'a'	`let regexB = /ir*/g`

La méthode match( )

- Méthode permettant de rechercher la présence ou non d'une séquence de caractères dans une chaine de caractère

```
9   <body>
10
11      <h1 style="text-decoration: underline; font-size: larger;
        ">Les regex c'est cool!</h1>
12
13      <p id="p1"></p>
14      <p id="p2"></p>
15      <p id="p3"></p>
16
17  </body>
18
19      <script>
20          let string = 'Petite phrase en exemple sur les regexp';
21          let string1 = /[^a-z]/g;
22          let string2 = /phrase/g;
23          let string3 = /\d/g
24
25          document.getElementById("p1").innerHTML = 'Caractères
            trouvés: ' + string.match(string1)
26          document.getElementById("p2").innerHTML = 'Caractères
            trouvés: ' + string.match(string2)
27          document.getElementById("p3").innerHTML = 'Caractères
            trouvés: ' + string.match(string3)
28
29      </script>
30  </html>
```

Les regex c'est cool!

Caractères trouvés: P, , , , , ,

Caractères trouvés: phrase

Caractères trouvés: null

La méthode search( )

- Méthode permettant d'effectuer une recherche dans une chaine de caractère et voir sa position

```
9   <body>
10
11      <h1 style="text-decoration: underline; font-size: larger;
        ">Les regex c'est cool!</h1>
12
13      <p id="p1"></p>
14      <p id="p2"></p>
15      <p id="p3"></p>
16
17  </body>
18
19      <script>
20          let string = 'Petite phrase en exemple sur les regexp';
21          let string1 = /[^a-z]/g;
22          let string2 = /phrase/g;
23          let string3 = /\d/g
24
25          document.getElementById("p1").innerHTML = 'Caractères
            trouvés à la position: ' + string.search(string1)
26          document.getElementById("p2").innerHTML = 'Caractères
            trouvés à la position: ' + string.search(string2)
27          document.getElementById("p3").innerHTML = 'Caractères
            trouvés à la position: ' + string.search(string3)
28
29      </script>
30  </html>
```

Les regex c'est cool!

Caractères trouvés à la position: 0

Caractères trouvés à la position: 7

Caractères trouvés à la position: -1

La méthode test( )

- Méthode permettant de rechercher des correspondances mais renvoie un booléen si la correspondance est trouvée ou non

```
18  </body>
19
20      <script>
21          let string = 'Petite phraase en exemple sur les regexp';
22          let string1 = /e/g
23          let string2 = /becode/
24          let string3 = /a{2}/g
25
26          let p1 = document.getElementById("p1")
27          let p2 = document.getElementById("p2")
28          let p3 = document.getElementById("p3")
29
30          if (string1.test(string)){
31              p1.textContent = "Correspondance trouvée"
32          } else {
33              p1.textContent = "Correspondance non trouvée"
34          }
35
36          string2.test(string) ? p2.textContent = "Correspondance
            trouvée" : p2.textContent = "Correspondance non trouvée"
37
38          string3.test(string) ? p3.textContent = "Correspondance
            trouvée" : p3.textContent = "Correspondance non trouvée"
39      </script>
40  </html>
```

Les regex c'est cool!

Correspondance trouvée

Correspondance non trouvée

Correspondance trouvée

La méthode replace( )

- Méthode permettant de rechercher un caractère ou une séquence et les remplacer par une autre expression

```
<body>

    <h1 style="text-decoration: underline; font-size: larger;
    ">Les regex c'est cool!</h1>

    <p id="p1"></p>
    <p id="p2"></p>
    <p id="p3"></p>
    <p id="p4"></p>

</body>

    <script>
        let string = 'Petite phrase en exemple sur les regexp';
        let string1 = /^[A-Z]/g;
        let string2 = /phrase/g;
        let string3 = /e/g
        let string4 = /.$/

        document.getElementById("p1").innerHTML = string.replace
        (string1, "")
        document.getElementById("p2").innerHTML = string.replace
        (string2, "~un autre exemple de remplacement~")
        document.getElementById("p3").innerHTML = string.replace
        (string3, 2)
        document.getElementById("p4").innerHTML = string.replace
        (string4, 0)

    </script>
</html>
```

Les regex c'est cool!

etite phrase en exemple sur les regexp

Petite ~un autre exemple de remplacement~ en exemple sur les regexp

P2tit2 phras2 2n 2x2mpl2 sur l2s r2g2xp

Petite phrase en exemple sur les regex0

Exemple d'application en Python

Faison un petit exercice. Quelle serait l'expression régulière obligeant un utilisateur de donner un numéro de téléphone correct? (de type belge xxx/xx.xx.xx)

In [3]:
```
chaine = ""
expression = r"^0[0-9][0-9][/](([0-9]{2}[.]){2}[0-9]{2}))$"
while re.search(expression, chaine) is None:
    chaine = input("Saisissez un numéro de téléphone (valide):")
```

```
Saisissez un numéro de téléphone (valide):048121212
Saisissez un numéro de téléphone (valide):048/12.12.12
```

Quelques petits jeux pour la route? ;)

- https://lesjoiesducode.fr/vous-aimez-les-regex-ce-jeu-est-fait-pour-vous
- https://www.codingame.com/start

Plus d'infos?

- https://github.com/CalcagnoLoic/aide_memoire/blob/main/R%C3%A9pertoire/chapJS/regex.md

Merci pour votre attention ☺