HOW SOFTWARE WORKS · BEGINNER · 18 MIN READ

Read code like a builder (with AI as your tutor)

Variables, functions, if/else, loops, objects and components: just enough JavaScript to read what AI writes and ask better questions.

Pass the check (4 of 5) online to earn the “Code Reader” badge:
https://digitalbuilders757.com/university/study/read-code-with-ai

You don't need to write it all. You do need to read it.

Vibe coding lets AI type the code, but you're still responsible for what it does. Being able to read a little code helps you spot when the AI did something odd, explain bugs precisely, and learn faster every time you build.

The examples here are JavaScript, the language of the web. Most languages use the same ideas with different spelling.

Variables: named boxes

A variable stores a value under a name. const price = 20 creates a box called price holding 20 that won't change. let count = 0 creates one that can change later (count = count + 1).

Values have types: numbers (20), strings of text ("hello"), booleans (true or false), lists called arrays (["red", "blue"]) and objects that group named values ({ name: "Ada", age: 36 }).

Functions: reusable recipes

A function is a named recipe that takes inputs (parameters) and returns an output. function total(price, tip) { return price + price * tip } takes a price and a tip rate and returns the total. You call it with total(20, 0.15).

You'll also see arrow functions, a shorter way to write the same thing: const total = (price, tip) => price + price * tip.

Decisions and repetition

- if / else: if (age >= 18) { showSignup() } else { showParentNote() }. Only one branch runs.
- Comparisons: === means 'is exactly equal', !== 'is not equal', and && means AND, || means OR.

- Loops repeat work: for (const item of cart) { sum = sum + item.price } adds up every item.
- Array helpers do common loops for you: items.map(...) transforms each item, items.filter(...) keeps some, items.find(...) gets one.

Components, imports and async

In React, a component is a function that returns what the screen should show, written in JSX (HTML-like tags inside JavaScript): function Greeting({ name }) { return <h1>Hi, {name}</h1> }. Props are the inputs to a component; state is data the component remembers and updates.

import pulls code from another file or package; export shares it. async and await handle things that take time, like loading data from a database: const notes = await getNotes() waits for the answer before continuing.

Use AI as a tutor, not just a typist

- 'Explain this file line by line for a beginner. Don't change anything.'
- 'What would happen if this value were empty?'
- 'Rewrite this function more simply and explain the difference.'
- 'Quiz me with three questions about this code.'
- Change one small thing yourself (a label, a number) and predict what will happen before you run it.

Key terms

Variable
A named value in code, like const price = 20.

Function
A named, reusable recipe in code that takes inputs and can return an output.

Array
A list of values in code, like ["red", "blue"].

Object (in code)
A group of named values, like { name: "Ada", age: 36 }.

Conditional (if/else)
Code that runs only when something is true, otherwise something else.

Loop
Code that repeats for each item in a list or until something changes.

Component
A reusable piece of interface, in React a function that returns what to show.

Props
The inputs passed into a React component.

State
Data a component or app remembers and updates, like what's in a cart.

Async / await
Code for slow tasks like loading data: await waits for the answer.

JavaScript
The programming language that makes web pages interactive.

Try it: Predict, then run

1. Ask an AI for a 20-line JavaScript tip calculator with a function, an if/else and a loop.
2. Before running it, write down what you think it prints.
3. Run it (an online playground or your browser's console works) and compare.
4. Ask the AI to quiz you on it, and change one line yourself to see what breaks.

KNOWLEDGE CHECK

Read code like a builder (with AI as your tutor)

5 questions. Circle one answer for each; 4 or more correct passes. Answers are at the back.

1. What does `const price = 20` do?

A. Creates a named value, price, set to 20
B. Prints the number 20 on the screen when the page loads
C. Deletes price
D. Sends 20 to the printer

2. What is a function?

A. A type of database
B. A reusable recipe that takes inputs and can return an output
C. A CSS colour
D. An error

3. In `if (age >= 18) { a() } else { b() }`, what happens when age is 15?

A. a() runs
B. Both run
C. b() runs
D. Neither runs

4. In React, what are props?

A. Rows loaded from the database when the page starts
B. The CSS files that style a component
C. Server logs
D. Inputs passed into a component

5. What does `await` do in `const notes = await getNotes()`?

A. Waits for getNotes to finish before continuing
B. Deletes the notes
C. Runs getNotes twice
D. Makes getNotes run in the background so the code doesn't wait

Official sites

MDN: JavaScript first steps:
https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting
React: Quick start: https://react.dev/learn

Last reviewed 2026-11-25. AI tools change fast; check each tool's own site for current features, plans and prices.

ANSWER KEY

Read code like a builder (with AI as your tutor)

1. A: Creates a named value, price, set to 20
const creates a variable that can't be reassigned; let creates one that can.

2. B: A reusable recipe that takes inputs and can return an output
Functions package steps you can run again with different inputs.

3. C: b() runs
15 is not >= 18, so the else branch runs.

4. D: Inputs passed into a component
Props are a component's inputs; state is what it remembers and updates.

5. A: Waits for getNotes to finish before continuing
await pauses until a slow task (like a database call) finishes.