

# Introducing website creation with ChatGPT

---

## **This chapter covers**

- Introducing ChatGPT
- Understanding what kinds of web pages you can create with the help of ChatGPT
- Getting to know the limitations of making web pages with ChatGPT
- Learning how ChatGPT enables you to create your own web pages

The British science fiction writer Arthur C. Clarke once formulated three truisms that came to be known as *Clarke's three laws*, the third of which is the one most often cited: “Any sufficiently advanced technology is indistinguishable from magic.” If you’ve always wanted to create web pages but were daunted by the technology, then the premise of this book—that you *can* create your own website with the help of ChatGPT—may seem like magic to you. That’s no surprise, because if any technology in recent memory deserves the description “sufficiently advanced,” it’s ChatGPT and similar so-called *generative AI*—artificial intelligence applications that can create poetry, prose, and, yes, web pages.

In this chapter, my aim is to demystify ChatGPT and make it seem a little less magical and a whole lot more practical. In the next few pages, you get a short explanation of the basics of ChatGPT, an overview of the kinds of pages you can create with ChatGPT’s help (and, for balance, some notes on the kinds of pages you *can’t* create), and a behind-the-scenes and mercifully nontechnical tour of how you can use ChatGPT to perform the seemingly magical—but in reality, merely mathematical—feat of taking a simple, typed request and turning it into a full-fledged, ready-to-be-surfed web page.

## 1.1 **What is ChatGPT?**

Unless you’ve been hunkered down in a hermitage over the past year or two, you’ve likely heard about GPT and/or ChatGPT, the artificial intelligence agents that have taken the world (or, at least, that part of the world that pays attention to social media) by storm. However, knowing *about* ChatGPT is one thing, but knowing what ChatGPT actually *is*, is quite another.

To help you get a feel for what ChatGPT is and what it does, it helps to break down each component of the name to see what it means. I’ll begin with GPT:

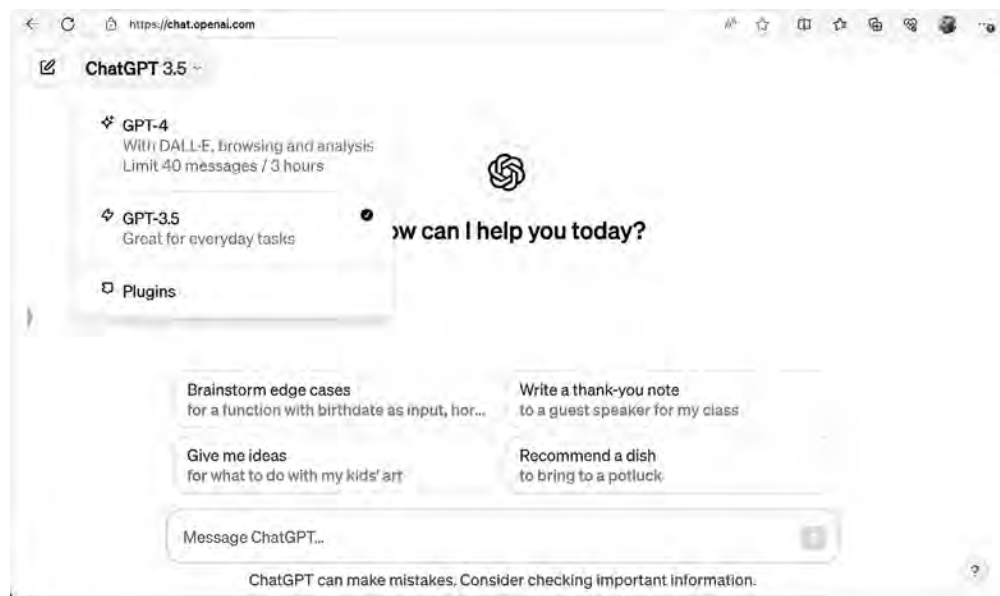
- *G*—The *G* in *GPT* stands for *generative*, which means GPT can create—or *generate*—new content. GPT is a *large language model (LLM)*, which generally means it’s designed to generate text, such as essays, stories, and even poems. More specifically for our purposes in this book, GPT’s generative capabilities also extend to programming code, particularly the code that underlies web pages.
- *P*—The *P* in *GPT* stands for *pretrained*, which means GPT was exposed to huge amounts of text (that’s what the “large” in *large language model* is referring to). During this *pretraining* process, GPT learned the patterns and structures in the language, such as how sentences are typically formed. In particular, given some existing text, pretraining enables GPT to predict what word or phrase usually comes next. In a sense, that’s all GPT is really doing: predicting the next word! GPT was also trained on an unimaginably large set of programming data; so, because programming code is usually more predictable than regular writing, GPT excels at generating code.
- *T*—The *T* in *GPT* stands for *transformer*, which means GPT is able to take some text input (such as a request to create a web page) and transform that text into its most important components while ignoring or giving lower priority to less important components of the text. This enables GPT to produce results with greater relevance and accuracy.

The *Chat* part of *ChatGPT* means you have conversational access to GPT, which means you can exchange messages with GPT more or less as you do in a chat conversation. In the context of creating web pages, these “conversations” will amount to you requesting some page component and ChatGPT providing the required code. However, ChatGPT also has a limited ability to “remember” previous messages in the current conversation, which can occasionally be useful in your website creation tasks.

As I write this, two main versions of GPT are available: GPT-3.5, released in November 2022, and GPT-4, released in March 2023. If you use the OpenAI app (discussed in a moment) either with a free ChatGPT account (learn more in appendix A) or with no account, you have access only to GPT-3.5; if you have a paid ChatGPT Plus account and you use the OpenAI app, you have access to both GPT-3.5 and GPT-4.

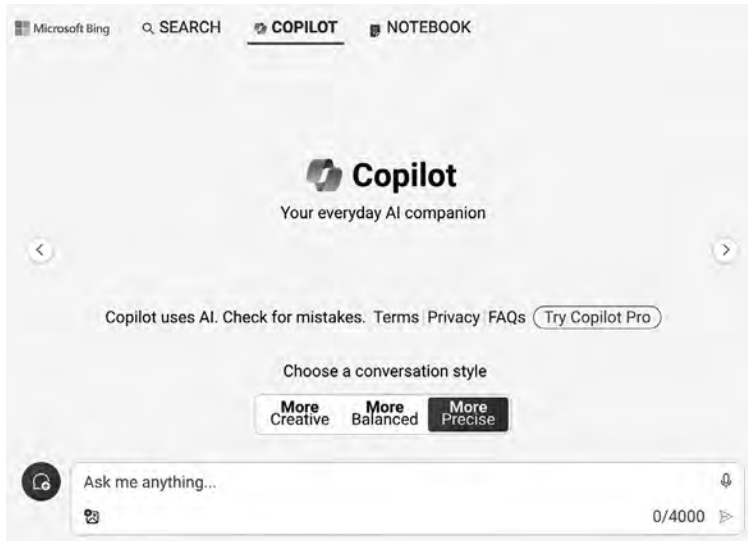
To access ChatGPT to help you create web pages, you have three preferred choices:

- *OpenAI app*—This is an online app operated by OpenAI, the creators of GPT and ChatGPT. The app is available at <https://chat.openai.com>. You don't need an OpenAI account to access it, but having an account removes certain restrictions (check out appendix A to learn how to create an account to access ChatGPT). If you have a paid ChatGPT Plus account, you get to choose between GPT-3.5 and GPT-4, as shown in figure 1.1.



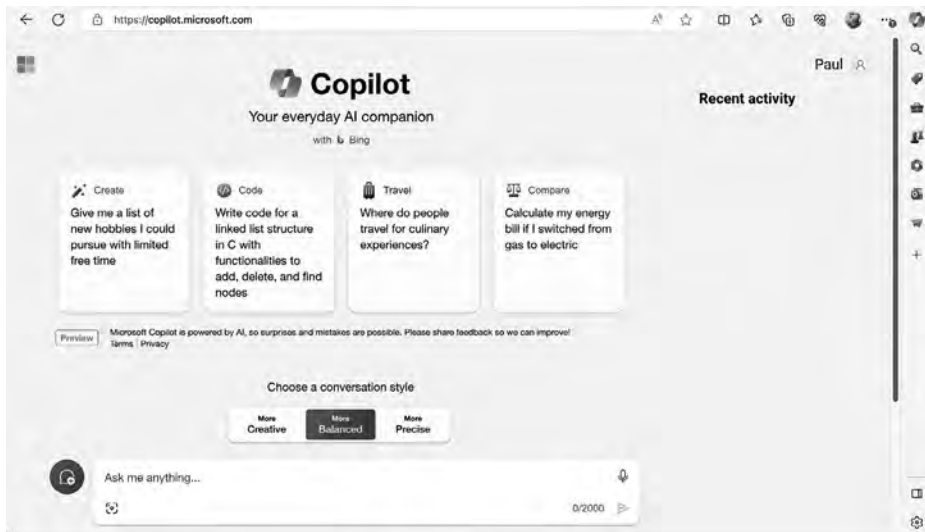
**Figure 1.1** With a ChatGPT Plus account, the OpenAI app gives you access to both GPT-3.5 and GPT-4.

- *Microsoft Copilot in Bing*—This is the AI-enhanced version of Microsoft's search engine, which offers a chat feature that uses GPT-4 behind the scenes and also has access to the web. Navigate to <https://bing.com> and select the Copilot tab to get started, as shown in figure 1.2. Note that you don't need a ChatGPT account to use Bing Copilot. (If you're wondering about the three "conversation style" choices, I explain them in detail in appendix A.)



**Figure 1.2** Use Bing Copilot to work with GPT-4.

- *Microsoft Copilot*—This is the standalone implementation of Microsoft’s version of ChatGPT, which uses the GPT-4 model. Navigate to <https://copilot.microsoft.com>, as shown in figure 1.3. Note that you don’t need a ChatGPT account to use Microsoft Copilot, but you do need a Microsoft account. (Again, I explain the three “conversation style” choices in appendix A.)



**Figure 1.3** If you have a Microsoft account, use Copilot to work with GPT-4.

**NOTE** There are quite a few other ways to access ChatGPT, either directly or indirectly. For example, there are ChatGPT apps for iOS and Android. These other methods are fine for messing around with ChatGPT, but for creating web pages, it's best to stick with the OpenAI app, Bing Copilot, or Microsoft Copilot because they give you easy access to the code generated by ChatGPT.

Which method should you use to access ChatGPT? When it comes to creating the relatively simple web page code that's covered in this book, it doesn't matter all that much. GPT-4 tends to produce more "modern" code, which is generally a good thing. (However, this may mean your pages don't work well in very old browsers such as Internet Explorer. My advice? Ignore those ancient browsers and embrace the modern web!). If you don't have a ChatGPT account and don't want the restrictions that the Open AI app imposes on non-account holders, Bing Copilot is the way to go.

## **1.2 ChatGPT enables anyone to create web pages**

I've never been lucky enough to have an assistant, but I can imagine that it's a sweet deal. After all, who wouldn't be happy to have someone dedicated to performing necessary but mundane tasks such as setting up virtual meetings; arranging flights, hotels, and other travel details; and putting together presentations? Having an assistant is even more valuable if you don't know (and don't want to know) how to set up a Zoom meeting, navigate the Expedia website, or build a PowerPoint presentation.

These tasks are nontrivial, for sure, but they seem like kindergarten exercises compared to designing, coding, and deploying a web page or even an entire website. Sure, there are templates and similar page-building tools out there that offer ready-to-roll web pages, but the results are almost always disappointing because you get only a limited amount of control over the output. You have a great topic for a site, and you have a mental image of how it should look, but there's a barrier between you and your vision: the web page code.

The three main types of web page code—HTML, CSS, and JavaScript—reside on a steep learning curve: a slope that for many people is too onerous a climb when all they want is to put up a page or three for themselves or for a team, hobby, project, charity, or event. If you want to create a presence on the web but have either been disappointed in the available prefabricated solutions or shuddered at the thought of learning HTML, CSS, and JavaScript, all is not lost, because now you can "hire" (for free, if you prefer) an assistant to generate web page code to your exact specifications. That assistant is ChatGPT, which sits well on the other side of the web development learning curve and is ready, willing, and very able to help you turn your web page vision into web page reality. By providing the model with a few simple, plain-English instructions, you can cajole ChatGPT into translating those instructions into working web page code. You then upload that code to the web (ChatGPT can even help you with that step), and you're done!

Okay, it's perhaps not quite as easy as all that (otherwise, there wouldn't have been much point in writing the rest of this book!). But the basic procedure is every bit as straightforward as I've outlined it here.

### **1.3 Understanding the types of pages you can (and can't) create**

By now, you may be thinking that transforming ChatGPT into a website-building robot assistant must come with some sort of catch. After all, to the uninitiated, building web pages seems like an almost quintessentially complex task, so claiming that you can farm out almost all that labor to an AI model without having to learn to code yourself must come with a rather large "gotcha." Surprisingly, there are no such flies in the ChatGPT ointment, but there are a few provisos to bear in mind.

First, you need to know that the types of web pages most easily created with the help of ChatGPT are those described as *static* in the web development trade. A static web page is one that contains text and data that don't change after the page is loaded. That may sound restrictive, but there's really no limit on the types of static pages you can ask ChatGPT to help you build. Here are 10 ideas:

- Personal home page
- Information page for a team, organization, or event
- Product landing page
- Hobby page
- Photo gallery
- Portfolio page
- Post page (essay, review, fan fiction, or whatever)
- Top-10 list
- How-to instructions
- Travel guide

I bet you can easily come up with 10 more on your own. These are the types of pages you learn how to create with the help of ChatGPT in this book. And the really great news is that unless you opt for a paid ChatGPT Plus subscription, you can do everything—from accessing ChatGPT to saving the generated code to deploying your pages—completely free.

Second, one of the characteristics of the types of pages I just listed is that they only require what web development geeks call *frontend* code, which means code (HTML, CSS, and JavaScript) that runs in the web browser. A much different beast is *backend* code, which means code that runs on a web server and is generally used to supply text and data for a *dynamic* web page, where the text and data change on the fly.

Technically, it's possible to ask ChatGPT to supply such backend code, but practically speaking, you're talking about setup tasks that are an order of magnitude more complicated, far more complex in terms of organizing and deploying the generated code, *huge* security risks because web servers are vulnerable to many types of attacks and advanced

coding techniques must be used to harden backend code against malicious users, and almost always added costs because web hosting accounts that enable access to a server usually require a paid subscription plan. For all these reasons, you don't learn how to use ChatGPT to generate backend code in this book.

## 1.4 How you use ChatGPT to help you build web pages

*Fritterware* is an old computing term that refers to any software that seduces the user into spending inordinate amounts of time playing around with the program's features and options. Once you have a ChatGPT account (which I cover in appendix A; however, remember that having an account is optional), you have instant access to one of the all-time great pieces of fritterware ever invented! It's easy to spend huge hunks of time getting ChatGPT to do all kinds of fun and silly things, but eventually, you'll want to stop frittering and start creating.

The ChatGPT creative process varies depending on what you're making, but for our purposes here, you need to know the overall process of getting ChatGPT to help you create web pages. The simplified version of that process is summarized in figure 1.4. The next few sections explain each step, and Chapter 2 takes you through a complete example of the process.

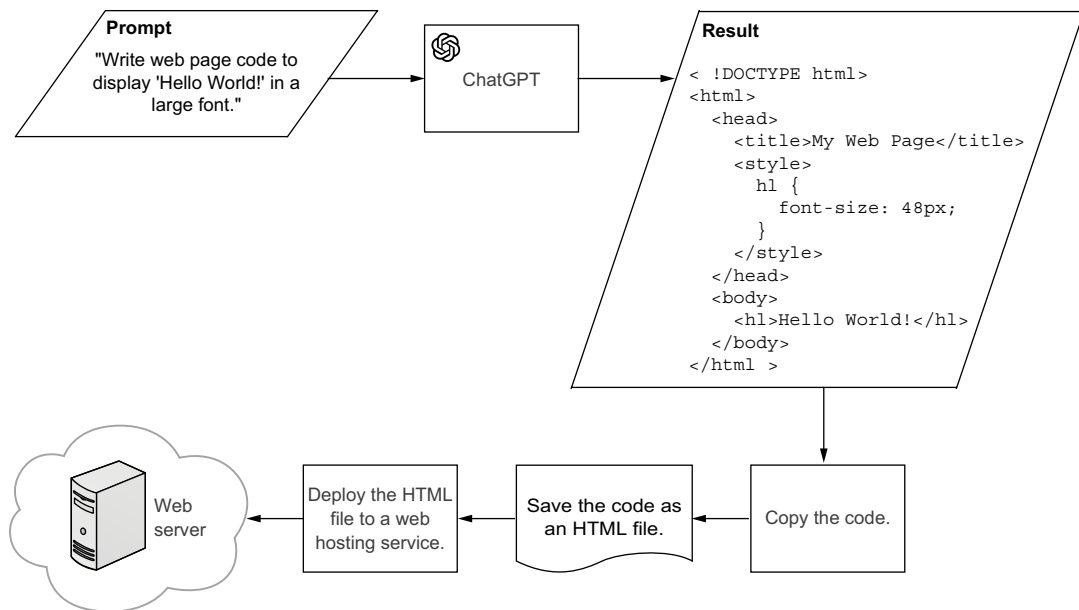


Figure 1.4 A simplified diagram of the process you use to get ChatGPT to help you create a web page

### 1.4.1 Prompting ChatGPT

The process always begins with a request that specifies what you want ChatGPT to create for you. This request is called a *prompt*. When you log in to the ChatGPT app (refer

to figure 1.1) or navigate to Bing Copilot (refer to figure 1.2), you see a text box where you type your prompt.

Prompts can be as simple as a single sentence (as demonstrated in figure 1.3) or as complex as multiple paragraphs. However, the sky is definitely not the limit here because most versions of ChatGPT will accept only up to 4,000 characters per prompt. That's a good 500 words or so, which ought to be more than enough for most of your web creation prompts.

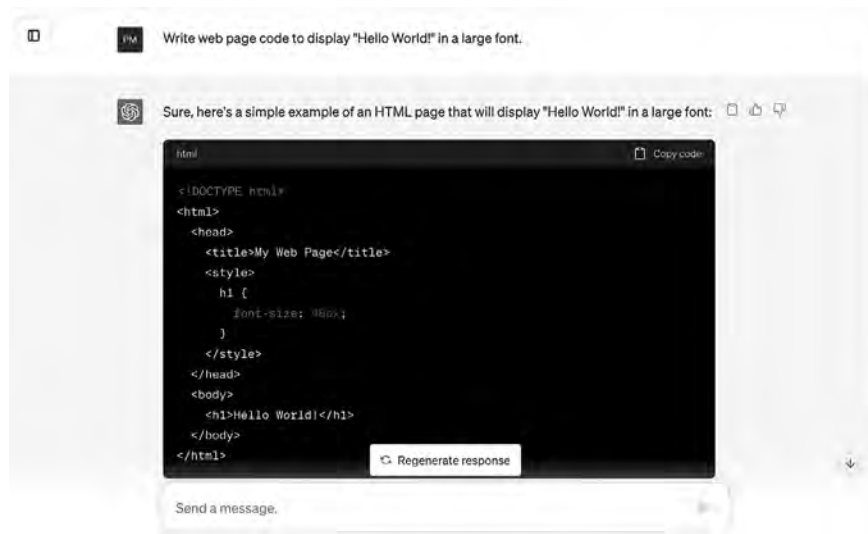
It's no exaggeration to say that prompting ChatGPT is the most important step because the quality of your prompt directly determines the quality of the result returned by ChatGPT. In a sense, this entire book is about providing you with high-quality prompts that get ChatGPT to perform specific web page creation tasks. I also devote quite a few pages in appendix C to explaining some best practices associated with prompting ChatGPT (a process called *prompt engineering* by the cognoscenti).

**WARNING** Although it's generally true that ChatGPT is a quality-in, quality-out model, it is, like all large language models, prone to occasionally producing unusable or downright weird results even when the prompt is a good one. I talk about some ways to troubleshoot such problems in appendix A.

### 1.4.2 Viewing ChatGPT's results

When you submit your prompt, ChatGPT sets to work and usually begins “typing” its response within a few seconds. Figure 1.5 shows an example response to the following prompt:

Write web page code to display “Hello World!” in a large font.



**Figure 1.5** In this example, ChatGPT's response to the prompt includes the requested code.



As shown in figure 1.5, the result in this case consists of a friendly response to your prompt followed by a box titled `html` that includes the web page code that was requested. This code will no doubt seem like near-gibberish to you, but trust me when I tell you that it does exactly what the prompt asked for: it displays the message `Hello World!` in a large (in this case, 48-point) font. Note that, for various technical reasons, ChatGPT may not return the same code each time you run the prompt. However, because there are often several ways to achieve the same result with HTML and CSS, the generated code will still produce a web page that looks the same (or at least very similar).

### 1.4.3 Getting the code into a file

The example prompt I used in the previous section is a simple one, for sure, but it's still more than a little awesome that within a few seconds of submitting this prompt, ChatGPT generated some workable web page code that satisfied the request. As you work on your web page projects with ChatGPT at your side, as it were, that sense of amazement will come up again and again as the model quickly and seemingly effortlessly generates the code you asked for.

However, the web page code produced by ChatGPT, no matter how accurate it is or how suitable to your needs, doesn't do anything. That's because, on its own, web page code is inert; it's just a collection of words and odd-looking symbols. For web page code to come alive, it must be displayed in a web browser; and before you can do that, you must get the code into a file that a browser can access.

Appendix A dives into the ins and outs of web page files in just the right amount of detail. Figure 1.6 shows the code generated by ChatGPT (refer to figure 1.5) pasted into an HTML file, which has been saved as `index.html`. At this point, you could load the saved HTML file into your favorite web browser (see chapter 2 for the details), but a true web project requires one more step: deploying the code to the web itself.

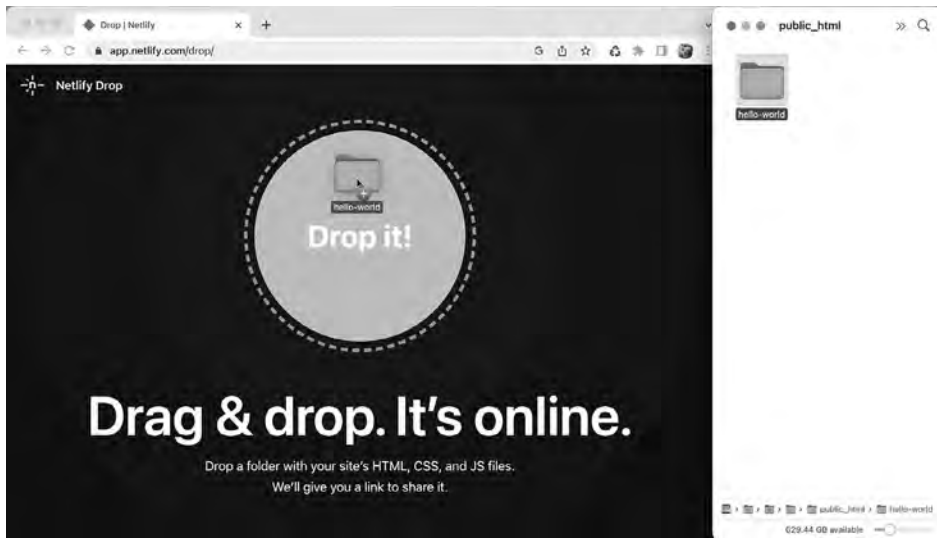


```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>My Web Page</title>
5     <style>
6       h1 {
7         font-size: 48px;
8       }
9     </style>
10  </head>
11  <body>
12    <h1>Hello World!</h1>
13  </body>
14 </html>
```

Figure 1.6 The code copied from ChatGPT and saved to an HTML file

### 1.4.4 **Deploying the HTML file**

The only way other people can view your pages is for you to put them on the web. For the relatively simple projects you create in this book, this deployment process involves copying the file or files that ChatGPT helped you create to a service that hosts web pages. I go into this process in more detail in appendix B, but for the most part, it just means uploading the folder in which you've stored your web page files to the host. Figure 1.7 shows an example of the process where I've dragged the hello-world folder from the Finder window on the right and am about to drop it on the window on the left. With your web page files uploaded, you can view them right away in your favorite web browser, as shown in figure 1.8.



**Figure 1.7** With some services, such as Netlify, shown here, deploying is a simple drag-and-drop affair.



**Figure 1.8** The web page created by ChatGPT is on the web.

The previous few sections took you through a prompt-copy-save-deploy process that has, I think you'll agree, a satisfying—and more than a little surprising—straightforwardness to it. However, not all of your projects will be this uncomplicated, particularly as you venture into creating web pages that go well beyond saying “Hello World!” For these more ambitious projects, you'll most often use two extra techniques: repeating the prompt-copy-save-deploy process to create multiple pages and page components; and refining and revising your ChatGPT prompts.

#### **1.4.5 Repeating as needed**

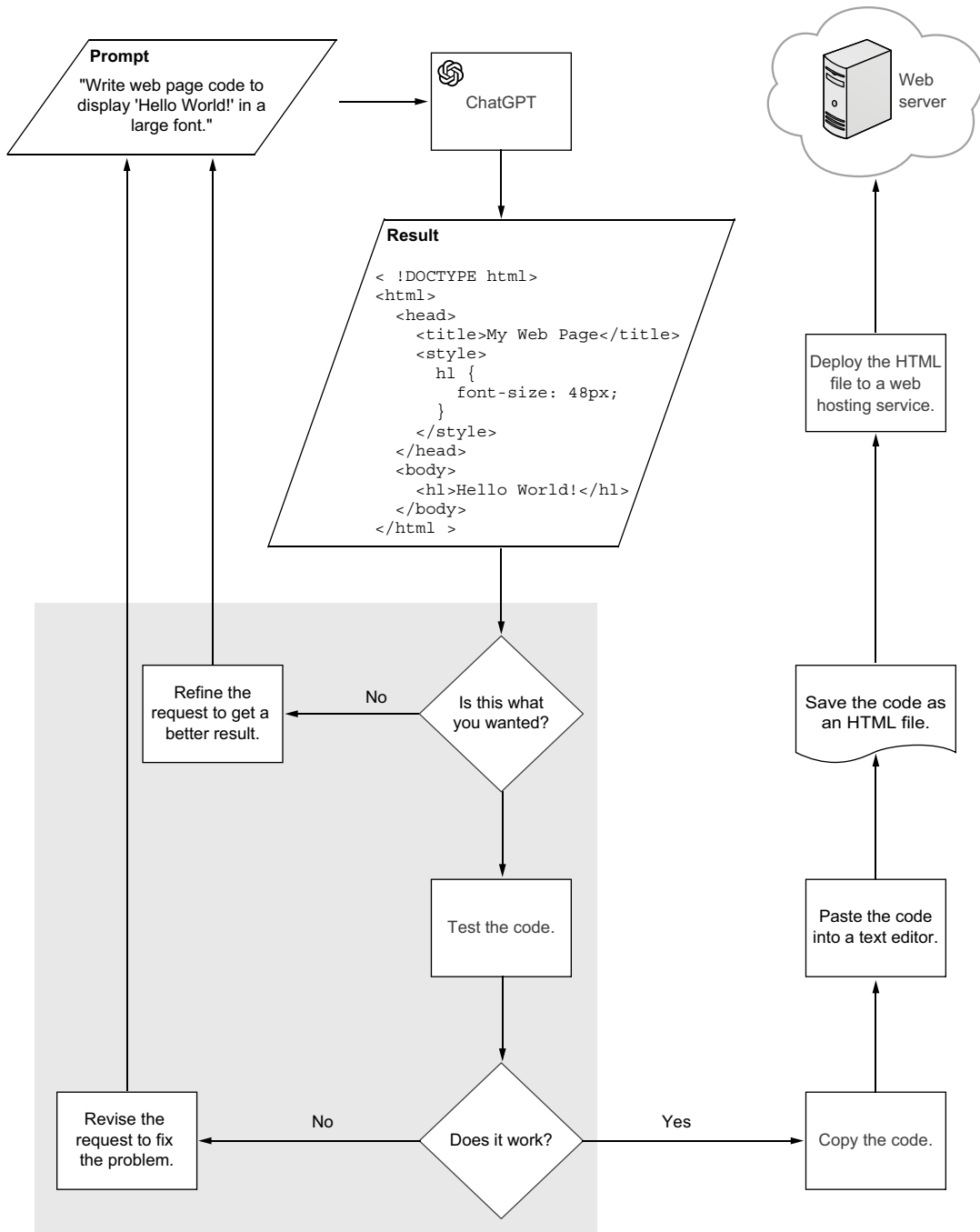
A single prompt that creates a single web page will be relatively rare in your web page creation career. It's far more likely that you'll have to perform the prompt-copy-save-deploy cycle multiple times before you end up with all the content you need. There are two scenarios to consider:

- *Creating multiple pages*—Lots of websites consist of a single page, but the far more common case is a website that consists of multiple pages. Even a modest website may consist of a home page, an About page (that describes you or your site), a Contact page (that lists the various ways site visitors can get in touch with you), and separate pages for content such as essays, photos, and portfolios. For such a website, you'll need to repeat the prompt-copy-save-deploy cycle for each page.
- *Creating multiple components for a single page*—Most modern web pages consist of multiple components, such as a header, a navigation bar or menu, a content area, a sidebar, and a footer. It's possible to include all the components you need in a single prompt, but you'll get more satisfying results if you use a separate prompt for each component. In this case, the process becomes one where you repeat the prompt-copy-save part of the cycle for each component, and the copied code gets inserted into the same HTML file. You then deploy the code only after you've added all your page components.

If all this seems a bit abstract or confusing now, don't worry: the rest of this book is all about getting you comfortable with the specifics of creating multiple pages and page components.

#### **1.4.6 Refining and revising your prompts**

The ChatGPT-assisted page creation process depicted earlier in figure 1.3 works for uncomplicated web pages and for those times when ChatGPT nails your request. But as your web pages—and your prompts—grow more complex, you'll almost certainly find yourself adding a few more steps to the process. I've included these extra steps in the shaded area of the process diagram shown in figure 1.9.



**Figure 1.9** Refining and revising your web page prompts

At first glance, this likely appears to be a much more complex process, but really, there are just two extra paths this new process can take. These two paths come in the form of questions:

- *Is this what you wanted?* You ask this question immediately after ChatGPT generates the code you requested in your prompt. Basically, you're examining the code (as best you can, anyway) to make sure it does the job for you. If the answer is "Yes," you move on to the next question; if the answer is "No," you refine the prompt in some way (for example, by making it more specific) and try again.
- *Does it work?* This question requires that you test the code by copying it and then pasting it into an online site designed for testing web page code, as I describe in appendix A. If the code checks out, you can proceed with saving it to an HTML file. If the code is wrong somehow, you need to fix the error by revising your prompt in some way and then resubmitting it; again, appendix A is the place to go for some troubleshooting suggestions.

Sure, it would be great if ChatGPT generated awesome code every time. And it's great that each new release of ChatGPT gets better not only at producing accurate code but also at creating code that matches even vague or general prompts. But although ChatGPT may someday routinely crank out perfect web page code every time, it's not there yet, so you'll need to refine and revise your prompts to get the pages you want.

## Summary

- GPT stands for *generative* (it's designed to generate text), *pretrained* (it was exposed to large amounts of text to learn language patterns and structures), *transformer* (it analyzes requests to give higher priority to the most important components).
- ChatGPT is an app that enables conversational access to GPT via the OpenAI app at <https://chat.openai.com>, via Bing Copilot at <https://bing.com> (select the Copilot tab), or via Microsoft Copilot at <https://copilot.microsoft.com>.
- GPT has been trained on massive amounts of programming code, including the code used to build web pages: HTML, CSS, and JavaScript. This training enables ChatGPT to generate web page code from plain-English instructions.
- ChatGPT is best used to create static web pages that don't require or rely on data stored on a web server.
- ChatGPT helps you create web pages using a basic prompt-copy-save-deploy cycle, where the *prompt* is the instruction that tells ChatGPT what type of page you want, *copy* means to copy the code generated by ChatGPT and paste it into a file, *save* means to save the code as an HTML file, and *deploy* means to upload the HTML file to a web hosting provider.
- For best results, you'll usually have to refine your prompts to get the structure and content of your page just right, and you'll usually have to revise your prompts to fix page problems.