

Quizz_2022_07

April 1, 2022

Ecrire une fonction `valeurb()` qui prend en paramètre une string (ne contenant que des 0 et des 1) et qui retourne la valeur de cette string interprétée comme un nombre écrit en base 2.

Exemples:

- `valeurb("101")` renvoie 5 ($1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$)
- `valeurb("1100")` renvoie 12 ($1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$)
- `valeurb("1001")` renvoie 9 ($1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$)

```
[8]: def valeurb(s):
      val = 0
      for i in range(len(s)):
          val += 2**i if s[len(s) - (i+1)] == '1' else 0
      return val
```

```
[9]: # on peut faire un calcul direct
      # avec la valeur entière du caractère '0' ou '1'
      def valeurb_int(s):
          val = 0
          for i in range(len(s)):
              val += 2**i * int(s[len(s) - (i+1)])
          return val
```

```
[11]: for m in ('01', '1010', '1000', '0010', '100001', '1001'):
        print(valeurb(m), valeurb_int(m), end= ' / ')
```

1 1 / 10 10 / 8 8 / 2 2 / 33 33 / 9 9 /

```
[5]: # pour visualiser le calcul
      def print_valeur(s):
          val = 0
          print("La chaîne %8s vaut " % s, end = '')
          for i in range(len(s)):
              val += 2**i if s[len(s) - (i+1)] == '1' else 0
              print("%s x 2^%d" % (s[len(s) - (i+1)], i), end = ' + ')
          print(" = %d" % val)
```

```
[6]: for m in ('01', '1010', '1000', '0010', '100001', '1001'):
        print_valeur(m)
```

La chaîne 01 vaut $1 \times 2^0 + 0 \times 2^1 + \dots = 1$
 La chaîne 1010 vaut $0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + \dots = 10$
 La chaîne 1000 vaut $0 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + \dots = 8$
 La chaîne 0010 vaut $0 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + \dots = 2$
 La chaîne 100001 vaut $1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 0 \times 2^4 + 1 \times 2^5 + \dots = 33$
 La chaîne 1001 vaut $1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + \dots = 9$

```
[14]: # Illustration de quelques cas possibles pour la gestion des indices:
# cas A: i in [0..N[ : indice = i ; puissance = N-(i+1)
# cas B: i in [1..N+1[ : indice = i-1 ; puissance = N-i
# cas C: i in [N..0[ : indice = N-i ; puissance = i-1
# cas D: i in [N-1..-1[ : indice = N-(i+1) ; puissance = i
def valb_A(s):
    val, N = 0, len(s)
    for i in range(N):
        val += int(s[i]) * 2**(N-(i+1))
    return val
def valb_B(s):
    val, N = 0, len(s)
    for i in range(1,N+1):
        val += int(s[i-1]) * 2**(N-i)
    return val
def valb_C(s):
    val, N = 0, len(s)
    for i in range(N,0,-1):
        val += 2**(i-1) * int(s[N-i])
    return val
def valb_D(s):
    val, N = 0, len(s)
    for i in range(N-1,-1,-1):
        val += 2**i * int(s[N-(i+1)])
    return val
```

```
[15]: for m in ('01', '1010', '1000', '0010', '100001', '1001', '101001', '00010000'):
    for f in (valb_A, valb_B, valb_C, valb_D):
        print(m,f(m))
```

01 1
 01 1
 01 1
 01 1
 1010 10
 1010 10
 1010 10
 1010 10
 1000 8
 1000 8