

The Role of Coding in Everyday Life

Format: Expository | Level: Middle School | ~500 words

Most people use Google Maps every morning without thinking about the algorithm deciding whether to route them down Main Street or the highway. That decision involves graph theory, real-time data, and a computational process completing in under a second on a server hundreds of miles away. Coding is not a specialized skill confined to software companies; it is the mechanism behind nearly every tool a person touches before they leave their house.

Consider the recommendation system on a streaming service. When a platform suggests a show based on viewing history, it is executing a collaborative filtering algorithm that compares the user's behavior to thousands of similar users and identifies patterns too complex for any human analyst to compute manually. The system does not know what is good. It knows what people with similar preferences chose to watch next, and it uses that information to make a prediction. That is the basic logic of machine learning, running invisibly behind an interface that looks like a simple list of suggestions.

Navigation applications work on a similar principle. The shortest route between two points is a classic problem in graph theory, where intersections are nodes and roads are edges with weights representing distance or travel time. When traffic data is added, the weights change in real time and the algorithm recalculates. A student who understands this concept does not just know how to use a map app; they understand why it sometimes takes them on routes that look strange on the screen but turn out to be faster.

Even a basic household appliance like a smart thermostat relies on code. The device collects temperature data, learns the household's schedule, applies a set of rules to predict when the house needs to be heated or cooled, and adjusts automatically. The programming involved is not sophisticated by professional standards, but it represents the same fundamental logic that underlies systems far more complex: if this condition is true, do this thing; otherwise, do something else.

Coding literacy does not mean everyone needs to learn how to build software. It means understanding enough about how computational systems work to interact with them thoughtfully, recognize their limitations, and evaluate their outputs critically. A student who knows that a recommendation algorithm is optimizing for engagement, not quality, is better equipped to decide what to watch, read, or believe. A student who understands how location data is collected and used is better equipped to make decisions about privacy. The argument for coding literacy is not that everyone should become a programmer. It is that the world already runs on code, and understanding

it is part of understanding the world.

This essay is provided by CollegeEssay.org as a learning example. For reference only — not for submission.



CollegeEssay.org