Image Manipulation in Numpy - Structuur van finale programma

# Deel 1

## Imports

## Basis Functies (#11)

* **imagetomatrix (file)**functie om beeld (.jpg) om te zetten naar (image)matrix
* **fliphor (image)**functie om (image)matrix horizontaal te spiegelen
* **flipver (image)**functie om (image)matrix vertikaal te spiegelen -
* **fliphorver (image)**functie om (image)matrix horizontaal en vertikaal te spiegelen
* **keepcolor (image,color),** color in (0,1,2)  
  functie om 1 kleur van imagematrix (rgb) te bewaren -
* **keepcolor\_w (image,color),** color in (0,1,2)  
  functie om wit + 1 kleur van imagematrix (rgb) te bewaren -
* **keepcolor\_rgb (image,color),** color in (r,g,b)  
  functie om 1 kleur van imagematrix (rgb) te bewaren
* **colorflip (image, cf\_type),** cf\_type in (r0, r1, …)  
  functie om 1 kleur van imagematric (rgb) te bewaren en tegelijkertijd te flippen
* **larger (matrix,f) larger (matrix,f)**functie om (image)matrix een factor te vergroten
* **repeat\_k (matrix,fk)**functie om (image)matrix horizontaal te herhalen
* **repeat\_r (matrix,fr)**functie om (image)matrix vertikaal te herhalen

## Advanced Functies

* **grid\_with\_flips (image, matrix)**   
  the argument “matrix” is a matrix containing 0, 1, 2, 3 which stand for flip types
* **create\_colorful\_big\_one (image, colors)**   
  the argument “colors” is a list of ‘b’, ‘r’, ‘g’
* **square\_colorsflips (image, n)**  
  the argument “n” is the dimension of the square of manipulations  
  the program will generate itself a random list of 4\*n-4 flip types (r0, r1, …)  
  e.g. n = 7 🡪 24 flip types; e.g. n = 10 🡪 36 flip types
* functies om (image)matrix horizontal/vertikaal te knippen  
  **cuthor (image, nb\_pieces, line\_thickness, line\_color)  
  cutver (image, nb\_pieces, line\_thickness, line\_color)  
  cuthorver (image, nb\_horpieces, nb\_verpieces, line\_thickness, line\_color)**

# Deel 2 : Hoofdprogramma

## Kiezen, inlezen en omzetten van een image naar numpy-matrix

Verschillende VIERKANTE foto’s in data/input

Foto’s tonen.

INPUT Gebruiker : keuze van foto  
Image.open(os.path.join(os.path.dirname(os.getcwd()),'data\input\*python.jpg*'))

## Opdracht Stap 4 : Opgelegde manipulaties

De door Tim voorgestelde manipulaties uitvoeren

* 3x8
* 4x6
* 4x4

## Opdracht Stap 5 : Generalize

* **grid\_with\_flips (image, matrix)**
  + matrix1 :[[0 for i in range(8)] for j in range(3)]komt overeen met eerste manipulatie van opdracht stap 4
  + matrix2 : [[j for i in range(6)] for i in range(4)]  
    komt overeen met tweede manipulatie van opdracht stap 4
* **create\_colorful\_big\_one (image, colors)**  
  + colors1 : ['b','b','b','b','r','r','g','g','g','g','r','r']  
    komt overeen met derde manipulatie van opdracht stap 4
* **square\_colorsflips (image, n)**
  + The user is asked to choose n, where n is the dimension of the square of image manipulations

## Opdracht Stap 6 : Eigen manipulaties

Knippen

* **cuthor (image, nb\_pieces, line\_thickness, line\_color)**chosen\_image\_cuthor = cuthor(chosen\_image\_array, 15, 10, [103,56,121])
* **cutver (image, nb\_pieces, line\_thickness, line\_color)**chosen\_image\_cutver = cutver(chosen\_image\_array, 4, 10, [103,56,121])
* **cuthorver (image, nb\_horpieces, nb\_verpieces, line\_thickness, line\_color)**chosen\_image\_cuthorver = cuthorver(chosen\_image\_array, 10, 10, 2, [103,56,121])

Valentine

The program will create a canvas with red manipulations being a heart.

Puzzel

The program will generate a puzzle, after asking the user for the number of horizontal and vertical pieces of the puzzle.

Next, the user is invited to solve the puzzle.