

# Conversion : Base 10 → base 2 (Méthode puissance de 2)

Exemple :  $184_{10}$

Puissances	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Valeurs	128	64	32	16	8	4	2	1
Tests / Restes	$184 \geq 128$ → reste 56	$56 < 64$ → reste 56	$56 \geq 32$ → reste 24	$24 \geq 16$ → reste 8	$8 \geq 8$ → reste 0	0 < 4	0 < 2	0 < 1
Bits	1	0	1	1	1	0	0	0

Résultat :  $184 = 10111000_2$

# Conversion Base 2 → Base 10 (Méthode des valeurs positionnelles)

Exemple :  $10101110_2$

Bits	1	0	1	0	1	1	1	0
Puissances	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Valeurs	128	64	32	16	8	4	2	1
Calcul	$1 \times 128$	$0 \times 64$	$1 \times 32$	$0 \times 16$	$1 \times 8$	$1 \times 4$	$1 \times 2$	$0 \times 1$
Résultat partiel	128	0	32	0	8	4	2	0

Somme :  $128 + 32 + 8 + 4 + 2 = 174$

Résultat :  $10101110_2 = 174_{10}$

## Le système binaire

Binaire (tout en 1)	Décimal	Nombre de bits	Valeurs possibles ( $2^n$ )
$1_2$	1	1 bit	$2^1 = 2$
$11_2$	3	2 bits	$2^2 = 4$
$111_2$	7	3 bits	$2^3 = 8$
$1111_2$	15	4 bits	$2^4 = 16$
$11111_2$	31	5 bits	$2^5 = 32$
$111111_2$	63	6 bits	$2^6 = 64$
$1111111_2$	127	7 bits	$2^7 = 128$
$11111111_2$	255	8 bits	$2^8 = 256$
$111111111_2$	511	9 bits	$2^9 = 512$
$1111111111_2$	1023	10 bits	$2^{10} = 1024$

## Résumé simple

### Base 10 → Base 2 (décimal vers binaire)

On utilise les puissances de 2 (128, 64, 32, 16, 8, 4, 2, 1).

Pour chaque puissance :

- si on peut soustraire on met **1**
- sinon on met **0**
- On continue avec le reste.

### Base 2 → Base 10 (binaire vers décimal)

On lit les bits de gauche à droite.

On additionne uniquement les puissances de 2 où le bit vaut 1.

### Valeur max selon le nombre de bits

Un nombre composé uniquement de 1 donne la valeur maximale pour n bits

(ex :  $11111111_2 = 255$ ).

## Le tableau type Hexadécimal → Binaire

Hexa	0	1	2	3	4	5	6	7
Binaire	0000	0001	0010	0011	0100	0101	0110	0111

Hexa	8	9	A	B	C	D	E	F
Binaire	1000	1001	1010	1011	1100	1101	1110	1111

Hexa	Décimal
A	10
B	11
C	12
D	13
E	14
F	15

Systeme	Ce que tu vois	Exemple
Décimal	chiffres 0–9	184
Binaire	seulement 0 et 1	10101110
Hexadécimal	lettres A–F	F2E6

## Résumé simple — Hexadécimal

### 1. 1 chiffre hexadécimal = 4 bits

Ex :

F → 1111

A → 1010

9 → 1001

### 2. L'hexadécimal va de 0 à F

- 0–9 = valeurs normales
- A = 10
- B = 11

- C = 12
- D = 13
- E = 14
- F = 15

### 3. Conversion Hexa → Binaire

On remplace chaque chiffre hexa par ses 4 bits.

Ex : F2E6 → 1111 0010 1110 0110

### 4. Conversion Binaire → Hexa

On regroupe les bits **par 4**, en partant de la droite.

Ex : 10101110 → 1010 1110 → A E

## Règle simple

1 chiffre hexadécimal = 4 bits

Ex : F2E6 → 1111 0010 1110 0110

## L'addition en binaire — Table de vérité

◆ Base 10

A	B	A + B
0	0	0 <sub>10</sub>
0	1	1 <sub>10</sub>
1	0	1 <sub>10</sub>
1	1	2 <sub>10</sub>

◆ Base 2

A	B	A + B
0	0	00 <sub>2</sub>
0	1	01 <sub>2</sub>
1	0	01 <sub>2</sub>
1	1	10 <sub>2</sub>

## Explication simple

- $0 + 0 = 0 \rightarrow 00_2$
  - $0 + 1 = 1 \rightarrow 01_2$
  - $1 + 0 = 1 \rightarrow 01_2$
  - $1 + 1 = 2 \rightarrow 10_2$  (1 en retenue, 0 écrit)
- 

## Mémo rapide — Les systèmes de numération

- **Décimal** : chiffres 0 à 9
  - **Binaire** : chiffres 0 et 1
  - **Hexadécimal** : 0 à 9 + A à F (A=10, B=11, ... F=15)
- 

## Addition binaire — Exemple complet

Exemple :

Additionner  $1011_2$  et  $0110_2$

```
  1011
+ 0110
-----
 10001
```

### Étapes (compactes et claires) :

- $1 + 0 = 1$
- $1 + 1 = 0$  (retenue 1)
- $0 + 1 + \text{retenue} = 0$  (retenue 1)
- $1 + 0 + \text{retenue} = 0$  (retenue 1)
- La retenue finale devient le bit le plus à gauche  $\rightarrow 1$

### Résultat :

$1011_2 + 0110_2 = 10001_2$

---

## Exemple pour se remettre les idées en place :

**34 en décimal  $\rightarrow$  100010 en binaire**

Ça, c'est juste une conversion de base.

Rien à voir avec AND/OR/NOT.

**111111 (63) → c'est juste un nombre**

Pas une opération logique.

**A AND B → c'est une opération logique**

Pas un nombre.

---

## **Rappel :**

- Le petit <sub>2</sub> = "ce nombre est en base 2"
- Le petit <sub>10</sub> = "ce nombre est en base 10"
- Le petit <sub>16</sub> = "ce nombre est en base 16"

C'est juste une **étiquette**, rien de plus.