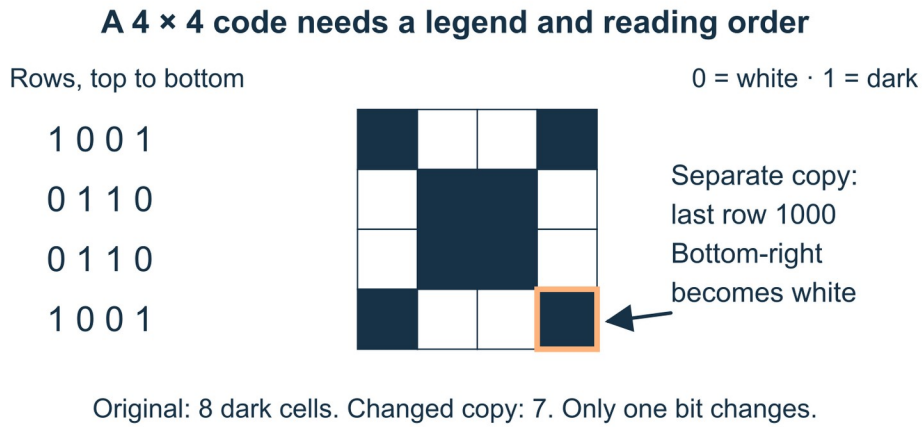
Encode a tiny pixel picture

Can someone rebuild a small image from an unambiguous written code?

Decode an original four-by-four bitmap with a complete legend, then check how one changed bit alters the picture.

Make | 15–25 minutes, or one row | The supplied example, read or heard, Optional scrap paper and a pen or pencil

Start here Use a 4-by-4 grid and these rows from top to bottom: 1001, 0110, 0110, 1001. Read each row left to right; 1 is dark and 0 is white.



A code needs an arrangement rule

1 Keep the header with the data

Write width 4 and height 4 beside the rows. Without a size and reading-order convention, a string of digits may be arranged in different ways.

2 Decode the original

Shade cells marked 1 and leave 0 white. The pattern has dark cells along both diagonals: eight dark and eight white cells.

3 Check a round trip

Read your shaded grid back into four digit rows. Compare every row with the supplied code rather than deciding only whether the picture looks familiar.

4 Change one bit

Make a separately labelled copy whose last row is 1000. The bottom-right cell becomes white; the other 15 cells stay the same. Keep the original for comparison.

Original bitmap a tiny diagonal pattern

Header: width 4, height 4; row order top to bottom; within each row left to right; 0 white, 1 dark.

• Rows: 1001/0110/0110/1001. Each row contains four bits.

• The original contains eight digits equal to 1, so it has eight dark cells.

• Changing only the last digit produces 1000 in the last row and seven dark cells overall.

What this example can show This is a monochrome encoding exercise, not every image format. A code needs its agreed dimensions and symbol meanings to be decoded correctly.

A different way

Decode only the first row or point to the changed bottom-right cell. You can explain the supplied grid without drawing or installing anything.

If you want to go further

Describe row 1001 as one dark cell, two white cells and one dark cell. Explain why counts without a starting colour would be ambiguous.

Something to notice

Did my code preserve both the picture data and the rules for arranging it?

Read or explore more

Horizon topic Pixels Images

https://launchhorizonhq.com/learn/pixels-images/

Horizon topic Binary Digital Data

https://launchhorizonhq.com/learn/binary-digital-data/

Horizon topic Formats Compression

https://launchhorizonhq.com/learn/formats-compression/

Computer Science Field Guide, University of Canterbury Images and Colours

https://www.csfieldguide.org.nz/en/chapters/data-representation/images-and-colours/

Computer Science Field Guide, University of Canterbury Run length encoding

https://www.csfieldguide.org.nz/en/chapters/coding-compression/run-length-encoding/