



Réseaux pour ingénieurs GLO-2000

Envoi de courriels avec Python

Professeur responsable:

Ronald Beaubrun
Ronald.Beaubrun@ift.ulaval.ca

Responsables des travaux pratiques:

Louis Fortier-Dubois et Jean-Alexandre Beaumont
glo2000a17@gmail.com

1 Introduction aux expressions régulières

Dans le jargon informatique, on désigne par « expressions régulières » (ou rationnelles) une syntaxe utilisée pour définir des motifs recherchés dans un texte ou une chaîne de caractères. Les expressions régulières sont communément désignées par le diminutif « Regex » pour « Regular Expression ». Nous nous en servons par exemple pour valider des adresses courriel.

En python, l'utilisation d'expressions régulières se fait généralement à l'aide de la librairie « re », qui permet de détecter des motifs à l'aide de « `re.search(r"<regex>", var)` ». `<regex>` doit être une chaîne de symboles parmi les suivants :

.	un caractère unique (n'importe lequel)
\s	un blanc (espace, tabulation)
\S	un caractère non-blanc (le contraire du précédent)
\d	un chiffre (0-9)
\D	un non-chiffre
\w	un caractère alphanumérique (a-z, A-Z, 0-9, _)
\W	un non-alphanumérique
[aeiou]	n'importe quel caractère de l'ensemble entre crochets
[^aeiou]	n'importe quel caractère sauf ceux de l'ensemble entre crochets
(foo bar baz)	n'importe laquelle des alternatives proposées
^	le début d'une chaîne de caractères
\$	la fin d'une chaîne de caractères

Des quantificateurs peuvent être utilisés pour indiquer combien des éléments précédents vous désirez, un élément désignant aussi bien un caractère littéral qu'un des méta-caractères énumérés plus haut, ou encore un groupe de caractères ou de méta-caractères entre parenthèses.

*	zéro ou plus
+	un ou plus
?	zéro ou un
{3}	exactement 3 fois l'élément précédent
{3,6}	entre 3 et 6 fois l'élément précédent
{3,}	3 ou plus des éléments précédents

1. Créez un fichier python et nommez-le *exercice1.py*
2. Tapez et exécutez les lignes suivantes :

```
import re

var = input("Donnez une valeur a var : ")

if re.search(r"World", var):
    print("var contient le mot World")
else:
    print("var ne contient pas le mot World")

if re.search(r".", var):
    print("var contient au moins un caractere")
else:
    print("var ne contient pas de caracteres")

if re.search(r"^\.?$", var):
    print("var contient un ou zero caractere")
else:
    print("var contient plus d'un caractere")

if re.search(r"\d{3}", var):
    print("var contient trois chiffres de suite")
```

```

else:
    print("var ne contient pas trois chiffres de suite")

if re.search(r"[a-zA-Z0-9]{3,12}", var):
    print("var contient entre 3 et 12 caracteres alphanumeriques de suite")
else:
    print("var ne contient pas entre 3 et 12 caracteres alphanumeriques de suite")

if re.search(r"[TWHQ]ello", var):
    print("var contient un T, un W, un H ou un Q suivi de ello")
else:
    print("var ne contient pas un T, un W, un H ou un Q suivi de ello")

if re.search(r"foo|bar|ello", var):
    print("var contient foo, bar ou ello")
else:
    print("var ne contient pas foo, bar ou ello")

if re.search(r"^(Hello)", var):
    print("var commence par Hello")
else:
    print("var ne commence pas par Hello")

if re.search(r"!!![0-9]{3}$", var):
    print("var se termine par !!! suivi de 3 chiffres")
else:
    print("var ne se termine pas par !!! suivi de 3 chiffres")

if re.search(r"^H.*[0-9]{3}$", var):
    print("var commence par un H et se termine par 3 chiffres")
else:
    print("var ne commence pas par un H ou ne se termine pas par 3 chiffres")

```

3. Amusez-vous à entrer différentes chaînes de caractères, dont « HelloWorld!!!345 ».

4. Ajoutez les lignes suivantes, puis entrez votre adresse courriel :

```

if re.search(r"^[^@]+@[^@]+\.[^@]+$", var):
    print("var est un courriel valide")
else:
    print("var n'est pas un courriel valide")

```

5. En principe, quand un symbole de l'expression régulière correspond à un symbole de la chaîne, la suite de l'analyse se fait à partir du symbole suivant de la chaîne, sans jamais revenir en arrière. Par exemple, l'expression « `[a-z]\d{3}` » reconnaît « a123 » mais pas « 123a », car bien que le « a » corresponde à « `[a-z]` », il est trop tard pour reconnaître les trois chiffres puisqu'ils étaient avant.

6. Il est toutefois possible de définir des assertions avant (*lookahead*). Cela permet aussi de repérer un motif dans la chaîne, mais une fois fait, l'analyse retourne à la position de l'assertion. La syntaxe est « `(?=<regex>)` ». Ajoutez les lignes suivantes et entrez diverses chaînes de caractères :

```

if re.search(r"(?=[^a-z]*[a-z])(?=[^d]*\d{3})", var):
    print("var contient une lettre et trois chiffres de suite (peu importe l'ordre)")
else:
    print("var ne contient pas de lettre ou trois chiffres de suite")

```

2 Script d'envoi de courriel

1. Créez un fichier python et nommez-le *exercice2.py*
2. Tapez et exécutez les lignes suivantes :

```
import smtplib
from email.mime.text import MIMEText

#remplissage des champs par l'utilisateur
mailfrom = input("Mail from : ")
rcptto = input("Rcpt to : ")
subject = input("Subject : ")

print("Data : ")
text = ""
temp = input()
while temp != ".":
    text += temp + "\n"
    temp = input()

#creation d'un objet courriel avec MIMEText
msg = MIMEText(text)
msg["From"] = mailfrom
msg["To"] = rcptto
msg["Subject"] = subject

#envoi du courriel grace au protocole SMTP et au serveur de l'universite Laval
try:
    smtpConnection = smtplib.SMTP(host="smtp.ulaval.ca", timeout=10)
    smtpConnection.sendmail(mailfrom, rcptto, msg.as_string())
    smtpConnection.quit()
    print("Message envoye")
except:
    print("L'envoi n'a pas pu etre effectue. ")
```

3. Envoyez un courriel (*Rcpt to*) à votre propre adresse et vérifiez que vous le recevez.
Note : Pour accéder au serveur *smtp.ulaval.ca*, vous devez soit être sur le campus et être connecté au wi-fi *Eduroam*, soit vous connecter au VPN de l'Université Laval : <https://www.dti.ulaval.ca/connexion-au-reseau-de-lul/reseau-distance>
4. Comme le champ *Mail from* accepte n'importe quelle adresse valide, il est très facile de *spoof* son adresse courriel (vous faire passer pour quelqu'un d'autre). Ceci dit, **ne le faites pas!**

3 Serveur qui envoie des courriels

1. Créez un fichier python et nommez-le *exercice3.py*
2. Tapez et exécutez les lignes suivantes :

```
import smtplib, re, socket, optparse, sys
from email.mime.text import MIMEText

#choisissez le port avec l'option -p
parser = optparse.OptionParser()
parser.add_option("-p", "--port", action="store", dest="port", type=int, default=1337)
port = parser.parse_args(sys.argv[1:])[0].port
```

```

#creation d'un socket
serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serversocket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
serversocket.bind(("localhost", port))

#demarre le socket
serversocket.listen(5)
print("Listening on port " + str(serversocket.getsockname()[1]))

i = 0

while True:
    #un client se connecte au serveur
    #s est un nouveau socket pour interagir avec le client
    (s, address) = serversocket.accept()

    #affichage du nombre de connection au serveur
    i += 1
    print (str(i) + "e connexion au serveur")

    #message de bienvenue
    msg = "Bienvenue dans mon serveur. \nA qui dois-je envoyer un courriel? "
    s.send(msg.encode())

    #reception du courriel et verification qu'il est valide
    emailAddress = s.recv(1024).decode()
    while not re.search(r"^[^@]+@[^@]+\.[^@]+$", emailAddress):
        msg = "Saisissez une adresse courriel valide : "
        s.send(msg.encode())
        emailAddress = s.recv(1024).decode()

    #creation du courriel
    courriel = MIMEText("Ce courriel a ete envoye par mon serveur de courriel")
    courriel["From"] = "exercice3@glo2000.ca"
    courriel["To"] = emailAddress
    courriel["Subject"] = "Exercice3"

    #envoi du courriel
    try:
        smtpConnection = smtplib.SMTP(host="smtp.ulaval.ca", timeout=10)
        smtpConnection.sendmail(courriel["From"], courriel["To"], courriel.as_string())
        smtpConnection.quit()
        msg = "Le courriel a bien ete envoye! "
        s.send(msg.encode())
    except:
        msg = "L'envoi n'a pas pu etre effectue. "
        s.send(msg.encode())

    msg = "Au revoir!\n"
    s.send(msg.encode())

    s.close()

```

3. Dans une autre console, utiliser ncet pour vous connecter au serveur : « ncet localhost 1337 », et essayez de vous envoyer un courriel à l'aide du serveur « exercice3.py ».