

# HLIN102 – Du binaire au web

## Représentation d'images

Sources :  
[images.math.cnrs.fr/Le-traitement-numerique-des-images.html](http://images.math.cnrs.fr/Le-traitement-numerique-des-images.html)

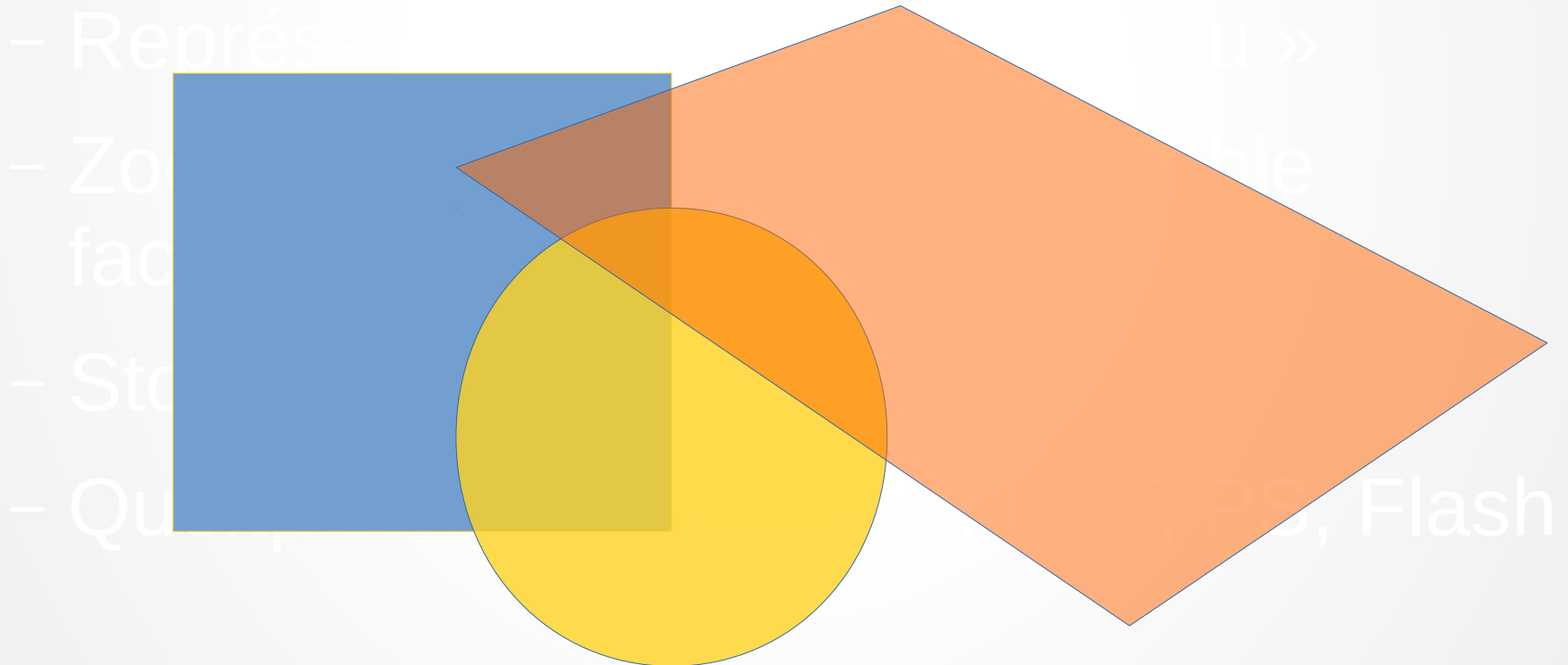
# Représentation d'images

2 familles de représentations, selon ce qu'exprime l'image :

- Image = expression d'un modèle abstrait (ex : formes géométriques, équations physiques) → représentation abstraite
- Image = imitation fidèle de la réalité (ex : photo, scan de document) → représentation matricielle

# Représentation abstraite

- Image vectorielle : une image est une superposition de formes géométriques



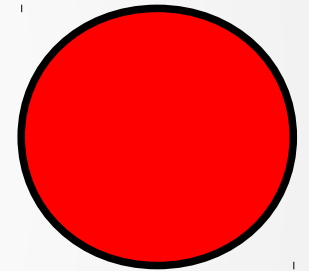
# Représentation abstraite

- Image vectorielle : une image est une superposition de formes géométriques
  - Représentation « de haut niveau »
  - Zoomable indéfiniment, modifiable facilement
  - Stockage économique
  - Quelques formats : SVG, PDF, PS, Flash

# Image vectorielle : SVG

- Exemple de code SVG dans une page Web :

```
<!DOCTYPE html>  
<html><body>  
  <svg height="100" width="100">  
    <circle cx="50" cy="50" r="40"  
      stroke="black" stroke-width="3"  
      fill="red" />  
  </svg>  
</body></html>
```



# Représentation abstraite

- Image vectorielle
- Modèle physique : l'image est le résultat d'un calcul issu d'équations physiques

- Images rendues par ordinateur (cinéma, jeux vidéo)
- Souvent ces images sont créées sous forme numérique



# Représentation abstraite

- Image vectorielle
- Modèle physique : l'image est le résultat d'un calcul issu d'équations physiques
  - Images réalistes → applications artistiques (cinéma, jeu vidéo) et scientifiques (simulations)
  - Souvent coûteux en temps de calcul → les images sont souvent calculées puis stockées sous forme matricielle

# Représentation d'images

2 familles de représentations, selon ce qu'exprime l'image :

- Image = expression d'un modèle abstrait (ex : formes géométriques, équations physiques) → représentation abstraite
- Image = imitation fidèle de la réalité (ex : photo, scan de document) → représentation matricielle



# Images matricielles

- Image = tableau à deux dimensions de points (ou *pixels*)



# Images matricielles

- Image = tableau à deux dimensions de points (ou *pixels*)
  - Exemples de formats :
    - Brut : RAW, TIFF
    - Compressé sans perte : PNG, GIF
    - Compressé avec perte : JPG

# Images matricielles

- Image = tableau à deux dimensions de points (ou *pixels*)
  - Exemples de formats
  - Chaque pixel est associé à un ou plusieurs nombres qui décrivent son apparence

# Images matricielles

- Ex : images brutes en niveaux de gris
  - Chaque pixel est décrit par son intensité lumineuse
  - ex sur 8 bits :

0 = noir



255 = blanc



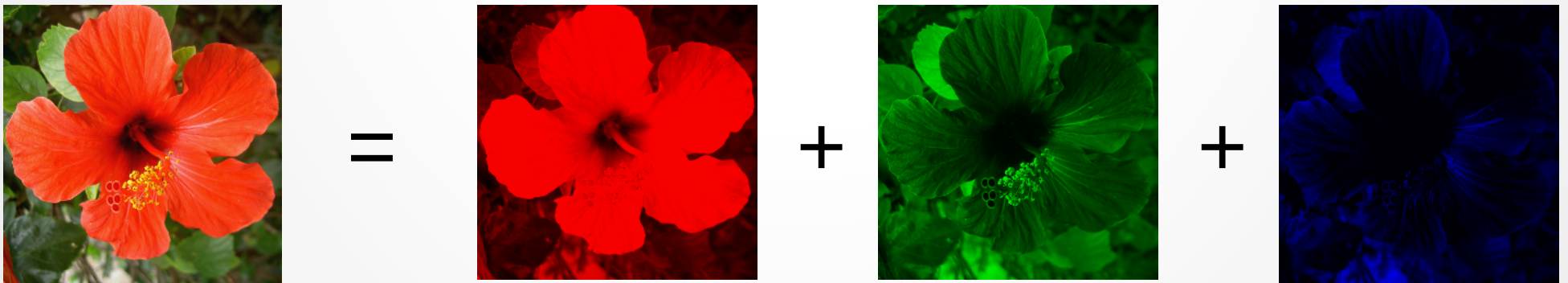
137	139	137	128	119	126	130	129
129	124	118	115	120	147	181	182
117	93	87	117	141	160	185	203
115	84	79	111	142	166	178	191
119	87	73	97	135	155	176	187
116	91	77	84	118	150	173	188
117	97	84	78	101	131	160	177
111	104	90	78	78	105	142	170

# Images en niveaux de gris

- Le nombre de bits par pixel donne le nombre de nuances de gris (« profondeur » de l'image)
- Le début (= en-tête) du fichier qui stocke l'image doit au moins préciser :
  - le nombre de lignes et de colonnes de l'image
  - le mode : niveaux de gris (sauf si c'est une caractéristique du format)
  - le nombre de bits par pixel (sauf si c'est une caractéristique du format)

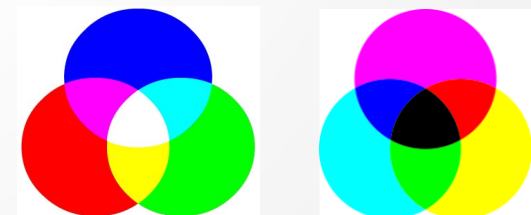
# Images matricielles

- Image en couleurs
  - Format brut : 1 image couleur = superposition de 3 images (« masques ») rouge, vert et bleu par synthèse additive



# Images en couleurs

- Si chaque composante RVB est codée sur 8 bits, 1 pixel est codé sur 24 bits soit  $2^{24} \approx 16,8$  millions de couleurs
- Autres représentations :
  - Cyan Magenta Jaune (CMJ) : composantes en synthèse soustractive, complémentaires des RVB :  
 $C=1-R$   $M=1-V$   $J=1-B$
  - Teinte Saturation Luminosité (TSL) : représentation adaptée à la perception humaine



# Images matricielles

- De la couleur au gris
  - Luminance =  $\text{moyenne}(R, V, B)$
  - 24 bits  $\rightarrow$  8 bits : perte de profondeur





# Alléger la représentation

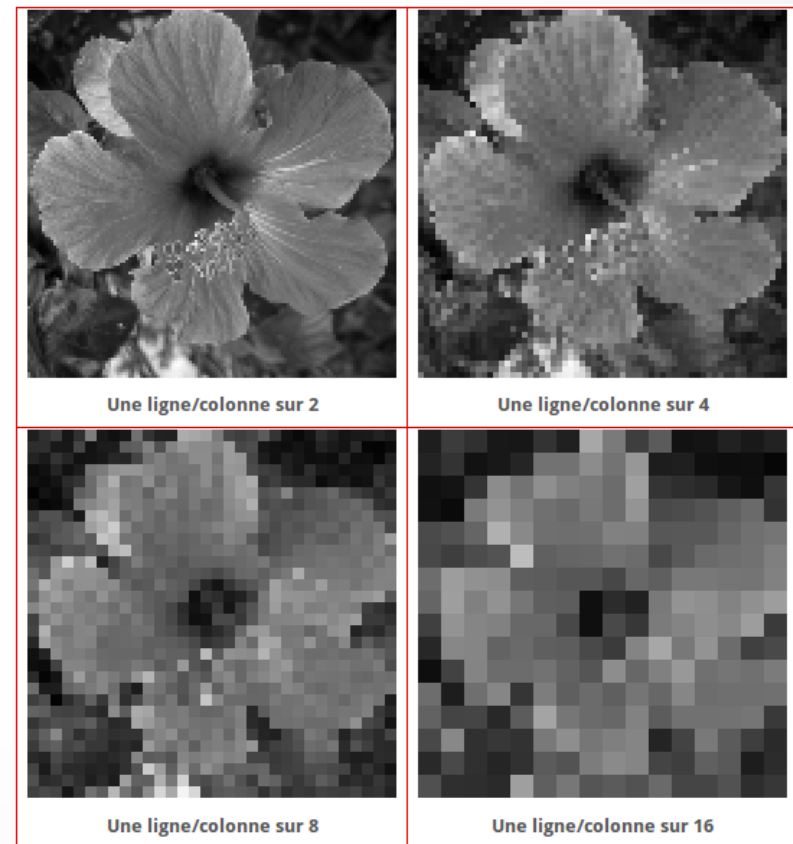
- Comment réduire l'espace occupé par une image ?
  - Réduire la précision
  - Repérer les répétitions pour décrire l'image plus brièvement
  - Décrire l'image comme l'expression d'un modèle abstrait

# Alléger la représentation

- Comment réduire l'espace occupé par une image ?
  - Réduire la précision
  - Repérer les répétitions pour décrire l'image plus brièvement
  - Décrire l'image comme l'expression d'un modèle abstrait

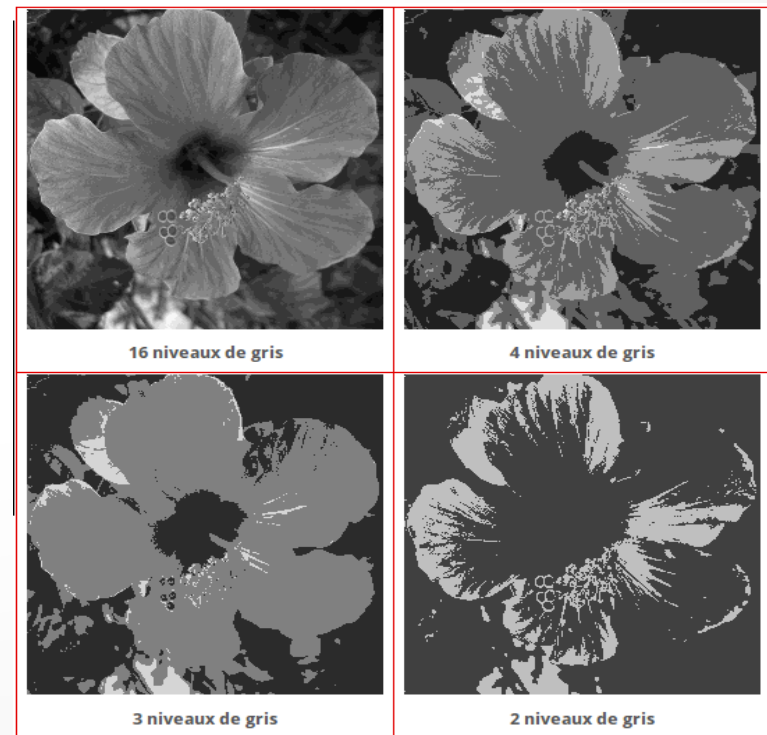
# Réduire la précision

- Réduction du nombre de lignes/colonnes
  - Cette réduction peut être adaptative selon les zones de l'image  
(ex : Google Maps)



# Réduire la précision

- Réduction de profondeur
  - Quantification (ex : 32 bits  $\rightarrow$  8 bits :  
[0;15]  $\rightarrow$  0    [16;31]  $\rightarrow$  1    [32;47]  $\rightarrow$  2    ...)

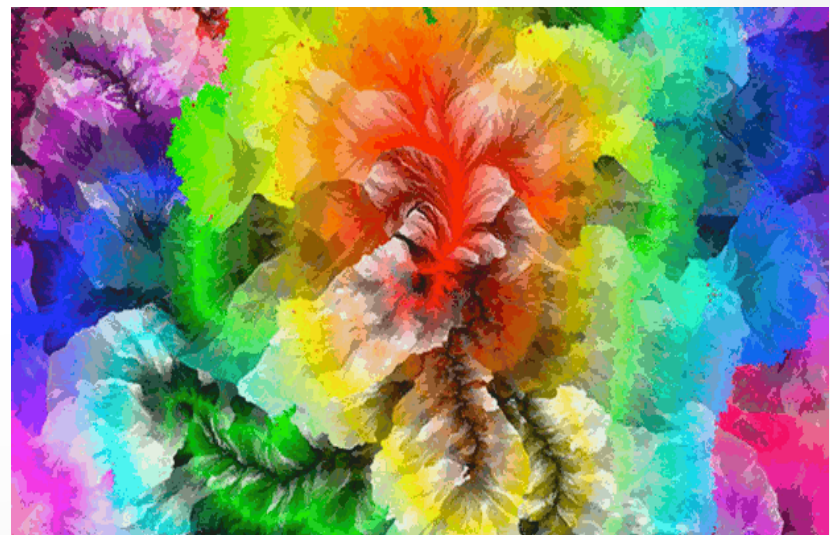


# Réduire la précision

- Réduction de profondeur
    - Optimisation de palette
- Exemple ci-dessous : 256 couleurs, chacune sur 24 bits  
→ « palette » à inscrire dans le fichier de l'image



16,8 millions de couleurs



256 couleurs

# Alléger la représentation

- Comment réduire l'espace occupé par une image ?
  - Réduire la précision
  - Repérer les répétitions pour décrire l'image plus brièvement
  - Décrire l'image comme l'expression d'un modèle abstrait

# Repérer les répétitions

- Un exemple simpliste

0	0	0	0	0	0
1	1	1	0	0	0
0	1	1	1	0	0
0	0	1	1	1	0
0	0	0	0	0	0



(5 6 1) 6 3 4 3 4 3



# Repérer les répétitions

- Un exemple simpliste
- Algorithmes de compression (sans perte) : recherche de motifs répétés dans l'image
  - Analyses fréquentielles (dans le plan et en profondeur)
  - Descriptions multi-échelles

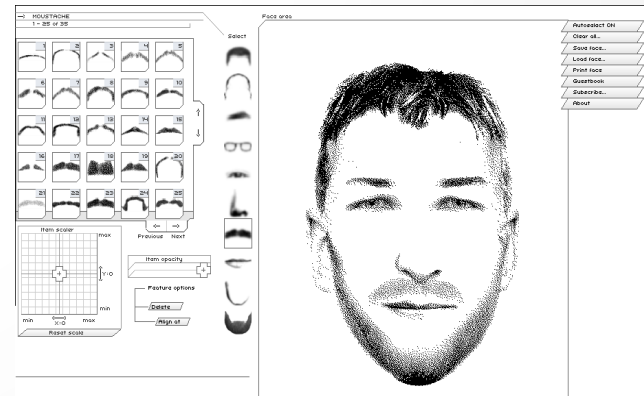


# Alléger la représentation

- Comment réduire l'espace occupé par une image ?
  - Réduire la précision
  - Repérer les répétitions pour décrire l'image plus brièvement
  - Décrire l'image comme l'expression d'un modèle abstrait

# Image réelle, modèle abstrait

- Utilisation de connaissances sur l'image ou le spectateur pour extraire les informations pertinentes
  - Ex : reconnaissance de visage, utilisation de phénomènes de masquage pour réduire la précision...



# Merci pour votre attention

