

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°9 : Les expressions régulières

BTS CIEL
2^{ème} année
Page 1 sur 2

Exercice 1 : Validation des plaques d'immatriculation

Vous êtes chargé de créer un programme Python pour valider des numéros de plaques d'immatriculation françaises.



Les critères de validation sont les suivants :

- Le numéro de plaque doit être composé de 7 caractères alphanumériques.
- Les deux premiers caractères doivent être des lettres majuscules suivie d'un tiret -.
- Les trois caractères suivants doivent être des chiffres.
- Les trois derniers caractères doivent être un tiret suivie deux lettres majuscules.

Instructions :

1. Utilisez une expression régulière pour représenter un numéro de plaque d'immatriculation valide en fonction des critères énumérés ci-dessus.
2. Utilisez le module re de Python pour rechercher toutes les occurrences de cette expression régulière dans la liste de numéros de plaques fournies.
3. Affichez les numéros de plaques valides.

Code de départ:

```
import re

# Liste de numéros de plaques d'immatriculation
license_plate_list = ["AB123-CD", "12-BTS-67", "IL-89Z-01", "BT-321-SN"]

# Votre expression régulière ici
pattern = re.compile(r"
```

Conseils :

- Utilisez des classes de caractères, des quantificateurs et des groupes pour gérer les différents formats possibles des numéros de plaques.
- Assurez-vous de prendre en compte les lettres majuscules et les chiffres dans votre expression régulière.

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°9 : Les expressions régulières

BTS CIEL
2^{ème} année
Page 2 sur 2

Exercice 2 : Extraction d'informations d'une trame NMEA

Supposons que vous ayez une trame NMEA stockées dans une chaîne de caractères. Votre tâche est d'extraire des informations spécifiques de cette trame.

Considérez la chaîne suivante contenant une trame NMEA :

```
nmea_data = "$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,*47"
```

Votre tâche est de créer une expression régulière et un programme Python pour extraire les informations suivantes de chaque trame :

1. L'heure (HHMMSS) de la trame GGA.
2. La latitude de la trame GGA.
3. La longitude de la trame GGA.

Instructions:

- Utilisez une ou des expressions régulières pour extraire les informations spécifiées de la trame NMEA.
- Utilisez le module `re` de Python pour rechercher toutes les occurrences dans la chaîne `nmea_data`.
- Affichez les informations extraites.

TRAVAUX PRATIQUES
La programmation en Python
T.P. n°9 : Les expressions régulières

BTS CIEL
2^{ème} année
Page 3 sur 2

Exercice 3 : Validation des Numéros de Téléphone Français

Supposons que vous ayez une liste de numéros de téléphone français stockés dans une chaîne de caractères. Votre tâche est de valider chaque numéro de téléphone selon certains critères.

Les critères pour un numéro de téléphone français valide sont les suivants :

1. Le numéro de téléphone doit être composé de 10 chiffres.
2. Il peut inclure des espaces, des tirets ou des parenthèses pour améliorer la lisibilité.
3. L'indicatif international, le cas échéant, doit commencer par le signe plus (+) suivi de l'indicatif du pays (33 pour la France) et ne doit pas dépasser 15 caractères.

Exemple :

Considérons la chaîne suivante :

```
phone_numbers = "+33 6 12 34 56 78, 0123456789, +33 (0)1 23 45 67 89, invalid_number"
```

Votre tâche est de créer une expression régulière et un programme Python pour filtrer et afficher uniquement les numéros de téléphone français valides.

Instructions :

Utilisez une expression régulière pour représenter un numéro de téléphone français valide en fonction des critères énumérés ci-dessus.

Utilisez le module `re` de Python pour rechercher toutes les occurrences de cette expression régulière dans la chaîne `phone_numbers`.

Affichez les numéros de téléphone français valides.

Code de départ :

```
import re

phone_numbers = "+33 6 12 34 56 78, 0123456789, +33 (0)1 23 45 67 89, invalid_number"

# Votre expression régulière ici
pattern = re.compile(r"") # À compléter

# Utilisez la méthode findall pour obtenir tous les numéros de téléphone valides
valid_phone_numbers = pattern.findall(phone_numbers)

# Affichez les numéros de téléphone valides print("Numéros de
téléphone français valides :", valid_phone_numbers)
```

Conseils :

- Utilisez des classes de caractères, des quantificateurs et des groupes pour gérer les différents formats possibles des numéros de téléphone.
- Assurez-vous de prendre en compte les caractères spéciaux tels que l'espace, le tiret et la parenthèse.
- N'oubliez pas que `findall` renvoie une liste de toutes les occurrences trouvées dans la chaîne.