Everything is a yes or no

Draw a single closed loop on a sheet of paper. A circle, a blob, any shape, so long as the line meets itself. Before you drew it, the page was undivided. After every point on the page is either inside the loop or outside it. Nowhere else. With one stroke, you've split the page into two regions and made the smallest possible distinction you can make.

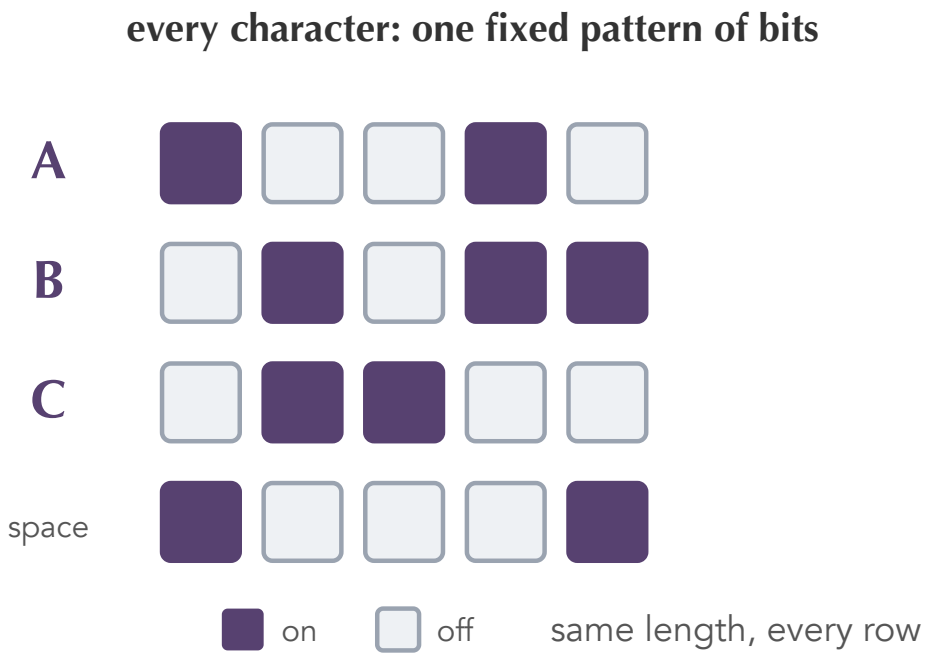
That single distinction, the one bit of "in or out", is what this chapter is about. Under everything that runs on a computer, under every photo, every song, every search result, every video call, is a long line of those distinctions. Yes or no. In or out. One or zero. Computing has exactly one alphabet, and it has two letters.

What is a bit, really?

> **THE LOOP ON THE PAGE**
>
> Your closed loop divides the page into exactly two regions: inside and outside. To say where a dot is, all you need to answer is one question: in or out? That single yes-or-no is the smallest possible piece of news. It cannot be split into smaller pieces. There is no half-answer.

The atom of information

A *bit* is that single yes-or-no. It is not a piece of metal or a wire in a machine; it is the smallest unit of information you can have, the smallest news you can pass. One bit, one distinction, one of two values: on or off, 1 or 0, yes or no, in or out. Smaller than that, there's nothing to say.[5]



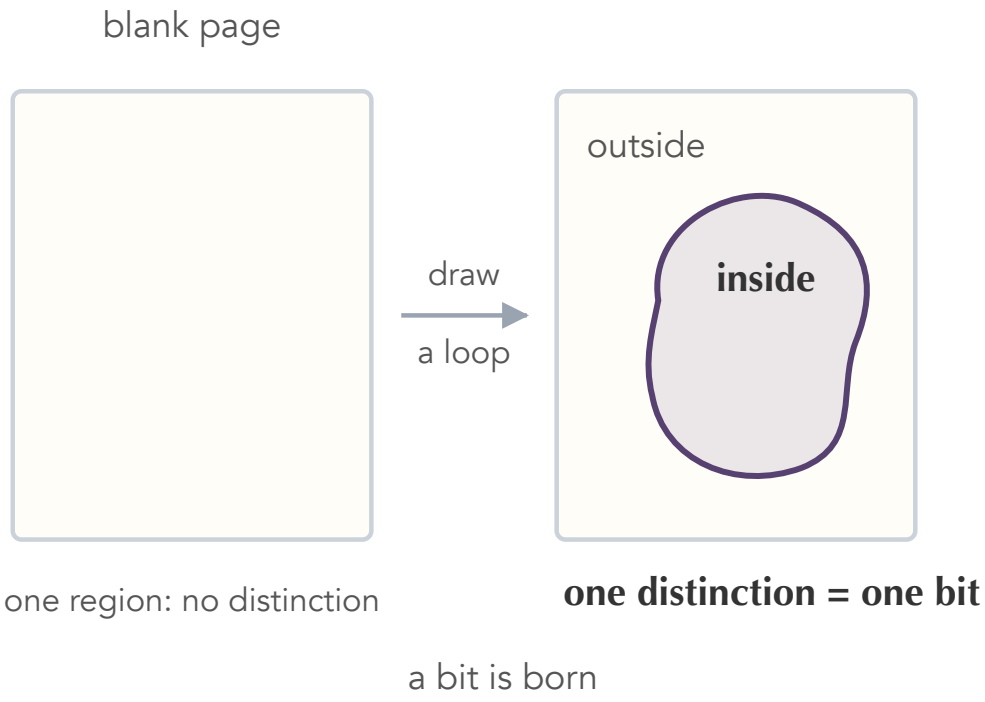
5.1: *Each character is one fixed pattern of bits*

5. Written formally: let *I(x)* denote the information content of *x*, measured in bits. The smallest non-trivial *I(x)* is exactly 1, the answer to a single yes-or-no; everything larger is built from those.

Why two, and not ten?

You might ask: why settle for two? Why not ten values per slot, the way we count in everyday life, or twenty-six, one per letter? The answer is that two is the floor. Two is the smallest number of options that still lets you tell anything apart at all. One option (every dot is "in") tells you nothing, because the answer never changes. Three or ten or twenty-six options can be built up from twos: a few yes-or-nos in a row can name any of them. But you can't go below two and still have a distinction. Two is the smallest stable unit of meaning, and that is why every machine we have built is made of stacks of it.

> **THE IDEA**
>
> *A bit is not machinery. A bit is the smallest piece of information you can have: a single resolved yes or no, a single distinction between this and not-this. It is the floor of information. Everything that a computer stores, sends, or computes is, in the end, a long line of these.*



5.2: *One line, the smallest distinction you can make*

How many questions to find one thing?

The world has more than two things in it. So the next question is the natural one: how many of these little yes-or-no atoms does it take to single out one thing in a big group of things?

We answered it in the last chapter without quite naming what we'd done. I was thinking of a number between 1 and a million, and you could pin it down in about 20 questions, each of the form "higher or lower than X?" Each question is a yes-or-no. So twenty bits are enough to identify any one of a million things.