j10

J10 : Objectifs

- Expressions régulières
- Bases de données
- SQL
- sqlite

Expression régulière

- Utilisation de caractères spéciaux ci-contre, des parenthèses et des quantifieurs
  - Détaillés plus loin

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

Par exemple :

- une adresse ip

```
^[0-9]{1,3}(\.[0-9]{1,3}){3}$
192.168.1.14
```

- Un prénom

```
^([A-Z][a-z]*)(\-[A-Z][a-z]*)?$
Marc-Antoine
```

- Un numéro de téléphone mobile à 10 chiffres

```
^0[67]( (\d){2}){4}$
06 12 68 40 06
```

- Une tautologie

```
^Un (?P<name>(\w)+) est un (?P=name)$
Un chat est un chat
```

Caractères spéciaux

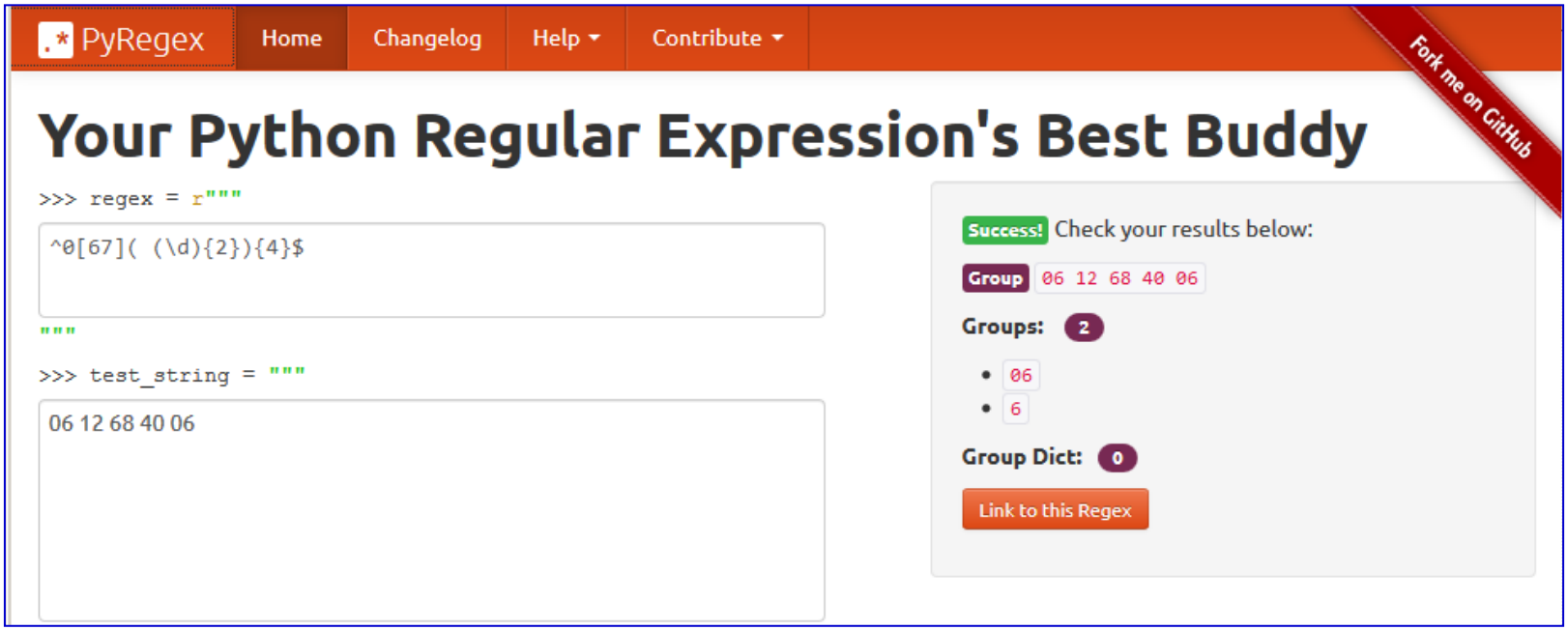
Caractère	Signification	Commentaire
.	tout caractère	
^	début de ligne	à l'intérieur de [ ] signifie "sauf"
$	fin de ligne	ou fin de "segment"
[xyz]	1 caractère parmi x, y, z	[^xyz] tout caractère sauf x, y, z
(ab|cde)	la chaîne ab ou cde	
[0-9]	un chiffre décimal	
\d	un chiffre décimal	
\D	tout sauf chiffre	
\s	un espace	inclut la tabulation, le saut de page, saut de ligne, retour à la ligne
\S	tout sauf espace	
\w	caractère alphanumérique	[a-zA-Z0-9_]
\W	tout sauf alphanumérique	
\	échappement	le caractère qui suit intervient pour lui-même
-	le tiret abrège l'écriture de séquences	a-z pour les lettres min, 0-9 pour les chiffres, A-Z pour les maj

Qualifieurs/Quantifieurs

Qual/Quant…	Signification	Commentaire
{n}	répété n fois	Les quantifieurs s'appliquent au caractère
{n,m}	répété de n à m fois	ou au groupe qui précède
+	répété au moins 1 fois	
*	répété 0 ou plus	
?	optionnel (0 ou 1 fois)	
(abc)	les éléments entre parenthèses constituent un groupe	numérotés dans leur ordre d'apparition
(?P<nom>regex)	Capture un groupe nommé	"nom" peut être réutilisé
(?P=nom)	Utilise un groupe capturé	

Exercice : PyRegex – Pythex est mieux

- Rendez vous sur www.pyregex.com ou regex101
- Mettez à l'épreuve les exemples proposés



Expressions régulières

- *Regular expressions*
- *regex*

```
import re


re.match(pattern, string, flags = 0)
```

Les Regex en Python

- Vérifier la correspondance
  - *matching*

```
import re
if re.match(r"BL(.)*", "BLANC"):
    # trouvé
    …
```

- Trouver toutes les occurrences

```
hits = re.findall(r"BL(.)*", "BLEU SUR BLANC"):
```

- Récupérer les groupes

```
phrase = "Sophie aime Rachid"
pattern = "^(?P<sujet>\w+) (?P<verbe>\w+)
(?P<complement>\w+)$"
m = re.search(pattern, phrase)
if m is not None:
   print(m.group('sujet'))
   print(m.group('verbe'))
   print(m.group('complement'))
```

- Compiler une regex

```
regex = re.compile(pattern)
m =  regex.match(phrase)
…
```

http://apprendre-python.com/page-expressions-regulieres-regular-python

Exercice N° Tel

- On veut vérifier des numéros de téléphone mobiles Français
  - Commencer par 06 ou 07
  - Avoir 10 chiffres en tout

  Par exemple : 0612345678

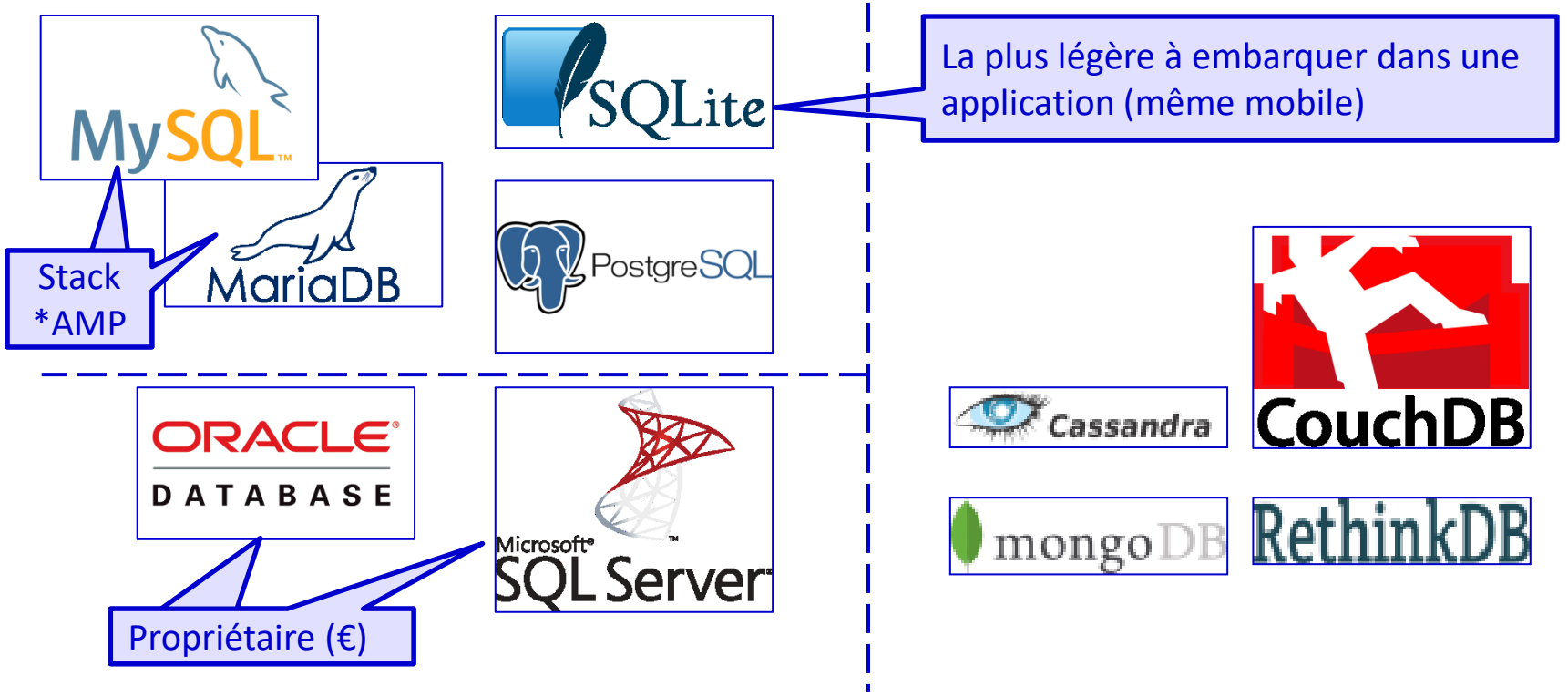
- Faire une fonction qui vérifie si une chaîne de caractères donnée en argument est ou non, un numéro de téléphone bien formé en utilisant une expression régulière

- Lire une chaine de caractères au clavier, dire si c'est un numéro de téléphone mobile correct et afficher le numéro par groupes de 2 chiffres séparés par des espaces, quelque soit la façon dont le numéro a été tapé.
  Par exemple : 06 12 3 4 5678
- Options : admettre les écritures avec le code Pays, telles que

  + 33 6 12 345678

  +33 (0)6 12345678

Bases de données



- Bases de données relationnelles
- Langage SQL
  - décrire et manipuler des tables

- NOSQL (Not Only SQL)
  - Documents plutôt que tables
  - JavaScript est le langage privilégié

SQL

- Structured Query Language
  - SELECT … FROM … WHERE …
  - UPDATE … SET … WHERE …
  - DELETE FROM … WHERE …
  - INSERT INTO … VALUES …

- Basé sur l'algèbre relationnelle définie par E.F Codd (1969), puis développé chez IBM et Oracle (à l'époque : Relational Software).

- SQL est un standard ANSI, ISO

- De légères variations de la syntaxe d'un SGBD à l'autre

  (Systèmes de Gestions de Bases de Données) *DBMS (Data Base Management Systems)*

SQL

- Créer une base de données

```
CREATE DATABASE maBase;        // pas en sqlite3
```

- Détruire une base de données

```
DROP DATABASE maBase;          // pas en sqlite3
```

- Créer une table

```
CREATE TABLE IF NOT EXISTS user_t (
   id INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,
   name_f TEXT,
   room_f INTEGER);
```

- Insérer une ligne dans une table

```
INSERT INTO user_t (name_f, room_f)
VALUES ('Georges', 1);
```

- Sélectionner les lignes d'une table

```
SELECT * FROM user_t  WHERE room_f=1;
```

- Mettre à jour une ligne dans une table

```
UPDATE user_t  SET room_f=2
WHERE name_f='Georges';
```

- Effacer une ligne

```
DELETE FROM user_t WHERE name_f='Georges';
```

- Effacer toutes les lignes

```
TRUNCATE user_t;
```

- Détruire une table

```
DROP TABLE user_t;
```

sqlite 1/3

- Ouvrir une base de données

```
import sqlite3
cnxn = sqlite3.connect('mabase.db')
```

- Récupérer un "curseur"

```
cursor= cnxn.cursor()
```

- Envoyer une requête

```
cursor.execute("""DROP TABLE IF EXISTS user_t""")
```

- Créer une table

```
cursor.execute("""
  CREATE TABLE IF NOT EXISTS user_t (
    id INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,
    name_f TEXT,
    room_f INTEGER
  )
""")
```

- Fermer la base de données

```
cnxn.close()
```

http://apprendre-python.com/page-database-data-base-donnees-query-sql-mysql-postgre-sqlite

sqlite 2/3

- Commit permet de synchroniser la base pour les autres connexions en cours à la même base

```
cnxn.commit()
```

- Peupler une table
  - Insérer un tuple

```
cursor.execute("""
    INSERT INTO user_t (name_f, room_f)
    VALUES (?, ?)
""",('Georges', 1))
```

  - Insérer une liste de tuples

```
users = []
users.append(('Bertrand', 1))
users.append(('Philippe', 2))
users.append(('Marie', 3))
users.append(('Zoe', 3))
print("Inserting many")
cursor.executemany("""
    INSERT INTO user_t (name_f, room_f)
    VALUES (?, ?)""",    users)
```

sqlite 3/3

- Voir le contenu d'une table par une requête SELECT

```
cursor.execute("""
    SELECT id, name_f, room_f FROM user_t""")
rows=cursor.fetchall()
for row in rows:
    print(row)
```

- Rechercher un tuple particulier
  - Ici, on cherche dans quelle pièce est Marie

```
name='Marie'
cursor.execute("""
    SELECT room_f
    FROM user_t
    WHERE name_f=?
""", (name,))
room=cursor.fetchone()
```

- Effacer un tuple

```
cursor.execute("""
    DELETE
    FROM user_t
    WHERE name_f='Georges'
""")
```

Exercice : sqlite

- Sur la base donnée en exemple, tenter l'insertion d'un user qui existe déjà.
- Observer l'exception levée
- Encadrer la tentative d'insertion d'un try: … except…. approprié
- Vérifier que les insertions de doublons sont désormais refusées sans interrompre le cours de l'exécution.

Merci !

- Restons en contact :
  - Georges Georgoulis – ggeorgoulis@alteractifs.org – 06 12 68 40 06

  - Alteractifs

    Coopérative d'activité et d'entrepreneurs www.alteractifs.org