basics of HTML and CSS (introduced on Days **2** and **3**) then look at the software that makes them possible: web servers like Apache, programming environments like C++.

Many are simpler than they look; with the right learning methods you can gain basic skills in a month. (Mastery is far harder, but it's surprisingly easy to get a grip on the fundamentals.)

Learn electronics by building a computer

It's simpler than you think to **build your own PC**. It can be cheap—just a couple of hundred £ for a capable desktop box. The basics are a **case** with **power supply**, into which you'll fit a **motherboard** with **processor**, **memory**, and **expansion slots**, plus **connection** options for WiFi and broadband. Also inside go a **hard drive** for storage, while outside go **keyboard** and **screen**.

(*Multiple* screens are a great upgrade for getting work done; graphics cards can now power several at 4K resolution, giving you a huge canvas to create on.)

Buying parts in the hope they'll work may sound risky, but the PC is a mature sector with long-established hardware formats, and all vendors have an interest in making sure their gear plays well with others.

Building your own box also teaches you a great deal about **electricity** and **electronics**. What amps, volts, and watts do; the small set of parts—resistor, capacitor, diode, transistor, amplifier—that join together to make logic gates and integrated circuits; the standards that let different devices talk to each other. The most complex computer is just a big assemblage of bits like these, and learning how they work gives you a healthy understanding of the machines the world runs on. And there's nothing like that feeling when you first switch on a PC you've made yourself.

Alternatively, explore the Raspberry Pi, a deliberately bare-bones hobbyist computer the size of a small paperback with much the same power and connectivity as a small desktop PC. Many can substitute for a full machine.

Get into virtual worlds with computer graphics

With an understanding of code, among the most absorbing areas to explore is **computer graphics**. How pixels and triangles relate to voxels and vertices; how realistic three-dimensional worlds result from texture maps and how augmented reality is blurring the real and unreal.

A top-end **graphics card** in your PC is the start; graphics chips are designed to handle visual elements and nothing else, and with other applications like AI driving the growth curve, there's no end in sight for performance. And you don't need to code your own experiences from the ground up. Countless games,