

Exercice 1 : Numération

1) Compléter les éléments suivants :

Base 10	Base 2				Base 16
0	0	0	0	0	0
1					1
2					2
3					3
4					4
5					5
6					6
7					7
8					8
9					9
10					A
11					B
12					C
13					D
14					E
15	1	1	1	1	F

$$(5)_{10} = (\quad)_2 = (\quad)_{16}$$

$$(12)_{10} = (\quad)_2 = (\quad)_{16}$$

$$(15)_{10} = (\quad)_2 = (\quad)_{16}$$

$$(128)_{10} = (\quad)_2 = (\quad)_{16}$$

$$(2021)_{10} = (\quad)_2 = (\quad)_{16}$$

aide

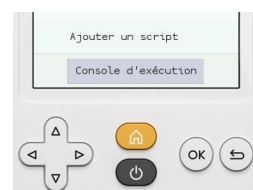
$$(2000) = 1024 + 512 + 256 + 128 + 64 + 16$$

$$(387)_{10} = (\quad)_2 = (\quad)_{16}$$

2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
		1024										

2) Conversions avec votre NumWorks :

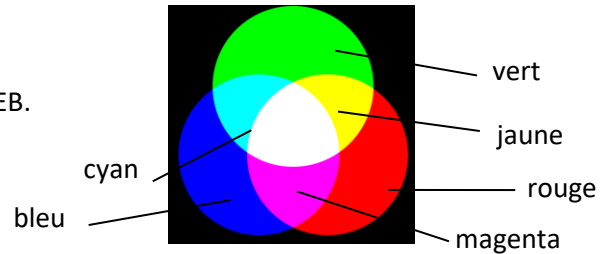
Sous une console Python, taper `hex(232)`, `bin(5871)`, `int("100011",2)` puis vérifier les conversions précédentes.



Exercice 2 : Les couleurs

Le codage hexadécimal se retrouve dans le code html des sites WEB.

Coder les couleurs ci-dessous en hexadécimal puis en base 10 :



	Rouge	Vert	Bleu	Magenta	Cyan	Jaune	noir	banc	Gris
Code ($)_{16}$	FF 00 00								
Code ($)_{10}$	255 0 0								128 128 128

Le codage RVB utilise octets, soit un mot binaire de bits.

Un octet par couleur primaire, soit 2 caractères en hexadécimale, donc caractères en tout.

$(FF)_{16} = (1111\ 1111)_2 = (255)_{10}$ $(80)_{16} = (1000\ 0000)_2 = (128)_{10}$

Exercice 3 : Code Wifi d'une Box



1) Donner le début de la clé Wifi en binaire:

.....

2) Le codage d'une clé WEP de la Livebox est codée en hexadécimal sur 26 caractères, on peut donc dire qu'une clé est codée sur un nombre de bits. Avec ce codage il y a codes possibles.

3) Si un système pirate peut tester 1 million de combinaisons par seconde, en combien de temps pourrait-il cracker le code ?

Exercice 3 : Les adresses IP



IP V4 codée sur : bits

IP V6 codée sur : bits

IP V6 (génération actuelle)	IP V4 (ancienne génération)
<p>An IPv6 address (in hexadecimal)</p> <p>2001:0DB8:AC10:FE01:0000:0000:0000:0000</p> <p>↓ ↓ ↓ ↓</p> <p>2001:0DB8:AC10:FE01:: Zeros can be omitted</p> <p>↓ ↓ ↓ ↓</p> <p>0010000000000001:000011011011000:1010110000010000:1111110000000001: 0000000000000000:0000000000000000:0000000000000000:0000000000000000</p>	<p>Une adresse IPv4 (numération décimale pointée)</p> <p>172 . 16 . 254 . 1</p> <p>↓ ↓ ↓ ↓</p> <p>10101100.00010000.11111110.00000001</p> <p>1 octet = 8 bits</p> <p>32 bits (4 x 8) ou 4 octets</p>

Avec la multiplication des objets connectés, Le stock d'adresse IPV4 créé en 1980 a presque été totalement distribué. Il a fallu inventer un autre type d'adressage permettant une quantité beaucoup plus vaste : l'IPV6.

1) Calculer le nombre d'adresses que permettait l'IPV4 :

2) Calculer le nombre d'adresses que permet l'IPV6 :