CODAGE DE L'IMAGE
&
REDUCTION DE DONNÉES

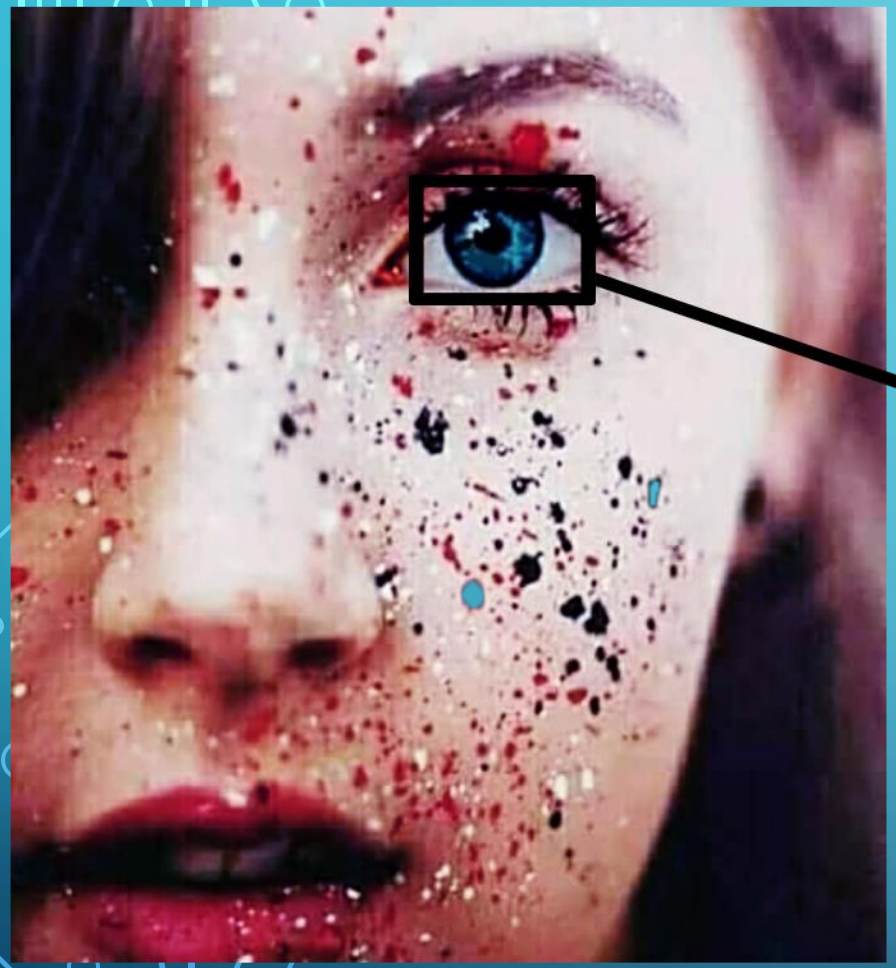
RÉALISÉE PAR : BENHAMZA SOFIANE

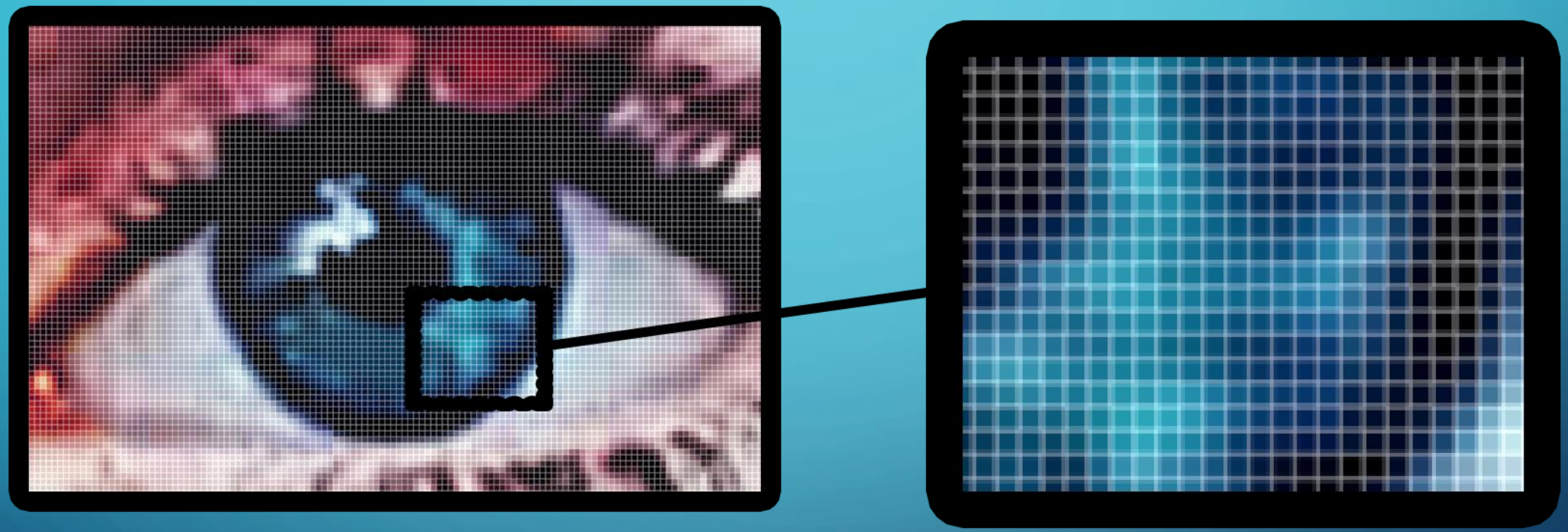
PLAN

- Notions de base
- Qu'est ce qu'une image
- Codage vectoriel
- Codage Matriciel
- RLE
- Gaspillage

Petites choses a savoir :

- Les images se compose par des trés petites boites qu'on appelle ¨ PIXELS ”
- La qualité d'une image augment avec le nombre des pixels qu'elle contient
- La taille d'une image est depend de sa qualité , type de codage, la distribution des couleurs , etc ...





Ces boites sont des Pixels

Une image est une ensemble des pixels qui se suivent l'une l'autre

Une Pixel est en un seul couleur

Comment on transfere ces images en code ?

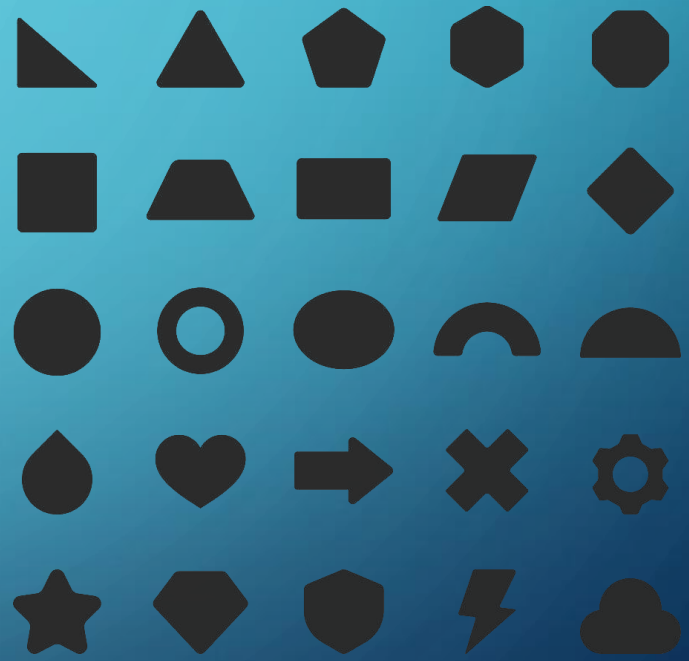
Codage Vectoriel

Basée sur une representation mathematique

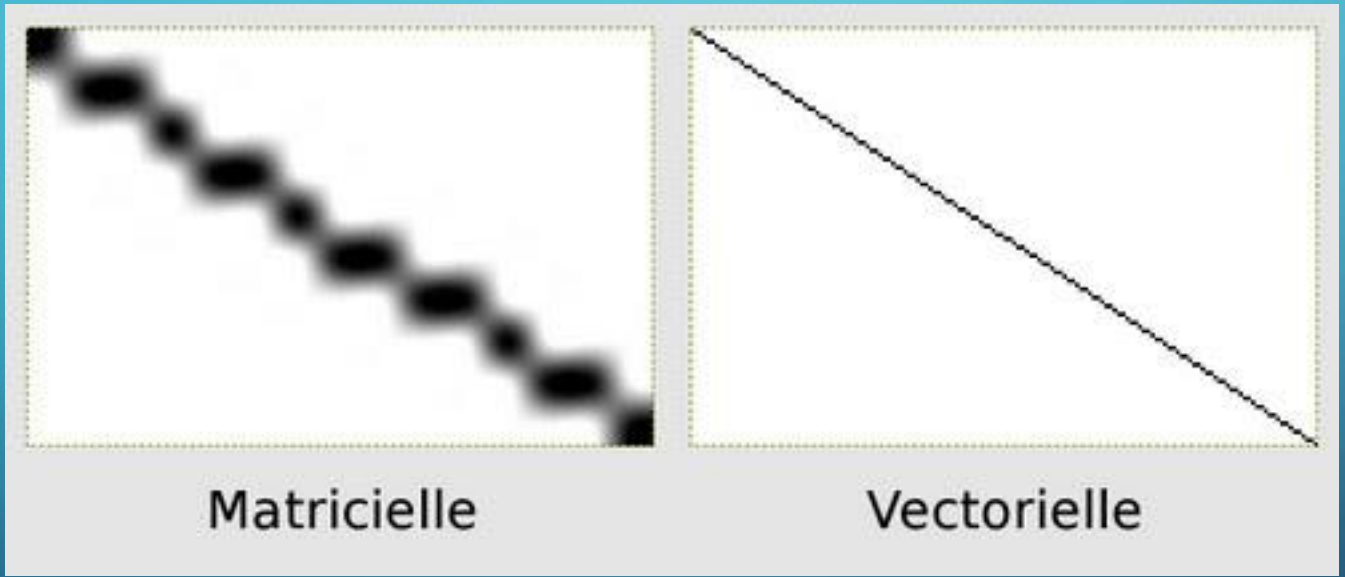
- Un Cercle est definie par un centre et un Royan
- Un Triangle est definie par trois points
- Un Rectangle, Carré , trosange ou

un trapézo est définie par 4 points

- etc …

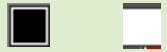
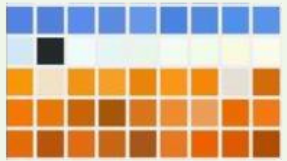
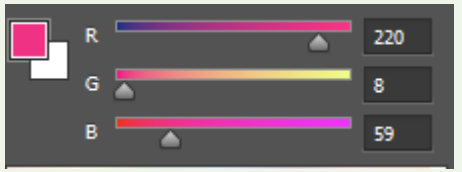


Contraire au codage matriciel , il n'y a pas de perte de qualité lors de la zoom



Codage Matriciel

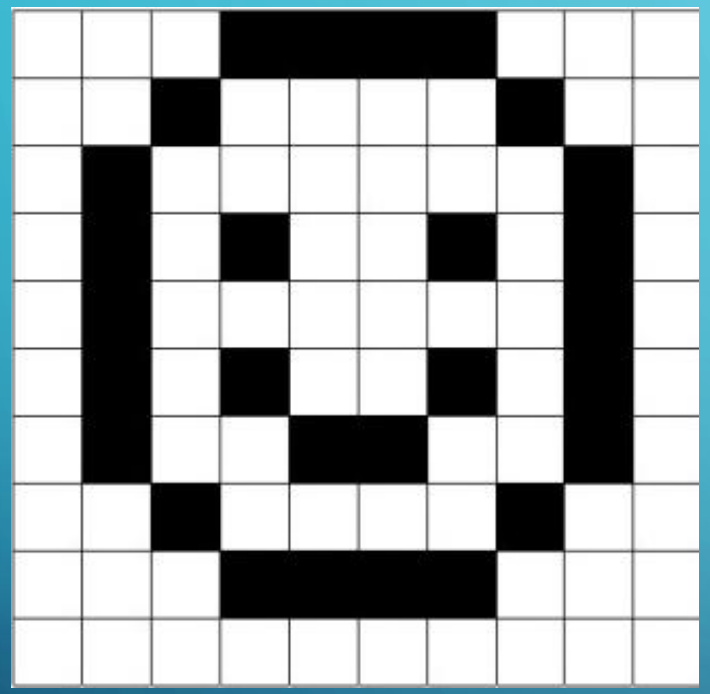
On peut considerer une image comme une Matrice de 2 (NxM) ou 3 dimensions (NxMxL)
Chaque pixel est une case dans la matrice

Niveau	Nbr Couleurs	Octet par Pixel	notation
1bit	2^1= 2 couleurs	1/8 = 0.125	
8bits	2^8= 256 couleurs	1	 .....
N bits	2^N Couleurs / possibilités	N/8	
24bits	2^{24} = 16 777 216 couleurs RGB true color	3	

Exemple :

codage d'une image sur 1 bit (noir et blanches)

■ = 0 □ = 1



Donc notre code (ligne par ligne) est

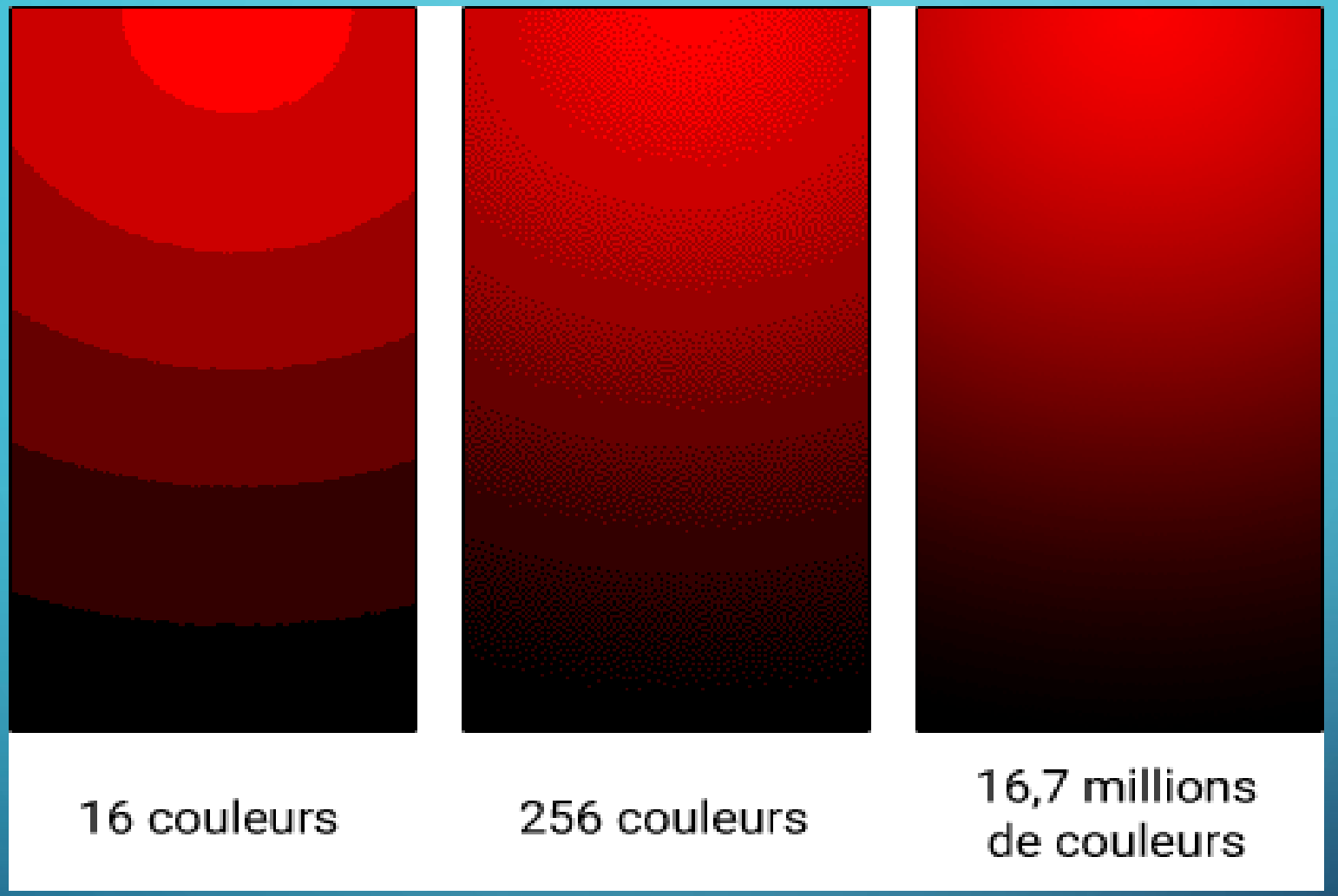
1110000111 1101111011 1011111101 1010110101
1011111101 1010110101 1011001101 1101111011
1110000111 1111111111

10 ligne x 10 colone = 100 pixel

100= 96+4 = 8*12 + 4

100 Pixel ⇔ 100 bits ⇔ 12 octet et 4bits =12,5 Octet sans compression

Taille d'une photo (en Octet) = nbr_ligne x nbr_colone x nbr_bits/pixel



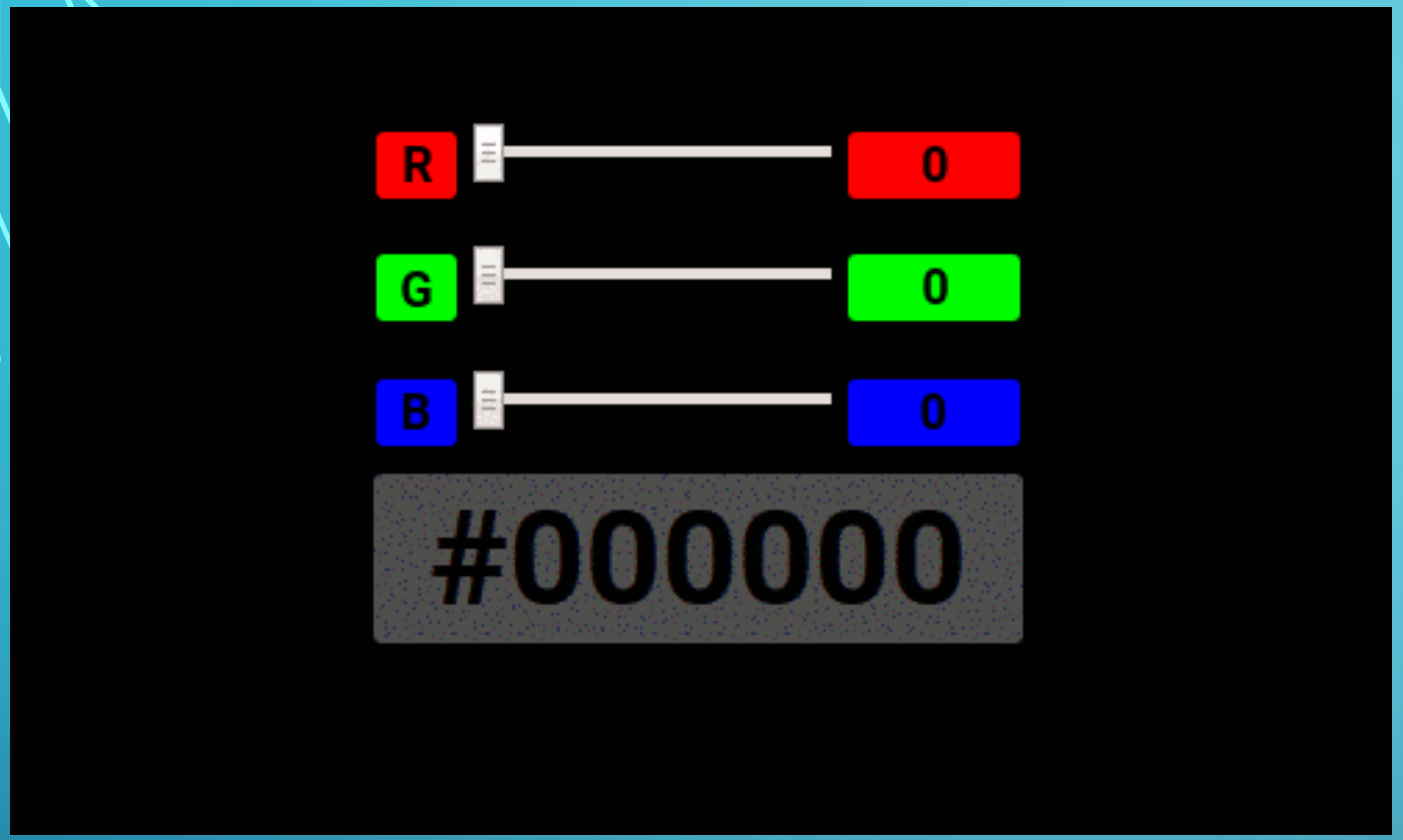
4bits 8bits 24bits

La meme photo codée sur des differents bits

La qualité augment avec la taille qui depend de nombre de bits

La Techique RVB/RGB

On peut construire n’import qu’elle Couleur avec un melange de rouge , vert et bleu



Exemple : #F8E02E

= RGB(F8,E0,2E)

0XF8 = 248 en decimal

0XE0= 224

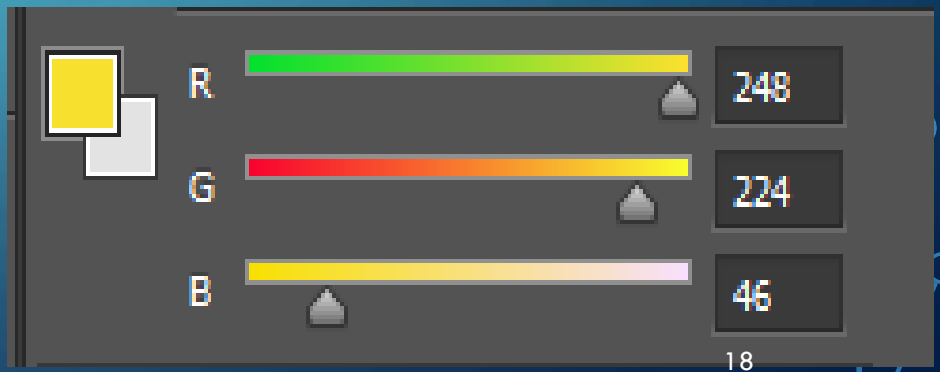
0X2E= 46

Red(248);green(224);bleu(46)

RGB a 24bits

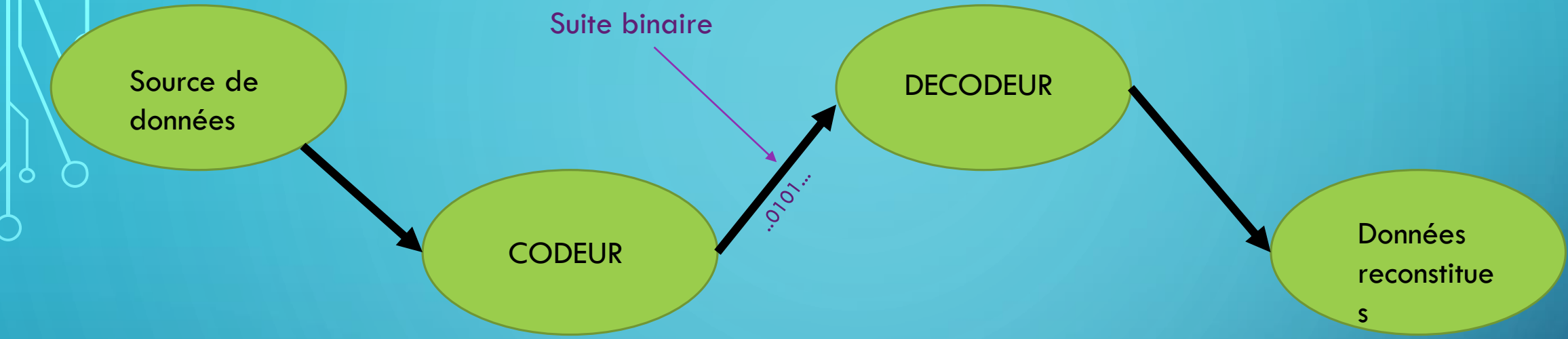
3 valeurs de 0 a 255

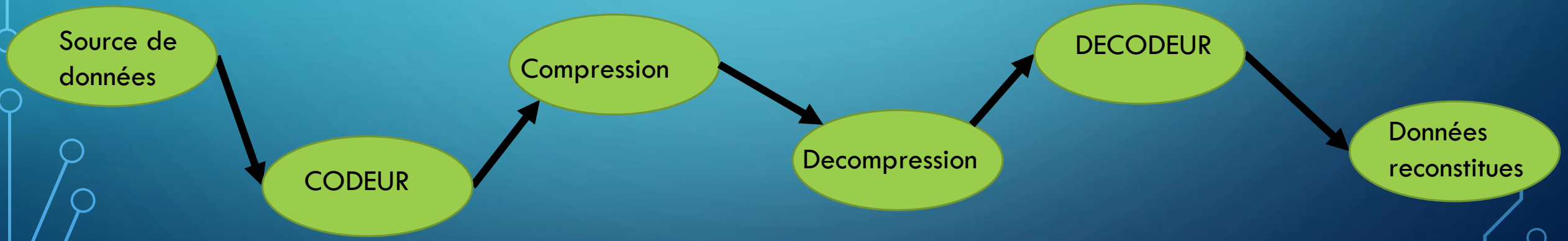
Chacun des trois couleurs est codée sur 8bits=1octet



Pour des raisons de stockage / Vitesse de transfert de données

On est obligé de diminuer la taille des images





On gagne du temps lors de partage et de l'espace

RLE (Run Length Encoding)

Consiste a réduire la taille d’un fichier

PPPPSSSSSSSSS => 4P9S

13 Octet 4 Octet

On a gagné (13-4 = 9) Octet

AFFOLEMENT => 1A2F1O1L1E1M1E1N1T

10 18

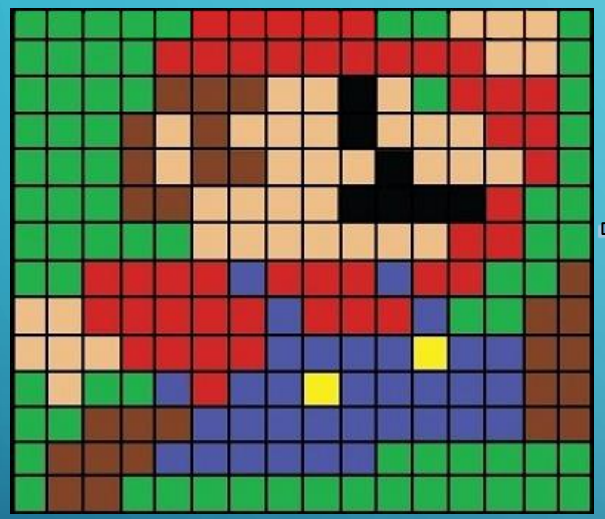
nous avons perdu (18-10 = 8) Octet

On pratique on fait ca just avec les lettres qui sont repetes plus que 3 fois

Alors on peut pas diminuer la taille d'un text puisque qu'il n'y a pas une mot qui contient 3 letttres identiques qui se suivent

Il est utile d'utiliser RLE pour les images qui ont des grandes parties Uniforme

16 pixel



14 pixel

16 x 14 = 224pixel

Rgb 3 Octet / pixel <=> 672 Octet

rgb(2,6,7)
00000010
00000110
00000111

rgb(74,89,174)
01001010
01011001
10101110

rgb(240,191,133)
11110000
10111111
10000101

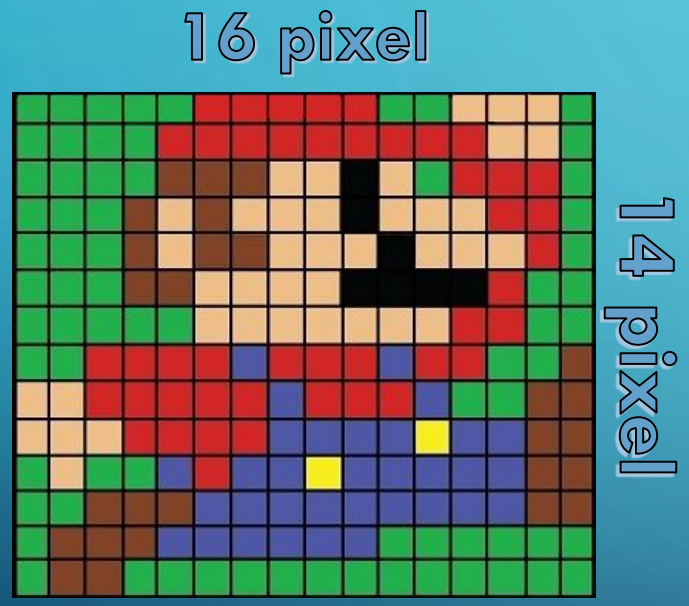
rgb(42,176,79)
00101010
10110000
01001111

rgb(128,68,40)
10000000
010000100
01000100

rgb(212,39,41)
11010100
00100111
00101001

rgb(248,238,27)
11111000 11101110 00011011

la 1er ligne qui se compose de 16pixel son code sans compression est :



00101010 10110000 01001111
00101010 10110000 01001111
00101010 10110000 01001111
00101010 10110000 01001111
00101010 10110000 01001111
11010100 00100111 00101001
11010100 00100111 00101001
11010100 00100111 00101001
11010100 00100111 00101001
11010100 00100111 00101001
00101010 10110000 01001111
00101010 10110000 01001111
11110000 10111111 10000101
11110000 10111111 10000101
11110000 10111111 10000101
00101010 10110000 01001111

16 Pixel * 3 Octet = 48 Octet pour 16pixel just !

Mais si on utilise l'RLE 

On donne un numero n > 2 avant chaque element qui se repete plus que 3 fois

On voie que 🟩 se répéte 5 fois , 🟥 5fois , 🟩 2 fois , 🟧 3 fois et un seul 🟩

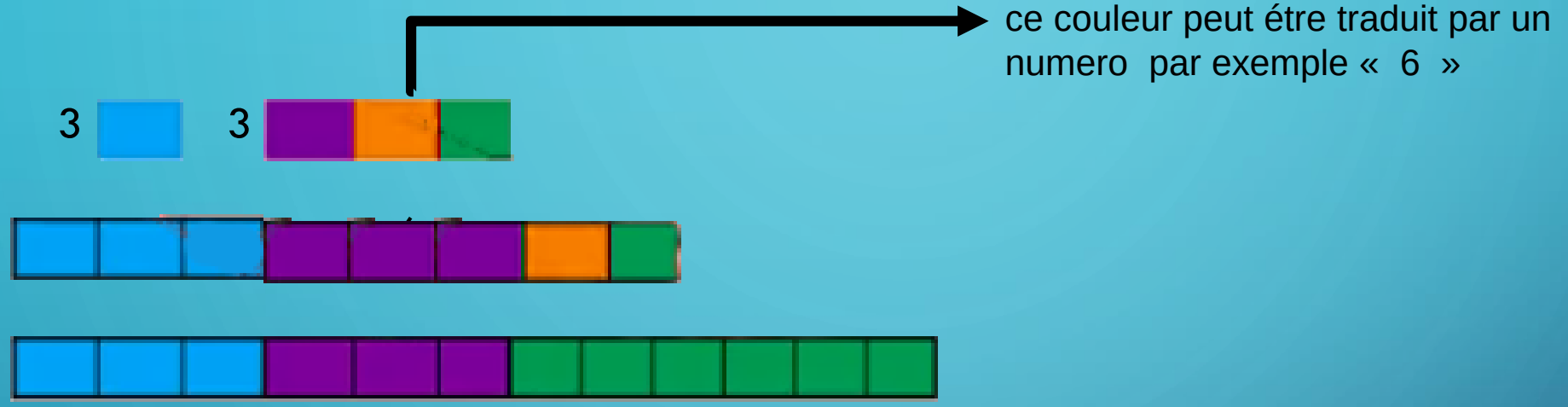
On peut traduire cette ligne par *5* 🟩 *5* 🟥 🟩 🟩 3 🟧 🟩

00000101 00101010 10110000 01001111 > 🟩
00000101 11010100 00100111 00101001 > 🟥
00101010 10110000 01001111 > 🟩
00101010 10110000 01001111 > 🟩
00000011 11110000 10111111 10000101 > 🟧
00101010 10110000 01001111 > 🟩

On a utilisé just 21 Octet << 48 Octet qu'on a trouvé sans RLE

Gaspillage :

supposons qu'on code sur 8bits (256 couleurs) Chaque pixel est Codé sur 1 Octet



Pour transcender cette erreur de decodage , on ajoute un Caractére special avant le nombre des pixel a ne pas repete

MERCI POUR VOTRE ATTENTION