



PREMIERE GENERALE

NUMERIQUE ET SCIENCES INFORMATIQUES

CREER UNE IMAGE ASCII ART

- ▷ Histoire de l'informatique
- ▷ Représentation des données : types et valeurs de base
- ▷ **Représentation des données : types construits**
- ▷ Architectures matérielles et systèmes d'exploitation
- ▷ Interactions entre l'homme et la machine sur le Web
- ▷ Architectures matérielles et systèmes d'exploitation
- ▷ **Langages et programmation**
- ▷ Algorithmique

1 – L'ASCII ART

L'ASCII Art est une forme d'art consistant à faire des dessins composés uniquement avec des caractères ASCII que l'on juxtapose.

Un dessin ASCII Art peut être créé à partir de 127 caractères reprenant toutes les lettres majuscules et minuscules, les chiffres, quelques symboles et signes de ponctuation. Pour être vu correctement, l'art ASCII suppose l'utilisation d'une police de caractères à espacement constant, c'est-à-dire une police dont tous les caractères ont la même largeur. C'est le cas de la police Courier par exemple.

[illegible]

Cette pratique a commencé avec les mainframes, où ce genre d'images permettait de montrer ce qu'il était possible de faire avec une imprimante bien commandée. Les images étaient alors d'une largeur de 120 ou 132 caractères pour une longueur variable, et autorisaient les superpositions multiples de caractères.

[illegible]

On peut réaliser de l'art ASCII avec un simple éditeur de texte mais il existe des logiciels automatisant le processus, à l'aide d'algorithmes de conversion d'image en texte. Évidemment ces images, si elles sont faites à la main, demandent beaucoup de temps et de talent, d'où le terme « art ».

2 – CREER UNE IMAGE ASCII ART

Le but de cette activité est de transformer une image numérique en dessin ASCII Art. Les caractères ASCII seront stockés dans un tableau python.



Le principe est le suivant :

- Convertir l'image RGB en image en niveaux de gris.
 - Découper l'image en zone de 2 pixels en largeur et 5 pixels en hauteur.
 - Calculer la moyenne du niveau de gris de cette zone.
 - Placer le caractère ASCII image de la moyenne de niveaux de gris dans la case du tableau correspondante.
 - Sauvegarder le tableau dans un fichier `.txt`.
1. Sachant que l'image à convertir a pour dimensions **largeur = 880** et **hauteur = 586**, **donner** le nombre de lignes et de colonnes du tableau qui doit contenir les caractères ASCII. Donner le nombre de lignes et de colonnes du tableau pour un image de dimensions quelconques **largeur** et **hauteur**.
 2. **Ouvrir** le fichier `art_ASCII.py`. **Compléter** la ligne du code n°4 permettant de récupérer les dimensions de l'image à convertir.
 3. **Compléter** la ligne du code n°6 permettant de créer, par compréhension, le tableau `img_ASCII` qui va contenir les caractères ASCII. Ce tableau sera rempli du caractère ' ' (espace).
 4. **Compléter** les lignes n°8 et 9 afin de parcourir l'ensemble des lignes et des colonnes du tableau `img_ASCII`.

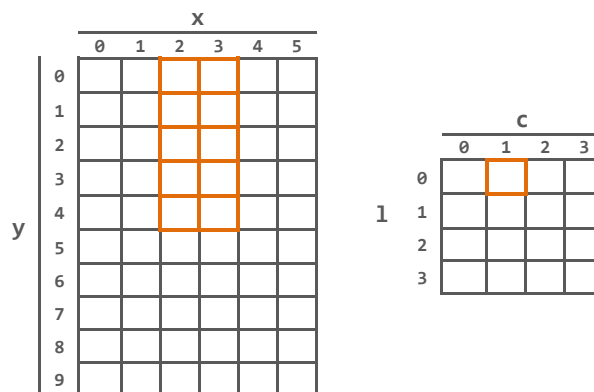
Dans la case `img_ASCII[0][0]` sera contenu un caractère correspondant à la moyenne des niveaux des pixels $(0, 0), (0, 1), (0, 2), (0, 3), (0, 4), (1, 0), (1, 1), (1, 2), (1, 3)$ et $(1, 4)$. x prend les valeurs de 0 à 1 et y les valeurs de 0 à 4.

		x					
		0	1	2	3	4	5
y	0						
	1						
	2						
	3						
	4						
5							
6							
7							
8							
9							

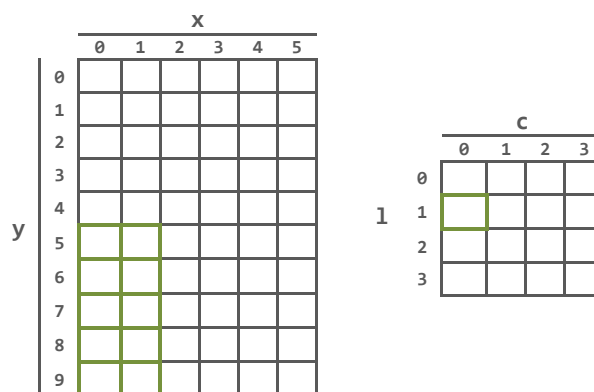
		c			
		0	1	2	3
l	0				
	1				
	2				
	3				



Dans la case `img_ASCII[0][1]` sera contenu un caractère correspondant à la moyenne des niveaux des pixels $(2, 0), (2, 1), (2, 2), (2, 3), (2, 4), (3, 0), (3, 1), (3, 2), (3, 3)$ et $(3, 4)$. x prend les valeurs de 2 à 3 et y les valeurs de 0 à 4.



Dans la case `img_ASCII[1][0]` sera contenu un caractère correspondant à la moyenne des niveaux des pixels $(0, 5), (0, 6), (0, 7), (0, 8), (0, 9), (1, 5), (1, 6), (1, 7), (1, 8)$ et $(1, 9)$. x prend les valeurs de 0 à 1 et y les valeurs de 5 à 9.



5. **Indiquer** quelles valeurs prendront les variables x et y pour une case quelconque `img_ASCII[1][c]`.
6. **Compléter** les lignes n°11 et 12 afin de parcourir l'ensemble des pixels de la zone en fonction de l et de c .
7. **Compléter** les lignes n°13 à 16 permettant de récupérer les composantes **R**, **G** et **B** du pixel défini par (x, y) puis de calculer le niveau de gris correspondant, de l'additionner à la variable **somme** et enfin lorsque les boucles sont finies, de calculer la moyenne de ces niveau de gris.

Nous allons nous intéressons considérer 8 gammes de niveaux de gris : 0 à 31, 32 à 63, 64 à 95, ... et 224 à 255. Chaque gamme sera représentée par un caractère ASCII.

8. **Choisir** 8 caractères permettant de représenter chaque gamme de niveau de gris. Par exemple, un niveau de gris compris entre 224 et 255 (proche du blanc) pourra être représentée par le caractère ' ' (espace).
9. **Compléter** les lignes 17 à 32 qui permettent d'affecter aux différentes cases du tableau le caractère en fonction du niveau de gris. L'affectation sera `img_ASCII[l][c] = caractere`.

Les lignes 35 à 40 permettent de sauvegarder la tableau `img_ASCII` dans le fichier `ascii.txt`.

10. Après avoir vérifié que le script est complet, **tester**-le (selon la taille de la photo le temps d'exécution peut être long). **Ouvrir** le fichier `ascii.txt` et **vérifier** que le dessin art ASCII correspond bien à l'image de départ.
11. **Tester**-le sur une autre image (Attention à ne pas prendre une image avec des dimensions trop grandes).



```
1 from PIL import Image
2
3 img = Image.open("yoda.jpeg")
4 largeur, hauteur = # dimensions de l'image RGB
5
6 img_ASCII =
7
8 for l : # Parcourir les lignes du tableau img_ASCII
9     for c : # Parcourir les colonnes du tableau img_ASCII
10         somme = 0
11         for y : # Parcourir les lignes de la zone
12             for x : # Parcourir les colonnes de la zone
13                 # récupérer les composantes R, G et B
14                 # calculer niveau de gris
15             somme =
16         moyNG = # calculer la moyenne des niveaux de gris
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34 # sauvegarde du tableau dans le fichier ascii.txt
35 f = open('ascii.txt', 'w')
36 for l in range(hauteur // 5):
37     for c in range(largeur // 2):
38         f.write(img_ASCII[l][c])
39         f.write('\n')
40 f.close()
```