

# Leçon n°2 : Codage binaire et codage hexadécimal.

## 1. le système colorimétrique RVB : différents codages.

On appelle système RVB ( ou RGB en anglais) un système de codage numérique des couleurs par addition des 3 couleurs primaires : Rouge, Vert et Bleu. Ce système est utilisé pour la représentation des couleurs pour les écrans.

Pour l'impression on utilise le système CYMK ou CMJN ( Cyan, Magenta, Jaune, Noir ) qui est basé sur le principe de soustraction des couleurs, plus efficace dans le monde de l'imprimerie.

**Le système RVB est codé sur 3 octets, chaque couleur primaire étant codée sur un octet.**

### Exercice 1 : RGB , écriture en % en en décimal.

1. Dans le système RVB, quelles sont les valeurs décimales possibles pour chaque canal de couleur ?
2. Combien de couleurs différentes possibles a-t-on avec le système RVB ?
3. En HTML, CSS ou SVG, on peut définir la couleur d'un élément à l'aide de la propriété :

```
1 | color : rgb(100%,80%,60%)
```

soit en écriture décimale :

```
1 | color : rgb( __ , __ , __ )
```

4. Déterminer les codes RVB correspondants aux couleurs noire et blanche :

```
1 | color : rgb( __ , __ , __ ) /* codage couleur du noir en % */
2 | color : rgb( __ , __ , __ ) /* codage couleur du noir en décimal */
3 | color : rgb( __ , __ , __ ) /* codage couleur du blanc en % */
4 | color : rgb( __ , __ , __ ) /* codage couleur du blanc en décimal */
```

### Exercice 2 : rgb, codage en hexadécimal

1. Donner l'écriture en hexadécimal des nombres 204 et 153.

2. Expliquer pourquoi les 2 codes couleurs suivants sont identiques :

```
1 color : rgb(255,204,153) /* codage couleur en décimal */
2 color : #FFCC99 /* codage couleur en hexadécimal */
```

3. Compléter les codes couleurs suivants :

```
1 color : rgb(105,16,42) /* codage couleur en décimal */
2 color : #           /* même couleur codée en hexadécimal */
```

```
1 color : #E0EDDA           /* couleur codée en hexadécimal */
2 color : rgb(   ,   ,   ) /* même couleur codée en décimal */
```

## 2. Passage du codage binaire au codage en hexadécimal.

### Exemple 1

Considérons le nombre 255, qui se code en binaire sur un octet par :  $\underbrace{11111111}_{8 \text{ bits}}$

On peut décomposer cette écriture en deux blocs de 4 bits :  $\underbrace{1111}_{4 \text{ bits}} \underbrace{1111}_{4 \text{ bits}}$

On peut coder facilement chacun de ces deux blocs en hexadécimal :

1111 représente le nombre \_\_ en décimal, soit le nombre \_ en hexadécimal.

On a alors :  $\underbrace{11111111}_{255 \text{ en décimal}} = \underbrace{ff}_{\text{en hexadécimal}}$

## Exemple 2

Considérons le nombre binaire codé sur 8 bits :  $\underbrace{01101101}_{8 \text{ bits}}$ ,

On peut écrire que :  $\underbrace{01101101}_{\text{bloc1}} = \underbrace{01100000}_{\text{bloc1}} + \underbrace{00001101}_{\text{bloc2}} = \underbrace{0110}_{\text{bloc1}} \times \underbrace{10000}_{16 \text{ en base 2}} + \underbrace{00001101}_{\text{bloc2}}$

On a alors :

$0110 = 6 \text{ en hexadécimal}$

$1101 = 13 \text{ en décimal} = D \text{ en hexadécimal}$

et donc  $01101101 = 6D \text{ en hexadécimal}$

Pour convertir un binaire en hexadécimal, il suffit de regrouper l'écriture par blocs de 4 bits et de traduire chacun de ces blocs en hexadécimal.

Ce que confirment ces commandes python :

```
1 >>> 0x6D
2 109
3 >>> 0b01101101
4 109
5 >>> bin(0x6D)
6 '0b1101101'
7 >>> hex(0b01101101)
8 '0x6d'
9 >>>
```

## 3. Calcul binaire : addition et multiplication.

### A. Addition en binaire.

Considérons l'addition en binaire suivante :

$$\begin{array}{r} 01001101 \\ + 01101100 \\ \hline = \end{array}$$

Que l'on peut vérifier en décimal :

$$\begin{array}{r} + \\ \hline = \end{array}$$

Les opérations arithmétiques en binaire se font de la même façon qu'en décimal.

## B. Multiplication en binaire.

Considérons la multiplication en binaire suivante :

$$\begin{array}{r} 1101 \\ \times 1001 \\ \hline \end{array}$$

La multiplication en binaire s'effectue par de simples décalages de l'écriture.