

# RegEx

Les expressions régulières décrivent des sous-ensembles de chaînes de caractères. Souvent peu appréciées pour leur syntaxe parfois alambiquée, elles offrent pourtant de réelles possibilités pour les développeurs et les référenceurs. Les regex (mot-valise issu de *regular expression*) répondent à d'innombrables usages, notamment en matière d'extraction de contenus ou de reconnaissance de chaînes, souvent utiles dans les outils ou programmes préférés des SEO.

- [Nouvelle page](#)

# Nouvelle page

---

## Perfectionnement des systèmes open source ?

Auteur : Dw4rF

---

## REGEX

### 4.1. Késaco ?

- **Regular Expression**
  - Aide **parsing** de fichier
  - Peut être **simple** ou **complexe**, selon le besoin
- 

### 4.2. Bases de compréhension (AND)

- `[]` : **Caractères** (ou type)
  - `{ }` : **Nombre** de caractères
- 

### 4.3. Bases de compréhension (OR)

- `()` : "switch"
  - `|` : Séparateur
- 

### 4.4. Joker(s)

- `.` : N'importe quel caractère (**UN** seul caractère)
  - `*` : N'importe quel caractère, pas de limite de nombre
  - `\` : **Escape** les caractères spéciaux
- 

### 4.5. Position

- `^` : Juste *après* le début de ligne
  - `$` : Juste *avant* la fin de ligne
- 

## 4.5. Practice !

Exemples à matcher :

- `06.xx.xx.xx.xx`
  - `07.xx.xx.xx.xx`
  - `06-xx-xx-xx-xx`
  - `07-xx-xx-xx-xx`
- 

## 4.6. Practice ! - correction

- `06.xx.xx.xx.xx`
- `07.xx.xx.xx.xx`
- `06-xx-xx-xx-xx`
- `07-xx-xx-xx-xx`

Regex : `(06|07)`

---

## 4.6. Practice ! - correction

- `06.xx.xx.xx.xx`
- `07.xx.xx.xx.xx`
- `06-xx-xx-xx-xx`
- `07-xx-xx-xx-xx`

Regex : `(06|07)(\.-)`

---

## 4.6. Practice ! - correction

- `06.xx.xx.xx.xx`
- `07.xx.xx.xx.xx`
- `06-xx-xx-xx-xx`
- `07-xx-xx-xx-xx`

Regex : `(06|07) (\.-)[0-9]{2}`

---