# **From electricity to bits**

In a computer, information travels over wires. The easiest way to convey information in a wire is to consider it "on" or "off", based on how much electricity is going through it.



A diagram of 2 wires. The top wire has lightning bolts going through it, representing electricity, and is labeled ON. The bottom wire is filled in black and is labeled OFF.

An "on" wire represents 111, and an "off" wire represents 000.



Diagram of two wires, one with electricity pulsing through it, another without. The first on wire has a 1 next to it, the second off wire has a 0 next to it.

This small piece of information is called a **"bit"**, and it's the smallest piece of information that computers process.

## **More wires = more bits**

A single wire can only represent one **bit**, one piece of information. We can represent the results of a coin flip with a single bit—by saying that 0 represents tails and 1 represents heads—but we usually need to represent much more information than that in a computer.

The solution? More wires! Each wire adds an additional bit of information, an extra bit that can be considered on or off, 1 or 0.

For example, let's say we want to represent which of three lightbulbs to turn on. We can use three wires, with each wire representing the on/off state of a lightbulb:



Diagram of 3 wires. The first wire has electricity pulsing through it, is labeled 1, and is shown next to a lit lightbulb. The second wire is off, labeled 0, and is shown next to an off lightbulb. The third wire looks like the first.

In computers, we use bits to represent numbers, using the **binary number system**. We'll dive deep into binary numbers in the next article.

## **Behind the abstraction**

In actuality, a wire isn't exactly "on" or exactly "off". That's an **abstraction** that simplifies the details of how computers work. We use abstraction often in computer science so that we can more easily understand the systems that we're building. Let's peek behind the hood to see how this abstraction works.

A wire can have varying amounts of electricity flowing through it, but a computer needs to be able to interpret the electricity in a wire as either definitely 000 or definitely 111.

In 1947, engineers invented the **transistor**, a tiny physical device that acts like a digital switch in computers. The transistor turns on when enough electricity flows through and stays off otherwise.

How much electricity is "enough"? That depends on the transistor and its threshold voltage. If an engineer uses a transistor with a threshold voltage of 4.5 volts, then any voltage of 4.5 or higher will turn the transistor on. At lower voltages, the transistor stays off.

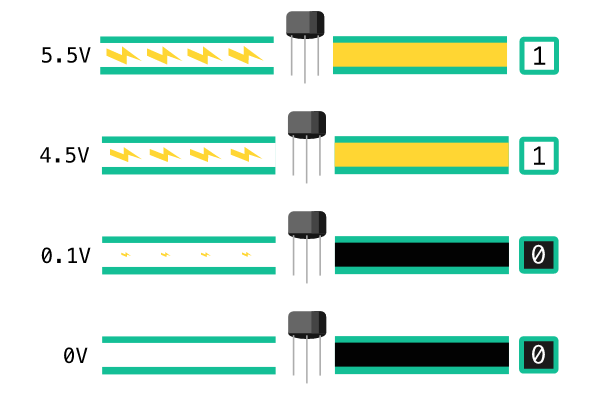


Diagram of 4 wires going through transistors. The top wire has electricity pulsing through it at 5.5V, and comes out of the transistor in the on state, labeled as 1. The second wire has electricity pulsing through it at 4.5V, and comes out of the transistor in the on state, labeled as 1. The third wire has a small amount of electricity pulsing through it at 0.1V, and comes out of the transistor in the off state, labeled as 0. The fourth wire has 0V of electricity going through it, and comes out of the transistor in the off state, labeled as 0.

Consider a computer that needs to determine whether a USB cable is plugged in. When you plug the cable of a mouse in the computer's USB port, circuitry in the mouse uses the voltage provided by the port to pull up the voltage in the cable above 3.3 volts. Inside the computer, a transistor detects the high voltage and translates it to "on" or 111. This bit of information tells your computer that a USB device is plugged into the port.

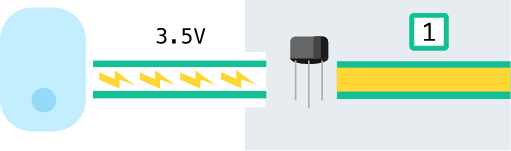
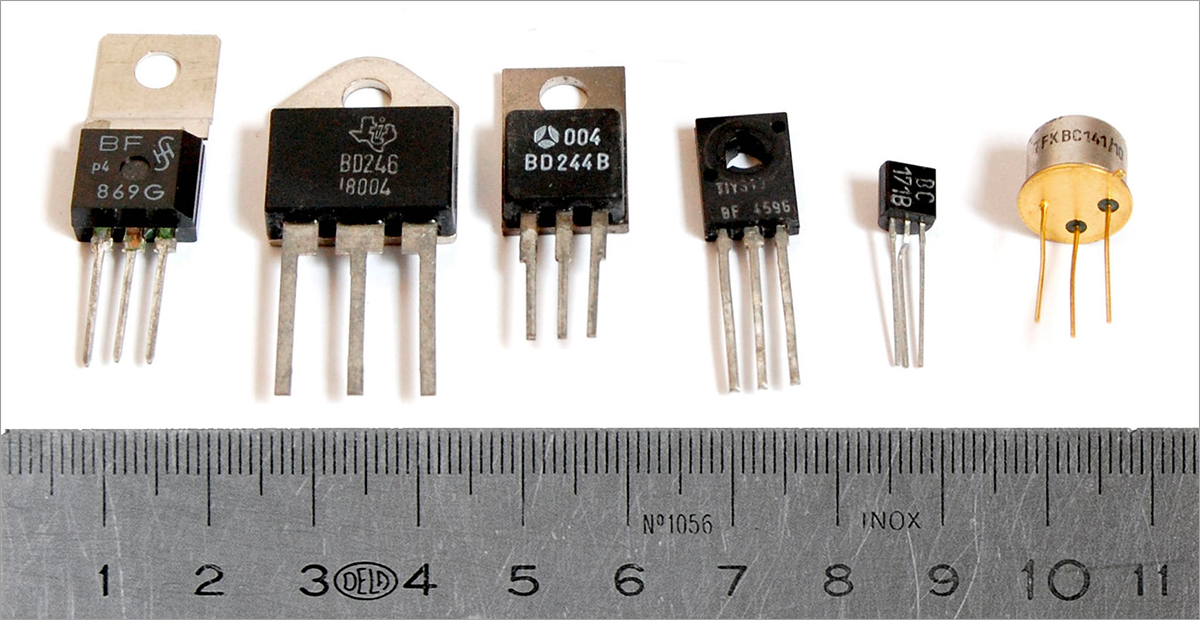


Illustration of a computer mouse with a wire labeled as 3.5 volts going into a USB port. Inside the computer next to the port, there's a transistor and a wire coming out labeled 1.

In this case, engineers used a transistor with the threshold voltage of 3.3 volts for the "on" state, and of 0.3 volts for the "off" state.

There's a huge variety in transistors. Engineers choose the transistors that are the best fit for the job, by considering characteristics like the threshold voltage, material, and size.

The transistors inside your computer are so small, we would need a high-powered microscope to see them. However, transistors are also used in other electrical projects and those transistors are ones you could pick up with your fingers. Here are a few examples:



6 transistors, each transistor has an encased rectangular top and three metal prongs emerging from it.

A lot of electrical engineering and physics goes into the physical construction of computer hardware like transistors, and we won't dive deep into that here. If you'd like to learn more, check out this video on [how a transistor works](https://www.youtube.com/watch?v=DXvAlwMAxiA).

An important takeaway here is that computers are built on layers of abstraction, like bits abstracting on top of transistors. Those layers enable computer scientists to use and control computers in predictable ways.