

# Corrigés de la semaine 6

regex-pythonid - Semaine 6 Séquence 6

```
1 # un identificateur commence par une lettre ou un underscore
2 # et peut être suivi par n'importe quel nombre de
3 # lettre, chiffre ou underscore, ce qui se trouve être \w
4 # si on ne se met pas en mode unicode
5 regex_pythonid = "[a-zA-Z_]\w*"
```

regex-pythonid - Semaine 6 Séquence 6

```
1 # on peut aussi bien sûr l'écrire en clair
2 regex_pythonid2 = "[a-zA-Z_][a-zA-Z0-9_]*"
```

regex-specials - Semaine 6 Séquence 6

```
1 # il faut commencer par exactement 2 underscores
2 # donc le caractère suivant doit être une lettre
3 # ensuite on peut mettre ce qu'on veut comme alphanumérique,
4 # mais avant les deux derniers underscores on ne peut pas avoir
5 # un underscore
6 # enfin pour traiter le cas où la partie centrale est réduite
7 # à un seul caractère, on met une option - avec ()?
8 regex_specials = "__[a-zA-Z](\w*[a-zA-Z0-9])?__"
```

```

1  # en ignorant la casse on pourra ne mentionner les noms de protocoles
2  # qu'en minuscules
3  i_flag = "(?i)"
4
5  # pour élaborer la chaine (proto1|proto2|...)
6  protos_list = ['http', 'https', 'ftp', 'ssh', ]
7  protos      = "(?P<proto>" + "|".join(protos_list) + ")"
8
9  # à l'intérieur de la zone 'user/password', la partie
10 # password est optionnelle - mais on ne veut pas le ':' dans
11 # le groupe 'password' - il nous faut deux groupes
12 password    = r"(:(?P<password>[^\:]+))?"
13
14 # la partie user-password elle-même est optionnelle
15 user        = r"((?P<user>\w+){password}@)?".format(**locals())
16
17 # pour le hostname on accepte des lettres, chiffres, underscore et '.'
18 # attention à backslaher . car sinon ceci va matcher tout y compris /
19 hostname    = r"(?P<hostname>[\w\.]*)"
20
21 # le port est optionnel
22 port        = r"(:(?P<port>\d+))?"
23
24 # après le premier slash
25 path        = r"(?P<path>.*)"
26
27 # on assemble le tout
28 regex_url = i_flag + protos + "://" + user + hostname + port + '/' + path

```