

# Les codages binaire, décimal, hexadécimal

## Plan :

I - Le codage binaire, décimal, hexadécimal

qu'est-ce que c'est ?

II - Comment passe-t-on de l'un à l'autre ?

# Le codage binaire

---

1 pour « vrai »  
0 pour « faux »

Ce codage de l'information est nommé base binaire. C'est avec ce codage que fonctionnent les ordinateurs.

Il consiste à utiliser deux états (représentés par les chiffres 0 et 1) pour coder les informations.

## Le bit « **binary digit** »

Il s'agit de la plus petite unité d'information manipulable par une machine numérique. Ce sont les 1 et 0 du code binaire.

## L'octet

C'est une unité d'information composée de 8 bits. Il permet par exemple de stocker un caractère, tel qu'une lettre ou un chiffre.

# Le codage binaire

---

## Les avantages :

- Ce code est facile à manipuler, il n'y a que deux valeurs
- Son interprétation est plus facile pour le processeur
- Le traitement d'information se fait rapidement

## Les inconvénients :

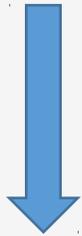
- Ce code reste très long, il manque de compacité
- Pour un développeur il est très compliqué de travailler avec ce code par sa longueur et sa forme

# Le codage décimal

---

Ce système est basé sur une logique à dix symboles, de 0 à 9  
On travaille donc sur une base de 10.

C'est un système positionnel



Le **2** de 5**2**3 n'a pas la même valeur que le **2** de 30**2**

## Comment ce codage fonctionne ?

Les milliers	Les centaines	Les dizaines	Les unités
$10^3$	$10^2$	$10^1$	$10^0$
1000	100	10	1
1	4	7	8

$$1478_{(10)} = 8 \times 10^0 + 7 \times 10^1 + 4 \times 10^2 + 1 \times 10^3$$

# Le codage hexadécimal

---

Le code hexadécimal permet une représentation simplifiée des nombres binaires.

**4 bits** permettent de coder **16 chiffres**, on en ajoute 6 aux chiffres de la base 10 donc pour cela on utilise les lettres de A à F.

Ce code est donc plus simple même si il reste difficile pour un développeur de travailler avec, il aura plus tendance à

Décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binai re	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Hexa décimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Chiffre hexadécimal	Code binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

## Conversion du binaire au décimal :

---

Le binaire ne peut prendre que 2 valeurs : 1 et 0. Il fonctionne donc sur une base de 2 :

$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
32	16	8	4	2	1
1	0	0	1	1	1

$$10\ 0111_{(2)} \square 32 + 4 + 2 + 1 = 39_{(10)}$$

# Conversion du décimal au binaire :

---

Comment compter en binaire ?

	16	8	4	2	1
1					1
2				1	0
3				1	1
4			1	0	0
5			1	0	1
6			1	1	0
7			1	1	1
8		1	0	0	0
9		1	0	0	1
10		1	0	1	0

Méthode par soustraction

Méthode par division

# Conversion du binaire à l'hexadécimal

1100 0011 1110

$2^3$	$2^2$	$2^1$	$2^0$
1	1	0	0
0	0	1	1
1	1	1	0

$$1100_{(2)} = 12_{(10)} = C_{(16)}$$

$$0011_{(2)} = 3_{(10)} = 3_{(16)}$$

$$1110_{(2)} = 14_{(10)} = E_{(16)}$$

$$1100\ 0011\ 1110_{(10)} \square C3E_{(16)}$$

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

## Conversion de l'hexadécimal au décimal :

---

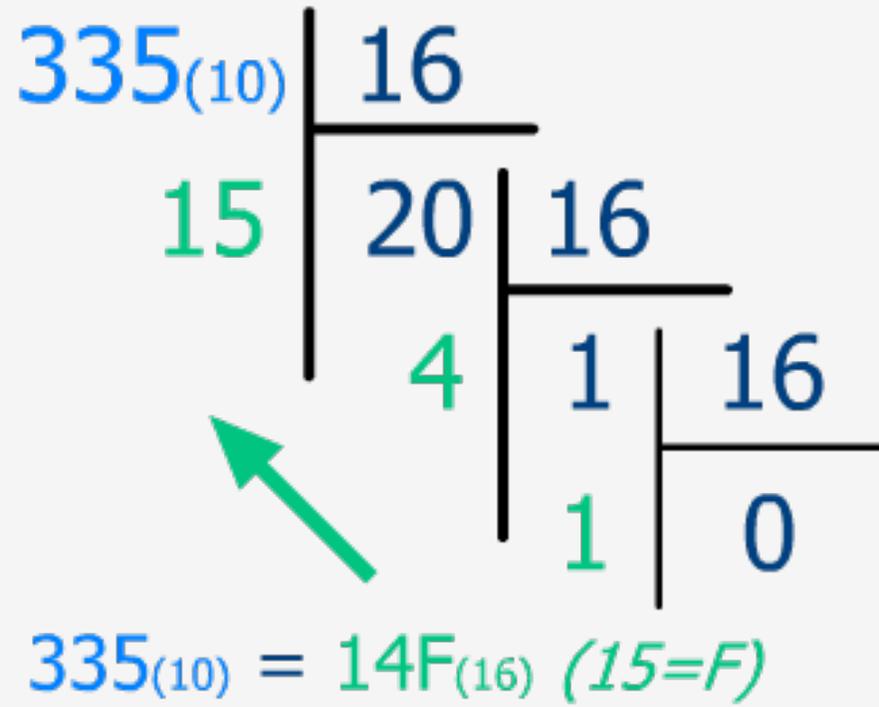
La même méthode que du binaire au décimal mais on utilise la méthode avec la base 16 :

$16^2$	$16^1$	$16^0$
1	7	D

$$17D_{(16)} = 1 \times 16^2 + 7 \times 16^1 + (\ll D \gg \times 16^0) \quad 13 \times 16^0 = 381_{(10)}$$

## Conversion du décimal à l'hexadécimal :

---



# SOURCES

<https://www.apprendre-en-ligne.net/crypto/images/bases.html>

<https://web.maths.unsw.edu.au/~lafaye/CCM/base/binaire.htm>

<http://info.blaisepascal.fr/isn-code-hexadecimal>

<https://www.youtube.com/watch?v=PZfMv8c4tPQ>