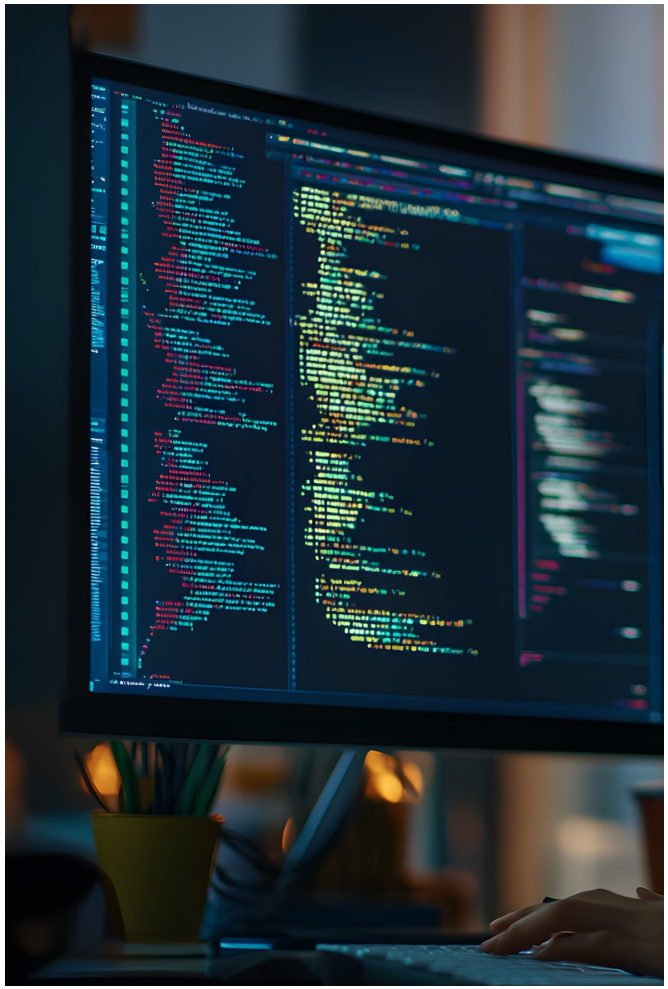
Information, Calcul et Communication

CS-119(k) ICC – Programmation
Semaine 10

Rafael Pires
rafael.pires@epfl.ch

Lausanne, 27.04.2025

Précédemment, dans... ICC-P



Planning

Cours et séries, partie programmation

P	1	2	3	4	5	6	7		8	9	10	11	12	13	
T	1	2	3	4	5	6			8	9	10	11	12	13	14

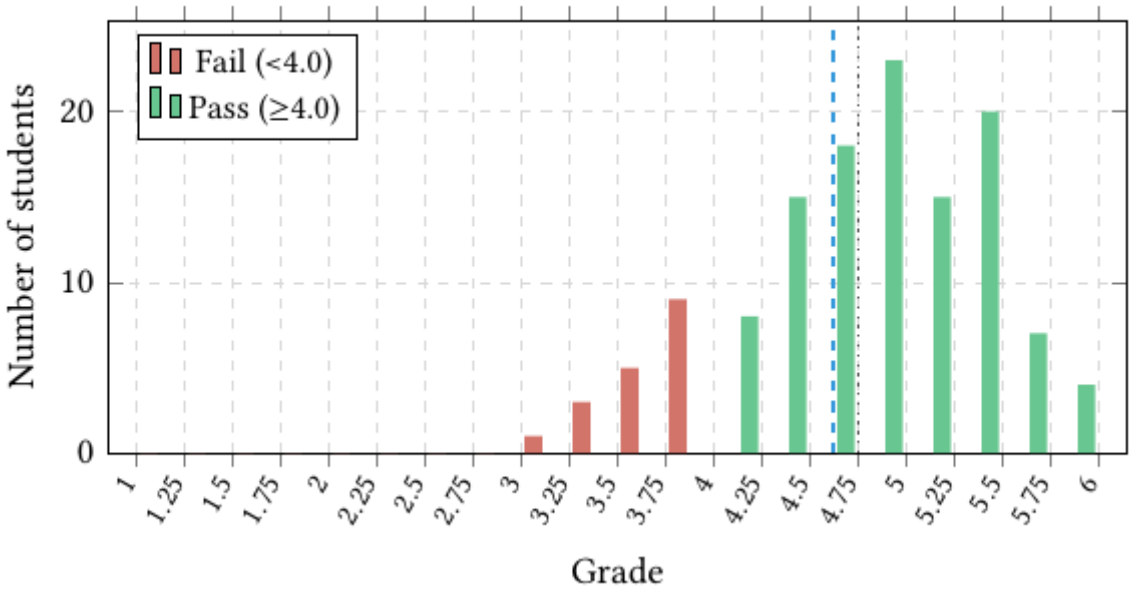
Cours et séries, partie théorique

01.05 Changement de salle : ELA 1

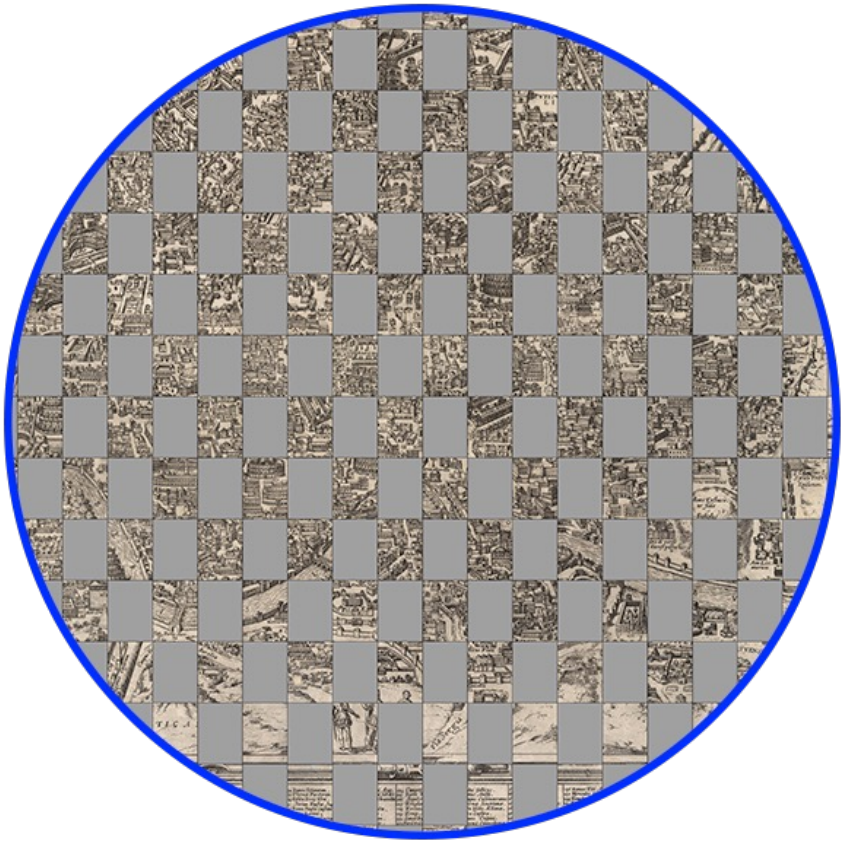
26.06 9h15 Examen **P (1/3)** et **T (2/3)** (**65%**)

Midterm

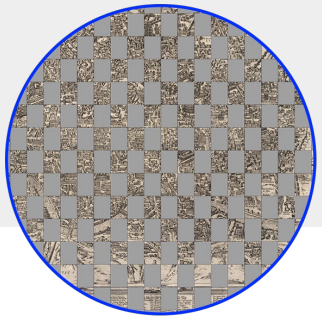
- Note moyenne : 72.4 % $\pm$ 12.2 %, 85.9% supérieur à 4.0
- Notes disponibles sur Moodle demain



Mini-Projet



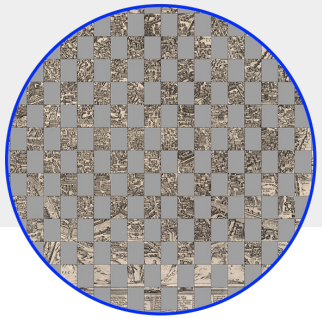
Charger une image : qu'obtient-on vraiment ?



```
img_uint8 = np.array(PIL.Image.open('imgs/europe.jpg'))
print('type :', type(img_uint8))
print('shape :', img_uint8.shape) # (H, W, 3)
print('dtype :', img_uint8.dtype) # uint8 ← directement depuis le fichier
```

```
type  : <class 'numpy.ndarray'>
shape : (1536, 1082, 3)
dtype : uint8
```

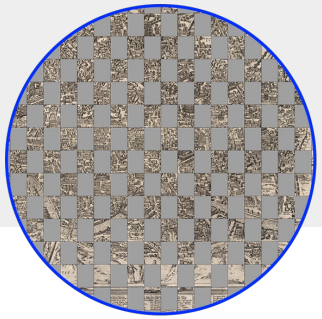
Charger une image : qu'obtient-on vraiment ?



```
img[ligne, colonne, canal]
      │       │       └── 0=R  1=G  2=B
      │       └────────── 0 … W−1  (gauche → droite)
      └────────────────── 0 … H−1  (haut  → bas)
```

Expression	Ce qu'on obtient
`img[0, 0, :]`	Triplet RGB du pixel en haut à gauche
`img[0, 0, 2]`	Canal bleu du pixel en haut à gauche
`img[100, 200, :]`	Triplet RGB du pixel à la ligne 100, colonne 200
`img.shape[0]`	Hauteur
`img.shape[1]`	Largeur

Pourquoi le projet utilise int16 et non uint8 ?



```python
a = np.uint8(10)
b = np.uint8(200)
print('uint8  10 - 200 =', a - b)  # débordement : 10 - 200 + 256 = 66  ← FAUX

a16 = np.int16(10)
b16 = np.int16(200)
print('int16  10 - 200 =', a16 - b16)  # -190  ← correct
```

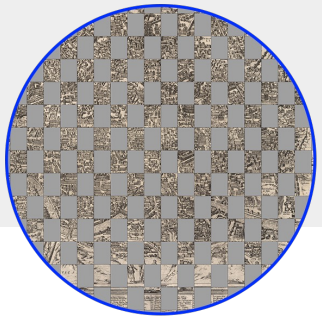
```
uint8  10 - 200 = 66
int16  10 - 200 = -190
```

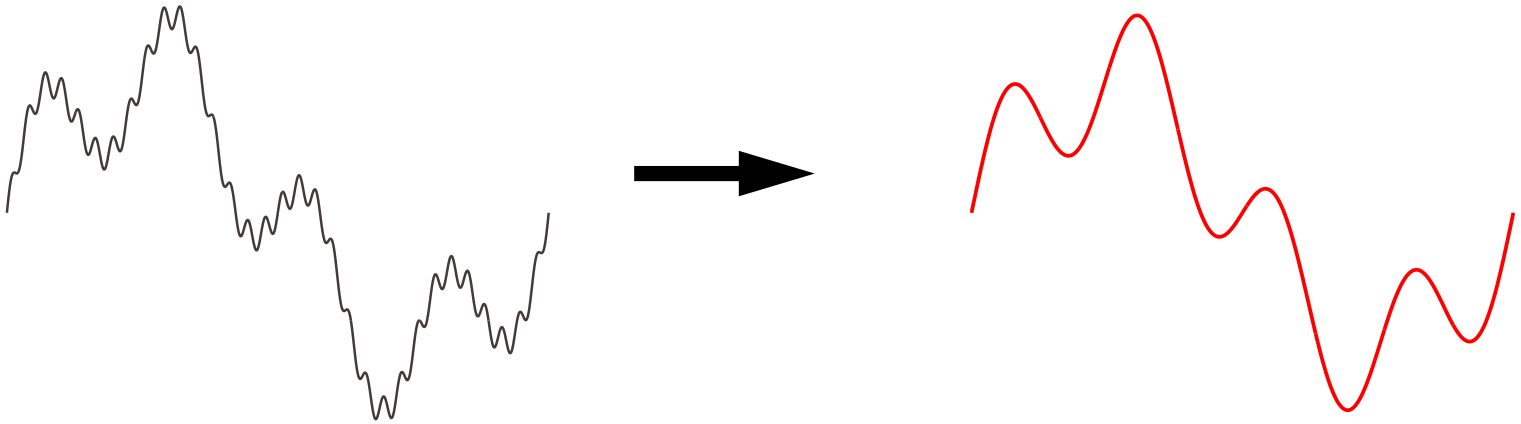
```
/var/folders/nq/bqjgm5t13f9ff40m2glp3s640000gn/T/ipykernel_83075/352342616.py:3:
RuntimeWarning: overflow encountered in scalar subtract
  print('uint8  10 - 200 =', a - b)  # débordement : 10 - 200 + 256 = 66  ← FAUX
```

```python
# Dans le projet, les tuiles sont toujours chargées en int16 :
tile = np.array(PIL.Image.open('tiles/tile_given_0_0.jpg')).astype(np.int16)
print('dtype de la tuile :', tile.dtype)   # int16
print('shape de la tuile :', tile.shape)   # (H, W, 3)
```

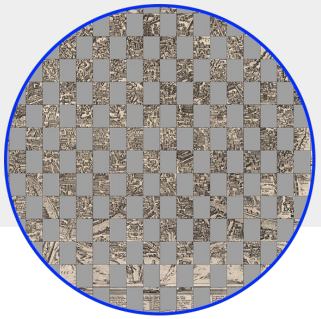
```
dtype de la tuile : int16
shape de la tuile : (304, 223, 3)
```

Filtrage d'un signal



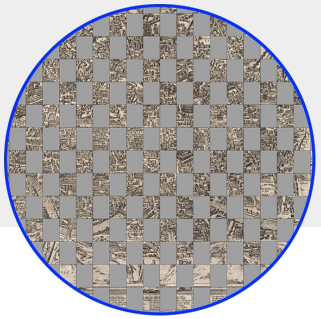


Convolutions

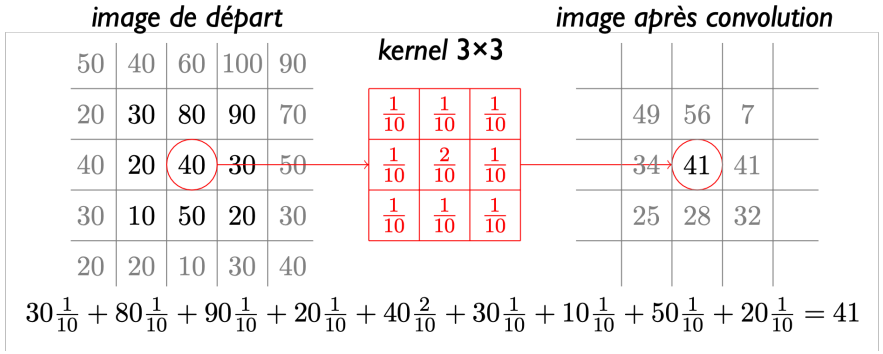


- Opération mathématique qui permet de transformer une image en appliquant un filtre (ou noyau).
- Ce filtre passe sur l'image comme un petit cadre, et à chaque position, il fait une sorte de moyenne pondérée des pixels, ce qui permet de :
  - Détection des contours
  - Réduction de bruit
  - Flou

La convolution, comment ça marche ?



- Choisir un filtre : par exemple une petite matrice 3x3 comme.



- Faire glisser ce filtre sur l'image, pixel par pixel.
- À chaque position, faire le produit élément par élément entre le filtre et la portion de l'image (3x3), puis faire la somme des résultats.
- Remplacer le pixel central par ce nouveau résultat.

Résumé Cours 10 – ICC-P

- Annonces :
  - date de l'examen (26.06 9h15-12h15)
  - changement de salle (01.05 ELA 1)
- Résultats du Midterm
- Filtrage d'un signal et convolutions
- Démo convolutions

rafael.pires@epfl.ch

Merci