

# Small group support

## Notation binaire et hexa – lien avec l'IP

### 1) Notation décimale

10 chiffres, 0 1 2 ...

Au-delà de 9 on change de rang

### 2) Notation binaire

Valeur en décimal :	Équivalent en binaire :	Explications :
0	0	Logique !
1	1	Simple !
2	10	Le premier rang a atteint le maximum autorisé ! Qu'à cela ne tienne, on passe au rang suivant. On met le second à 1 et on remet le premier à 0.
3	11	On re-remplit le rang 1.
4	100	Le rang 1 est plein, mais le 2 aussi ! On passe donc au troisième et on remet les précédents à 0 (comme on le fait lorsque l'on passe de 0999 à 1000, par exemple).
5	101	
6	110	On procède de même.
7	111	
8	1000	On entame le quatrième rang.
9	1001	On recommence au premier...
10	1010	On remplit les rangs.

On entame le rang suivant quand celui en cours est plein !

### 3) Conversion décimal → binaire

Décomposition en puissance de 2

Puissance de 2	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Valeur	128	64	32	16	8	4	2	1
35	0	0	1					
$35 - 32 = 3$				0	0	0	1	
$3 - 2 = 1$								1
$35 =$	0	0	1	0	0	0	1	1

$$35 = 32 + 2 + 1 = 0 * 128 + 0 * 64 + 1 * 32 + 0 * 16 + 0 * 8 + 0 * 4 + 1 * 2 + 1 * 1 \rightarrow 00100011$$

$$84 = 64 + 16 + 4 = 0 * 128 + 1 * 64 + 0 * 32 + 1 * 16 + 0 * 8 + 1 * 4 + 0 * 2 + 0 * 1 \rightarrow 01010100$$

### 4) Conversion binaire → décimal (beaucoup plus facile !)

Puissance de 2	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Valeur	128	64	32	16	8	4	2	1
	1	0	0	1	1	0	1	0

$$10011010 \rightarrow 1 * 128 + 0 * 64 + 0 * 32 + 1 * 16 + 1 * 8 + 0 * 4 + 1 * 2 + 0 * 1 = 154$$

## 5) Notation hexadécimale

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

Tableau récapitulatif : **Décimal**, **Hexadécimal**, **Binaire**

Dec Hx

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	F
16	10
17	11
18	12
19	13
20	14
21	15
22	16
23	17
24	18
25	19
26	1A
27	1B
28	1C
29	1D
30	1E
31	1F

## 6) Lien avec les adresses IP

Ecriture courante : 192.168.1.10 / 24

/24 → indique un masque de sous-réseau sur 24 bits ( $3 * 8$  octets)

a) Exemple 1 : masque à partir de l'@IP

Prenons une adresse IP (@IP) : 192.168.1.10 / 24

En binaire : 11000000 . 10101000 . 00000001 . 00001010

/24 indique les bits de « poids fort » (les bits à gauche) → partie réseau de l'adresse IP

Les autres bits restant (à droite) correspondent à l'hôte.

/24 = 24 bits donc **11111111.11111111.11111111.00000000**

Cette adresse IP correspond au masque de sous réseau.

**11111111.11111111.11111111.00000000** → en décimal : 255.255.255.0

Donc 192.168.1.10 / 24

Veut dire :

@IP : 192.168.1.10

Masque de sous réseau : 255.255.255.0

b) Exemple 2 : partie hôte de l'@IP par le masque de sous-réseau

Prenons cette configuration réseau :

@IP : 192.168.1.30

Masque de sous réseau : 255.255.0.0

En binaire, cela donne :

**@IP → 11000000.10101000.00000001.00011110**

**MASK → 11111111.11111111.00000000.00000000**

On va multiplier les bits 1 à 1 :

Pour le 1<sup>er</sup> octet :

**@IP → 11000000**

**MASK → 11111111**

$$\begin{aligned} \Rightarrow 1 * 1 &= 1 \\ \Rightarrow 1 * 1 &= 1 \\ \Rightarrow 1 * 0 &= 0 \end{aligned}$$

Donc le résultat est 11000000

Si on fait ça pour chaque octet de l'@IP :

**@IP → 11000000.10101000.00000001.00011110**

**MASK → 11111111.11111111.00000000.00000000**

**11000000.10101000.00000000.00000000**

⇒ Ici on a

255 \* 255 possibilités d'@IP

L'adresse IP est 192.168.1.30 / 16 → 16 bits de poids fort soit les 2 premiers octets