

## Corrigé – Les chaînes de caractères

```
1  # Ex 1
2  s = "a b"*2 + "ba" + "b"*3
3  print(s, len(s)) # affiche 'a ba bbabbb' 11
4
5  #####
6
7  # Ex 2
8  # La fonction convert renvoie la même lettre en majuscule
9
10 #####
11
12 # Ex 3
13 def rang(l):
14     return ord(l) - 97
15
16 def lettre(r):
17     return chr(r + 97)
18
19 #####
20
21 # Ex 4
22 s = "abn\\nn\\\\"
23 print(len(s)) # affiche 7
24
25 #####
26
27 # Ex 5
28 s = "abcdefghij"
29 n = len(s)
30 print(s[1], s[5]) # affiche 'b' 'f'
31 s = s*2
32 print(n) # affiche 10
33 print(s[12]) # affiche 'c'
34
35 #####
36
37 # Ex 6
38 ch = "abcdef"
39 n = len(ch)
40 x = ch[n-1]
41 # ou
42 x = ch[len(ch)-1]
43 # ou
44 x = ch[-1]
45
46 #####
47
48
```

```

49  # Ex 7
50
51  def est_numerique(car):
52      pt_code = ord(car)
53      if pt_code >= 48 and pt_code <= 57:
54          return True
55      return False
56
57  def est_numerique(car):
58      return ord(car) >= 48 and ord(car) <= 57
59
60  res1 = est_numerique("k")
61  print(res1)
62  res2 = est_numerique("4")
63  print(res2)
64
65  #####
66
67  # Ex 8
68  def est_une_date_valide(chaine):
69      if len(chaine) != 10:
70          return False
71      if chaine[2] != "/" or chaine[5] != "/":
72          return False
73      for idx in range(10):
74          if idx != 2 and idx != 5:
75              if not est_numerique(chaine[idx]):
76                  return False
77      return True
78
79  #####
80
81  # Ex 9
82  def affiche_coordonnées(x, y):
83      print("Ce point a pour coordonnées (%s, %s)." % (x, y))
84
85  #####
86
87  # Ex 10
88  s = "abcde"
89  for idx in range(len(s)):
90      print(idx, s[idx])
91  # 0 a
92  # 1 b
93  # 2 c
94  # 3 d
95  # 4 e
96
97  #####
98
99
100

```

```

101  # Ex 11
102  def nb_e(s):
103      compteur = 0
104      for caract in s:
105          if caract == "e":
106              compteur +=1
107      return compteur
108
109  #####
110
111  # Ex 12
112  def est_binaire_valide(s):
113      for c in s:
114          if c != "0" and c != "1":
115              return False
116      return True
117
118  print(est_binaire_valide("001010"))
119
120  #####
121
122  # Ex 13
123  # 1)
124  def valeur_decimale(s):
125      n = len(s)
126      somme = 0
127      for idx in range(n):
128          if s[idx] == "1":
129              somme = somme + 2**(n-1-idx)
130      return somme
131
132  print(valeur_decimale("00010"))
133
134  # 2)
135  def valeur_decimale_2(s):
136      n = len(s)
137      somme = 0
138      for idx in range(n):
139          c = s[idx]
140          pt_code = ord(c)
141          if pt_code >= 48 and pt_code <= 57:
142              digit = int(c)
143          elif 65 <= pt_code <= 70:
144              digit = pt_code - 55
145          somme = somme + digit*16**(n-1-idx)
146      return somme
147
148  print(valeur_decimale_2("45EA"))
149
150  #####
151
152

```

```

153 # Ex 14
154 def est_palindrome(s):
155     n = len(s)
156     for idx in range(n//2):
157         if s[idx] != s[n-1-idx] : # ou s[idx] != s[-idx-1]
158             return False
159     return True
160
161 #####
162
163 # Ex 15
164 def somme_chiffres(n):
165     s = str(n)
166     somme = 0
167     for c in s:
168         somme += int(c)
169     return somme
170
171 def somme_tous_les_chiffres(n):
172     somme = 0
173     for k in range(n):
174         somme += somme_chiffres(k)
175     return somme
176
177 #####
178
179 # Ex 16
180 chaine = "abc"
181 s = ""
182 for char in chaine:
183     s = s + char + s
184 print(s) # affiche 'abacaba'
185
186 #####
187
188 # Ex 17
189 def str_filter(s, mask):
190     res = ""
191     for carac in s:
192         if carac not in mask:
193             res = res + carac
194     return res
195
196 #####
197
198 # Ex 18
199 def a_l_envers(s):
200     res = ""
201     for c in s:
202         res = c + res
203     return res
204

```

```

205 #####
206
207 # Ex 19
208 def chiffre_vigenere(msg, key):
209     res = ""
210     n = len(key)
211     idx = 0
212     for c in msg:
213         decalage = rang(key[idx])
214         r = rang(c)
215         res = res + lettre((r + decalage) % 26)
216         idx = (idx + 1) % n
217     return res
218
219 def dechiffre_vigenere(msg, key):
220     res = ""
221     n = len(key)
222     idx = 0
223     for c in msg:
224         decalage = rang(key[idx])
225         r = rang(c)
226         res = res + lettre((r + 26 - decalage) % 26)
227         idx = (idx + 1) % n
228     return res
229
230 print(chiffre_vigenere("lyceelangevinwallon", "nsi"))
231 print(dechiffre_vigenere("yqkrwtnfornqaoiydwa", "nsi"))
232
233 #####
234
235 # Ex 20
236 reader = open("encodage.txt", "r", encoding = "latin-1" )
237 s = reader.read()
238 reader.close()
239 writer = open("encodage-utf8.txt", "w", encoding="utf-8")
240 writer.write(s)
241 writer.close()

```