

Module re - descriptif basé sur le livre de Harold Erbin

Syntaxe Tout caractère alphanumérique n'a pas de signification spéciale : A correspond simplement à la lettre A, 1 au chiffre 1, etc. Quant aux principaux symboles spéciaux, ils sont :

- . : désigne n'importe quel caractère ;
 - ^ : indique que le début de la chaîne doit correspondre ;
 - \$: indique que la fin de la chaîne doit correspondre ;
 - {n} : indique que le caractère précédent doit être répété n fois.
 - {n,m} : indique que le caractère précédent doit être répété entre n et m fois.
 - *
 - +
 - ?
- ab et a.

Les positions “fin de ligne” et “début de ligne” ont un sens différent selon si on est ou non en mode **MULTILINE**. Par défaut, le début de ligne signifie le début de la chaîne ; si on veut qu'il signifie le symbole suit un `\n`, il faut ajouter le *flag* `re.MULTILINE` (de même pour la fin de ligne).

L'antislash (`\`) permet d'échapper tous ces caractères spéciaux. Les crochets `[]` permettent d'indiquer une plage de caractère, par exemple `[e-h]` correspondra à e, f, g ou h. Finalement, il reste quelques caractères spéciaux assez utiles :

- `\w` : il correspond à tout caractère alphabétique, c'est à dire qu'il est équivalent à `[a-zA-Z]` ;
- `\W` : il correspond à tout ce qui n'est pas un caractère alphabétique ;
- `\b` : il correspond à la frontière (début ou fin) d'un mot ;
- `\d` : il correspond à tout caractère numérique, c'est à dire qu'il est équivalent à `[0-9]` ;
- `\D` : il correspond à tout ce qui n'est pas un caractère numérique.
- `\s` : il correspond à tout caractère d'espacement, le retour à la ligne compris ;
- `\S` : il correspond à tout ce qui n'est pas un caractère d'espacement.

Utilisation

`re.search(pattern, string)` Cherche le motif dans la chaîne passée en argument et retourne un `MatchObject` si des correspondances sont trouvées, sinon retourne `None`.

`re.match(pattern, string)` Cherche le motif au début de la chaîne `string` passée en argument et retourne un `MatchObject` si des correspondances sont trouvées, sinon retourne `None`.

`m.group(0)` Retourne le motif correspondant à l'expression régulière dont le `MatchObject m` résulte.

`re.split(pattern, string)` Découpe la chaîne `string` selon les occurrences du motif.

```
>>> re.split(r'\W', 'Truth is beautiful, without doubt.')
['Truth', 'is', 'beautiful', '', 'without', 'doubt', '']
```

`re.findall(pattern, string)` Retourne toutes les sous-chaînes de `string` correspondant au motif.

`re.finditer(pattern, string)` Retourne un itérateur contenant toutes les sous-chaînes de `string` correspondant au motif.

`re.sub(pattern, repl, string)` Retourne la chaîne `string` où le motif a été remplacé par `repl`.