

## Percorso Matrici e immagini digitali con Python - Introduzione e indicazioni operative

Un'immagine non è altro che una griglia di pixel, ovvero di piccoli quadratini con associato un colore.

Ma come vengono rappresentati i colori di ogni pixel? Nel caso più semplice, ovvero di immagini in bianco e nero, ogni pixel è semplicemente un numero, 0 o 1, a seconda del colore; l'immagine, quindi, è una matrice (per esempio, 1024 x 768) di zero e uno. Ammettendo numeri tra 0 e 255, invece, è possibile costruire un'immagine in "scala di grigi": il nero è 0, il bianco 255 e più alto sarà il numero inserito più il pixel sarà "chiaro". Il valore determina pertanto l'intensità del colore rappresentato, in questo caso il nero. Per introdurre i colori esistono diversi formati: il più comune è quello "RGB" (Red Green Blue) caratterizzato da una terna di numeri per ogni pixel (di fatto quindi da tre matrici) indicanti rispettivamente l'intensità del rosso, del verde e del blu.

Proprio a partire dalla struttura numerica delle immagini digitali, nel percorso proposto si svolgerà un'attività diretta di programmazione con il coinvolgimento degli studenti

### Obiettivi

- Consolidare, attraverso l'uso di Python, alcune competenze di tipo matematico su vettori e matrici e sulle loro operazioni, sulle trasformazioni geometriche e sugli algoritmi;
- Realizzare semplici programmi in grado di elaborare e manipolare immagini digitali.

### Prerequisiti

- Vettori, matrici e loro operazioni;
- Piano cartesiano;
- Trasformazioni geometriche nel piano.

Il percorso può essere introdotto a partire dal secondo biennio di Scuola Secondaria di Secondo grado e favorisce, alla stessa stregua di tutta la programmazione, la creatività, il ragionamento e la capacità di risolvere i problemi.

Si è scelto di utilizzare Python, in quanto linguaggio di programmazione facile da imparare e introdotto in molte scuole sin dai primi anni, e la piattaforma Colab di Google (una interfaccia interattiva su cui è possibile scrivere i programmi e vederli girare senza dover scaricare Python sul proprio pc) attraverso la gestione di singoli Notebook tematici, uno per ogni argomento.

La flessibilità di Colab consente al docente di decidere, in totale autonomia, di tagliare, modificare o spostare intere celle di testo e codice, potendo così scegliere quali argomenti trattare tra quelli elencati, se eliminare una parte del codice fornendo invitando lo studente a completarlo oppure se fornire agli studenti esercizi aggiuntivi sulla base di quelli proposti. Il docente può, in definitiva, comporre il proprio percorso didattico smontando e ricostruendo i Notebook Colab offerti (per questo viene fornito ai docenti anche un Tutorial per l'utilizzo didattico dei Notebook).

Il percorso può essere proposto in due modalità, in base al progresso della classe:

**Prima modalità:** gli studenti arrivano a questo percorso dopo aver svolto, nel primo biennio, quanto proposto nella prima parte del progetto "Matrici e immagini digitali", ovvero gli studenti possiedono già i prerequisiti necessari e hanno già sperimentato altri strumenti di programmazione, quali GeoGebra o Snap!.

**Seconda modalità:** il docente intende partire direttamente da questo percorso in linguaggio Python; in tal caso occorre accertarsi che gli studenti siano in possesso dei prerequisiti sopra menzionati. Qualora gli studenti conoscano già gli elementi di base del linguaggio Python si può ignorare il Notebook NB0.

Indipendentemente dalla modalità che il docente intende seguire:

- in ogni Notebook è richiamata brevemente la parte di teoria relativa all'argomento trattato;
- in ogni codice sono riportate le spiegazioni relative alle funzioni utilizzate.

Il percorso è articolato in cinque attività, ciascuna associata a uno o più specifici Notebook (per ognuno dei quali è fornito il link corrispondente):

### **Attività 1: Introduzione all'uso del linguaggio Python in generale**

Nel Notebook (**NB0**) si illustra come importare moduli, come implementare cicli e alternative, come creare un vettore o una matrice e come creare un grafico nel piano cartesiano. Sono riportati, inoltre, semplici esempi per familiarizzare con i comandi base e prepararsi alle sfide successive. Si tratta di un modulo da usare qualora non sia stata già svolta un'attività simile.

<https://colab.research.google.com/drive/18Jd2S6P6j7aqhPedgpgmjxOKriBf6Lby?usp=sharing>

### **Attività 2: Introduzione al calcolo vettoriale e matriciale con Python**

Si illustra come effettuare le operazioni tra vettori e matrici (**NB1**). Le matrici, inoltre, vengono trattate come operatori per trasformazioni geometriche di semplici figure piane, con particolare riguardo alle isometrie (traslazioni, simmetrie, rotazioni), alle trasformazioni affini e alle omotetie (**NB2**).

[https://colab.research.google.com/drive/1\\_MT8FnHvX1jCa5Shsi5DSEDwUexgyJ7b?usp=sharing](https://colab.research.google.com/drive/1_MT8FnHvX1jCa5Shsi5DSEDwUexgyJ7b?usp=sharing),

<https://colab.research.google.com/drive/1LGLa19ooPP0hwKwcfTdicKRhijiroQTt?usp=sharing>

### **Attività 3: Dalla matrice all'immagine digitale e viceversa**

Nel Notebook (**NB3**) dopo aver richiamato i concetti di pixels, immagine vettoriale e immagine bitmap, si illustra come ottenere in Python le immagini binarie (bianco e nero), in scala di grigi e a colori (RGB). Ci si sofferma sui vari formati di compressione (jpeg, png, bmp, gif, tiff), per poi analizzare il passaggio dalla matrice all'immagine e viceversa, nonché sulla costruzione e analisi di semplici immagini binarie, sulla mappa dei colori principali nel sistema RGB e sulla scomposizione di un'immagine nelle tre bande.

<https://colab.research.google.com/drive/1pzX7e9NllsMWgzpgxZscJ-A5Dq4idgMx?usp=sharing>

### **Attività 4: Manipolazione di immagini digitali**

Nel Notebook (**NB4**) si illustrano le azioni ottenibili con semplici manipolazioni delle tabelle delle immagini, quali la copertura, il taglio o la scalatura di porzioni di un'immagine e il collage, e le azioni sulle bande dei colori per variarne le intensità. Si passa poi ad analizzare gli effetti soglia, il passaggio da RGB a scala di grigi, la transizione da un'immagine a un'altra (somma di immagini) e lo shift di un pixel per l'individuazione dei bordi (differenza di immagini).

<https://colab.research.google.com/drive/1fNKEs5N1IQKPIsM1fmzLEunP20aeCGuB?usp=sharing>

### **Attività 5: Modulo di approfondimento su alcune elaborazioni complesse di immagini digitali**

Nel Notebook (**NB5**), si discutono le idee matematiche sottostanti ad alcune elaborazioni complesse di immagini. Ci si sofferma per esempio sui filtri di convoluzione, lineari e non lineari, e sulle azioni locali attorno a ogni pixel con effetto di sfocamento (blur), per determinare i contorni

(edge detection) o per rimuovere il rumore (noise filtering). Si analizzano, infine, le trasformazioni geometriche applicate alle immagini digitali e la compressione delle immagini, in particolare la decomposizione SVD.

<https://colab.research.google.com/drive/1J1VdKUvMKiyQvEg61zE29a8WcQgRQdr4?usp=sharing>