
Expressions régulières

Principe

La classe `NSRegularExpression` est utilisée pour représenter des expressions régulières

Création d'une expression régulière

- ❖ `init(pattern:options:)`

Recherche d'une chaîne de caractères correspondant

- ❖ `numberOfMatches(in:options:range:)`
- ❖ `matches(in:options:range:)`

26

Expressions régulières

Les motifs de recherche

^	Indique le début de la chaîne exemple : <code>^chat</code> reconnaît une ligne qui commence par chat
\$	Indique la fin de la chaîne exemple : <code>chat\$</code> reconnaît une ligne qui finit par chat
.	Le point indique n'importe quel caractère

*	Indique 0, 1 ou plusieurs occurrences du caractère exemple : <code>a*</code> reconnaît « pas de a », <code>a</code> , <code>aa</code> , <code>aaa...</code>
+	Indique une ou plusieurs occurrences du caractère exemple : <code>a+</code> reconnaît <code>a</code> , <code>aa</code> , <code>aaa...</code>
?	Indique 0 ou une occurrence du caractère exemple : <code>a?</code> reconnaît « pas de a » et <code>a</code>

27

Expressions régulières

Les motifs de recherche

[...]	tous les caractères énumérés entre les crochets exemple : <code>[aeiouy]</code> reconnaît toutes les voyelles
[^...]	Tous les caractères sauf ceux énumérés exemple : <code>[^aeiouy]</code> reconnaît tous les caractères sauf les voyelles
	alternative - ou reconnaît l'un ou l'autre exemple : <code>aa ab</code> reconnaît les chaînes <code>aa</code> ou <code>ab</code>
[...-...]	intervalle de recherche exemple : <code>[0-9]</code> reconnaît tous les chiffres de 0 à 9

28

Expressions régulières

Les motifs de recherche

<code>a{3}</code>	correspond exactement à <code>aaa</code>
<code>a{2,}</code>	correspond à un minimum de deux <code>a</code> consécutifs soit <code>aa</code> , <code>aaa</code> , <code>aaaa</code>
<code>a{2,4}</code>	correspond uniquement à <code>aa</code> , <code>aaa</code> , <code>aaaa</code>
<code>(...)</code>	Limite la portée d'un masque exemple : <code>a(b){2}</code> reconnaît la chaîne <code>abb</code>

Certains caractères sont des caractères spéciaux, ce sont les métacaractères:

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

Pour les rechercher dans une chaîne de caractères il faut les échapper avec `\\`

Exemple : `\\[.]{3}\\\\` reconnaît une suite de trois caractères entre crochets

29

Expressions régulières

Utilisation de la classe `NSRegularExpression`

`ViewController.swift`

```
override func viewDidLoad() {
    super.viewDidLoad()

    let s1 = "123abc"
    let s2 = "abc123"
    let s3 = "abc"

    for s in [s1, s2, s3] {
        do {
            let regex = try NSRegularExpression(pattern:"^[1-9]{3}",
                                                options: .caseInsensitive)
            let match = regex.numberOfMatches(s,
                                                options: .reportProgress, range: NSRange(location:0,
                                                length:s.characters.count))

            if (match > 0) {
                print("\(s) correspond à l'expression régulière")
            }
        } catch {
            print("Erreur")
        }
    }
}
```