



## TP 6 – Codage numérique des caractères

### Objectifs :

#### 1. Numérisation

Coder un nombre, un caractère au travers de codes standards (UTF-8 et ASCII), un texte sous forme d'une liste de valeurs numériques.

#### 2. Représentation binaire

Manipuler à l'aide d'opérations élémentaires les trois unités de base : bit, octet, mot.

#### 3. Formats

Identifier quelques formats de documents.

**Rappel :** Un ordinateur ne manipule que des .....  
**Comment alors code-t-il du texte ?**

### I. Le code ASCII

#### A. Une première approche

1. On va représenter la feuille de calcul ci-dessous, en respectant les consignes données plus loin, dans le Google Sheet fourni dans l'énoncé du TP :

A	B	C	D	E	F	G	H	I	J
N	O	M		P	R	E	N	O	M
78	79	77	32	80	82	69	78	79	77

- Sur la ligne 1 du tableau, écrire vos NOM et PRÉNOM en **MAJUSCULE**, en saisissant une lettre par case et en laissant un espace entre le NOM et le PRÉNOM.
- Sur la ligne 2 du tableau, entrer la formule **=code(A1)** puis recopier vers la droite :

Remarque : Chaque lettre, ainsi que l'espace, sont codés par un nombre selon le **code ASCII**.

2. Sur la ligne 3 du tableau, écrire vos NOM et PRÉNOM en **MINUSCULE**, en saisissant une lettre par case et en laissant un espace entre le NOM et le PRÉNOM.

Dans la cellule **A4** du tableau, entrer la formule permettant d'obtenir le code ASCII de la lettre saisie en A3. Recopier cette formule vers la droite.

3. Que constate-t-on ?

.....

## Culture Informatique...

Un des premiers codages standardisés est le code du « Morse » au XIXe siècle. Pour ce qui est de l'informatique, c'est le code ASCII dont les premières versions datent du début des années 1960.

**ASCII** signifie **American Standard Code for Information Interchange**.

Le principe est assez simple, il consiste à associer à chaque lettre ou chiffre un entier entre 0 et 127.

Ce codage doit être compris par tous: il est normalisé.

**L'ISO** (International Organization for Standardisation) est un organisme de normalisation international et **l'AFNOR** (Association française de normalisation) est l'organisme français.

Chaque caractère est identifié par un code unique qui est un entier naturel et la correspondance entre le caractère et son code est appelée un **Charset**.

Le code n'étant pas utilisable tel quel par un ordinateur qui ne comprend que le binaire, il faut donc représenter les codes par des octets: C'est l'**ENCODING**

**5. Petit calcul :** Sachant que l'on dispose de **127 caractères en ASCII**, combien de bits sont nécessaires pour coder tous les caractères du code ? .....

**6.** Dans la cellule **A5** du tableau, à l'aide de la fonction **DEC2BIN()**, entrer la formule permettant d'obtenir la représentation en binaire du caractère saisi en A4. Recopier cette formule vers la droite.  
Vérifier le nombre de bits utilisés pour chaque lettre.

### A retenir :

Chaque caractère est codé sur un .....

Le bit de poids fort est toujours un ....

Exemple : le caractère y est codé \_ \_ \_ \_ \_

## B. Taille d'un texte

**1.** On considère la phrase suivante : «Enfin ! Je viens de comprendre ce qu'il se produit.»

En considérant les lettres, symboles et espaces, quelle est la taille en octet de cette phrase ? ..... bytes

**2. On va vérifier la réponse donnée à la question précédente.**

**a.** Taper cette phrase dans **Notepad ++** et enregistrer ce texte sous le nom *textebrut.txt*

**b.** Vérifier la taille en octets du fichier (clic droit sur l'icône du fichier puis sur Propriétés).

**d.** Écrire la même phrase dans Word et observer la taille du fichier : ..... bytes

**e.** Mettre en gras le verbe comprendre et sauvegarder à nouveau.

Observer la taille du fichier modifié : ..... bytes

Que se passe-t-il ?

.....  
.....  
.....

### C. Utilisation de la table ASCII

Pour les questions suivantes, on travaillera sur **EduPython**.

On pourra utiliser les fonctions suivantes :

- **bin()** : cette fonction renvoie .....

.....

Par exemple : .....

- **int('x',b)** : cette fonction renvoie .....

.....

Par exemple : .....

- **chr(n)** : cette fonction renvoie .....

.....

Par exemple : .....

- **ord('c')** : cette fonction renvoie .....

.....

Par exemple : .....

**Exercice 1 :** Quel est le code (en ASCII puis en binaire) du caractère 1 ? du caractère \* ?

.....  
.....  
.....

**Exercice 2 :**

**a.** Écrire un script *code\_chaine* qui demande une chaîne de caractères à l'utilisateur et qui retourne la liste des entiers codant chacun des caractères de la chaîne dans le code ASCII.

Par exemple, si l'utilisateur rentre la chaîne *NSI*, le script affiche 78 83 73.

**b.** Compléter le script précédent afin d'afficher la liste des octets correspondants.

Par exemple, si l'utilisateur rentre la chaîne *NSI*, le script affiche 78 83 73 puis 0b1001001 0b1010011 0b1001110.

**c. Application 1 :** coder en binaire : *L'an qui vient !*

**d. Application 2 :** coder en ASCII : *Un âne est-il passé par là ?*

Qu'observe-t-on ?

.....  
.....  
.....

## Culture Informatique...

La norme ASCII permet de représenter un texte en anglais ou un programme informatique, mais elle n'est pas adaptée pour les autres langues.

La table ASCII a été étendue pour pouvoir coder de nouveaux caractères. Le codage des caractères a été étendu sur **8 bits**.

Ainsi, on peut citer la norme ISO 8859-1, définie dans les années 1980, aussi appelée « latin-1 » ou Europe occidentale, qui permet de représenter les caractères accentués (é, è, ç, à, ù, ô) mais omet quelques caractères fort utiles (ainsi, la ligature *oe* n'y figure pas).

### Exercice 3 :

a. Combien le fait d'avoir 8 bits amène-t-il de nouvelles possibilités de codage de caractères?

.....

b. Voici le code binaire d'un texte (que vous trouverez dans Classroom, nommé *codesecret.txt*)

```
01000010 01110010 01100001 01110110 01101111 00101100 00100000 01110100 01110101
00100000 01100001 01110011 00100000 011100 00 01110010 01100101 01110011 01110001
01110101 01100101 00100000 01110100 01101111 01110101 01110100 00100000 01110100
01110010 01101111 01110101 01110110 11101001 00101110 00101110 00101110
```

Écrire un script *décode\_chaine* qui demande une suite d'octets à l'utilisateur et qui retourne le texte décodé en ASCII.

Par exemple, si l'utilisateur rentre la suite **01000010 01110010 01100001 01110110 01101111**, le script affiche **bravo**

## II. Quand le net s'affole (à la maison : à préparer pour la séance suivante)

Nous avons tous un jour reçu un courriel bizarre ou lu une page web telle que celle-ci :

*Prenons l'exemple typique de la lumière mise par un phare maritime : elle est d'abord indivisible, son coût de production étant alors indépendant du nombre d'utilisateurs ; elle possède une propriété de non-rivalité (elle ne se détruit pas dans l'usage et peut donc être adoptée par un nombre illimité d'utilisateurs) ; elle est également non excluable car il est impossible d'exclure de l'usage un utilisateur, même si ce dernier ne contribue pas à son financement.*

1. Pour expliquer ce problème d'affichage, faire des recherches sur les différents formats UNICODE, UTF-8 et ASCII.

2. a. Ouvrir l'ancienne page web du LFS avec un navigateur. (fichiers sur Classroom)

b. Appuyer sur CTRL + U pour afficher le code source de la page, et observer sur la ligne 4 l'encodage utilisé.

c. Avec un clic droit, enregistrer cette page avec le type Webpage, HTML only.

d. Ouvrir cette page avec *Notepad++*, modifier l'encodage en ASCII à la ligne 4 et sauvegarder.

e. Lancer cette page dans Google Chrome : que constatez-vous ?