

LA BASE 16

1. Вы говорите по-русски ?

Voici ci-contre un extrait de la table Unicode des caractères cyrilliques russes. On lit ce tableau en associant l'étiquette de la ligne sur laquelle il se trouve avec l'étiquette de la colonne sur laquelle il se trouve. Les symboles 'U+' indiquent qu'il s'agit de la norme Unicode.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
U+0400	Ё	Ђ	Ѓ	Ќ	Є	Ѕ	І	Ї	Ј	Љ	Њ	Ћ	Ќ	Ў	Ў	Џ
U+0410	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
U+0420	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
U+0430	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
U+0440	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э	ю	я
U+0450	ё	ђ	ѓ	ќ	є	ѕ	і	ї	ј	љ	њ	ћ	ќ	ў	џ	џ
U+0460	Ѡ	ѡ	Ѣ	ѣ	Ѥ	ѥ	Ѧ	ѧ	Ѩ	ѩ	Ѫ	ѫ	Ѭ	ѭ	Ѯ	ѯ
U+0470	Ѱ	ѱ	Ѳ	ѳ	Ѵ	ѵ	Ѷ	ѷ	Ѹ	ѹ	Ѻ	ѻ	Ѽ	ѽ	Ѿ	ѿ

Exemples de lecture:

- Le caractère 'и' correspond au code 0438
- Le code 0414 correspond au caractère 'Д'

Exercice 1:

1. Quel est le code du caractère 'ж' ?
2. Quel est le code du caractère 'П' ?
3. En utilisant la fonction `chr()` avec python avec les codes précédents, essayer d'afficher ces caractères. Que se passe-t-il ?

In [6]:

```
#Tests de la fonction chr():  
chr(416)
```

Out[6]:

'Ѓ'

Réponses:

- 1.
- 2.
- 3.

Explication :

Les codes indiqués ici ne sont pas exprimés en base 10...Mais en base 16 !

2. La base 16 (hexadécimale)

Qu'est-ce que c'est ?

La base 16 comporte 16 "chiffres" : 0 1 2 3 4 5 6 7 8 9 A B C D E F

Ils permettent par exemple de représenter les entiers de 0 à 15. Ainsi :

- Le caractère 'A' représente le nombre 10.
- Le nombre 13 est représenté par le caractère 'D'.
- 'F' représente le nombre 15.
- Les 10 premiers caractères (0 à 9) correspondent aux entiers de 0 à 9.

Pourquoi ?

- 16 est un multiple de 8. Cela permet de rendre plus compréhensible qu'en base 2 la représentation des octets (1 octets = 8 bits)
- Disposer de plus de caractères permet de représenter les grandes valeurs avec moins de caractères.
- Exemples :
 - 14 en base 10, c'est 1110 en base 2 et 'E' en base 16.
 - 255 en base 10, c'est 11111111 en base 2 et 'FF' en base 16.

3. De la base 10 vers la base 16

Méthode :

- On écrit le nombre en base 10.
- On le divise par 16 et on note le reste de la division euclidienne (soit un entier de 0 à 15, c'est à dire un caractère entre 0 et F).
- On refait la même chose avec le quotient précédent, et on met de nouveau le reste de côté.
- On réitère la division, jusqu'à ce que le quotient soit 0.
- Le nombre en base 16 se lit à l'aide des restes en commençant par celui de la dernière opération effectuée.

Exemples :

1. Conversion de 125 en base 16.
 - $125 = 16 \times 7 + 13$: on note 'D'
 - $7 = 16 \times 0 + 7$: on note '7'
 - Le nombre 125 s'écrit '7D' en base 16
1. Conversion de 41 en base 16.
 - $41 = 16 \times 2 + 9$: on note '9'
 - $2 = 16 \times 0 + 2$: on note '2'
 - Le nombre 41 s'écrit '29' en base 16

Exercice 2 :

1. Convertir le nombre 234 en base 16
2. Convertir le nombre 1000 en base 16

Réponses :

1.

-

2.

-

-

4. De la base 16 vers la base 10

Quelque soit la base, la position du chiffre indique sa valeur.

Ecrit en base 10, le nombre 125 se décompose comme suit : $1 \times 10^2 + 2 \times 10 + 5$.

Il en va de même en base 16, avec les puissances de 16. Ainsi, pour retrouver l'écriture en base 10 du nombre '43E', on décompose comme suit :

$$4 \times 16^2 + 3 \times 16^1 + 14 \times 16^0 = 4 \times 256 + 3 \times 16 + 14 \times 1 = 1024 + 48 + 14 = 1085$$

Exercice 3 :

Donner l'écriture en base 10 des nombres :

- 'A0' :
- '34' :
- 'FFFF' :

Remarques :

- Connaître quelques valeurs particulières (puissances de 16 par exemple), donne de bons repères.
- Le nombre 12 est écrit 'C' en base 16, mais en informatique, il est important de connaître le nombre de bits ou d'octets avec lequel on travaille. Ainsi sur un octet (8 bits) le nombre 12 peut être écrit '0C'. Les '0' au début de la représentation ne sont pas si inutiles que cela.

5. Exercices

Les exercices ci-après ont pour objectif de manipuler cette nouvelle base en utilisant les méthodes présentées ci-dessus mais aussi de donner quelques repères et représentations mentales qui permettent d'éviter le recours systématique à ces méthodes. De plus, on pourra tester ou vérifier les résultats avec les instructions pythons ci-dessous :

In [7]:

```
#Représentation d'un entier naturel en écriture hexadécimale.  
#Les caractères '0x' renvoyés par python sont à ignorer dans la lecture du nombre.  
#Remplacer 42 par l'entier de votre choix et exécuter la cellule.  
hex(42)
```

Out[7]:

'0x2a'

In [8]:

```
#La fonction int renvoie un entier écrit en base 10  
#Paramètres :  
#-La représentation du nombre à convertir sous forme de chaîne de caractères  
#-La base dans laquelle est représentée le nombre à convertir  
#Remplacer la chaîne de caractères par une autre et exécuter la cellule  
int('0x2a',16)
```

Out[8]:

42

Exercice 5 :

Le code Unicode du caractère cyrillique russe 'Ж' est U+0416. Retrouver la valeur correspondante en écriture décimale, puis l'afficher en utilisant la fonction `chr()`.

Réponse :

In [1]:

```
#Test de la fonction chr()
```

Exercice 6 :

La valeur décimale de l'entier correspondant au symbole '€' est 8354 en Unicode. Déterminer la représentation hexadécimale de cet entier.

Réponse :

-
-
-
-

L'Unicode du symbole € est

In [2]:

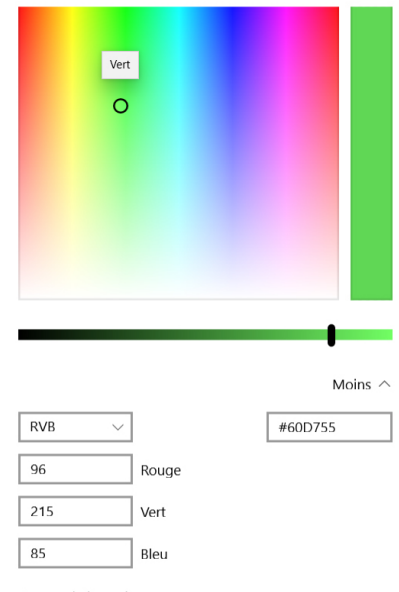
```
#Vérification
```

Exercice 7:

Le mode RVB (Rouge Vert Bleu, RGB en anglais) est un système de représentation des couleurs pour un ordinateur qui se base sur le fonctionnement de notre rétine principalement sensible à trois types de radiations: rouge, verte et bleue.

Dans ce système une couleur peut être exprimée par un triplet d'entiers écrits en base 10, compris entre 0 et 255 représentant les intensités des composantes rouge, verte et bleue. Ainsi :

- Le noir est (0,0,0)
- Le blanc est (255,255,255)



Partie A :

1. Quel est le codage du :
 - rouge:
 - vert :
 - bleu :
 - jaune :
1. Combien de couleurs peut-on représenter dans ce système ?
2. Comment peut-on coder du gris ?

Partie B :

1. Dans l'image ci-dessus, justifier la représentation '#60D755 '
2. Donner alors la représentation des couleurs :
 - noire :
 - blanche :
 - rouge :
 - verte :
 - bleue :

Partie C :

1. Donner le triplet d'entiers correspondant à la représentation #0078D7.
2. Donner le triplet d'entiers correspondant à la représentation #ABCDEF.

Réponses : Partie A :

1. Codage du :

- rouge :
- vert :
- bleu :
- jaune :

1.

1.

Partie B :

1.

-
-
-

1. Donner alors la représentation des couleurs :

- noire :
- blanche :
- rouge :
- verte :
- bleue :

Partie C :

1. #0078D7 :

-
-
-

- Réponse :

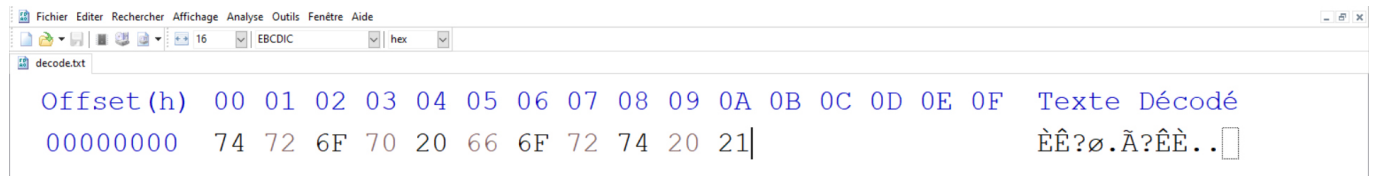
1. #ABCDEF:

-
-
-

- Réponse :

Exercice 8 : fichier

Un éditeur hexadécimal est un logiciel permettant de visualiser et éditer le contenu de toutes sortes de fichiers, en affichant leurs données en octets présentés sous forme de caractères hexadécimaux. A l'aide d'un éditeur hexadécimal, il sera possible de consulter le contenu intégral de n'importe quel fichier. Ci-dessous, le contenu d'un fichier ouvert avec l'éditeur 'HxD' contenant un texte qui ne s'affiche pas correctement à droite et dont peut voir la suite d'octets à gauche sur la deuxième ligne.



Décodez ce message (on pourra utiliser la fonctions `chr()` comme aide

In []:

```
#tests
```

Réponse :

-
-
-
-
-
-
-
- Le texte est :

Exercice 9 :

En remplaçant les caractères `?`, compléter la fonction `hexa(n)` qui prend en paramètre un entier naturel positif et qui renvoie la chaîne de caractères contenant la représentation en base 16 de cet entier. Ainsi :

- `hexa(32)` doit renvoyer `'20'`
- `hexa(255)` doit renvoyer `'FF'`

In [16]:

```
def hexa(n):
    base = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F']
    quotient = n
    reste = 0
    resultat = ''
    while quotient > 0:
        resultat = ?
        quotient = ?

    return resultat
```

`hexa(32)`

File "<ipython-input-16-4f665090f3a3>", line 7

`resultat = ?`

^

SyntaxError: invalid syntax

In [4]:

#Réponse :

Exercice 10 :

En remplaçant les caractères `?`, compléter la fonction `entier16(chaine)` qui prend en paramètre une chaîne de caractères contenant la représentation en base 16 d'un entier et qui renvoie l'entier correspondant. Ainsi :

- `entier16('20')` doit renvoyer 32
- `entier16('FF')` doit renvoyer 255

Aide : La méthode `index()` renvoie l'index de l'élément d'une liste.

- Dans la liste `L=['x','y','z']`, `L.index('y')` renvoie 1

In []:

```
def entier16(chaine):
    base = ['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F']
    n = 0
    N = len(chaine)-1 # plus grand exposant
    i = 0
    while i < len(chaine):
        n = n + ?
        i = i +1

    return n

entier16('FF')
```

In [3]:

#réponse

FIN