

Exemple simple d'expressions régulières					
1 ^{ère} chaîne	m	a	m	a	n
2 ^{ème} chaîne	p	a	p	a	
Expression régulière	[mp]	a	[mp]	a	\$n ?
Signification	m ou (<i>métacaractère crochet</i>) p	seulement a	m ou (<i>métacaractère crochet</i>) p	seulement a	Eventuellement (<i>métacaractère ?</i>) n en fin de chaîne (<i>métacaractère \$</i>)
L'expression régulière générale est : [mp]a[mp]a\$n ?					

Test "Les expressions régulières en python" => "TRUE" / "FALSE"

EXPRESSION	CHAINE	TRUE	FALSE	SOLUTION
GR(.)+S	GRIS	TRUE	FALSE	
GR(.)?S	GRS	TRUE	FALSE	
GRA(.)?S	GRAS	TRUE	FALSE	
GAS(.)?	GRAS	TRUE	FALSE	

1. La méthode `match()` recherche une correspondance au début de la chaîne de caractères.

```
import re
chaîne = 'Chaîne de caractères dans laquelle rechercher le motif'
correspondance = re.match('motif', chaîne)
print (correspondance)
```

Si une correspondance est trouvée, cette méthode renvoie une instance "`MatchObject`" indiquant la position du début et de fin de la correspondance et la sous-chaîne de caractères avec laquelle elle correspond => exemple : `<re.Match object; span=(0, 13), match='motif'>`

=====

2. La méthode `search()` recherche une correspondance dans toute la chaîne de caractères et/ou dans la première ligne du texte.

```
import re
chaîne = 'Chaîne de caractères dans laquelle rechercher le motif'
correspondance = re.search('laquelle', chaîne)
print (correspondance)
```

Si une correspondance est trouvée, cette méthode renvoie une instance "`MatchObject`" => exemple : `<re.Match object; span=(21, 29), match='laquelle'>`

=====

3. Ces deux méthodes renvoient l'opérateur logique "`None`" si aucune correspondance n'est trouvée => l'intérêt de ces deux méthodes est d'effectuer un **test logique** ("vrai" - "faux") avec des **chaînes de caractères**.

Exemples :

```
correspondance = re.match(r"SSBOND", chaque ligne d'un fichier PDB)
if correspondance is not None: # si la recherche en début de ligne trouve le terme SSBOND
=> alors imprimer la ligne donnée
```

```
correspondance = re.search(r"^SSBOND", chaque ligne d'un fichier PDB)
if correspondance is not None: # si la recherche en début de ligne (métacaractère ^) trouve le terme SSBOND
=> alors imprimer la ligne donnée
```