Boucles

July 22, 2014

0.1 Boucles inconditionnelles

0.1.1 Codage d'un message secret

Méthode inverse

```
In [17]: # -*- coding: utf-8 -*-
         message = u"Mon message secret."
         secret = ""
         for c in message:
             secret = c + secret
         print secret
```

```
.terces egassem noM
```

Un message très facile à déchiffer !!!

Méthode style Caesar

Remplacer une lettre par une autre avec un décalage constant vers la droite

```
In [2]: message = u"Mon message secret."
        key = 5
        secret = ""
        for c in message:
            c = ord(c) + key
            secret += chr(c) #unichr() pour l'unicode
        print secret
```

```
Rts%rjxxflj%xjhwjy3
```

Des caractères inconnus peuvent apparaître en raison de problème d'encodage des caractères

Une solution est de borner l'intervalle des caractères

```
In [5]: message = u"Mon message secret."
        key1 = 123
        mini, maxi = 33, ord('z')
        secret1 = ""
        for c in message:
            c = ord(c) + key1
            if c > maxi:
                c = mini + (c-maxi)%(maxi-mini)
            secret1 += unichr(c)
        print secret1
```

```
o87B6.<<*0.B<.,;.=P
```

0.1.2 Décodage

On a besoin de la clé de codage

```
In [20]: decodage = ""
         print u"message codé: "+ secret
         for c in secret:
             c = ord(c) - key
             decodage += chr(c)
         print decodage == message #opérateur booléen
         print "texte du message: " + decodage
```

```
message codé: Rts%rjxxflj%xjhwjy3
True
texte du message: Mon message secret.
```

0.1.3 Conserver les espaces

```
In [50]: #un texte très long
         message = "Letter frequencies, like word frequencies, tend to vary, both by writer and by subj
         +"One cannot write an essay about x-rays without using frequent Xs,"\
         +"and the essay will have an idiosyncratic letter frequency"\
         +"if the essay is about the frequent use of x-rays to treat zebras in Qatar."\
         +"Different authors have habits which can be reflected in their use of letters."\
         +"Hemingway's writing style, for example, is visibly different from Faulkner's."\
         +"Letter, bigram, trigram, word frequencies, word length, and sentence length can be calculate
         +"and used to prove or disprove authorship of texts, even for authors whose styles are not so
         #message = u"Mon message secret."
         key = 5
         mini, maxi = 33, ord('z')
         secret = ""
         for c in message:
             if c != " ":
                 c = ord(c) + key
                 if c > maxi:
                     c = mini + (c-maxi)%(maxi-mini)
                 secret += unichr(c)
             else:
                 secret += c
         print secret
```

```
Qjyyjw kwjvzjshnjx1 qnpj #twi kwjvzjshnjx1 yjsi yt "fw%1 gtym g% #wnyjw fsi g% xzgojhy3Tsj hfssty #wnyj
```

0.1.4 Piratage de la clé de codage

Solution force brute : on teste toutes les clés possibles

```
In [3]: for key in range(10):
            message=""
            for c in secret:
                if c != " ":
                    c = ord(c) - key
                    message += chr(c)
                else:
                    message += c
            print message
```

```
Rts rjxxflj xjhwjy3
Qsr qiwweki wigvix2
Prq phvvdjh vhfuhw1
Oqp oguucig ugetgv0
Npo nfttbhf tfdsfu/
Mon message secret.
Lnm ldrr‘fd rdbqds-
Kml kcqq_ec qcapcr,
Jlk jbpp^db pb‘obq+
Ikj iaoo]ca oa_nap*
```

Solution plus élégante basée sur la fréquence des lettres

```
In [54]: oc = 0
         lettre = ""
         for c in secret:
             if c != " ":
                 if secret.count(c) > oc:
                     oc = secret.count(c) #nombre d'occurence du caractère c
                     lettre = c
                     print lettre, oc
         #print lettre, oc
         key = abs(ord(lettre) - ord('e')) # français max(freq)=e (14%) et anglais max(freq)=e (13%)
         print u"clé = " , key
```

```
Q 2
j 74
clé =  5
```

Différence de complexité pour trouver la clé:

```
In []:
```