# 0.1 1-NSI-Dictionnaire

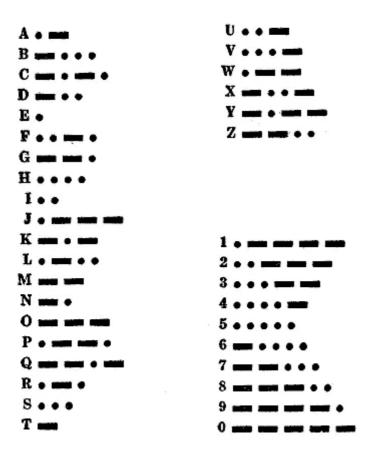
## 1. Autour du code morse...

L'alphabet morse, ou code morse, est un code permettant de transmettre un texte à l'aide de séries d'impulsions courtes et longues. Inventé en 1835 par Samuel Morse pour la télégraphie, ce code assigne à chaque lettre, chiffre et signe de ponctuation une combinaison unique de signaux intermittents. Considéré comme le précurseur des communications numériques, le code morse a depuis le 1er février 1999 été délaissé au profit d'un système satellitaire pour les communications maritimes.

C'est en 1838 que naît l'alphabet " morse " que nous connaissons. Deux types d'impulsions sont utilisés. Les impulsions courtes (notées " . ", point) qui correspondent à une impulsion électrique de 1/4 de seconde et les longues (notées " - ", trait) à une impulsion de 3/4 de seconde.

#### INTERNATIONAL MORSE CODE

- 1. A dash is equal to three dots.
- 2. The space between parts of the same letter is equal to one dot.
- 3. The space between two letters is equal to three dots.
- 4. The space between two words is equal to five dots.



#### Par exemple:

```
-°-° --- -°°° / -- --- °-°°°°°° C O D E (espace) M O R S E
```

On représente le code morse à l'aide d'un dictionnaire, on ne s'intéresse qu'aux lettres en majuscules non accentuées. Pour l'espace on utilise le slash (par exemple).

```
morse = {' ': '/', 'E': '°', 'I': '°°', 'S': '°°°', 'H': '°°°°', 'V': '°

°°-', 'U': '°°-', 'F': '°°-°', 'A': '°-', 'R': '°-°', 'L': '°-°°', 'W':

'°--', 'P': '°--°', 'J': '°---', 'T': '-', 'N': '-°', 'D': '-°°', 'B':

'-°°°', 'X': '-°°-', 'K': '-°-', 'C': '-°-°', 'Y': '-°--', 'M': '--',

'G': '--°', 'Z': '--°°', 'Q': '--°-', 'O': '---'}
```

#### 1.1.1 Les principales méthodes sur les dictionnaires

```
morse = {' ': '/', 'E': '°', 'I': '°°', 'S': '°°°', 'H': •••
°°°', 'V': '°°°-', 'U': '°°-', 'F': '°°-°', 'A': '°-', 'R':
In [ ]:
          '°-°', 'L': '°-°', 'W': '°--', 'P': '°--°', 'J': '°---',
          'T': '-', 'N': '-°', 'D': '-°°', 'B': '-°°°', 'X': '-°°-',
          'K': '-°-', 'C': '-°-°', 'Y': '-°--', 'M': '--', 'G': '--
          °', 'Z': '--°', '0': '--°-', '0': '---'}
         # afficher la valeur d'une clé
         print(morse['S'])
         # tester la présence d'une clé
         if 'S' in morse.keys():
              print('Yes')
         # tester la présence d'une valeur
         if '°°°' in morse.values():
              print('Yes')
         #tester la présence d'une clé
         if 'S' in morse:
              print('Yes')
         # afficher les clés
         print(morse.keys())
         # afficher les couples (clé, valeur)
         print(morse.items())
         # itérer sur les clés
         for keys in morse:
              print(keys, end=" ")
         print(' ')
         # accéder à une clé à partir d'une valeur
         for key, value in morse.items():
              if value == '°°°':
                  print(key)
```

#### 1.1.2 À faire 1:

Écrire une fonction encode\_lettre(lettre) qui prend en paramètre une lettre et le dictionnaire morse et qui renvoie son code morse.

Exemple de résultat attendu :

# à faire 1

```
print(encode_lettre('S', morse))

Affiche:

ooo

In []: def encode_lettre(lettre, morse):
```

#### 1.1.3 À faire 2:

Rajouter au dictionnaire la clé "" avec comme valeur ", qui nous servira à séparer les lettres.

**Rappel:** Pour ajouter un couple (clé,valeur) à un dictionnaire, il suffit d'exécuter mon\_dico[clé]=valeur

```
In []: # à faire 2
```

### 1.1.4 À faire 3 :

Modifier la fonction précédente pour qu'elle renvoie la lettre codée et une étoile.

Exemple de résultat attendu :

```
print(encode_lettre('S', morse))
```

Affiche:

000\*

```
In []: def encode_lettre(lettre, morse):
# à faire 3
```

### 1.1.5 À faire 4:

Écrire une fonction encode\_mots(mots, morse) qui prend en paramètre un message et le dictionnaire et qui renvoie le message codé en morse.

Exemple de résultat attendu :

#### 1.1.6 À faire 5:

Écrire une fonction decode\_mots(code, morse) qui prend en paramètre un message codé et le dictionnaire et qui renvoie le message décodé.

*Aide*: On pourra utiliser la méthode list(phrase.split(caractère)) qui permet de lister les mots d'une phrase en repérant le caractère comme séparateur.

Exemple ce code:

```
phrase = "hello world"
print(list(phrase.split(" ")))
print(list(phrase.split("o")))
```

#### affiche:

```
['hello', 'world']
['hell', 'w', 'rld']

In []: def decode_mots(code, morse):
# à faire 5
```

### 1.1.7 À faire 6:

#### Décoder le message suivant :

```
In []: # à faire 6
```