NSI 1ère – Mr Fleury – Mars 2020

Représentation des données : Types et valeurs de bases

Cette fiche de révision appartient au chapitre « Représentation des données ». Les notions suivantes sont abordées : représentation de l'information et Codage et décodage en base 2, 10 et 16.

Représentation de l'information :

En informatique, un format de données est la façon dont est représenté (codé) un type de données, sous forme d'une suite de bits (des 0 ou des 1). La répartition des charges positives ou négatives d'un transistor, composé de silicium permet de stocker les 2 états (0 ou 1) d'un bit.

Une suite de bits est appelée un nombre binaire ce qui correspond à un nombre en base 2 en Mathématiques. Une suite de huit bits est appelée **un octet** (ou byte en anglais). C'est l'unité de base permettant de définir la dimension d'un espace de stockage. Le stockage devant contenir de plus en plus d'informations, il faut rapidement utiliser des unités plus grandes :

- le **K-Octet** (Kilo octet = 2^{10} octets = 1024 octets)
- le **Méga-Octet** (1024 K, soit environ un million d'octets)
- le **Giga-Octet** (1024 Mégas, soit environ un milliard d'octets)
- le **Tera-Octet** (1024 Gigas, soit environ mille-milliards d'octets)

L'information (image, vidéo, audio, fichiers, …) est donc stockée sous forme de **fichiers** qui sont eux-mêmes stockés sous forme de suite de bits. **L'extension** d'un fichier, composé de 3 (ou 4) lettres après un point, permet de savoir de quel type est une information : « toto.txt » est un fichier texte, voici quelques exemples en fonction des types d'extensions :

- type texte : doc, txt, rtf, odt, pdf, …
- de type audio : mp3, wav, wma, aac, ogc, …
- de type vidéo : mp4, mov, avi, flv, vob, mpeg, …
- de type image : gif, jpg, png, ai, psd, tif, svg ...

Les textes sont formés d'une suite de caractères et ponctuations en nombres finis, on utilise des systèmes de codages sophistiqués ***ASCII***, ***Unicode*** ou chaque symbole correspond à un numéro. La mise en page ou la mise en forme des textes peuvent être représentées par des instructions, comme dans Word, le langage HTML ou UML.

Les images sont découpées en points élémentaires « ***Pixels*** » ou la couleur est représentée par un nombre qui correspond à une palette décomposée dans les 3 couleurs primaires : Rouge, Vert, Bleu. Par exemple un bleu (R=0;V=0;B=255) se note en Hexadécimal #0000FF.

Les sons sont découpés en format brut ou compressés ou alors par partie en flux (Stream).

En **programmation**, on utilise aussi les types pour différencier les variables (les nombres entiers, relatifs, les listes, les chaînes de caractères, ...) en **Python** on a :

- les types numériques : int, float, long, complex
- les types d'objets itérables : str, unicode, tuple, list, dict, set, file, ...
- les autres types d'objets : vide, object, bool, exception, fonction, module, …

Il est possible de convertir une variable dans un autre type par exemple :

- **int**() : permet de modifier une variable en entier
- **long**() : transforme une valeur en long

- **float**() : permet la transformation en flottant
- **str**() : permet de transformer des variables d'un autre type en chaînes de caractère

Codage et décodage en base 2, 10 et 16 :

- En base 10, il existe 10 chiffres, on peut compter de 0 jusqu'à 9 : [0,1,2,3,4,5,6,7,8,9]
- En base 2, il existe 2 chiffres, on peut compter de 0 jusqu'à 1 : [0,1,]
- En base 16, il existe 16 chiffres, on peut compter de 0 jusqu'à 9 puis de A jusqu'à F : [0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F]

DÉCIMALE à BINAIRE:En écriture binaire un nombre est décomposé selon les puissances de 2
26= 16 + 8 + 2 donc 26= **1**×2^4 + **1**×2^3 + **1**×2^1 donc 26= **1**×2^4 + **1**×2^3 + **0**×2^2 + **1**×2^1 + **0**×2^0
Il est important de ne pas oublier les puissances dont les coefficients sont zéro.

Finalement, pour obtenir le nombre 26 en binaire, il suffit de mettre les coefficients qui sont devant les puissances de 2 à la suite. On obtient : 11010. On écrit : $(26)_{dec} = (1\ 1010)_{bin}$

Une autre méthode consiste a effectuer les divisions euclidiennes par 2 successives et on note le reste de la division (c'est soit un 1 soit un 0) jusqu'à ce que le quotient soit zéro. Il faut lire les restes en partant de la fin pour obtenir le binaire.

BINAIRE à DÉCIMALE: À l'inverser si le nombre binaire est 101 0110,
on a donc 0×2^0 + 1×2^1 + 1×2^2 + 0×2^3 + 1×2^4 + 0×2^5 + 1×2^6.

EN HEXADÉCIMALE : ici on décompose en puissances de 16. ex :1680 = 6×16^2 + 9×16^1 + 0×16^0.

Les nombres réels sont représentés en binaire par : (10.125 en format IEEE 754 simple précision)

- 1 bit de signe : 1
- 1 exposant :1000 0010
- 1 mantisse : 010 0010 0000 0000 0000 0000

le **ET** logique : une instruction ne sera exécutée que si les deux conditions sont vraies.
le **OU** logique : une instruction ne sera exécutée que si l'une des deux conditions sont vraies.
Le **XOR** est le **OU exclusif** : une instruction ne sera exécutée que si l'une des deux conditions sont vraies mais pas les 2 en même temps.
Il est possible de combiner les opérateurs logiques : (A **ET** (B **OU** C)) est vraie si A est vraie ET B est vraie ou alors si A est vraie et C est vraie et sera fausse dans tous les autres cas.

1ère loi de Morgan: $\overline{a \cup b} = \overline{a} \cap \overline{b}$ 2ème loi de Morgan: $\overline{a \cap b} = \overline{a} \cup \overline{b}$

CE QU'IL FAUT RETENIR

Pour stocker les informations de plus en plus grandes il faut avoir recours à la ***compression*** qui permet de transformer les données pour quelles prennent moins de place. ***Hufman***, propose un algorithme de compression de données sans pertes d'informations (En donnant un code à chaque caractère de taille inversement proportionnel à sa probabilité d'apparition).

Dans le monde, il y a 10 catégories de personnes : celles qui connaissent le binaire et celles qui ne le connaissent pas (seul les informaticiens comprendront).