

# Représentation du texte

test

1ere NSI

- Pourquoi encoder le texte ?
- Comment l'encoder ?
- Quels soucis ?

- Pourquoi encoder le texte ?
  - Comment l'encoder ?
  - Quels soucis ?
- 

## Objectifs:

- Identifier l'intérêt des différents systèmes d'encodage.
- Convertir un fichier texte dans différents formats d'encodage.

- 1 L'encodage du texte sur machine
- 2 Un encodage déjà utilisé ?

# L'intérêt de l'encodage

- Permet de définir comment encoder et décoder une information.
- Permet d'avoir un standard que l'on peut utiliser pour partager des informations.

## 3 normes d'encodage

- Nous allons voir 3 normes d'encodage communes.
- Comment encoder une information en accord a ces normes.
- Comment récupérer l'information, la décoder.
- Les normes que nous allons voir sont: ASCII, ISO-8859-1, Unicode

- Une des premières normes.
- Utilisée dans la majorité des langages de programmation.
- 128 caractères (soit 7 bits).
- Suffisante pour l'anglais.

Une annexe est fourni sur e-lyco montrant la table des caractères.

- 256 caractères (soit 8 bits).
- Suffisante pour la majorité des langues latines.
- Quasiment suffisante pour le français, il manque uniquement "œ".

- 1,114,112 de caractères disponibles.
- Différentes manières de l'encoder.
- Remplie pour l'instant à un peu plus de 10%.
- encodée par les formats UTF-8, UTF-16, etc.

- 1 L'encodage du texte sur machine
- 2 Un encodage déjà utilisé ?

On a parler de l'encodage en machine, mais vous avez peut-être déjà utiliser un encodage: le code César.

La phrase suivante utilise le code César:

"Rgtuqppg pg vtqwxgtc eg oguucig".

Avez vous une idée de ce qui a été fait pour encoder ce texte ?

Quel est le texte de base ?

Le code César consiste à décaler toutes les lettres d'un texte d'une certaine valeur. par exemple si on choisit de décaler de 2:

- a devient c,
- b devient d,
- c devient e,
- ...
- x devient z,
- y devient a,
- z devient b.

# Code César: PYTHON ! (1/2)

En python, on a accès a 2 fonctions qui permettraient d'implémenter le code césar:

`ord("a")` renvoie la valeur du caractère a en code ASCII (97).

`chr(97)` renvoie le caractère associé a 97 en code ASCII (a).

## Code César: PYTHON ! (2/2)

En python on peut aussi utiliser une boucle pour parcourir chacune des lettres d'un texte de la manière suivante:

```
1 texte = "bonjour je suis cesar"  
2 for lettre in texte:  
3     print(lettre)  
4
```

Pour ce tp, nous allons réaliser un programme pour coder et décoder un texte en code César (nous n'utiliserons que des caractères en minuscule). Dans un premier temps, mettre au point une fonction qui décale tout les caractères d'un texte de 2 places, comme vu dans l'exemple.

Pour faire cela il faudra utiliser une boucle pour parcourir le texte lettre par lettre.

Dans un second temps, utiliser les fonctions `ord()` et `chr()` pour décaler les valeurs.

Pour faire cela:

- Stocker le retour de la fonction `ord()` de la lettre.
- Ajouter 2 a la valeur, pour décaler le texte.
- Utiliser la fonction `chr()` pour récupérer la nouvelle lettre.
- Ajouter le résultat au bout d'une nouvelle chaîne de caractère.
  - en faisant, par exemple: `texte_decode += lettre_decode`

Bonus 1: Vérifier que le code marche bien avec les lettres qui sont en limite de l'alphabet (par exemple: avec un décalage de 2, que 'z' devienne bien 'b').

Astuce: Si la valeur de la lettre dépasse la dernière lettre du code ASCII correspondant aux lettres minuscules, revenir au début de l'alphabet avant de rajouter le reste.

Bonus 2: Modifier le code pour qu'il marche avec tout décalage, par exemple que l'on puisse décaler le texte de 5, de 10, ou de toute autre valeur.

Bonus 3: En se basant sur les étapes précédentes du tp, trouver la phrase de base de ce texte "drobo sc yxo swzycdyb kwyxq ec".

Bonus 4: Gestion des caractères en majuscule.