< html >

My First Website

{ }

WordPress, no-code, AI builders,
HTML/CSS and Shopify
— how to build your first website in 2026

MzdrowY

2026

< /html >

My First Website

My First Website

WordPress, no-code, AI builders, HTML/CSS and Shopify — how to build your first website in 2026

MzdrowY

2026

My First Website

WordPress, no-code, AI builders, HTML/CSS and Shopify — how to build your first website in 2026

Author: MzdrowY

Table of Contents

Introduction

Do you remember the moment when you first thought: “I want my own website”? Maybe it was when you started your own business. Maybe you wanted a place for your portfolio, a blog, or a simple shop for your handmade products. Or maybe you just got tired of telling friends “the website is still under construction” every time they asked about your business.

Then you started looking for information. And you hit a wall of conflicting advice. One article says you need to learn programming. Another says WordPress is outdated. A third recommends a platform you’ve never heard of, and a fourth tells you to pay a specialist thousands of dollars before you do anything on your own.

This book exists to cut through that chaos.

You won’t find a claim here that there’s one best way to build your first website. Because there isn’t. There are several good ways, and the right choice depends on your specific situation — how much time you have, whether you plan to sell products, whether you want full control, or whether you just want it done. Instead of pushing you toward one solution, this book will help you figure out which one fits you, and walk you through it step by step.

I’m writing this in mid-2026, and that matters. The internet I describe on these pages is different from the one in tutorials from five or ten years ago — not because old rules stopped working, but because new, real options have emerged. WordPress still powers nearly half of all websites in the world, but for the first time in its history, it’s steadily losing ground to simpler, faster solutions. At the same time, an entirely new category of tools has grown — builders where you simply describe in words what your website should look like, and artificial intelligence generates it in minutes. Skipping this topic in a guide written today would be unfair to you as a reader.

Who is this book for? Anyone who has never built a website before and doesn't want to sign up for a six-month programming course to do it. You don't need to know anything about computer science to use it — every technical term that appears here is explained the moment it's needed, and at the end of the book you'll find a glossary you can return to.

What will you learn? We'll go through all the real paths to building a website in 2026 — WordPress, writing your own code in HTML and CSS, no-code and AI builders, and Shopify for online stores. I give each of them as much attention as it truly deserves, based on real market data, not guesses. You'll also learn how to choose a domain and hosting — a common element for most of these paths — and how to tell whether you chose well before you invest weeks of work into it.

One thing I ask at the start: you don't have to read this book cover to cover before you start building. Chapter 1 will help you quickly figure out which path is yours — and from there you can jump straight to the matching chapter. The best website is the one that finally gets published, not the one whose construction you plan forever.

Let's begin.

Chapter 1. Market Overview — Status vs. Trend

1.1 How Websites Are Built Today — Four Paths

Imagine you want a house. You have a few options. You can buy a ready-made blueprint and hire a construction company that knows what it's doing — you get a solid, proven home, but within the catalog they offer. You can buy a lot and build it brick by brick with your own hands — maximum control, but also maximum effort. You can order a modular home that arrives in sections and gets assembled in a few days. Or — the newest option — you can simply describe what your dream house should look like, and someone (or something) builds it for you.

Building a website works the same way. In 2026, anyone who wants to launch their first site chooses between four paths.

WordPress. By far the most popular choice in the world — according to W3Techs, an independent firm that tracks web technologies, WordPress powered nearly 42% of all websites worldwide as of June 2026. It's a ready-made system where you add a theme (the look) and plugins (the features), a bit like flat-pack furniture — you don't have to design the screws, but you still put it all together yourself.

The 42% figure comes from W3Techs, which surveys roughly 10 million of the most popular websites and counts each domain (e.g., myblog.com) as one result, including subdomains under the main entry. There's also a second source — HTTP Archive — which draws on Chrome User Experience Report (CrUX) data and counts every subdomain separately. According to that source (GravityKit data, April 2026), WordPress holds about 33% of the market. The roughly 9-percentage-point gap comes from methodology, not error — both sources are credible, they just measure different things.

HTML and CSS from scratch. This is building brick by brick. You write the code yourself that tells the browser what to display and how. It gives you full control and is a great way to understand how the web actually works — but it takes the most time and study before you see results.

No-code and AI builders. This covers classic drag-and-drop tools (Wix, Squarespace, Webflow) and their younger, fast-growing cousins — builders where you simply describe your site in plain language and artificial intelligence generates it for you (e.g., Lovable or Wix AI). This is the fastest path from idea to a live website.

Shopify (and similar platforms). If your goal isn't just a website but an online store, this is a separate category with its own rules — a ready-made e-commerce platform where you pay a monthly subscription plus a sales commission, and in return you get a shopping cart, payment processing, and inventory management working from day one.

There's one element that applies to every single one of these paths without exception: a domain (your site's address) and hosting (the place where your site "lives" on the internet). That's the topic of Chapter 6 — no matter which path you choose, at some point you'll need to buy the lot and set the address.

1.2 Market Status vs. Trend: Why "Popular" Doesn't Mean "Worth Choosing Today"

Here I want to save you from a trap that's easy to fall into when reading internet statistics.

When you see the number "WordPress powers 42% of all websites," you're looking at a snapshot of the entire existing web — including millions of sites built ten or fifteen years ago that are still running simply because nobody had a reason to change them. That's largely a result of inertia, not active choice. If someone who built a site in 2012 were starting from scratch today, there's no guarantee they'd pick the same solution.

And you're starting from scratch right now. That's why a more important question than "what's most popular" is "what are people choosing when they start today, just like you."

And here the picture gets more interesting. WordPress, despite still holding a massive share, is in 2026 losing market share systematically for the first time in 20 years — down from over 43% as recently as 2025. Meanwhile, Wix is growing at over 30% year over year, and in the e-commerce space, more than half of all new online stores launched in 2025 chose ready-made SaaS platforms (Search Engine Journal, 2025 data) rather than a self-hosted CMS.

The youngest but fastest-growing category is AI builders — tools where you describe your site in words. The market for these tools is already worth over $3 billion in 2026, and — this is especially relevant to you — on the Hostinger Horizons platform, as many as 93–95% of users are people building their very first website. That's exactly your situation.

The takeaway isn't that WordPress is a "bad" choice — for many situations it's still the best option, and I've devoted the most pages in this book to it for a reason. The takeaway is that your choice should come from your situation, not from what the majority of the internet happened to adopt over the past two decades. Every chapter in this book is designed to help you match the path to you — not the other way around.

1.3 Which Method Is Right for You? A Quick Quiz

Before you go further, answer four questions honestly. There are no wrong answers — each one leads you to a different chapter.

1. What do you want to achieve?

If your main goal is selling products → go to Chapter 5 (Shopify).

If you want a business card site, blog, portfolio, or company page → keep reading.

2. How much time do you want to spend learning before your site is ready?

As little as possible, ideally today → Chapter 4 (no-code and AI builders).

I can spend a few weekends learning in exchange for more control → Chapter 2 (WordPress).

I want to truly understand how it all works, even if it takes longer → Chapter 3 (HTML/CSS).

3. Do you care about full control and independence (your own hosting, no monthly platform fees), or do you prefer simplicity and not worrying about technical details?

Control and independence → WordPress or HTML/CSS.

Simplicity, even with a monthly subscription → no-code or AI builders.

4. Should your website grow and evolve over the years, or do you need something "right now"?

For the long haul, with growth in mind → WordPress.

Right now, with the option to change later → no-code/AI builders.

If you're still not sure after this quiz — that's a good sign, not a problem. Chapter 7 contains a detailed decision tree and a comparison table of all paths that you can return to after reading the individual chapters.

1.4 How to Use This Book

You don’t need to read this book cover to cover. Chapter 1 and Chapter 7 (the one you’re reading now and the summary at the end) are always worth reading — they give you the map and help you check whether you chose the right path. Between them, pick the chapter that matches your path from the quiz above and focus on it. You can browse the remaining chapters later, when your needs change — or when you’re simply curious how things look from the other side.

One thing I ask: don’t treat the numbers and statistics in this book as a final verdict. The internet changes fast — what’s true in mid-2026 may look different in two years. The numbers are here to help you understand the general direction, not to replace your own judgment of the situation.

Let’s keep going.

Chapter 2. WordPress

2.1 What is WordPress and who is it for

WordPress is a content management system — in the industry, they call it a CMS (Content Management System). That means you don't write your site from scratch in code. Instead, you log into a dashboard, click "add new page," type your text, insert images, and hit publish. All the technical wiring — how the page gets stored, displayed, and served — runs in the background, and you focus on what your visitors actually see.

It started as a simple blogging engine back in 2003. Two decades later, it powers nearly 42% of all websites in the world — from small local business pages and personal blogs to major media outlets and corporate sites.

Who is it a good fit for? Anyone who wants real control over their site and its growth, doesn't want to pay a monthly platform fee (though you still need hosting), and is willing to spend some time learning the dashboard. If you're planning a site that will grow with you over the years — new subpages, a blog, maybe an online store down the road — WordPress is one of the safest choices, precisely because so many people already use it. Whatever problem you run into, someone has already solved it and written about it online.

2.2 WordPress.com vs WordPress.org — the key difference

This is the first trap nearly every beginner falls into, because the names are deceptively similar and they're actually two very different things.

WordPress.com is a hosted service — think of it like renting a furnished apartment. You sign up, pay (or use a limited free plan), and get a turnkey site running on Automattic's servers. Convenient, but restricted: on lower-tier plans you can't install your own plugins or themes, and your site will always be partly subject to the platform's rules.

WordPress.org is software — a free program you download and install on your own hosting (more on hosting in Chapter 6). It's like buying a house: you're responsible for the plumbing and repairs, but you have full control — you can install any plugin, any theme, and change literally anything.

In this book, when we say "WordPress," we mean the .org version — that's the one that gives you real flexibility, and that's what the rest of this chapter covers. If all you care about is simplicity and you don't plan to expand your site in the future, WordPress.com might be enough — but then you might also want to look at the tools in Chapter 4.

2.3 Installation — step by step

Good news: in 2026, practically nobody installs WordPress "by hand" anymore by uploading files to a server. Most hosting companies offer a one-click installer. Here's the typical path:

As of May 2026, WordPress 7.0 is available. Most hosting installers default to this version — so you get native AI integration right out of the box (WP AI Client built into core), an improved editor, and better performance. The AI thread is one of the running themes of this book — with WordPress 7.0, you have it from day one, no extra plugins needed.

- Buy hosting and a domain (Chapter 6 explains how to choose). Most WordPress-focused hosting companies include it in their plan price.
- In your hosting dashboard, look for "Install WordPress" — usually a clearly labeled button or icon, sometimes called "Softaculous" or "Auto Installer."
- Fill out a short form: site name, admin username, password (use a password manager and pick something truly strong — this is your first line of defense against break-ins), and email address.
- Wait 2–5 minutes while the system installs the files and database.
- You'll get your admin panel URL — usually yourdomain.com/wp-admin. Save it — you'll use it every time you visit.

That's it. If your host doesn't have an installer like this (rare in 2026, but it happens), consider switching hosts — manual installation is possible, but for your first site it unnecessarily complicates things.

2.4 First settings after installation

Right after your first login, take care of a few things before you start working on the look of your site:

- General settings (Settings → General): site title, tagline, timezone, and dashboard language.
- Login security: change the username from the default "admin" to something unique if the installer didn't do it automatically — it's the most attacked account name in the world.
- Settings → Permalinks: change the URL structure to "Post name" — the default structure with dates and query strings is unreadable and bad for SEO.
- Remove sample content: WordPress ships with a sample "Hello World!" post and a sample page — delete them so they don't accidentally get published.
- Check your WordPress version and enable automatic updates for minor security patches (Settings → General, update section) — this is one of the most important things protecting you from break-ins.

Those five minutes at the start will save you hours of trouble later.

2.5 Choosing a theme — what to look for in 2026

A theme is the visual design of your site — colors, layout, typography, sometimes ready-made page templates. In 2026, block themes clearly dominate, designed to work with the Gutenberg editor, which we'll get to in a moment.

What to pay attention to when choosing:

- Lightweight and fast. A trendy, "heavy" theme packed with dozens of built-in effects can slow your site down, which hurts both users and SEO. Proven, lightweight themes (like GeneratePress, Kadence, or Astra) are the professional standard in 2026 for exactly this reason.
- Built for the block editor, not the older template system — this ensures easier editing without code.

- Responsiveness — the theme must look great on a phone. Check it in the preview before installing, ideally on an actual phone, not just the dashboard preview.
- Support and updates — a theme with a changelog that hasn't been updated in a long time is a risk (security bugs, incompatibility with new WordPress versions).

A free theme from the official WordPress.org directory is enough for a first site in most cases. Premium themes (usually $50–$100 one-time) give you more ready-made templates and technical support — worth considering if the free options don't satisfy you visually.

2.6 The Gutenberg editor — basics

Gutenberg is the default content editor in WordPress, where you build pages and posts from blocks — every paragraph, image, button, or heading is a separate block that you can move, edit, and style independently.

The basic workflow:

- Click "+" to add a new block — you can choose from paragraphs, headings, images, lists, buttons, columns, and more.
- Each block has its own settings panel on the right — that's where you change colors, spacing, and alignment.
- Group and Column blocks let you build more complex, responsive layouts — for example, three service columns side by side that automatically stack on a phone.
- The page editor works the same way as the blog post editor — it's the same logic in both places.

The most common beginner mistake: going overboard with the number of different blocks and colors on a single page. Keep it simple — one or two fonts, a limited color palette, plenty of white space. It not only looks better, it loads faster too.

Spend an hour just playing around with the editor on a test page before you start building the real thing — you'll quickly get a feel for how it works.

2.7 Pages vs posts — when to use which

This distinction confuses many beginners, but it's simple once you get it.

Pages are static content that rarely changes: "About," "Contact," "Services," the homepage. They have a fixed spot in your menu and aren't arranged chronologically.

Posts are dynamic, time-based content: blog articles, news, announcements. By default they display from newest to oldest and can be organized with categories and tags.

Simple rule: if the content answers "what is this / who are we" — it's a page. If it answers "what happened / what's new" — it's a post. A business site without a blog can get by with pages alone. If you plan to publish regularly and build visibility in Google through new articles, posts (and a blog) are your most important tool.

2.8 Menus, widgets, footer — building navigation

Content alone isn't enough — your visitors need a way to move between pages.

Menu (Appearance → Menus) is the main navigation, usually visible at the top of the site. You build it by adding selected pages in your chosen order. In block themes, you edit it directly in the full site editor (Appearance → Editor).

Widgets are smaller functional blocks you can place in specific areas of your theme — most often in a sidebar or footer. Examples: a list of recent posts, a contact form, social media icons.

Footer is the bar at the bottom of every page — it usually contains contact info, links to the privacy policy and terms of service (legally required in the EU), and sometimes a sitemap. In block themes, you edit it just like any other section, in the full site editor.

Good navigation shares one universal trait: a user should reach any important piece of information in three clicks. If your main menu has more than 6–7 items, you should probably group them.

2.9 Essential plugins

Plugins are add-ons that extend WordPress's functionality — they're what give WordPress its reputation as a "does everything" platform. The official directory has tens of thousands, but for a first site you really only need a few categories:

- Contact form — e.g., WPForms or Contact Form 7. Without one, visitors can't reach you except through the contact info in your footer.
- SEO — e.g., Yoast SEO or Rank Math. They walk you through optimizing each page for Google (more in section 2.10).
- Security — e.g., Wordfence. Blocks basic break-in attempts and scans your site for malicious code.

- Backups — e.g., UpdraftPlus. Automatically saves a copy of your site so you can restore it if something goes wrong.
- Caching / performance — e.g., WP Rocket (paid) or WP Super Cache (free). Speeds up page loading, which matters for both users and SEO.

The golden rule of plugins: install only what you actually need, and regularly remove ones you don't use. Every extra plugin is a potential security risk and added load on your site — ten plugins installed "just in case" do more harm than good.

2.10 SEO basics in WordPress

SEO (Search Engine Optimization) is a set of practices that help your site rank higher in Google search results. It's a topic for a separate book, but a few fundamentals are worth knowing from day one:

- Every page and post should have a unique title and meta description — that's exactly what the SEO plugin from the previous section manages, walking you through title and description fields below each piece of content you edit.
- Headings have a hierarchy — one main heading (H1) per page, H2 headings below it, H3 below those. The block editor makes this easy to set for each heading block.
- Page speed matters — that's why you need the caching plugin from the previous section and a lightweight theme from 2.5.
- Write for people, not for algorithms — in 2026, Google increasingly rewards genuine value and expertise in your text, not artificial keyword stuffing.
- Every page accessible from the menu — "orphan" pages with no links pointing to them are harder for search engines to find.

These five principles are enough to start. The rest — link building, keyword research, content marketing — is work for the months ahead, not the first day.

2.11 Security and backups

WordPress, precisely because it's so popular, is also the most frequently targeted by bots scanning the internet for weak passwords and outdated versions. That's not a reason to panic — it's a reason to build a few simple habits:

- Keep everything updated — WordPress, your theme, and all plugins, ideally with automatic updates for minor patches. Most break-ins exploit known, already-patched bugs in outdated software.
- Strong, unique password and, if your host supports it, two-factor authentication for dashboard login.
- Regular offsite backups (e.g., automatic saves to Google Drive or Dropbox via a backup plugin) — if something goes wrong, you can restore your site in minutes instead of rebuilding it from scratch.
- Limit admin accounts — every account with access is a potential weak point.

Think of it like locking your front door: none of this guarantees 100% security, but it eliminates 95% of potential problems with minimal effort.

2.12 Most common beginner mistakes

Based on the experience of thousands of people building their first WordPress site, a few traps come up again and again:

- Installing too many plugins “just in case” — slows down the site and increases the risk of conflicts between plugins.
- Ignoring updates for months, until something breaks or the site gets hacked.
- No backups — until the day something goes wrong and there’s nothing to restore.
- Going overboard with visual effects — animations, sliders, bright colors on every section — that look “interesting” in the editor but actually distract visitors and slow down the site.
- Writing content for yourself, not for the visitor — a business site full of vague talk about “passion and professionalism” without any concrete information about what you actually do or how to reach you.
- Skipping the phone test before publishing — most visitors will see your site on a small screen first.

None of these mistakes are catastrophic if you catch and fix them early — it gets worse when they go unnoticed for months.

2.13 How much does it cost — a realistic budget

WordPress as software is free, but a WordPress site is never truly "free." Here's a rough budget for the first year:

- Hosting: from about $30–$50 per year for a basic shared plan, to several hundred dollars per month for dedicated WordPress hosting at higher traffic levels.
- Domain: $10–$20 per year for popular extensions (.com, .org).
- Theme: $0 (free from the directory) to $75–$100 one-time for a premium theme.
- Premium plugins (optional): individual paid plugins usually cost $25–$75 per year, but at the start most features you need are covered by free versions.

Realistically, a modest WordPress business site costs about $50–$150 in the first year if you build it yourself. That doesn't include hiring a specialist — which, depending on scope, typically starts at a few hundred dollars and can go into the thousands.

2.14 When WordPress is NOT the right choice

Despite all its strengths, WordPress isn't a universal answer to every need. Consider a different path if:

- You need a site "right now," within an hour, and don't plan to grow it — a no-code or AI builder from Chapter 4 will get you there faster with less effort.
- Your main goal is a large-scale online store — Shopify (Chapter 5) is built specifically for e-commerce and takes many technical decisions off your hands that you'd otherwise manage yourself in WordPress through the WooCommerce plugin.
- You want to deeply understand how the web works, not just have a functioning site — then start with HTML and CSS from Chapter 3, even if you eventually switch to WordPress later.
- You absolutely don't want to deal with updates, security, and backups yourself — managed hosting exists that handles this for you (at a higher price), but it still demands more attention than fully hosted platforms from Chapter 4.

WordPress is like a well-equipped workshop — it gives you the tools to build almost anything, but you still need to learn how to use them. For many people, that's an investment worth making. For others, it's unnecessary overhead when a simpler solution would do just fine.

Chapter 3. HTML/CSS — How a Website Works

3.1 How a Browser Reads a Page — Mental Model

Before you write your first line of code, it helps to understand what actually happens between typing a URL into your browser and seeing a page appear on screen. It's simpler than you might think.

When you enter a web address, your browser sends a request to the server where that page "lives": give me your files. The server sends back usually three types of files:

HTML — the skeleton and content of the page: what elements are on the page and in what order (heading, paragraph, image, button).

CSS — the appearance: colors, fonts, spacing, how elements are positioned relative to each other.

JavaScript — behavior and interactivity: what happens when you click a button or scroll the page.

The browser reads HTML from top to bottom and builds something like a family tree of elements — in the industry, it's called the DOM. Then it "paints" that skeleton according to the rules from CSS. Finally, if there's any JavaScript, the browser runs it, adding interactivity.

In this chapter, we'll focus on the first two layers — HTML and CSS — because those are enough to build a complete, working static page. A "static" page simply means: the same files are sent to every visitor, with no database and no admin panel login like in WordPress. It's the simplest possible form of a website — and a good starting point for understanding everything that happens "under the hood" in the remaining chapters of this book.

3.2 HTML — Tags and Document Structure

HTML (HyperText Markup Language) describes content using tags — words in angle brackets that tell the browser what a given piece of content is. Most tags come in pairs: an opening tag and a closing tag, with content between them.

```
<p>This is a paragraph of text.</p>
```

Every HTML page has a fixed skeleton:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Page title in the browser tab</title>
</head>
<body>
  <h1>This is the main heading</h1>
  <p>This is a paragraph of text below the heading.</p>
</body>
</html>
```

Several parts of this skeleton always look the same: <!DOCTYPE html> tells the browser it's dealing with modern HTML. The <head> section contains information "for the browser and search engines" — not visible on the page itself. The <body> section holds everything the visitor actually sees.

The most commonly used tags that are enough for a good start:

Tag	What it does
<h1> to <h6>	headings, from most important to least important
<p>	a paragraph of text
<a href="...">	a link to another page
<img src="..." alt="...">	an image
and	a bulleted list
	a universal container — a "box" for grouping elements

In 2026, semantic HTML is the standard — using tags that describe the meaning of a page section on their own, instead of generic s everywhere: <header> for the top bar, <nav> for the menu, <main> for the primary content, <section> for a thematic section, <footer> for the footer. This makes the page better understood by search engines and screen readers for visually impaired users — and as the author, your code is simply easier to wrap your head around.

3.3 CSS — Styling Elements

HTML alone gives you a page that works, but looks like a text document from the 1990s — black text on a white background, with no styling whatsoever. CSS (Cascading Style Sheets) is the language that tells the browser how that skeleton should look.

Basic CSS syntax looks like this:

```
selector {
  property: value;
}
```

For example:

```
h1 {
  color: navy;
  font-size: 32px;
}
```

This tells the browser: "make every <h1> heading navy blue and set the font size to 32 pixels."

To style individual elements rather than all elements of a given type, you use classes — labels that you assign yourself in HTML:

```
<p class="warning">Attention, this is important information.</p>
```

... and then in CSS:

```
.warning {
  color: red;
  font-weight: bold;
}
```

Notice the dot before the class name in CSS — that's what says "find the element with this class," as opposed to targeting the tag name itself.

You can attach CSS to your page in three ways: directly in a tag (style="..."), in the <head> section inside a <style> block, or — the best and most common practice — in a separate .css file linked to the HTML with a single line:

```
<link rel="stylesheet" href="style.css">
```

A separate CSS file means a single color change applies to every subpage of your site at once, instead of having to fix each one individually — that's one of the biggest advantages of separating content (HTML) from appearance (CSS).

3.4 The Box Model and Flexbox — Layout Basics

Every element on a page — a paragraph, an image, a button — the browser treats as a rectangular "box." Understanding what that box is made of is one of those moments where CSS suddenly starts making sense.

Every box has four layers, from the inside out:

- Content — the actual content (text, image).
- Padding — the inner spacing between the content and the edge of the box.
- Border — the outline, if you set one.
- Margin — the outer spacing, separating this box from neighboring elements.

```
.card {
  padding: 16px;
  border: 1px solid #ddd;
  margin: 24px;
}
```

This explains why two elements sometimes "touch" each other and sometimes have a nice gap between them — it all comes down to padding and margins.

The next important topic is flexbox — a tool for arranging multiple boxes relative to each other, for example in a row side by side or evenly distributed in a column. You set it on the parent element that contains several child elements:

```
.menu {
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

This single block of code says: "lay out the child elements in a single row (display: flex), spread them evenly across the full width (justify-content), and center them vertically relative to each other (align-items)." This is exactly the type of code behind most navigation bars at the top of websites.

Flexbox arranges elements in a single row or a single column — it works in one direction. CSS Grid is the complement: it handles two dimensions at once, meaning rows and columns simultaneously, like a spreadsheet table. Which one to use? Flexbox when elements should line up in a line. Grid when you're creating a checkerboard — a grid of rows and columns. When starting out with CSS, flexbox is enough for most situations, but it's worth knowing that Grid is ready and waiting when you run into a two-dimensional layout.

You don't need to memorize these properties — most people who write CSS look them up in documentation or cheat sheets until they become second nature. What matters more is understanding the logic itself: boxes inside boxes, with spacing and layout that you control through CSS.

3.5 Responsiveness — The Page on a Phone

Most visitors will see your page on a phone first, not on a desktop. A page that doesn't adapt to a small screen is one of the fastest ways to lose a visitor today.

The first essential step is a single line in the <head> section:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Without this, mobile browsers display the page "zoomed out" by default, as if it were a full desktop view in miniature — exactly that annoying effect that forces you to pinch-zoom to read the text.

The second piece is media queries — CSS rules that only apply at a certain screen width:

```
.container {
  display: flex;
}
@media (max-width: 600px) {
  .container {
    flex-direction: column;
  }
}
```

This example says: "normally arrange elements in a row, but if the screen is narrower than 600 pixels, stack them one below the other." Thanks to this, three columns side by side on a desktop become a readable single column on a phone.

Best practice in 2026 is mobile-first thinking — designing and writing styles for the narrow screen first, then adding rules that expand the layout for larger screens, not the other way around. Regardless of your approach: always test your page on a real phone before publishing, not just in a desktop preview.

3.6 Browser Developer Tools

Every modern browser has a built-in set of tools for inspecting and debugging page code — and you don't need to install anything to use them.

You open them with the F12 key (or right-click on a page element → "Inspect"). Two tabs will be most useful to you at the start:

- Elements — shows the live HTML code of the page and, when you click on an element, exactly which CSS rules apply to it. You can temporarily change a value (like a color or spacing) right in the browser and see the effect without saving the file — great for quick experiments.

- Console — shows errors in the code, for example when a CSS file failed to load or there's a typo in a class name.

A practical skill worth developing right away: when something on the page "doesn't look right," instead of guessing, right-click that element, choose "Inspect," and check what styles are actually applied to it. Very often it turns out that some other CSS rule is "overriding" the one you're trying to apply — and the dev tools show this clearly, instead of leaving you to guess.

3.7 Project: Your First Static Page from Scratch

Time to put it all together into one complete project — a simple, single-page portfolio site with a header, an "about me" section, a list of services, and a footer with contact details.

HTML structure (file index.html):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Jane Doe — Photographer</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header class="header">
    <h1>Jane Doe</h1>
    <nav>
      <a href="#about">About</a>
      <a href="#services">Services</a>
      <a href="#contact">Contact</a>
    </nav>
  </header>


  <main>
    <section id="about" class="section">
      <h2>About Me</h2>
      <p>I've been photographing for over ten years, specializing in wedding and
portrait sessions.</p>
    </section>
    <section id="services" class="section">
      <h2>Services</h2>
      <ul>
        <li>Wedding sessions</li>
        <li>Individual portraits</li>
        <li>Family sessions</li>
```

```
      </ul>
    </section>
  </main>


  <footer id="contact" class="footer">
    <p>Contact: jane@example.com | (555) 000-0000</p>
  </footer>
</body>
</html>
```

Styles (file style.css):

```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  color: #222;
}


.header {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 16px 24px;
  background-color: #2c3e50;
  color: white;
}


.header nav a {
  color: white;
  margin-left: 16px;
  text-decoration: none;
}


.section {
  padding: 40px 24px;
  max-width: 700px;
  margin: 0 auto;
}


.footer {
  background-color: #f4f4f4;
  text-align: center;
  padding: 16px;
}
```

```
@media (max-width: 600px) {
  .header {
    flex-direction: column;
    text-align: center;
  }
  .header nav a {
    margin: 4px 8px;
  }
}
```

Save both files in the same folder and double-click index.html — that's all it takes to see the result in your browser, with no server or installation needed. Try changing colors, adding your own photo (the <img> tag), or writing another section — that's the best way to learn: break real code and fix it.

When you're happy with the result, you can publish a page like this completely free through GitHub Pages — we cover that in detail in Chapter 6.

3.8 Why This Matters Even If You'll Use WordPress or Wix

If after reading this chapter you still plan to build your site on WordPress or in a no-code builder — it wasn't wasted time, and it's no coincidence that this chapter is in this book regardless of your final decision.

Every WordPress theme, every Wix template, every page generated by an AI builder — in the final version it's still HTML and CSS, just generated for you. By understanding these basics:

- You can fix small problems more easily — a misaligned element, a wrong color, text that won't fit — without waiting for a specialist's help.
- You communicate better with people you hire to work on your site — you understand what they're talking about when they mention "padding" or "media query."
- You can confidently use the "custom CSS" option that nearly all builders from Chapter 4 offer, for small tweaks that go beyond the standard settings.
- You understand why some things are harder than they seem — for example, why perfectly aligning three elements in a row at every screen size isn't a matter of "one click," but a tradeoff between different widths.

In other words: even if you end up building your site without writing a single line of code, this chapter gives you something no builder can — an understanding of what's actually happening beneath the surface of your page.

Chapter 4. No-Code and AI-Native Builders

4.1 The Lay of the Land: Drag-and-Drop vs. AI-Native

This chapter covers two generations of tools that share one thing: neither of them requires you to write a single line of code.

Classic drag-and-drop builders — Wix, Squarespace, Webflow — work like a visual editor: you pick an element from a panel (image, text, button), drag it onto the page, and style it by clicking through options instead of writing CSS. You're still the one placing every element by hand.

AI-native builders — Lovable, Wix AI, Framer AI, and similar tools — are a newer category where you describe what you want in plain language, and the tool generates the site for you. No dragging elements around — you have a conversation instead.

The line between these two groups blurs every month — classic platforms are adding their own AI features, and AI-native tools are giving you more visual control after the initial generation. In this chapter, we'll walk through the key players from both generations so you can pick the one that fits your working style.

4.2 Wix — Who It's For and How to Start

Wix is the most popular classic builder in the world and one of the fastest-growing in terms of market share for new websites. Its strength is a massive template library and an editor that makes it practically impossible to "break" the entire site with a single click.

Who it's for: people who want a fully working, good-looking site up and running in a single afternoon, with zero prior learning, and who don't mind a monthly subscription in exchange for that convenience.

How to start:

- Sign up at wix.com and choose whether to start from a template or use the built-in AI (Wix asks you a few questions about your business and goals, then suggests a ready-made site draft).
- Edit the design — click on any element to change its text, image, color, or spacing.
- Add custom subpages from the top "Pages" menu if you need more than one.
- Publish with a single button. On the free plan, your site runs on a Wix subdomain and shows Wix ads — to connect your own domain and remove the ads, you need a paid plan.

Wix works well for business cards, portfolios, and smaller shops. It's a weaker fit if you're planning a highly unusual, custom layout or a future migration to another platform, because exporting out of Wix is severely limited.

4.3 Squarespace — Who It's For and How to Start

Squarespace has a smaller template library than Wix, but every template is designed with strong attention to aesthetics — it's especially popular among photographers, designers, artists, and small boutiques, where that "instantly professional" visual impression matters a lot.

Who it's for: people who want a more polished, refined default look than the average free Wix template offers, and who are willing to trade some editing flexibility for that result.

How to start:

- Pick a template matched to your industry — the library is smaller, but well-organized.
- Edit the page section by section — Squarespace walks you through pre-built content "blocks" (image with text, gallery, form) that are easy to rearrange but harder to completely redesign.
- Configure basic SEO settings and social media connections in the settings panel.
- Publish and connect your domain — Squarespace doesn't have a free ad-supported plan like Wix. Every plan is paid from the start, but there's no forced branding on your site.

Squarespace works well as a portfolio business card or a small shop with polished design. Less flexible than Wix if you want a highly unconventional page layout.

4.4 Webflow — When It's Worth Going Deeper

Webflow is a different beast from Wix and Squarespace, even though technically it's also a "no-code builder." It's designed for people who want control at a level close to writing their own CSS — just visually, by clicking rather than typing.

If you read Chapter 3, you'll feel right at home here: Webflow operates on the exact same concepts — box model, flexbox, media queries — just through a settings panel instead of a code editor. That means a much steeper learning curve than Wix, but also a much higher ceiling of what's possible. Webflow is the standard in agencies building more demanding, animated marketing sites.

Who it's for: people who've worked through the basics in Chapter 3 and want a visual interface instead of writing code from scratch, or those planning a more complex, unique site and willing to invest more time in learning the tool.

Webflow is more expensive than Wix and Squarespace on comparable plans, but it also lets you export clean HTML/CSS code on certain plans — which significantly reduces the risk of getting "locked in" compared to Wix.

If this sounds like "too much" for your first website — you're probably right. Webflow is a great choice for your second or third site, not your first project.

4.5 AI Builders — You Describe It, AI Builds It

This is the youngest and fastest-growing category in this book — and, as we mentioned in Chapter 1, aimed squarely at people building their very first website. It includes tools like Lovable, Wix AI (the chat-based version of Wix, separate from its classic editor), and Framer AI.

The principle is similar across all of them: you write in plain language what you need — for example, "a website for my yoga studio, with a class schedule, a sign-up form, and a section about the instructors, in a calm, natural color palette" — and the tool generates a complete site draft within seconds or minutes.

Typical workflow:

- Describe the site, as specifically as you can — the more detail you provide (industry, sections, tone, sample content), the better the first draft.

- Review the generated version and refine it with follow-up chat messages: “change the primary color to green,” “add a customer testimonials section,” “shorten the headline text.”
- After a few rounds of revisions, fine-tune the small details — sometimes in a simple visual editor built into the tool.
- Publish — depending on the platform, either on your own domain right away or by exporting the finished code.

Here’s an important technical difference to know before you pick a specific tool: some AI builders (like Lovable) generate real, exportable code (most often in React) that you can take and host anywhere. Others (like the chat version of Wix) create a site locked within their own platform, with no full export option — much like classic Wix.

One practical note: the quality of the first generated draft can be hit or miss — sometimes you get a solid design right away, sometimes you get something very generic that takes several rounds of fixes before it stops looking “like everything else.” That’s normal, and still faster than building from scratch — but don’t treat the first result as final.

4.6 Pros and Cons: Vendor Lock-In, Costs, Data Ownership

Before you make a decision, consider three things that are easy to overlook in the initial excitement of a quick result.

Vendor lock-in — the question of whether you can easily take your site elsewhere in the future. Classic builders (Wix, Squarespace) mostly don’t allow full export — your site “lives” inside their platform, and if you decide to switch providers, you’re effectively building a new site from scratch. Webflow and some AI builders offer real code export, which significantly reduces this risk.

Costs over time — unlike WordPress, where after paying for hosting and a domain you don’t have to pay for the software itself, builders in this chapter typically charge a fixed monthly or annual subscription that you keep paying as long as the site needs to stay online — and it usually goes up when you need more features (online store, higher traffic, removing the branding).

Data ownership — if the platform changes its pricing, cuts features, or (rarely but it happens) goes out of business, your site disappears with it unless you had an export option. Before choosing a specific tool, especially a newer AI builder, spend five minutes checking its documentation to see whether and how it exports data.

None of this is a reason to avoid this category of tools — for many people, the convenience and speed are more than worth the trade-offs. The point is to make it a conscious choice, not a surprise you discover a year from now.

4.7 Which Builder to Pick — A Checklist

A quick checklist to go through before you sign up for any of these tools:

- How much time do I have for a first result? Minutes → AI builder. Hours → Wix or Squarespace. Days or weeks with full control → Webflow.
- Do I need a unique, custom design, or is a solid ready-made template good enough? Ready-made template → Wix or Squarespace. Unique layout → Webflow or an AI builder with a good prompt.
- Do I have a clear vision of the site, or is it easier to describe it in words than to assemble it visually? Easier to describe → AI builder. Easier to see it and click things around → classic builder.
- Might I want to move the site to a different platform or my own hosting in the future? Yes, that matters → look for code export options (Webflow, some AI builders). No, I'll probably stay on the same platform for years → less important criterion.
- What monthly budget feels comfortable? Wix and Squarespace have plans starting at around $16–25 per month, Webflow is usually a bit higher, and AI builder costs vary widely between tools — check the current pricing before you commit, because this category changes its pricing models more often than anything else in this book.

If you've gone through this checklist and you're still torn between two options — pick the one that sounds like less work to get started. None of these decisions are irreversible, and a working site that's actually live is always better than a perfect plan that never shipped.

Chapter 5. Online Store — Shopify

5.1 Shopify vs. WooCommerce — How to Know Which One You Need

If the main goal of your website is selling products, you essentially have two paths — and it's worth choosing between them deliberately before you start building.

Shopify is a fully hosted platform built exclusively for e-commerce. You pay a monthly subscription, and in return you get a shopping cart, payment processing, inventory management, and shipping tools that work right out of the box — no configuring hosting or updating software.

WooCommerce is a free plugin that turns a regular WordPress site (from Chapter 2) into an online store. You only pay for hosting, but you manage updates, security, and all the technical pieces yourself.

Choose Shopify if: selling is the sole or primary purpose of your site, you want things running as fast as possible without dealing with hosting technicalities, you plan to sell across multiple channels at once (your own site, Instagram, Amazon), and you're willing to pay a higher fixed monthly cost in exchange for fewer responsibilities.

Choose WooCommerce if: you already have or plan to build a WordPress site with other content (a blog, a company section) and want to add a store to it, you care about lower long-term costs, and you don't mind managing updates on your own as described in Chapter 2.

5.2 Your First Store — Quick Start

Setting up a basic Shopify store usually takes anywhere from a few hours to a single working day, depending on how many products you have. The typical path:

- Sign up at shopify.com and choose a store name (you can change it later).
- Pick a theme — Shopify has its own library of free and paid themes tailored to different industries (fashion, food, electronics, handmade goods).
- Add products — for each product, you fill in the name, description, price, and photos. If the product has variants (size, color, etc.), you configure them in a separate section.
- Set up payments — the easiest route is Shopify Payments (built in), or you can connect an external payment provider if Shopify Payments isn't available in your country.
- Configure shipping methods and costs — you can set a flat rate, offer free shipping above a certain order total, or connect a carrier calculator.
- Connect your domain and publish the store.

The checkout process in Shopify is one of the platform's strongest features — it's well-tested and optimized to minimize the number of customers who abandon their cart at the last step. That's a real advantage over a self-configured checkout in WooCommerce.

Just like WordPress has plugins, Shopify has its own app marketplace (the Shopify App Store) — add-ons for email marketing, loyalty programs, product reviews, and dozens of other features that you install about as easily as WordPress plugins.

5.3 Your First Product — Adding a T-Shirt

I'm assuming you already have a Shopify account, either past the trial or on a paid plan. You log in to the admin panel at yourstore.myshopify.com/admin. On the left side you'll see the menu — click **Products**, then hit the **Add product** button.

The add product screen is split into several sections. At the very top, you enter the **title** — for example, "Graphic Tee — Northern Wolf." Below the title is the **description** field. Here you can insert plain text as well as formatting — bold, lists — and even images and embedded videos. The editor feels a lot like the one you'd use in a web email client.

The product description matters not just for shoppers, but also for search engines. Take a moment with it — write what material the shirt is made of, what the fit is like, whether the print is screen-printed or DTG, how to wash it. Shopify doesn't limit description length.

In the right column, you set the **price**. Shopify shows three fields: the compare-at price (for the crossed-out original price — the classic "was $29.99, now $24.99"), the selling price, and the cost per item. That last one is only used to display margins in your reports — customers never see it. You enter $24.99 as the selling price. The system accepts both $24.99 and 25 formats — don't overthink it.

Below the pricing section is **Variants**. Click "Add options like size or color." Shopify suggests a default structure: size and color. Leave both. In the size field, enter comma-separated values: S, M, L. In the color field: Black, Charcoal, Navy. Once you confirm, Shopify automatically generates nine variants — 3 sizes × 3 colors — each with its own price, SKU, and inventory count.

For each variant, you can set an individual price — useful when a larger size costs more — and assign a dedicated photo. A customer clicking "Navy, L" will see exactly that combination, not the main photo in a different color. It's a small detail that genuinely reduces return rates.

Move to the **Media** section. Drag the t-shirt photo from a folder on your computer. Shopify accepts JPEG, PNG, and WebP — no conversion needed. After uploading, you can crop the image and set it as the main photo. A good practice is to add several shots: front, back, a close-up of the print, the care label. You can drag photos in the gallery to reorder them — the first photo becomes the thumbnail visible in the store catalog.

The **Inventory** section. Check "Track quantity" and enter the available units for each variant. Shopify can automatically decrease stock after each order and send an alert when a product runs out. If you sell digital products or make everything to order — select "Don't track inventory," and the product will always show as available.

One last thing before saving: **Product visibility**. By default, the product is hidden and only accessible through a direct link. Change this to "Active" and choose your sales channels — online store, and if you've connected them, also Facebook, Instagram, Google. Click **Save**.

The whole operation — from clicking "Add product" to saving — takes about five minutes the first time. By your fifth product, you'll be under a minute. Shopify walks you through every step, doesn't hide options behind nested tabs, and what you see in the admin panel maps directly to how the product page looks in the store.

5.4 Shopify Plans — Which One Is for You

Shopify offers three subscription plans for independent sellers: **Basic**, **Standard** (often just labeled “Shopify” in the admin panel), and **Advanced**. There’s also a **Starter** option at a lower price that lets you sell through social media links without a full storefront — but for a proper online store, you need at least Basic.

The primary difference between plans is the **number of staff accounts**. Basic gives you 1–2 accounts — enough when you’re running the store solo. Standard allows up to 5, Advanced — up to 15. This matters when you hire a customer support person, a warehouse worker, or a marketing specialist and each of them needs their own login with appropriate permissions.

The second factor is **reporting**. Basic offers standard reports: sales, orders, products. Standard unlocks reports on customer behavior, segmentation, and marketing performance. Advanced gives you access to custom reports that you configure yourself — useful when you’re analyzing margins at the individual variant level or comparing sales channels against each other.

The third layer — and for many sellers the decisive one — is **transaction fees**. When using **Shopify Payments**, the rates decrease as you move to a higher plan: Basic has the highest rate, Standard — a mid-range rate, Advanced — the lowest. For a store processing several hundred orders a month, the difference in transaction fees between Basic and Advanced can easily exceed the difference in subscription price many times over. The more you sell, the more a higher plan pays for itself.

All three plans include the same core features: unlimited products, hosting, SSL certificate, shopping cart, and 24/7 technical support. Shopify doesn’t artificially cap the number of orders or site traffic based on your plan — the difference lies solely in advanced features and fee rates.

Practical strategy: start with **Basic**. It’s the lowest entry cost, and Shopify lets you change plans at any time without penalties or data loss. When your store reaches a volume where the savings on transaction fees cover the subscription difference — move up to Standard or Advanced. There’s no point paying for fifteen staff accounts when you’re the only person running the store.

5.5 Shopify Payments and External Providers

How you accept payments directly affects your monthly cost of running the store — and many beginners don’t discover this until their first billing statement.

Shopify Payments is Shopify's built-in payment system, available in the US since the platform's early days. When you use it, Shopify doesn't charge an additional platform transaction fee — you only pay the standard card processing fee. That fee goes to the card networks (Visa, Mastercard) and typically amounts to a small percentage of the transaction plus a minor fixed charge per transaction. You can set up Shopify Payments in a few clicks, without signing a separate agreement with a payment processor.

External payment providers — Stripe, PayPal, Square, Authorize.net — are the alternative. Setup requires creating an account with the provider, going through verification, and connecting it to Shopify through the admin panel. The financial difference is significant: Shopify adds its own transaction fee on top of every payment processed through an external provider. On top of that, you pay the provider's own fee. So you're paying twice — Shopify for facilitating the transaction on their platform, and the provider for actually processing the money.

In the US, here's how it plays out: Shopify Payments handles Visa, Mastercard, American Express, and Discover cards — covering the majority of online card transactions. But American customers increasingly expect digital wallet options like Apple Pay and Google Pay, as well as buy-now-pay-later services (Affirm, Klarna, Afterpay). Some of these work through Shopify Payments, but others — and certain niche payment methods — require an external provider.

Each of these providers has its own fee — usually between 1.5% and 3.5% of the transaction amount — to which Shopify adds its platform fee. That's why many US Shopify stores use a hybrid setup: Shopify Payments as the primary card channel, alongside PayPal or Square as an additional option. The customer at checkout chooses whether to pay with a card (lower cost for you) or through an alternative method (higher cost, but preferred by some shoppers). Shopify lets you activate multiple providers simultaneously.

A concrete example: on a $25 transaction through Shopify Payments, you pay roughly $0.75 in processing fees. The same transaction through an external provider costs about $0.75 to the provider plus an additional Shopify platform fee — depending on your plan, that could add $0.15 to $0.75 more. Across hundreds of orders per month, the difference adds up to a serious amount.

5.6 Hidden Costs — Apps

The first time you open the Shopify App Store, it's hard to shake the feeling that everything you could ever need is right at your fingertips. The catalog lists over 13,000 apps, from marketing

automation to wholesale integrations. The catch is that many of them run on a subscription model, and the bills pile up faster than you'd expect.

Email marketing is usually the first app you install beyond the basics. Shopify has its own simple email tool — good enough to start. When you need automation — an abandoned cart reminder email, a welcome sequence for new subscribers — you reach for an external app. **Mailchimp** offers a free plan up to a few hundred contacts. **Klaviyo** — a tool built specifically for e-commerce — also has a free tier, but with a larger customer list the monthly cost climbs to $20–30 or more.

Product reviews are the second category that quickly lands on the "must-have" list. Shopify retired its own free reviews app. If you want customers to rate products and add photos, you install **Loox** (has a free starter plan) or **Judge.me** (feature-rich, paid from a certain order volume). Reviews alone boost trust, but social media integration and customer photo uploads often cost extra.

More categories: **loyalty programs** (point systems, purchase rewards), **store translations** for other languages, **dropshipping tools** like Oberlo or Spocket (both with monthly subscriptions), **live chat** with customers, **advanced reporting** beyond what your subscription plan includes. The list of apps that "would definitely come in handy" grows every week you run the store.

Experienced Shopify sellers are blunt about it: apps can double your monthly platform cost before you even notice. The rule that will save you money at the start: only install what you literally cannot sell without. You have basic email marketing built in. You can collect reviews manually by asking customers for feedback in a post-purchase email. Shopify Payments covers most payment needs. Treat every app with a monthly subscription as a separate expense that has to pay for itself — if an app costs $15 a month, it needs to generate at least that much in additional revenue to be worth it.

5.7 Costs and Fees — What to Watch For

It's worth thinking about your Shopify budget in three layers, not just the subscription price shown on the homepage:

- Monthly subscription — Shopify has several pricing tiers that grow with the number of available features (reporting, staff accounts, lower transaction fees). Check the

current pricing at shopify.com before making a decision — it changes more often than most other things in this book.

- Transaction fees — if you use Shopify Payments, the fee is lower than with external payment providers, plus the standard card processing charge (typically a small percentage of the amount plus a minor fixed fee per transaction).
- Apps — many useful features (advanced email marketing, loyalty programs, custom reporting) are available as paid apps with their own subscriptions, added on top of your base plan cost.

For comparison: WooCommerce has no subscription fee for the software itself, but you still pay for hosting (Chapter 6) and a payment provider's fee (Stripe, PayPal, or a local processor) — though you typically don't face an additional "platform" fee the way you do with non-native payment methods on Shopify.

Practical advice for getting started: before committing to a full paid plan, use Shopify's free trial to set up your store and confirm the platform meets your needs before you start paying.

Chapter 6. Hosting and Domain

6.1 How to choose and buy a domain

A domain is your website's address — what visitors type into the browser, like yourcompany.com. No matter which path from the earlier chapters you chose, at some point you'll need to buy one.

A few practical rules for picking a name:

- Short and easy to remember — if you have to spell it out over the phone, a long name with tricky words becomes a problem fast.
- Avoid hyphens and numbers if you can — they're easy to forget or miscommunicate when speaking.
- Match the extension to your audience — .com is the universal standard if you're thinking beyond one country; a country-specific extension like .co.uk or .de signals a local business.
- Check availability across several extensions at once — domain registrars show you right away whether the same name is available as .com, .net, .org, and so on.

You buy domains from registrars — companies like Namecheap, GoDaddy, Google Domains, or Cloudflare, often bundled together with hosting (section 6.2). Here's the key thing to understand: you don't buy a domain forever. You rent it for a year (or several years upfront). If you forget to renew, someone else can register it once it expires. Turn on auto-renewal right when you buy — one less thing to remember.

Also check whether the registrar offers privacy protection for the domain owner's data (sometimes free, sometimes for an extra fee). Without it, the contact information you provide during registration may be publicly visible in the WHOIS database.

6.2 Types of hosting

Hosting is the "place" where your website files physically live — a server that responds to requests from visitors' browsers. For a first website, three types really matter:

Shared hosting — the cheapest option, where your site shares server resources with many other sites. Good enough for most business cards and smaller blogs at the start. This is where you should begin if you don't yet know how much traffic your site will get.

Managed WordPress hosting — a pricier version of shared hosting, optimized specifically for WordPress: automatic updates, built-in caching, and extra security already configured, no tinkering on your part. If you chose the path from Chapter 2 and would rather not worry about the technical details in sections 2.11–2.12, this is a solid investment.

VPS (Virtual Private Server) — a dedicated slice of a more powerful server, with greater control and resources, but requiring more technical knowledge to configure yourself. This is a later-stage option, for when your site has outgrown shared hosting — not a day-one choice.

For a first website from this book: shared hosting (or managed WordPress hosting if you chose Chapter 2) is more than enough in 95% of cases. Builders and AI tools from Chapter 4, and Shopify from Chapter 5, include hosting in their subscription price — this section mainly applies to the WordPress and HTML/CSS paths.

6.3 SSL, domain email, DNS basics

Three technical things worth knowing about, even if you don't dive into the details:

SSL (security certificate) encrypts the connection between a visitor's browser and your server — it's what changes your site's address from http:// to https:// and shows the padlock icon next to the URL. In 2026, this is an absolute standard. Without it, browsers warn visitors that your site is "not secure," and Google treats it as a negative ranking signal. The good news: nearly every hosting provider in 2026 installs a free SSL certificate (usually through Let's Encrypt) automatically, with no extra setup on your end.

Email on your own domain (like contact@yourcompany.com instead of yourcompany@gmail.com) looks far more professional and builds trust. Most hosting packages let

you set up a mailbox like this in the control panel within minutes, though storage space can be limited — if you have heavier email needs, consider a dedicated service (like Google Workspace) connected to the same domain.

DNS (Domain Name System) is the mechanism that connects your domain (the name) to the actual server where your site lives (the technical address). When you buy hosting separately from your domain, you need to "point" the domain to it in the domain panel — usually by entering the nameserver addresses your hosting company gave you. DNS changes aren't instant — they can take anywhere from a few minutes to 24–48 hours to propagate across the internet, so don't panic if the site doesn't work "right away" after changing the settings.

6.4 GitHub Pages and other free options — when they make sense

If you built your site following the project from section 3.7 — plain HTML and CSS, no database — you have access to a category of hosting that simply doesn't exist for the other paths in this book: free static site hosting.

GitHub Pages is a free service that publishes your site straight from a code repository on GitHub. In practice: you upload your index.html and style.css files to a repository, enable GitHub Pages in the settings, and your site is live at a free URL within minutes — with the option to connect your own domain later.

Similar free (to start) options include Netlify and Vercel — slightly more feature-rich, with automatic deployment that pushes changes as soon as you update your files. They're also popular for hosting code exported from some of the AI builders in Chapter 4.

When this makes sense: hobby projects, portfolios, simple personal sites, technical documentation — anything that doesn't need a database or user login. When it doesn't: WordPress (which requires PHP and a database, neither of which these services support) or an online store with a dynamic shopping cart. It's a real, fully professional option for the right use case — but it applies exclusively to the HTML/CSS path from Chapter 3, not the other roads in this book.

Chapter 7. What to Choose — Summary

7.1 Decision tree: from question to recommendation

If you read this book from the start, you probably already know which direction you are heading. This section is a shortcut for anyone who wants to quickly double-check their choice or is returning to the book after a longer break.

Question 1: Is the main purpose of the site selling products?

→ Yes: go to Question 1a.

→ No: go to Question 2.

Question 1a: Do you want a blog or extended business content alongside the store?

→ Yes, store and content together → WooCommerce on WordPress (Chapters 2 and 5).

→ No, store only, no extra content → Shopify (Chapter 5).

Question 2: How much time do you want to spend before the site is ready?

→ Minutes to hours, as little effort as possible → go to Question 2a.

→ I can spend a few weekends in exchange for more control → go to Question 3.

Question 2a: Is it easier for you to describe the site in words, or to lay it out visually by clicking?

→ Describe in words → AI builder (Chapter 4.5).

→ Lay out visually → Wix or Squarespace (Chapters 4.2–4.3).

Question 3: Do you want full independence (your own hosting, no recurring platform fee) and long-term growth for the site?

→ Yes → WordPress (Chapter 2).

→ Not entirely, but I want strong visual control without writing code → Webflow (Chapter 4.4).

→ I really want to understand how it works, regardless of the time → HTML/CSS from scratch (Chapter 3).

None of these answers is set in stone for the entire life of your site. Many people start with a no-code tool or an AI builder to get up and running fast, then after a year or two, when the site and their needs have outgrown that starting point, they move to WordPress for more control. That is a normal path, not a failure of the first choice.

7.2 Comparison table of all paths

	WordPress	**HTML/CSS**	**No-code (Wix/ Squarespace)**	**AI builders**	**Shopify**
Time to launch	Days	Weeks (learning)	Hours	Minutes–hours	Hours–a day
Upfront cost	Low (hosting + domain)	Very low	Low–medium	Low–medium	Medium
Long-term cost	Low, no platform fee	Very low	Recurring subscription	Recurring subscription	Recurring subscription + fees
Control & flexibility	Very high	Highest	Medium	Medium, tool-dependent	High within e-commerce
Learning curve	Medium	Steep	Very low	Very low	Low–medium
Vendor lock-in risk	Low	None	High	Varies (check export)	Medium
Best for	Sites growing over years	Learning & full control	A fast, polished business card	Fastest first result	Stores as the primary goal

The table does not replace the chapters it comes from — it is a cheat sheet to use after reading, not instead of it.

7.3 What to do after your first site

Regardless of which path you chose, a few pieces of advice apply equally to all of them.

A published site is better than a perfect plan. It is easy to fall into the trap of polishing every detail before launch. You are better off publishing a version you are 80% happy with and improving it as you go than waiting for 100%, which will never come.

Set up basic analytics from day one — for example, Google Analytics and Google Search Console (both free, work with every path described in this book). Within a few weeks they will show you real data: where visitors come from, which pages they read, where they drop off. That is a far better guide for your next changes than your own gut feelings.

Revisit other chapters when your needs change. A site you built on Wix as a simple business card might need a blog and better SEO a year from now — that is when Chapter 2 becomes relevant again. A store you started on WooCommerce might eventually outgrow your technical capacity — that is when Chapter 5 and Shopify are worth another look. None of these paths is a forever decision.

One last thing: this book describes the state of the market in mid-2026. Specific tools, prices, and platform names will change. What probably will not change are the questions in Section 7.1. If you come back to this book for your next site in a few years, those questions will still be a solid starting point, even if the answers on the right side need updating.

Good luck with your first website.

Glossary

AI builder — a tool where you describe in words how your site should look, and artificial intelligence generates it for you (e.g., Lovable, Wix AI, Framer AI). Chapter 4.5.

Backup — a saved copy of your entire site that you can restore if something goes wrong. Chapter 2.11.

Box model — the way a browser treats every page element as a rectangular "box" made up of content, padding, border, and margin. Chapter 3.4.

Cache — a mechanism that speeds up page loading by temporarily storing its elements so they don't have to be regenerated on every visit. Chapter 2.9.

Checkout — the process of completing an order in an online store, from the shopping cart to payment confirmation. Chapter 5.2.

CMS (Content Management System) — a system for managing your site's content through an admin panel, without writing code from scratch. WordPress is the most popular example. Chapter 2.1.

CSS (Cascading Style Sheets) — a language that describes the visual appearance of a site: colors, fonts, spacing, and element layout. Chapter 3.3.

Domain — your site's address on the internet, e.g., yourcompany.com. It's rented year by year, not bought once and for all. Chapter 6.1.

DNS (Domain Name System) — the system that connects a domain name to the actual server where the site is hosted. Chapter 6.3.

DOM — the tree-like structure that a browser builds from the page's HTML code before applying CSS styles on top. Chapter 3.1.

Flexbox — a CSS feature for laying out elements relative to each other, e.g., in a row or a column. Chapter 3.4.

GitHub Pages — a free hosting service for simple, static sites (HTML/CSS) served directly from a code repository. Chapter 6.4.

Gutenberg — the default content editor in WordPress, where you build pages and posts out of blocks. Chapter 2.6.

Hosting — the place where your site's files physically reside and which responds to visitors' browser requests. Chapter 6.2.

HTML (HyperText Markup Language) — a language that describes the structure and content of a page using tags. Chapter 3.2.

Media query — a CSS rule that applies specific styles only at a given screen width — the foundation of responsive design. Chapter 3.5.

Theme — the visual design of a site in WordPress or another builder: colors, layout, typography. Chapter 2.5.

No-code — tools for building sites without writing code, using visual drag-and-drop interfaces (e.g., Wix, Squarespace, Webflow). Chapter 4.1.

Permalink — the URL structure of a given page or post in WordPress. Chapter 2.4.

Responsiveness — a site's ability to adapt its appearance to different screen sizes, especially on mobile phones. Chapter 3.5.

SEO (Search Engine Optimization) — practices that increase a site's visibility in Google search results. Chapter 2.10.

SSL / HTTPS — encryption of the connection between the browser and the server, visible as a padlock next to the site's address. Chapter 6.3.

VPS (Virtual Private Server) — a dedicated, more powerful slice of a server with greater control, suitable for advanced sites. Chapter 6.2.

Vendor lock-in — a situation where it's difficult to move your site to another provider because the platform doesn't allow a full data export. Chapter 4.6.

Viewport — the visible area of the page on a device's screen; the viewport setting in your code determines how the site behaves on a phone. Chapter 3.5.

WooCommerce — a free plugin that turns a WordPress site into an online store. Chapter 5.1.

WordPress.com vs WordPress.org — WordPress.com is a hosted service with limitations; WordPress.org is free software you install on your own hosting, with full control. Chapter 2.2.

Plugin — an add-on that extends WordPress functionality, e.g., a contact form, security features, or SEO tools. Chapter 2.9.

End of book

My First Website

Notes