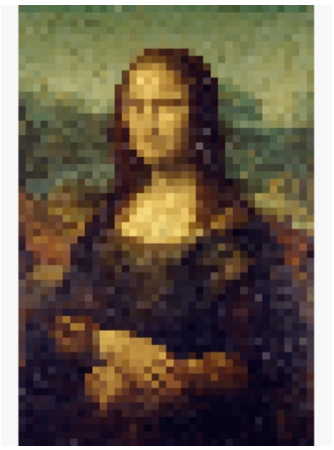
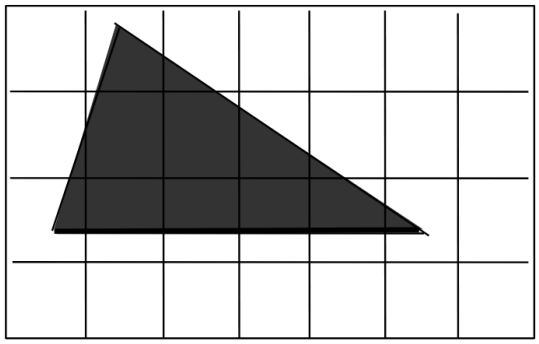


Bitmap di un’immagine:

L’immagine da rappresentare viene suddivisa in una griglia più o meno fitta, che ne determina la **risoluzione**. Ogni quadratino della griglia è detto **pixel**

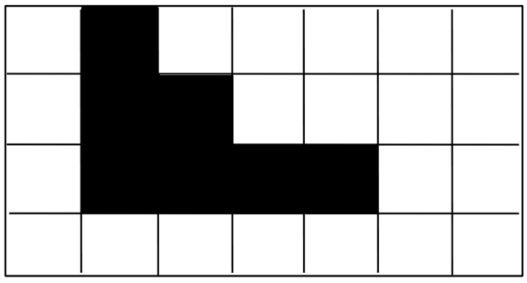
0000000 0111100 0110000 0100000



0_{22}	1_{23}	0_{24}	0_{25}	0_{26}	0_{27}	0_{28}
0_{15}	1_{16}	1_{17}	0_{18}	0_{19}	0_{20}	0_{21}
0_{8}	1_{9}	1_{10}	1_{11}	1_{12}	0_{13}	0_{14}
0_{1}	0_{2}	0_{3}	0_{4}	0_{5}	0_{6}	0_{7}

Usando la sequenza di bit a 0 o a 1 che rappresentano quadratini bianchi o neri, dalla stringa

0000000011110001100000100000 otteniamo:



Se vogliamo una accuratezza migliore, dobbiamo aumentare la risoluzione, ossia usare una griglia più fitta (ad esempio, 1024 x 768, oppure 2880 x 1864)

Se vogliamo rappresentare pixel colorati, un solo bit non ci basta. In informatica vengono usate soluzioni diverse. Una delle più note è il formato RGB: ogni pixel viene viene colorato con una combinazione di **tre colori di base**:

Red – Green – Blu = **RGB**

(ma esistono anche altri formati, come il C-M-Y-KB e il H-S-V)

1 byte viene usato per indicare la quantità di colore presente per ognuna dei tre colori di base:

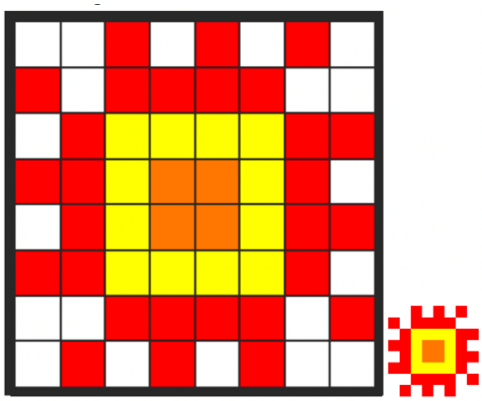
00 = componente di colore completamente assente;
FF = componente di colore completamente presente;
in totale, 256 livelli di colore per ognuno dei colori di base.

Così, possiamo rappresentare 256*256*256 = 2^24 ≃ 16 milioni di colori diversi

- BIANCO: tutte le componenti **presenti**
- NERO: tutte le componenti **assenti**
- GRIGIO: tutte le componenti a **metà**
- GIALLO: ROSSO + VERDE
- ARANCIONE: ROSSO + metà dose VERDE
- MAGENTA: ROSSO + BLU
- CIANO (azzurro): VERDE + BLU

https://colorizer.org/

rosso	verde	bianco	nero	grigio	giallo	arancione	magenta	azzurro	blu
FF0000	00FF00	FFFFFF	000000	808080	FFFF00	FF8000	FF00FF	00FFFF	0000FF



Ogni pixel viene rappresentato dal codice esadecimale del suo colore RGB, ad esempio partendo dal pixel più in alto a sinistra:

prima riga:
FFFFFF FFFFFF FF0000 FFFFFF FF0000 FFFFFF FF0000 FFFFFF
...
quarta riga:
FF0000 FF0000 FFFF00 FF8000 FF8000 FFFF00 FF0000 FFFFFF

In tutto ci vorranno 8 * 8 * 3 byte = 192 byte per rappresentare l'immagine

Per diminuire lo spazio occupato dall'immagine, **il formato GIF (Graphic Interchange Format)** inserisce in una tabella (o *palette*) i soli colori utili a rappresentare una certa immagine. Nel nostro esempio:

palette: 0=FFFFFF = BIANCO
1=FF0000 = ROSSO
2=FFFF00 = GIALLO
3=FF8000 = ARANCIONE

codifica della quarta riga: 1 1 2 3 3 2 1 0

Se anche usiamo un byte per ogni colore (e notate che ci basterebbero 2 bit), in tutto abbiamo bisogno di:

8 * 8 + 4 * 3 byte = 76 byte. Molto meglio di 192.

Nel caso reale, le immagini GIF usano una palette con 256 colori diversi, per cui abbiamo effettivamente bisogno di un byte per rappresentare il colore di ogni pixel (saranno i 256 colori predominanti dell'immagine)

Altri formati, ad esempio TIFF, permettono una accuratezza maggiore, ma al costo di file più grossi. I formati JPG e PNG usano forme di compressione più sofisticate, ottenendo un buon compromesso tra accuratezza e dimensione del file.