DIRECT RESPONSE
SITE DESIGN

EVERYTHING YOU NEED TO KNOW
TO GET YOUR WEBSITE UP AND RUNNING FAST!

Contents

1

WHY WEBSITES?

If history can be reduced to one overriding principle, it'd be this: every new technology change society forever. The development of ironworking changed military and construction forever, enabling us to wage longer wars, to build taller buildings, and to develop tools to move beyond agriculture for the first time in history. The development of the printing press allowed ideas to be spread to the common man, pushing us toward democracy and putting a market value on ideas. And the development of corporate law allowed industries to develop to the point that ever-higher levels of industrial development and research were possible, completely transforming the world in the space of about 150 years.

Today, the new technology is the Internet. And in the fifteen-odd years that the Internet has been available to the masses, it's created nearly as many opportunities for promotion, communication, and business as all the massive technological development throughout history.

In this book, we will focus on one of the most profitable opportunities available to the prospective online businessman today--direct response website marketing. We'll talk about just what direct response site design is, and how to design your web pages in order to convert as many visitors as possible into satisfied customers.

We will talk about the nuts and bolts of building a website, whether you want to do it yourself or whether you want to hire a professional coder to do it for you. We'll talk about what to include (and what not to include) on your website, and we'll give you the tools you need to write a sales letter for any product you choose to promote. And finally--and most importantly--we'll help you come up with some ideas to promote your website that'll bring in the traffic you need to make your business idea into a proven success.

Sounds good? Then let's get started by talking a little bit more about your chosen medium--the Internet.

A BRIEF HISTORY OF THE INTERNET

Some of the revolutionary technologies we talked about at the start of this chapter--iron refining in particular--had their roots in military applications. It's a strange irony of human life-commented on by several philosophers, Friedrich Nietzsche not least among them--that war, the most destructive of human drives, is the one that most often requires people to stretch thin resources far enough in order to develop new, world-altering technologies. And the Internet is far from the least of these military "success stories."

The seeds of the Internet were first planted in America in the 1950s, during the height of the Cold War. In response to fears of Soviet domination of two critical technology races--the Space Race and the nuclear arms race--the United States government created ARPA, the Advanced Research Projects Agency, to develop countermeasures to the perceived Soviet threat. One of ARPA's first major successes was to develop rules of communication--the ancestors of today's network protocols--that allowed all the United States' radar systems to be linked together, which would enable them to communicate rapidly and retain infrastructure even in the event of an unexpected nuclear attack.

Fortunately, that nuclear attack never came--and equally as fortunately, the scientists at ARPA began to realize the non-military potential for the linking of vast computer systems. The first plans for ARPAnet--the predecessor of the Internet--came about as part of an effort to make it easier for vital research-based computers to communicate with one another, allowing scientists to coordinate their research efforts across vast geographical distances.

After some years of work defining some of the basic principles on which the Internet works (a discussion of which is well outside the scope of this book), ARPAnet finally went live near the end of 1969, with four "nodes" active at four prominent universities. By 1971, the first emails could be sent across the network (and soon email was taking up some 75% of all network traffic--some things never change!), and by 1973, it was possible to send files from computer to computer through the FTP protocols that the Internet still uses today.

Although computer processor speed, memory capacity, and general power advanced exponentially throughout the years, one of the most significant events in the development of the Internet came in 1978 when Western Union, the British Post Office, and a network company all came together to create the first truly international computer network. At around this time, the number of network nodes on ARPAnet were growing into the hundreds--making the project far more ambitious than the initial plan to simply connect a handful of powerful research computers.

But the greatest step in the Internet's development came in 1983, when the simpler ARPAnet protocols for transferring information were supplanted by the TCP/IP protocol still used today--which transformed the ARPAnet into what we recognize today as the Internet.

THE HISTORY OF THE COMMERCIAL INTERNET

The basic idea of the Internet--allowing people to communicate and share data across vast geographical distances--was too good to keep away from the public for long. And in 1985, the government recognized this by opening ARPAnet--now more properly known as the Internet--to commercial interests.

Those commercial interests initially did little beyond allowing anyone who wanted to--rather than anyone with military clearance or academic credentials--to access the existing information networks that comprised the Internet. Companies like CompuServe and Prodigy offered anyone who was willing to pay for the privilege to use phone lines to "dial-up" computers connected to the larger Internet, which would then allow users at home to access the same database of information used by the academic computers--or at least a proprietary version of the same information.

However, while computer technicians on ARPAnet or computer enthusiasts on the long-running Telnet might have been willing to learn the sometimes-arcane process of accessing information on the Internet, the majority of potential users tended to view the Internet in the same way as a casual library user might view an encyclopaedia--nice to look at sometimes, but ultimately not worth the trouble of owning.

That was largely changed by the efforts of the fledgling Netscape company, whose Mosaic web browser (one of the earliest versions of the once-popular Netscape Navigator) made use of a new protocol for accessing information on the Internet: HTML. Before HTML, the process of sending information on the Internet was usually limited to sending large text files (that required a separate program to open) or to using a proprietary form of encoding information--proprietary forms that usually skimped on things like presentation, graphics, or other features that we take for granted on the Internet today.

HTML, by contrast, allowed nearly anyone with a basic knowledge of computer programming (or anyone who was willing to learn a few simple formatting instructions, or "tags") to create web pages that looked nice, create images to illustrate product offers or scientific concepts, and to "link" people to other

websites. (If you're interested in learning more about HTML, stay tuned--in a later chapter, we'll discuss more about the ins and outs of HTML than you may ever have wanted to know about--and if you don't want to know about it at all, we'll discuss some ways to get your
HTML coding done for you.)

The advent of HTML allowed the Internet to grow exponentially. And grow exponentially it did--by the mid-1990s, the Internet had moved beyond a few simple "bulletin boards" and academic research compendia. Now, companies were starting to sell their products online--artists were distributing their work in vastly populated forums--even families were publishing their own "web pages" to announce family events to the world at large. The age of the Internet had truly arrived.

SO WHY THE INTERNET?

If you're reading this book at all, it's likely that you've already decided to use the Internet as a major distribution tool for your product (if not the exclusive distribution tool for your product.) But for anyone who hasn't yet made that decision--or for anyone who wants to know just why the Internet is so ideal for business purposes--we'll talk briefly about why the Internet is a perfect vehicle for virtually any good or service you might want to sell.

As of 2007, over one billion people worldwide use the Internet. However good the offline marketing for your product is, that's too large of a potential market to ignore. More than that: if you promote your product by traditional means, the number of people you can reach is limited by the amount of money you can spend and the number of media outlets you can reasonably contact-meaning, most usually, the media in your city alone. But by using the Internet, you can not only reach people around the world directly--but by making use of business referrals from satisfied online customers, you can reach people outside of your geographic range who'd never even think of using a computer.

Additionally, traditional advertising only affords you a limited amount of space to talk about exactly what your product is and why people should buy it from you. But with the Internet--and specifically with direct response marketing--you have unlimited space to discuss your product in detail, to give yourself and your business a distinct identity, and to convince people more effectively to buy your product.

And finally--as we'll talk about in more detail in a later chapter-online business radically simplifies some of the major costs of running a business in the first place. If you want to start a traditional business, you'll need to apply for a license, you'll need to acquire a storefront or office space, and you'll need to devote huge amounts of time to the tedious process of filing tax documents, managing a payroll, and balancing your accounting books.

With online commerce, you may still need the business license (research business law in your area to find out if this is the case), but you're no longer required to find business premises (beyond registering a domain and buying some web space), and with online payment services and software, most of your accounting takes care of itself. All you need to do is develop a great product or service, set up your basic website and payment options, write a killer sales letter--and watch your product start to sell itself.

Sounds appealing? Then let's learn a little bit more about one of the most effective ways of promoting products in the modern Internet age--direct response website marketing.

2

THE BASICS OF DIRECT RESPONSE

By putting your business on the Internet, you stand to reach a much wider audience than you can through traditional channels, you're able to simplify your business operations considerably, and you can present more information about your product than you can through any other medium.

But as good as all that sounds, there's more to online business than simply putting your product out there and waiting for sales. As with any business, you're going to have to put some thought into how you plan to market your business. After all, there are millions of websites out there, and if you don't set your site apart from the rest, you stand a very real chance of vanishing into the crowd--no matter how good your product or service is.

And one of the best ways to promote your website online is through direct response website marketing.

WHY DIRECT RESPONSE?

In order to understand what differentiates direct response from other commercial websites, it's helpful to think of the mass communications medium of yesterday--television.

When most of us think of television programming, we tend to think of "shows". Television "shows" have two distinctive features in common:

They focus on providing content to an end user (the viewer). The quality of the content determines the quality of the show. They make their money by convincing other people to invest in the show.

Neither of these points determines what the content of the show will be, or what, exactly, will make the investors--whatever form they take--give their money to the producers of the show. The show could be a drama, a comedy, a variety show, a "reality show", a documentary, or any of a hundred other categories. The investors might pay for advertising space on the show, buy tapes or DVDs of old episodes of the show, or even send money directly to the producers in response to solicitations (as with the pledge drives public television uses to raise funds.) Whatever form the show, or the revenue-building strategy takes, however, one key principle holds: content over marketing.

If traditional web design can be thought of as "shows", direct response website marketing can be thought of as infomercials.
Infomercials have the following two features in common:

- They focus on describing a product to the end user. All content in the show is subordinated to the central goal of describing, promoting, and ultimately selling the product.

- The show makes its money through direct product sales.

Again, neither of these points determines absolutely what form the infomercial will take. An infomercial might include a short dramatic sketch, a mock "panel discussion", a comedy act, or simply a straightforward list of features and advantages for a given product. But as the first point states, all these entertaining or informative pieces of content must be subordinated to the central goal of selling products. A "show’s central point is, essentially, to get people to watch the show (and to make their money based on the number of people who watch the show.) Whereas an infomercial's central point is ultimately to get people to stop watching the show and to start ordering the product.

So again: why direct response? In traditional web design, the point is to sell the customer to the advertisers, or to sell the customers on your website (and to make enough of an impression that they buy your products or otherwise subsidize the site.) This can be more satisfying aesthetically, yes--but for a traditionally designed website to be successful, it needs to offer a wide variety of content from which users can freely choose.

In simple terms, you need to offer your potential customer 20 things to do on your website--only one of which is buying your product. Ideally, that gives you a 5% chance to turn a visitor to your website into a sale. The basic idea behind direct response marketing is to give your potential customer just one thing to do on your website--buy the product. Ideally, this gives you a 100% conversion rate. If each type of website gets the same amount of traffic--and assuming that your goal here is to sell products rather than to entertain casual web surfers--direct response websites, simply make more sense as a marketing tool.

Staunch television fans might find the comparison between direct response marketing and infomercials troubling--after all, very few people turn on the television in order to watch an infomercial, and the ability to eliminate commercials and other televised advertising from "shows" is one of the central features used to market the successful TiVo product. But if you intend to run a successful online business, it's important to remember to think in these terms: you're no longer a consumer of products; you're a producer. In television terms, you're no longer watching shows for entertainment, or even producing entertaining shows--you're producing commercials for a product, and the success of your business depends on how successfully you can convert viewers into customers. And infomercials, for all the disadvantages they have in terms of entertainment, are infinitely more effective than "shows" when it comes to producing customers. (Yet by no means does your own "infomercial" have to be devoid of art, entertainment, or other such content--a topic which we will get into further in a later chapter.)

BASIC PRINCIPLES OF DIRECT RESPONSE

Just as a few fundamental principles differentiate "shows" from infomercials, there is a parallel set of principles that differentiate traditional websites from direct response sites:

- Traditional websites focus on the website itself. All money made from the website is based on outside advertising, donations to support the site, or sometimes selling products from the site.

- Direct response websites focus on selling a product. Most of the money made from the website should be directly related to product sales.

And just as in the television world, the fundamental principles behind a type of program dictates to some extent the form the program will take, your goals for your website will determine what you can and can't do when marketing your product.

More fundamental than even these principles, though, are the limitations of the medium itself. In the television world, all programs will have a few common elements: all programs will either be filmed with cameras or generated by computers, all viewers will tune in to all programs by selecting a channel on a TV set, and all programs will be sensitive to time constraints imposed by the TV station's programming schedule.

To highlight the differences in approach between these two forms of commerce, let's first look at some of the common elements among all types of website, whatever its purpose:

- All websites are based on HTML code (the analogue of a camera or a computer filmmaking program in the television world.)

- All websites are accessed either through links from other websites or through a customer typing in the URL (web address.)

- Websites are not, as a rule, time sensitive. A viewer can browse a website for as long as he or she wants before leaving or buying a product.

We'll talk some more about how to use these common elements of all websites most effectively throughout the rest of this book. But for now--taking into account both these common elements and the fundamental principles that drive each form of web commerce--let's look at some of the specific differences between traditional web design and direct response marketing websites.

Basic Rule #1: No External Links

In traditional web design, your website benefits to some extent from the amount of links you offer--viewers have more of an incentive to visit your site. (This is the logic behind the successful "portal" websites that sometimes offer little beyond a very good and well-organized collection of links.) In direct response web design, your website is harmed by offering viewers external links, or any distractions from buying the product.

Therefore, the first basic rule of direct response web design is this: no external links. Once your customers get to your website, they should only leave once they've bought your product (or decided that it isn't for them.)

Basic Rule #2: Small User Base, High Conversion Rate

Traditional web design's aim is to attract and retain a consistent user base. If a traditional website's hit count is high, the website is more attractive to potential advertisers, allowing the website to make money. But that high hit count carries with it a cost: bandwidth fees. (Think of bandwidth fees as your electronic "rent.") The more users go to a website, the more data your web server is responsible for sending and receiving--which means a higher basic operating cost.

In direct response web design, you also need a high hit count-initially. What you don't want is a high hit count that keeps coming back for more without ever buying a product (unless you want your viewers to constantly return to your site in order to check for new products and offers--for some specific notes on what to do if this is your business model, see the next chapter.)

In order to accomplish this, make your website as simple as possible. Give your viewers some basic facts about the product, some information about how this product stands up against its competitors, a few positive testimonials or a photo or two, and then an ordering procedure. That way your viewers can read your information, think about whether they want to buy the product or not, and then either buy it or leave--keeping your bandwidth costs down by keeping out people who won't be of value to your product sales.

Basic Rule #3: KISS

To go back to the television metaphor for a moment: imagine that you're watching an infomercial for a new kitchen knife. You hear the expert testimony, you watch the knife do its work, and you find the price reasonable. You decide, based on all of this, to buy the knife. So, you wait for the end of the infomercial in order to learn how to order this wonderful knife--and you're told to dial a certain 1-800 number, to navigate a complicated menu, to enter a 12-digit confirmation code from the infomercial along with a social security number, and finally to talk to a sales representative about shipping information and payment. By the time you've gone through all of this and are about to read off your credit card numbers, a thought strikes you: was one knife worth all of this trouble?

This is exactly what you don't want your customers to think at any point while they're using your website. If your website is built on flashy but hard-to-use menus, if your ordering information is buried under a maze of links and supplementary pages full of product details, and if your ordering procedure is complicated and full of bugs, your customers are going to be too frustrated with your website to order your product--and worse, they might start to think of your products and your business as equally customer-unfriendly.

Computer scientists have an acronym to combat this possible problem: KISS, or "Keep It Simple, Stupid." If your website is easy to navigate, if the information about your product is clearly presented on as few pages as possible, and if your ordering procedure is bug-free, then viewers won't be scared off by your website--which leads to a much higher conversion rate, and thus a successful direct response marketing strategy.

We've talked about what direct response website marketing is, why it's an excellent strategy for your business, and above all a few basic principles of its implementation. And if you like what you've read so far, then you've got everything you need to start thinking in practical terms at last--and it's time to start designing your direct response website, which we'll talk about in the next chapter.

3

DIRECT RESPONSE SITE DESIGN

So, we've talked about the Internet in general, and we've talked about some of the key rules of direct response site design to keep in mind when you're actually building and publishing your website.

As a reminder, here are those rules again:

- No external links
- A small user base and a high conversion rate
- Keep it simple, stupid

If you have your product ready and a good amount of marketing material related to it, then it's finally time to start building your direct response site--and it won't be long before you'll start raking in the profits!

THE FIRST STEP: SITE MAP AND DESIGN

To return to our tried-and-true television metaphor: imagine this.

You're a young producer for NBC, and you've been given the task of creating a new half-hour drama program for prime time. You're alone in your office, all set to make your first crucial notes that will become the finished program. So, before you even think of some basic structural components--what's the concept behind the show? What characters will I feature? How will I fill thirty minutes every week, and keep viewers coming back? --you take out a blank piece of paper and start doodling costume designs for your Christmas special.

You create a great design, throw some story and characters around it, and start filming your first episode. On the appointed day, the viewers tune in, hang around for two minutes, and then tune out. Your costume designs are just as great as you intended, yes--but by choosing the wrong starting point, you couldn’t control every aspect of the viewer's experience of your show--and they respond by leaving your network, never to return.

This kind of approach is problematic for any kind of creative work, but with enough work put into a project from any starting point, you can achieve some success. But when you're talking about a programming project--and all websites are, essentially, programming projects--choosing the wrong starting point is disastrous. If you start work on your website by firing up an HTML editor and blindly layout out pages, you'll be rewarded with an inconsistent, buggy, and hard-to-use website--which violates one of our most important rules for site design.

In order to follow those rules--and thus to achieve success with your online business--start at the right place: by creating a workable site map. A site map is just what it sounds like: a diagram showing exactly what content your website has, and how that content connects to other content through hyperlinks.

In order to understand how to build one, let's look at the simplest possible example of a site map: the classic "Hello, world" program used by beginning programmers worldwide. "Hello, world" programs consist of a single screen with a single message: "Hello, world!" The site map for this would be a single page with a single piece of content on it, saying exactly this.

A more complicated site map might involve creating two pages: a "Hello, world!" index page, followed by a page with other information. The site map for this would be the same single page with "Hello, world!" and a link to the next page, followed by another page with other content. A mark should be made somewhere designating the "Hello, world!" page as the usual entry point to this bare-bones website.

You don't need any technology fancier than a pen and paper to make a site map: just draw out the pages you want, determine and note what should go on each page (including text, images, links, and tools for ordering your product), and connect the pages with lines to show how your viewers will navigate your site. If you want to get fancier with your design, there are also cheap (or even free) programs for mapping out web sites before you build the pages. If you're doing your own coding, investigate this option: some of the most popular web authoring clients integrate site mapping functions into their page editing software, which can save you money and reduce the amount of time it takes to learn how to use a new piece of software.

Site mapping may seem like an unnecessary step in the design process, especially with the simple sites that direct response marketing demands. But without a fully-detailed site map, you run the risk of increasing your costs beyond your original budget when you find out that the page you've designed doesn't work properly, or that you need some additional content or images but aren't sure where to put it. At best, this leads to costly revisions (in terms of time if you're doing your own coding, and in terms of money if you're hiring a coder), and at worst it can mean paying for a complete overhaul of your website. Think of your site map as your business plan: write it first, then stick to it unless you have a very good reason to change it once the site goes live.

SITE DESIGN AND THE THREE CLICKS RULE

Before you finalize your site map, let's think in more detail about how a good direct response website should be organized.

Our first rule forbids the use of external links. Therefore, you can eliminate any links pages, sidebars with site affiliates, advertising banners, or such things from your site map. Although selling advertising space on your website may seem like a good idea for building revenue, it distracts your viewers from your central goal: selling the product. So, keep it simple, and leave the advertising off.

Our second rule dictates that content on your site should be limited to only that which is essential for persuading people to buy your product. The policy that goes along with this rule is to consolidate most of your site's information onto as few pages as possible. This reduces the chance that viewers will come to your page, click a link that takes them to another page on your site about your product's features, and then forget to come back to your main page to actually buy the product. Of course, there are situations where you'll want to divide your content among several smaller web pages on your site as opposed to putting everything in one massive index page--if you have a wide variety of technical data about your product, as well as photos and testimonials, you run the risk of boring your viewer long before he gets to the crucial "Buy" link.

So, follow this guideline: if your viewers don't have to scroll down more than one or two times in order to read all of your product information, put everything on a single index page. If you have to scroll down too often to read all of your site content, then split the content into separate pages--but make use of pop-up windows in order to keep your main page open, or make sure that there's a link to your actual "Buy" page in a prominent place on every one of your sub-pages.

Above all, remember our third rule: keep it simple, stupid. One easy guideline for doing this is to follow the three-clicks rule:

- Upon arriving at your site, your viewers should never have to click more than three links in order to buy your product.

One way to implement the three-clicks rule might be this: your viewers start at an index page that describes the product information. They then click a "next" link to take them to a page about prices and ordering information. They then click a link to start ordering the product. That's two clicks in total. Another way to implement the three-clicks rule: your viewers start at an index page that talks in general terms about the product. They click on one of your subpages (features, testimonials, pricing, etc.-whatever best suits your specific product) to learn more about the product. They then click on a "Buy Now" button to learn about pricing, and then they click on a button to begin ordering the product. Three clicks. There are any number of other possible configurations--experiment with your site map until you come up with something that gets across all your information while maintaining this same simplicity. Your customers will thank you-and you'll thank yourself when you see the sales figures.

COMMERCE SYSTEM

The one site feature that you're unlikely to be able to provide on your own is a working system for buying products, sending shipping orders, and transferring customer's payments to your account. Systems like this are extremely complicated to code, difficult to integrate into a page, and above all risky in terms of security. You don't want to take a chance on building your own commerce system from scratch, starting to take customer orders, and then finding out that someone has hacked your system and reduced your business bank account to zero.

You have two options for getting around this problem: hiring a very, very good coder with experience in this area (which we'll talk about more in the next chapter) or going with a proprietary coding system. The former option--hiring an experienced coder and building your system from scratch--is riskier and costlier up front but has some advantages in terms of site design and simplifying your accounting operations. The latter option--using a proprietary system--is safer in terms of site security and more familiar to many users (who tend to use systems like PayPal or Bit Pass for several different types of online purchase), but also comes with a price: hefty transaction fees and possible content restrictions. Which you choose is up to you, but if you think you have an excellent coder and a good system for taking in revenue, go with the former option: you'll lose some money up front, but you'll tend to gain more in the long-term in saved transaction fees.

BELLS AND WHISTLES

Once you have your basic site design, you have a skeleton--and a skeleton is nothing without flesh. We'll talk more about developing your text content in a later chapter, but at this point we'll think in terms of the basic look and feel of your page: images, sounds, and Flash effects that you may want to use in order to enhance the look of your page and build your business's personal identity.

Here are some good rules to follow:

- Use graphics that you already use in promoting your business as a basic template for your site design. This builds a brand identity for you, which can be invaluable in attracting customers and building sales.

- Don't get too complicated. Too many beginning web designers feel the need to show off their skills, drag every HTML/CSS trick they know out in order to make a flashy, impressive page, and end up with an inconsistent mess that drives customers away. Choose a simple look for your page with a few colors and a general feel, and either communicate this to your coder or build it yourself.

- Avoid music on your page (unless you're selling CDs, instruments, or audio equipment.) Music is often irritating to web surfers, makes your page load more slowly, and doesn't add much to your online image.

- Keep load times on your page low. Any images you use should be optimized for web use (a good Photoshop tutorial or an experienced web designer/coder can tell you how to do this), background images should be avoided whenever possible, and complicated scripts running on your page (counters, message boards, or Java effects) should be kept to a minimum or eliminated entirely. Every second your viewers spend waiting for a page to load is a second in which they can easily decide that your product isn't worth the effort--avoid this at all costs.

So now you've mapped out your site with an eye to the rules of direct response design, you've decided on the images your page will use and the basic look of your site, and you've made some preliminary steps to deciding on what commerce system you'll use.

At this point, it's time to start laying bricks, as the builders say. And even if you plan to hire a coder to build your site for you--as many people choose to--it's a good idea to know a little bit about what materials your page will depend on. In the next chapter, we'll learn just that.

4

DIY SITE CODING

For some of you, this will be the best chapter in this book. For others of you, this will be a temporary detour into an interesting world, but not one that you'll need to know everything about in order to be successful with direct response websites.

For both groups, much of this information will be helpful. If you know something about how web pages work, then you'll know what to ask for--and what not to ask for--when it comes to hiring a coder to build your website. If you want something from your page that standard HTML can't deliver, you may find yourself facing extra charges for CSS or even Flash design--or you may find yourself striving to build a page that simply can't be built with current technology. And if you're not familiar with exactly how computers interpret and display website code, then you may run into costly revisions down the road when customers complain that your website doesn't work with their browser--as well as lost sales.

So, if you're interested in building your direct response site yourself, this is the place to start. If you're more interested in saving yourself some effort and hiring a coder, familiarize yourself with a few of these terms before moving on to the next chapter, where we talk in detail about how to find, retain, and communicate with your coder in order to build the best page possible.

HOW WEB PAGES WORK

All web pages and websites on the Internet exist fundamentally as text files saved with the HTML extension. Web browsers read these text files, decided based on them (and based on HTML defaults) which formatting choices to use on a page, generate all content from the text file and link to all images, and ultimately display the page.

Of course, it's impossible to talk to a web browser in natural language: saying in plain language to "make the margins one inch on all sides" is easy to understand for a human layout editor, but impossible for a computer. Which is why--as we touched on in the first chapter--standard protocols for online systems were a necessary condition for the growth of the Internet. For two computers to talk to one another--and in order to ensure that they're talking in the way that the user intends--it's necessary to speak a common language, which is essentially what HTML and other internet protocols are.

HTML (or "Hypertext Markup Language") is the most successful online formatting protocol yet devised and should be the basis for any good website. It's also a simple language to learn (as opposed to object-oriented languages like Java or C#, which require a much greater working knowledge of computers and a much greater willingness to spend time organizing a program.) HTML is essentially a formatting guideline rather than a true programming language, which explains some of its versatility and ease of use.

So, in order to learn DIY coding (or to learn how to talk intelligently to your web designer), we first have to learn some key features of HTML.

KEY HTML TAGS

When you view an HTML file in a web browser, the computer follows this logic to interpret your file:

Files are read in order from beginning to end.
If the file contains a tag, change the formatting rules to reflect that tag until the tag is closed.
If the file doesn't contain a tag, format the text according to HTML standards and browser options (most commonly twelve-point Times
New Roman.)

A "tag" is just a formatting instruction to your computer. All tags look like this:

<tag> </tag>

Where <tag> tells the computer to start formatting according to the tag, and </tag> tells it to stop formatting according to the tag. There's a little bit more to it, of course (and a good HTML tutorial can help you with some of the finer points of this), but essentially this is all that tags are: formatting instructions. Think of an HTML document as a document in a foreign language, with your browser as the translator.

The most important tag in any HTML file is the <html> tag. The first line of any .html file is always <html>, and the last line is always </html>. This tells the computer that any information between these tags should be interpreted according to the standard HTML rules, as coded into the web browser.

The second most important tag in any HTML file is the <body> tag. Any information within the <body> tag will be displayed within the browser window when the web page is loaded. The critical tag that complements the <body> tag is the <head> tag, which determines information that the browser should know about the content of the page--most commonly the <title> tag to determine the information to be displayed in the title bar of the browser window, and the <meta> tag to give information on the page's content to search engines. (A full discussion of the <meta> tag is well beyond the scope of this book but will be dealt with in a bit more detail in the marketing chapter.)

The most commonly seen tag in most HTML files is the <p> tag (which appears only in the <body> section of the file, as will all the tags below.) The <p> tag tells the browser to start a new paragraph of text, including a full line break between paragraphs. Within the <p> tag, it's possible to place other formatting tags, including <b>, <i>, and <u>--which declare bold text, italic text, and underlined text, respectively. (If you're not doing your own coding, just know that you can use tags to modify the appearance of text in order to emphasize certain key words in your sales letter or other content--often a useful marketing tactic.)

The <image> tag is used to place an image into a document. The <image> tag contains within it the web address of the image to be placed (for example, http://yourbusiness.com/yourgraphic.jpg), as well as optional rules for resizing the image to whatever height and weight you want (allowing you to include even large images like photographs on your main page.)

The <a> tag is placed around an element on the page--usually either text or an image--to turn that element into a hyperlink to another section of the page. This is most commonly used to connect pages to one another or to allow you to view larger photos at their full size--you might resize a 1200 x 900-pixel photo to 120 x 90 pixels on your main page, but at the same time turn that picture into a link. Customers can then click on the picture, which will allow them to view the picture at full size in a new window.

The <table> tag is used for displaying text or other page elements in tables on your page. A common use of the <table> tag is also to control the layout of your page: if the table borders are set to zero (consult an HTML reference to find out how to do this, if you're going the DIY route), you can create the illusion of small

text "fields" throughout your page--think of a newspaper with various columns, photos, and advertisements laid out on a single page
(with the <table> trick) versus a single page of a book (without the
<table> trick.)

And finally--but perhaps most importantly, if you're building your own commerce system or if you're trying to include user feedback and testimonials on your site--there's the <form> tag. This allows users to input text on your page, click on buttons, or select items from a dropdown list--an ideal way to keep your page simple while still giving your customers the maximum possible options. (The <form> tag can be extremely tricky for HTML novices and can also create some security problems on your page if used improperly--be sure to consult a good reference and to experiment before you start using this tag on your main page.)

TAGS IN DIFFERENT BROWSERS

To return to the "translator" metaphor: not all translators are equally good at what they do. Any fan of world literature knows that the same basic text can come out in wildly different ways when translated by different people--and the same is just as true for browsers.

A few tags in particular can create problems if your site is designed for one browser but viewed by a potential customer in another browser altogether. The <table> tag is notorious for these kinds of problems, and while the <form> tag will usually behave in the same way across different browsers, it may have a radically different appearance for different users--and not always one that you like.

There are two solutions to this problem. One is simply to test your page in as many browsers, screen resolutions, and operating systems as you possibly can, and tinkering with your basic HTML code to the point where it looks acceptable (and works in the way you intend) for most of these. The other solution is to choose one or two popular browsers to focus on, optimize your website for these browsers, and then include a prominent disclaimer on all the pages on your site: "Optimized for Browsers X and Y, versions a and b." This might cost you some traffic, but it isn't likely to be a significant drain on your potential revenue.

CSS

A recent development in the field of web design is the popularity of CSS formatting. CSS stands for "Cascading Style Sheets" and can be thought of as a series of rules for the browser to follow when interpreting HTML.

For example: the basic HTML tag <a> always formats hyperlinks in the same way--blue underlined Times New Roman text. You can manually change the font, color, and size of this text every time you use the <a> tag in your files, yes, but this gets timeconsuming and is limited in its possible effects. CSS files, by contrast, can change the definition of the <a> tag for your browser. For example, you might write a CSS sheet that defines the <a> tag as invariably green, Arial, bold, and without the standard underline. With a little bit more effort, you might even define rules for the <a> tag that allow it to change color or become highlighted whenever the user's mouse moves over the hyperlink-a cool little feature that, if done well, can add to the look and feel of your page without costing you any effort or violating any of our three basic rules.

CSS programming isn't as difficult as hard-core programming languages like Java, true, but it's still substantially less forgiving than HTML, and thus requires a much greater investment of time to learn and use effectively. If you want to experiment with CSS, be sure to give yourself some time to learn the syntax of the language, work out your page implementation, and be sure that you don't go overboard and make your design either confusing or unattractive. You might also do some web searches for CSS layouts--there are plenty of CSS formatting packages available for free online, including professional suites by design studios (released for promotional purposes)--and of course you can always look at other people's CSS files and get some ideas about how to implement your own, as long as you don't directly steal someone's hard coding work.

PLAIN TEXT HTML VS. AUTHORING TOOLS

Building a web page from the raw HTML file up can be deeply rewarding as an intellectual challenge, is certainly the most costeffective solution, and gives you total control over the look and feel of your page. But it can also be extremely time-consuming--not only to build, but to maintain. If you want to radically revise your product information at some point, you'll have to go back in and change lots of very specific HTML formatting in your raw text files-not the most effective option, certainly.

An alternative to text authoring is to invest in an actual HTML authoring tool like Dreamweaver or Frontpage. These have significant drawbacks, of course, not least of which are the very high price and the lack of total control over your website. But they do allow you to change your text through copying and pasting without changing the basic formatting or underlying HTML code for every single HTML file you use, and they simplify some of the more complex layout coding that you might choose to use to make your page look more stylish and unique.

Some HTML authoring tools actually give you the best of both worlds--Dreamweaver, for example, lets you edit the basic HTML coding of a page at the same time as you edit the page text, images, and layout directly (without manually changing HTML tags), which can satisfy both the desire to work quickly and efficiently and the desire to have total control over a page's layout. Any tool, however, is only as good as the person who uses it, so just use the tools that you prefer and that work the best for your coding style.

If this sounds complicated, it's because DIY coding is complicated. There are definite benefits to it, of course--but for many business owners, those benefits aren't worth the time invested in learning the language. And even if you know HTML and CSS backwards and forwards, you might believe that your basic design skills can't give you exactly what you want for your company's website. If this describes you, then move directly along to our next chapter, where we talk in detail about just how to find, evaluate, and hire a coder or designer who can help your website--and your product along with it--stand out from all the rest.

5

FINDING A CODER

In the last chapter, we gave you a great deal of information about the basics of HTML coding: browser theory, tag management, browser interoperability, and the more advanced coding world of CSS. In fact, we gave you so much information that some of you may have decided, on the strength of the last chapter alone, to give up on coding and look for an outside coder instead. That's perfectly fine--and even those of you who have substantial HTML/CSS experience might be interested in finding a coder in order to save time, or to give a website a certain look or feel of which you aren't yet technically capable. So, let's get started on learning how to find the perfect coder for you.

WHAT TO LOOK FOR

The most important quality to look for in a coder is experience. This is true of almost any job, yes, but it holds especially true for any job that involves producing goods, rather than simply providing services--and websites are actual goods. And as valuable as experience is for, say, a carpenter, the value of experience is at least doubled for programmers--if a carpenter builds a bookcase incorrectly, the potential damage is limited to your books falling over, while if your website coder builds the website incorrectly, you stand to lose massive business, compromise the security of your commerce system, or even exposure your web server--and thousands of computers besides--to nasty viruses or worse. If you look for a coder based on credentials alone--a degree from a design institute, for example, or a certification course--then you ensure that you have someone who at least knows about all the concepts that website design and coding requires--but you can't be sure that your coder will know what to do in a real-world situation, where things can and do go wrong as often as possible. So, if at all possible, get someone who's done HTML work before and save yourself some problems down the road.

It's also important to choose your coder based on the kind of experience they have. You don't necessarily have to choose a coder who's done direct response site work before--if you followed our instructions in chapter three, your site map should be good enough to communicate to your coder exactly what your site needs to maximize your conversion rate--but it can be helpful, especially if your coder has resolved some direct response site problems before and can help your site avoid similar problems. What you do need is a coder with substantial HTML experience, especially experience with sites that involve integrated commerce systems. Buyers will forgive a few lapses in the composition of your page but lapses in your commerce system interfere with the simplicity of the buying process, potentially compromise buyer security (and the security of your own accounts), and give your site an extremely bad name--and thus a low conversion rate.

Depending on how you've designed your site and site map, you may want to hire a coder based on CSS, Flash, or Java experience as well. If your site design calls for effects that only these types of coding can provide (see the previous chapter for a quick discussion of what CSS can do that HTML can't), you'll certainly want a coder who can bring those skills to your project. Even if your site doesn't explicitly call for these skills, consider hiring a designer with experience in this area anyway--you don't know how you might choose to revise your page in the future, and if you do decide to incorporate some advanced styling or effects in a later iteration of your page, you'll want to use the same coder in order to ensure a greater familiarity with the material and an idea about the specific problems that may arise from implementing more complicated layers of coding.

What's more, a good CSS/Flash/Java coder can suggest ideas for improving your website that you may not yourself have thought of. For example, you might be using a proprietary commerce system in order to handle your online orders--until you meet a coder who can create a more secure, easier-to-use commerce system with no service charges and that goes with your overall site design. A good CSS coder can also overhaul the look of your page with comparatively little effort, bringing in more paying customers and usually giving them a better experience of your page.

Whatever qualifications your coder has, it's always a good idea to check out all of the sites they've worked on, as well as any HTML examples they include in their portfolio. Look at the kind of work your prospective coder does: are there any special layout tricks that they tend to use over and over? Any stylistic choices that you just don't like? Does the site work well, or is it confusing to navigate or use?

Don't overthink this part of the hiring process: if you navigate your potential coder's portfolio on instinct, you can get a closer approximation of how your eventual customers will use sites designed by that coder. And unless the site is simply a mess (either functionally, stylistically, or in terms of navigation), don't necessarily take the coder out of consideration: what you might perceive to be a personal lack of taste or foresight could easily be a result of bad decisions taken on the part of this coder's former clients. As long as you're providing the basic site map and stylistic ideas--and as long as the coder obviously knows what he or she is doing on some level--you can avoid the pitfalls your competitors might have made and ensure that your site will be successful and striking.

Although it's often overlooked, there's one factor you should take into account when choosing a coder: his or her personality. This isn't as much of a consideration for a short-term website project (like the average direct response website) --but depending on your business plan, a coder with a good personality can be an asset in the long run. As we'll talk about in a later chapter, you're likely at some point to want to do some revision on your site--whether to add new products, resolve some functionality issues, or just to give the site a nice graphical overhaul. And it's far easier to make these kinds of changes if you know and trust your coder already--easier on you, since you don't have to go to the trouble of searching out and hiring a new coder, and easier on the coder, since they already know your basic business plan, site needs, and preferences--and they also know their own code well enough to start working on your revisions immediately without having to spend a great deal of time familiarizing themselves with someone else's work.

WHERE TO FIND CODERS

So now you know what to look for in a coder--but a much greater problem for many people is the problem of where to find the perfect coder for your project. It isn't simply a matter of posting an advertisement in a local classified section and waiting for responses--that might get you some potential candidates, but it removes one of your best tools for assessing the suitability of a coder: the portfolio.

One useful method is to post your project on freelancing programming sites, one of the most prominent of which is rentacoder.com. Rentacoder.com allows software buyers--such as yourself--to post details of your project on their directory of projects, along with some idea as to the rate you're willing to offer. Coders can then bid on your project, giving you portfolio examples, any certifications they may have, and their ideal rate for the work. Once you've checked out what they can do, you can approve their bid, place your payment in escrow, and just wait for the coding work to be done. As soon as the coder sends you the work (and as soon as you approve it), the money is released to the coder, and you can both go on your ways--your coder with his cash and experience, and you with your functioning direct response website.

There are a number of advantages to this method. Most importantly, there's the wide talent pool from which to choose--just as putting your business on the Internet gives your product a much wider potential audience than you could achieve through traditional channels, looking for contract employees (like coders) over the Internet gives you a much wider selection and a much greater chance of finding someone with the perfect skills for your job. Additionally, services like rentacoder.com greatly simplify the process of interviewing potential coders and determining prices: most of the things that you need to know about a coder (namely, their skills and their price) is available at the rental site, just waiting for you to sort through the options and make your decision.

But it's important to keep in mind some of the disadvantages to services like rentacoder.com as well. For one, it's very difficult to get a good sense about a potential coder's personality from their rentacoder.com profile or even their skills set. Again, this isn't a problem for short-term work, but as we've discussed, the ideal relationship with a coder is a long-term relationship. Not only does rentacoder.com make it more difficult to create such a relationship by masking coder personalities, but it also makes it more complicated to hire a coder on an ongoing basis. Rentacoder.com only allows you to bid on a coder for a single project--e.g. a single website--with no simple provision for providing ongoing work.

Fortunately, these obstacles aren't insurmountable--it just requires more work on your part in order to build and nurture a relationship with your coder. Rentacoder.com automatically releases personal details like phone number and email address for all projects above $500, allowing you to contact your coder directly--once the project is in motion, unfortunately. Before the hiring process--or if your design work costs less than $500, which it shouldn't (see below)-you can post messages to your coder on the rentacoder.com message boards or chat rooms, which is the ideal way to contact them--and there's nothing that says you can't ask for a phone number or email address in such communications to help you make the informed decision before renting a coder.

Are there other options? Of course--there's always classified advertising (on or offline), there are personal references from other business contacts, there's the possibility of emailing the designers of high-profile sites that you've seen and liked, and there's the ubiquitous Craigslist posting. But if you're willing to put in the effort to build a personal relationship with your coder (and to assess their personality and skills before hiring), sites like rentacoder.com simply offer too much variety and too much talent to ignore altogether.

WHAT TO SPEND

Web design isn't a cheap proposition. Although it's certainly possible to find someone to design your entire site for around $150, the adage in this case holds true--you get what you pay for-and you're unlikely to get a high level of commitment or talent for those rates. Alternatively, it's certainly possible to find someone to design your entire site for around $3,000--but again, although you're nearly assured a high-quality site, it's difficult for many start-up businesses--or even established businesses--to drop that kind of money on a single project.

Expect to pay anywhere from $1,000 to $2,000 for a good direct response site--the more features you want on your site, the higher the price. If you want a good commerce system programmed to order, expect to pay at least $2,000 or higher--one strong argument for using a proprietary system. This may seem high but consider what you're getting for the money. Web designers with enough experience to be valuable to you make their entire living from their designs, as a rule. If you underpay them, then they're forced to take on other work in order to make a basic living--which means time taken away from your site and a lack of willingness to work with you in the future. This undermines many of the basic objectives of your business and can easily lead to your spending much more money down the line to fix any problems that are created by underpaying your programmers. So, do it right the first time and pay for the level of quality you want to get (within reason, of course, and within your budget.)

Once you have a coder, you've made a huge step toward getting your site ready to go live. In order to go the rest of the way, keep reading as we talk about the next phase of the process--working with your coder in order to ensure the best site possible.

6

CODER COLLABORATIONS

If you've followed our advice so far, you've got a good site map and site design in your head, you've got an excellent coder at a reasonable price (for both your coder and for you), and of course you've got an outstanding product to sell. At this point, it's time to put all of these elements together to create a revenue-generating site that'll turn a profit. But in order to do that, you'll have to master one crucial skill--collaborating with your coder in order to create the perfect website for your purposes.

THE BASICS OF COLLABORATION

The first step in a fruitful coder collaboration is always to make your needs known as specifically as possible in the first communication. Ideally, your coder will be familiar with the basic principles of direct response site design, which keeps them on the same page as you throughout the process and which eliminates some of the initial effort of explaining yourself. If your coder isn't familiar with the basics of direct response site design, make sure to communicate it to them as soon as possible in the collaboration process. The same three rules we gave you earlier in this book will be perfectly fine for your coder as if you back it up with all the rest of the work you've been putting into your design ideas as well.

Of immense help in the collaboration process will be your basic site map. This gives your coder the basic template in which he or she can start bringing your more complicated ideas to life, as well as a sense--a not very fleshed out sense, true, but a sense--of what the finished product will look like, as well as how it'll behave. Don't think of it as condescending to give a site map to your coder, either--it isn't, and he or she is likely to thank you for it. It makes your coder's job infinitely easier, speeds up the overall process of building and revising the site--and saves you from any unwanted additional charges down the line.

Once your designer has the basics of direct response design down, and once you've given him or her a copy of the all-important site map, it's time to start thinking more specifically in terms of overall style and feel. Here's where the process of collaboration can start to resemble a true collaboration, rather than you simply telling your coder exactly what you want and expecting delivery.

There are two ways to play this--or rather, there are two extremes you can take when working with your coder on the overall look and feel of a page:

The coder makes the page look exactly like you want it to look. The coder, using his or her individual design skills and abilities, designs the look of your page for you from the ground up.

The best method of collaboration lies somewhere between those two extremes.

To some, the first extreme is the obvious choice. You know the product, after all, you know the principles of direct response site design, you've built the site map, and you've put a great deal of work into designing the site in order to take full advantage of the basic principles while giving customers easy access to all portions of your website. Any work the coder puts into your page design at this point can only serve to complicate things: it can make the site much more difficult for customers to use, for example, or it can put an undue amount of strain on your web server or overall bandwidth, or it simply won't look right to you. After all, the coder is your employee: you're paying him or her to produce a website for you. Since you're putting out the money, you should expect to get exactly what you've designed--exactly what you want.

To others, the second extreme is the obvious choice. Yes, you know the product, and yes, you designed the site map, but you're not at all sure that you know how that site map should be fleshed out. You have ideas, of course, but your coder is a professional in the field of web design, has plenty of experience designing other sites (if you followed our advice in the previous chapter about finding a coder), and brings his or her own distinct personality and visual style to the product.

The work the coder puts into your page design may solve some of the problems that your original design ideas didn't anticipate: that work might improve the efficiency of your site's load time, it might give customers additional unobtrusive methods for getting to all of the important content on your sight, or it might simply dazzle you and your customers with its innovative look and feel. The coder is your employee, yes: but part of what you want when you pay that employee is their instinct and expertise, not merely their ability to follow your orders. You expect to get the basic site you've designed, yes--but you also expect to get something that exceeds your expectations, something they could only come from your specific coder.

The best method for your site and your particular coder is probably somewhere in the middle. Talk to the coder about your visual and other design ideas--give them any sketches, notes, or other work you've put into the site. Explain to them some of your general concepts and talk about what you absolutely don't want to see on the site for whatever reason. Then--once you've communicated some of the specific things you want and don't want on your site-let your coder loose on the project.

The coder will feel better about the project: you're not leaving every decision up to them (which can paralyze a less-experienced coder, or which can result in a site that you as the client absolutely hate), but you're still giving them a good measure of autonomy in working on your site (which engages their creativity, encourages them to seek out and solve any design problems in their own personal style--and encourages them to work with you again on any future projects or site expansions that may be necessary in the future.)

Of course, you shouldn't cut your coder completely loose once they have your basic instructions and design ideas. The easiest way to guarantee success with a page design is to maintain a regular schedule of communications with your coder throughout the project in order to ensure that he or she isn't wasting time implementing features that you ultimately don't want to use in the site--and to ensure that your coder is staying on task throughout the project.

What's the right schedule to keep for regular communications with your coder? This depends on the overall size of the project. Any direct response website is going to require less work than a sprawling site full of image galleries, articles, or content nodes. For the average direct response site, expect to give your coder about a week to build a first draft of the site, and schedule another meeting either one week from the start of the project or a day or two after the first draft of the site is turned in. (If you choose this latter option, make sure to schedule a meeting one week from the start of the project whether the first draft of the site is turned in or not-you don't want your coder to stall indefinitely on the site, costing you time and money, and a reasonable but firm deadline for a progress report guarantees that you won't be wasting that time or money.) Keep up that regular schedule of contact until you and your coder both arrive at a version of the site that you can be happy with.

And above all: make sure your coder has access to all the resources he or she needs in order to complete the site. This means giving them any photos they need, any product specifications or testimonials you want to include on the site, details on the commerce system you'll be using, and of course the all-important sales letter (which we'll talk about in detail in the next chapter.) Ideally you should get this to your coder immediately upon hiring them and holding your first design meeting with them. If that simply isn't an option for whatever reason, set a specific date when you'll provide the information and hold to it.

Under no circumstances should you expect the coder to design your entire page without having access to these crucial resources, even if you know enough HTML to plug them in yourself once they're available. Minor differences in the size of an image, the length of a sales letter, or even the commerce system you intend to use can result in hours of work on the part of your coder as he or she tries to adjust the site design to accommodate your photos and resources--hours of work that you'll end up paying for. So, either get your resources to your coder as soon as possible, or simply delay hiring your coder until you have all of your resources available. There's no point in asking someone to build a website for you if you can't give them all of the tools, they need to build it.

Additional CHARGES: HOW TO HANDLE THEM

Don't be afraid of additional charges. (Don't be totally comfortable with them, either--see the next section.) As good as your basic site map and strategy are, they won't always be able to stand up against the various problems that occur when translating a good idea into a reality. And when those problems occur, it'll cost you and your coder both money and time in order to correct them.

Additional charges can crop up for any number of reasons, but always crop up from only one of two sources: you or your coder. You might realize at some point in the design process that there's a better way to organize your basic site map, that there's a certain angle for selling your product that you'd like to incorporate into your website, or that your original design ideas pale in comparison to what you've just come up with. Your coder, on the other hand, might build your site exactly to specifications, test it out, and find out that there's a fundamental problem with your collective solution to the three basic design problems of direct response site design. The interface might not be wholly intuitive, for example, or your commerce system might not be integrating with the page properly in order to give customers a seamless ordering experience.

You have two tasks when responding to additional charges:

- Are they necessary?
- How much are they worth?

The first task requires you to look at the website as it stands and determine whether your customers' experience will be negatively impacted by whatever problem you or your coder has come up with. This isn't just a matter of determining whether the site works or doesn't work--it's also a creative problem of figuring out if you can make the site work without requiring a great deal of additional effort on the part of your coder. If your basic site map is too complicated, is there a way to untangle it by moving around existing HTML files and changing a few links, or is it necessary to rebuild the site from scratch? If your user interface is complicated, is there a way to scale it back--from Flash to HTML, say--without requiring a great deal of additional coding and testing? If your commerce system is buggy, can you find another one to drop in without requiring a total redesign? These kinds of issues will have to be dealt with on a case-by-case basis according to your specific site and product.

Once you've determined that you have a legitimate reason to authorize additional charges, you'll have to determine just what to charge. Listen to your coder's advice, balance that against a list of standard design fees and the total cost of your project and approve whatever your budget is capable of approving (with an eye to any more unforeseen charges that may come up.)

Whatever you decide on, make sure that you don't simply allow your coder to act like an auto mechanic--make the repairs first, bill you for it later. This is another reason why it's essential to maintain regular contact with your coder while the site is being built. Make it clear that if your coder encounters any potential problems, he or she is to report it to you first before making any changes. This limits the coder's autonomy to some extent--but it also saves you a great deal of money, which is usually worth the slight time delay that approving each change will take.

WHEN IS THE PROCESS COMPLETE?

The simple answer to the problem of when you can consider your site complete: when it's done.

The more complicated answer depends on one of two things happening:

- You have the site you want.
- You can't afford to pay the coder for any more revisions.

Unless your site simply doesn't work--meaning that it actually crashes browsers when it loads, or that the individual pages don't link together properly--don't be afraid to accept the second outcome. Remember our TV analogy: the goal of a direct response web site isn't to be beautiful, but to sell a product. Yes, making your site beautiful, intuitive, and efficient can help with this ultimate goal--but it's more helpful to have a website that works and does its basic job.

And remember that you're running a business, not an art museum. The basic rule of business is not to let your costs exceed your profits--and until your site has been up for a while and has plenty of referrer links and other promotion, your profits aren't going to be massive, and certainly not high enough for you and your coder to spend infinite time picking at the tiny flaws in your website. If you exceed or come close to exceeding your design budget and you have a site that functions, contact your coder, pay what you owe, and call an end to the process.

But make sure to leave the door open--once you're making some money from the site, you may want to go back to that same coder in order to finally get your site to the level of perfection that you want--or to expand your site into something else altogether. Just because it doesn't make good business sense to refine your website now doesn't mean that you won't want to refine it in the future-and possibly a nearer future than you think, if your product and your overall strategy are where they should be.

So far, we've covered the basic principles of direct response site design, we've covered how to build your own basic site, and we've covered the process of hiring and working with a coder to make your site all it should be. But there's one crucial element of a good direct response site that we've left out: the sales letter. In the next chapter, we'll discuss just that: the final piece of the direct response puzzle.

7

THE SALES LETTER

The sales letter is at the heart of any good direct response website. As we touched on in the chapter about basic site design, your website always needs a good front page--and your sales letter should always be the bulk of that front page. Without a good, prominent sales letter, you may have the best website on the Internet--and if you follow our advice in the previous chapters, you'll certainly have a good shot at making that claim--but you won't be able to effectively convert website traffic into actual customers. And that fails the basic purpose of direct response site design: selling the product. So, make sure that your sales letter is prominent--and using the techniques and theories in this chapter, make sure that it's good.

SALES LETTER FORMAT

Think about what a sales letter does. Your sales letter is responsible for three things:

Introducing the customer to your product.
Arguing persuasively about the merits of that product.
Inspiring the customer to purchase the product.

This seems like a simple structure, but it requires you to use three distinct modes of writing throughout, and to seamlessly transition from one into the other. It also requires you to do this in a fairly brief span of time (about which we'll talk more in just a moment.)

The first section of the sales letter is your introduction. Within the introduction, you want to state clearly exactly what your product is. This is not the place to get persuasive about your product: this is simply the place to describe its essential nature so that any customer knows, right up front, what it is that you're selling. A new type of corkscrew should be described, right up front, as a new type of corkscrew. A new online fantasy novel should be described, right up front, as a new online fantasy novel.

Our third basic principle of direct response site design holds true here: keep it simple, stupid. Online customers have millions of other websites that they could be visiting, and if your sales letter starts off too fancily, obscurely, or densely, they're going to leave your site for one of those other millions. So, don't try to dazzle your readers by going into the long history of woodcarving in order to promote your handcrafted wind chimes, and don't sermonize about the history of art and information in order to promote your web design software package. Your first line should always read something like this: "Foo is a new type of Widget from the brilliant designers at Acme"--where Foo is the product, Widget is its description, and Acme is your company's name. Immediately your customer knows what you're selling, how that product might fit into their life, and who's doing the selling. You've laid your cards on the table--and with this level of simplicity, your customers will be much more willing to pick them up and play.

The second part of your sales letter--the argument--is where you can start getting fancy, introducing some dazzle into your presentation of your product. Your customers know what you're selling: now they're waiting for you to sell it to them. Use whatever tools are appropriate to your product. For a mechanical gadget or appliance, you might talk about how your product's specifications outperform many of the leading brands in your market.

For a piece of art, you might talk about the high level of training and craftsmanship on the part of the artist, or about the prestige and delight that a piece of well-made art can give. For a piece of software, you might discuss the benefits your product gives in terms of compatibility, efficiency, and usefulness--all while emphasizing your cutting-edge technology and your company's history of innovation in software marketing. Anything goes, as long as it's persuasive--and as long as you don't get too long-winded at any point. You want to convince your readers, not to lecture them-and you certainly don't want to bore them into leaving your site.

Once you've said your piece, it's time to move smoothly into the third and final part of your sales letter: the conclusion, which turns your reader from a passive admirer of your product into an active consumer of that product. Like the first part of your letter, don't get fancy with this. A simple call to action will do: "Don't wait. Try Foo today by clicking here." Should you use the imperative voice: instead of telling your customer that "You can click here to try Foo", command them to "click here in order to download/order/whatever."

That switch from the descriptive to the imperative--from telling to commanding--is often all that's necessary to decide the issue in the mind of an undecided customer. You can--and should--dress it up a bit, of course: you might close with a pithy line, reiterate your product's slogan or motto, or simply fall back on a standard closer like "Try it today!" If you don't bore your reader or spend too much time distracting them from the business of clicking on your "purchase" link, anything goes. (You might also direct them to the other sections of your website in order to learn more, if it's appropriate to your product--if you followed our earlier advice and made it simple to order the product from any point on your site, of course.)

Sound simple? It is, and it isn't. As long as your sales letter takes into account these three basic points, it'll be somewhat effective-but if you can not only give your customers persuasive arguments but interesting expressions, if you can seamlessly transition from one section of your sales letter to the next without alerting your customers, and if you can use your language to activate your reader's emotions without making them aware of it, then you'll delight and persuade your readers still more--and they'll respond by clicking on your "purchase" link more. So, it's worth taking some time to make your sales letter all it can be--or it's worth spending some money on a good marketing copywriter who can do the same. Your sales letter is the heart of your site, after all--make sure it beats.

FORMATTING ISSUES FOR WEBSITES

A general rule of sales letters is to keep them as short as you can. But of course, we aren't dealing with general sales letters--we're dealing with a very specific type of sales letter, one designed to be viewed and read online. And that brings with it some additional rules in order to ensure that your sales letter not only works at converting readers into customers, but that it functions within your website while considering all of the possibilities available on the Internet that traditional sales letters aren't capable of.

One of these advantages is the ability to delegate parts of your content to different parts of your site. In traditional commerce, your sales letter is your one chance to tell potential customers everything they might need to know about your product. One of the advantages of the Internet, however, is that you have an entire website's worth of space to convince people of the merits of your product by whatever means necessary. (Bearing in mind the limits we imposed in the chapter on basic site design, of course.) But that greater freedom imposes a greater responsibility on you to decide what you should include in your sales letter and what you should leave out. A lengthy testimonial from a satisfied customer might make a crucial difference in making the sale, of course--but do you really want to make all of your customers read through a 600-word product testimonial in order to get to your final call to action? Conversely, you might be able to write several pages' worth of information about the new compression technology in your new digital video playback software--but shouldn't at least some of that go into your sales letter?

A simple rule you can use to resolve these problems is this: make sure that any lengthy (but persuasive) section of your sales argument gets a page to itself. But at the same time, you should at least touch on that information in the sales letter in as brief and as persuasive a manner as possible. For example: one of the selling points of your graphics software package might be its intuitive, attractive user interface. You should show off that interface by including a detailed gallery of screenshots or videos, allowing your customers to see for themselves what your product has to offer. But you should also include in your sales letter a mention of this interface: "What's more, Foo contains one of the most intuitive user interfaces currently on the market: in just sixty seconds, you can learn all you need to know about how it works, giving you the edge you need to create dynamic graphics."

This rule holds true for any product: if you're selling art, mention some of its most prominent admirers in addition to linking to their testimonials. If you're selling soap, give your customers a picture of it in action and mention its lathering power in your letter. This not only reinforces the selling points that you want to make, but it ensures that if your readers just proceed directly from the sales letter to the purchasing decision without taking advantage of the rest of your site, you haven't left out any of the arguments that allow them to make that decision in your favor.

One more practical formatting issue brings back the question of building and coding your website. A long sales letter can bore your readers, yes, but it can also make your carefully designed site look terrible on any browser by spilling the text over your images or out of your frames.

This is going to be a problem that you and your coder will have to work out as part of the overall design and coding process, so make sure to get your first useful draft of your sales letter to your coder early--and make sure that you know how to cut it down, rearrange it, or otherwise allow it to fit comfortably into your site design if the need arises. (And one minor technical point: if you write your sales letter in a word processing program like Word or OpenOffice, you'll have access to formatting options like bold text, italics, different fonts, or even colors.

Be very, very careful about using these, because they often don't translate well from your word processor to your website's HTML file and can in some cases even create additional charges as your coder tries frantically to incorporate your formatting decisions into your site's actual code. Changing the formatting is often a crutch: weak writers change the style of their fonts or text in order to achieve effects that their words alone can't achieve. Save yourself some trouble: keep your text simple and free of formatting tricks. Your coder will thank you--and your sales letter will be better as a result.)

MUST A SALES LETTER BE BORING?

To return to the TV analogy: when many of us think of infomercials, we think of the same basic structure: a desk, a host, a product demonstration, a few arguments in favor of the product, and then ordering information. We tend to think along these lines because most of the infomercials we see never reach beyond this basic, safe format. And when we think of typical sales letters, we tend to think of basic, safe letters: letters that get the job done without necessarily entertaining the reader.

But there have been other infomercials on the airwaves. For example, a popular "adult" infomercial in the 1990s tried to sell a male enhancement cream to customers not by offering a product demonstration--a dicey proposition on any channel or network, considering the product--but by making their infomercial into something entertaining. The infomercial producers hired adult film stars, built sets, and turned what could have been a boring (yet salacious) infomercial into a quiz show, complete with innuendo, double engenders, and genuinely engaging content. The result was a memorable infomercial--which meant a memorable product and increased sales.

Your sales letter can aspire to the same level. If you keep to the three basic parts of any good sales letter--inform, persuade, convert readers into customers--you have infinite freedom in terms of content. You might present your sales letter in comic strip form, for example, or you might write your sales letter in engaging verse. You might write your sales letter in dialogue form, or you might write about a new piece of software as if it had come through a time portal from a technologically advanced future.

Don't bore the reader, of course, by getting too cute with your sales letter--but don't bore them by keeping your sales letter bland, either. What your readers think of your sales letter will be, if you do your job right, what they think of your product. So, if you can pull off a unique, entertaining sales letter--or if you're willing to pay for the services of someone who can--do it. A simple sales letter will get the job done, yes--but an entertaining sales letter will get the same job done better.

Once you have your sales letter, your website, and of course your product, your work is almost done. It's time to take a look at the last (and from the point of view of selling products, the most important) component of your successful direct response website: your commerce system and managing the overall revenue and costs of your site both online and offline.

8

MANAGING COMMERCE

Up to this point, we've always talked mostly about the design and presentation issues of direct response website building--with an eye kept on the bottom line of converting viewers into customers, of course. But now, it's important to remember one vital fact: the business of direct response website design isn't ultimately design, but business. In this chapter, we'll talk about the money side of that business: how to integrate and manage your website's commerce system, how to keep your account books in the black, and most importantly, how to bridge the gap between the online portion of your business and the offline portion--how to translate web sales into a healthy revenue stream, and hopefully healthy profits to go with it.

ONLINE COMMERCE AND OFFLINE BANKING

The basic premise behind online commerce systems is simple. You sign up for the service and the service creates an account for you, much like a bank. You can then add funds to your account by any number of methods: transferring money from your account, charging a credit card, or selling products and receiving payments. In order to get your money offline, you can either use a debit card linked directly to your account on the commerce system, or you can link your online account to an offline bank account, allowing you to transfer money freely between the two accounts--with a necessary time delay, of course. (This model of commerce won't be followed in exactly this way by every commerce system, of course--check the policies of your commerce system to get specific information for you. If you coded your own commerce system, it'll be a matter between you and your bank, of course.)

The basic premise is simple, yes--but in terms of actual implementation, there are some complicated details. Being aware of and working around these details is going to be the key to success for any good online commerce operation--and as we've said, if you're going to sell products online you need a good online commerce operation.

One important issue is the time delay in transferring your money from your online account to your offline account--not a major issue much of the time (the actual delay is anywhere from one to four business days), but at certain critical moments--if you need your money close to the end of the month in order to pay rent on office space or to pay bandwidth costs, for example--it can become a problem. So, make sure that you take into account the time delay when you're scheduling payments or purchases according to your business plan--or make sure that you have a debit card or other payment option from your commerce system, eliminating the time delay altogether.

Another issue (that hopefully won't be an issue in the future) is the problem of online security. We talked about this in terms of your website earlier in the book. But it's also vital to think about it in terms of your commerce system bank account--and your bank account in general. Again, we talked earlier about how commerce systems that you design yourself need to be secure enough to protect you and your customers from hackers, which not only give your business a bad name, but could wipe out your entire balance in minutes. Commerce systems that you design yourself don't have as much of a problem in this regard--usually a large staff of people exists to protect the integrity of online accounts. Depending on your protection policy, however, this can create new problems that you'll need to be aware of.

Certain online commerce systems take extremely harsh measures when dealing with compromised accounts: not only is your account locked and all the balance within it seized, but the balance of any bank account linked to your online account is seized as well. This makes some amount of sense--if hackers can access your online account, they can just as easily access any linked offline accounts through bank transfers. And if you follow up with your commerce system on any identity theft or hacking issues, you can often get the balances returned to you. But following up takes some time, and in some cases--serious identity theft cases--you may not be able to get the money back at all.

So, you'll need to integrate your bank account and your online account in such a way as to protect yourself from disaster--a rare event, of course, but one that can wipe out your business if you aren't careful. The simplest measure is this: set up two offline bank accounts for any online commerce system you're using. Link only one of those bank accounts to your online account. Whenever you receive payment online, transfer it to your linked offline account immediately--and then transfer it from your linked offline account to your other account, which has no direct link to the online part of your business. Even if your account is compromised and your accounts locked or seized, you can keep your money safe so that it can keep your business running. (Better yet--check out your online commerce system and its account integrity policies thoroughly before you sign up and commit your business to it.)

SHIPPING ISSUES

One of the problems that online businesses face is the problem of product shipping. (If your business doesn't deal with physical products, skip to the next section.) It's one thing to sell a product in a store, of course--the customer sees it, looks it over, buys it, and takes it with him. But selling products online introduces a whole new range of concerns: how can you make sure that the product remains undamaged from your warehouse to your customer's door? How will you handle shipping costs? And how will you communicate with your shipping employees (if you have some) whenever you get a new sale to minimize the amount of time between the customer's order and the receipt of the product?

The first concern is more a function of the shipping method you choose than anything else. Choose a reputable carrier and be willing to pay any necessary money for secure shipping if your project is in any way fragile--furniture or hand-blown glassware, for example, need an extra level of security that something like a CD or a new hammer doesn't. Whatever the fragility of your product, you'll always want to pay whatever it takes in order to track your shipments as they make their way to your customers.

The key to online commerce is trust: trust that your product is valuable to customers, and the customers' trust that you'll give them value for their money--and that you'll get the product to them in a timely, secure fashion. It's inevitable that problems with shipping will happen from time to time, and the more successful your business gets--the more products you ship--the more likely it becomes that you'll fall victim to one of these shipping problems. So, pay to track your shipments--make sure that if the worst happens, you can find the product, get it back on track, and maintain that level of trust with your satisfied customer.

Handling shipping costs is another important matter. Many online commerce systems give you the option to automatically add shipping costs onto any order, either by adding a flat fee onto the purchase price of any product or by adding a percentage of the total price of the product. The better option is almost always the first one: that way, you can figure out exactly how much it typically costs to ship a product (with tracking paid for) and charge customers only that amount.

This protects you from charges of "hidden costs", makes sure that you're not losing money on product shipments, and simplifies your ultimate accounting procedures--instead of having to determine what percentage of each transaction is devoted to shipping, you can simply deduct the shipping fee from all products when you enter net revenue into your accounting system. (For overseas orders, make sure that you deal directly with customers on a caseby-case basis--overseas shipping is wildly expensive, and there's simply no way to program even the most complicated commerce system to take into account variable rates to every country on Earth. Make sure that overseas customers know to deal with you directly by mentioning it on your main page before the customer even clicks the purchase link--don't waste their time by making them start the transaction and then cancel it once they realize the prohibitive overseas shipping costs.)

The problem of communicating each new sale to your shipping department is obviously easy if you're the shipping department. If your business is large enough that you have an actual warehouse or shipping employee, however, you'll need to make sure that they're notified as quickly as possible of each new sale as it gets processed.

One solution is to set up an automatic message forwarding system to send all order receipts from your commerce system to your shipping department as soon as they come into your mailbox--or simply to make all order receipts go directly to the shipping department, who are then responsible for forwarding them back to you so that you can do the accounting. There are other methods, and the one you choose will depend on what works best for your specific business organization--but it's crucial to have some method for communicating orders to the right people as soon as they happen. It improves the overall efficiency of your business, makes customers happier, and in the long run allows your business to grow.

SOFTWARE PIRACY AND FIGHTING IT

If your product isn't physical in nature--a new font, a piece of digital art, or a software suite, for example--then shipping issues don't enter into the equation. What does enter into the equation, however, is the problem of piracy.

The great virtue of an information economy is that its products require very little in the way of production costs to distribute--all you have to do is pay the bandwidth costs necessary to send the file to your customers. But the great vice of an information economy is exactly the same thing: files on a computer can be produced so easily that stealing your product is as simple as finding a way to download it without your knowledge. (As anyone involved in the music industry already knows.) This is a problem under any circumstances, of course, but it becomes an exponentially greater problem with time. If one person downloads your product without your knowledge, that isn't a serious issue. If that person then posts your product on a public file server or peer-to-peer network, however, then hundreds of people can download it--which is a very, very serious issue for you.

So you'll need to take measures to prevent piracy. A full discussion of that is outside the scope of this book, of course, but here are a few ideas:

Never provide a direct download link for your product--meaning never link to the product's file on any server. Use redirect techniques to keep people from knowing exactly where your product is on the Internet--and from then disseminating that information. A good coder can help you with this. Only make your product available in an encrypted form, and require customers to enter a code in order to unlock the content. Customers can then download the product freely--you'd just be selling the code to unlock the freely-downloadable file. Require customers to register your product, and digitally "mark" their file once they register it with you. This is basically the same technique used to combat physical theft: putting unique serial numbers on the product. This way, if you start seeing pirated copies of your content on the Internet, you can check the pirated copy against your records, determine who distributed the full version of your product, and take whatever measures are necessary.

TO REINVEST OR NOT TO REINVEST?

The answer to the question of "to reinvest or not to reinvest" is simple: reinvest!

That is, reinvest as much as you can to keep the business profitable for yourself: if your online business is your sole source of income, reinvesting all of your money is not a good idea. But if you're starting your online business in addition to a day job, definitely reinvest as much of the profits as you can into improving your business.

The logic behind this is simple: the more money you put into your business, the faster it can start to grow. If you improve your website server or pay for a new commerce system or redesign, you can improve your customer's experience at your site, which makes them more likely to buy your product and improve your business. If you refine the product itself, then you make it more attractive to customers and boost sales even more. (It may even lead to a new idea or an expanded product line down the road.) Taking the profits for yourself beyond your basic expenses is an attractive option, of course--and usually the reason we go into business in the first place. But putting the money you make back into your business allows you to make more money eventually, to increase your overall profits, and to keep your customers coming back, extending the life of your business significantly and keeping the revenue stream steady. A central tenet of modern capitalism is that if your business isn't growing, it's failing: make sure that your business grows by putting as much money back into it as you can.

Above all, put money into promotion for your site and your product--a topic which we'll cover in more detail in the next chapter.

9

PROMOTING YOUR SITE

By this point, you have your product ready to ship or distribute, you have your site and sales letter online, you've got your commerce system in order--in theory, you've got everything you need in order to start selling products and making some money. The problem, however, is that you can now sell products only in theory. In order to sell your product in practice, you'll have to get people to actually visit your site so that all of your careful design work and sales writing can have their effect and turn visitors into customers. And in order to do that, you'll need to learn how to promote your site.

BASICS OF DIRECT RESPONSE WEBSITE PROMOTION

Promoting any website means promoting the website address. In traditional advertising, you can promote your business perfectly well by promoting its name, products, or business description--as long as you give customers an easy way to get in touch with you, of course. In online advertising, all of those attributes should be included within your website--leaving you only the method of contacting you, the link, to promote. A link is not only your digital "phone number", but with a well-designed website, it's an advertisement in and of itself.

Direct response website promotion still follows this basic rule, but varies it to some extent, pushing it more in the direction of traditional advertising. This is because with direct response marketing, you're not ultimately selling your website to people (from which they can then order any products you have to offer), you're selling the product itself. The website is only a means to an end. And if you simply market your link without any information about the product you're selling, your customers aren't necessarily going to be interested in your product once they get to your site-making the work that your site has to do much harder.

The simplest solution to this is simply to market your link as a short description of your product. A typical link might look like this: http://yoursite.com. People click on the link, go to your site, and your promotion is successful. A direct response link should instead look like this: New Foo From YourCompany at yoursite.com, in natural language. When people click on those words, set up the link to take them to your site. This way, you get two benefits for the price of one: your URL is still being mentioned (raising the possibility that people will remember it and type it in independently in the future), but you're also letting customers know what you're selling before they even get to your site.

If you have the opportunity to give a brief product description as well as your link, take it. These descriptions shouldn't be on the same order of complexity as your website is, of course, but in a sentence or two you can get across to customers the basic nature of your product, a notable feature, and possibly a price (which, if it's low enough, may spark interest all on its own.) This makes your link more attractive to customers, making them more likely to click on it--and once they click on it, if you've done your job right, they'll be infinitely more likely to purchase your product.

DIRECT ADVERTISING

Just as there are two types of website (direct response website and traditional websites), there are two types of website marketing: direct marketing and indirect marketing.

Direct marketing is the simpler form: make your link available to as many people as you can. The easiest way to do this is just to distribute your link to any of your online contacts. This has a few advantages: no one is likely to be upset with you for marketing, and you'll probably get a few sales just on the strength of the personal connection. The disadvantage, however, is massive: no matter how many people you keep in touch with on a daily basis, you always have a much, much larger group of people who could buy your product--if only they knew about it. So relying on your contacts alone is obviously not a good overall strategy.

A more effective option is to post your link on various forums, blogs, on sites. This reaches a large number of people and allows you to target your marketing to the people most likely to buy your product: a graphic design forum, for example, would be a good place to promote your graphics software package. A blog about handcrafted art would be a good place to promote your wood carvings. A website devoted to literary reviews would be a good place to promote your book. Any number of options exist, depending on your product and the people most likely to buy it.

The disadvantage to this type of direct marketing, however, is the problem of reputation. Directly placing your link in as many places as possible may give you a much larger pool of potential customers, but if done improperly, it can easily give you and your business a bad name. At best, this makes it impossible to promote on a single forum. At worst, people will start actively disparaging you, your business, and your product on other forums--giving you a bad name before people even see your link.

The solution to this problem is to follow general rules of good online behavior. This means:

- Don't post your links in places unrelated to your product.

- Don't interrupt forum conversations in progress in order to promote your product.

- Be careful when promoting your product in forums that you don't regularly visit. Read the forum policies and act accordingly.

- Don't react negatively if anyone disparages you or your product after you make your link available. Talk to the person--publicly, if you can--and respond to their criticism.
  This not only negates any resentment that exists toward you, but actually gives you something of a positive reputation as a reasonable person--which makes forum dwellers more likely to buy your product.

One easy workaround to the problems of etiquette is to promote your product passively, simply by participating in a forum or discussion. Most forums or blogs have a "signature" option that's appended to any posts you make on that forum or comments on that blog. Simply include your website link in that signature and comment normally on the forum, referring to your product only when it's appropriate to do so. You're not calling attention to your link by doing this--but you are making it available online, and that's going to result in some site traffic and some purchases.

INDIRECT ADVERTISING

Indirect advertising involves one of the following:

- Getting people to promote your product for you.
- Promoting your product on wide-reaching advertising networks.

This has the advantage of making your link available to even more people than direct marketing can reach, even if you're posting on widely-traveled forums or highly-trafficked blogs. Each of these can reach a good number of people, true--but they're still limited by the number of people who visit those forums or blogs on a regular basis. What's more, the number of blogs or forums you can visit is limited by your own searches and your own ability to think up new places to promote your product. With indirect marketing, you can promote to a far wider audience without having to think about each individual blog, forum, or other venue for promotion: you can do all of that work automatically, or you can get other people to do it for you.

There are many ways to get people to promote your product for you. The obvious way, of course, is just to ask people to do it, or to hire a permanent marketing employee whose job it is to find potential opportunities for promotion and to post your link. A better way, however, might be to offer incentives: a discount on your product, for example, if a certain number of sales can be traced to a customer's marketing efforts.

This means that you might take a loss (or simply break even) on one sale, but it means a guaranteed number of other sales at full price, plus some publicity for your site and company (for future expansions of your site or product line.) You might also offer incentives for bringing people to visit your site with a higher threshold--100 referred visitors would equal one discount, for example. This can be problematic if those visitors don't actually buy the product, of course, but if you've done your work well when building the site, you can convert a good number of those visitors into paying customers and still come out ahead on the deal.

You can also promote your link indirectly by making use of advertising networks. Google Ads is one of the largest networks currently available, and can be doubly advantageous for you in that you get a certain amount of money if people click on your link on top of the money you'll get if those same people then buy your product. Other marketing networks like Project Wonderful can publish your link on a wide variety of sites for a nominal cost, with the cost depending on the average traffic of the site. If you have the money to invest in paying for advertising--and if your site is good enough to convert visitors into customers--then this can be an excellent option for promoting your site.

AFFILIATE MARKETING

Outside of promoting your link, you can still make some headway on the Internet by selling your product directly through affiliate marketing networks. Affiliate networks sell your product directly through their website, taking a small percentage per sale (usually about 8%). The advantage of this, however, is the ability to set commissions for people who market your product for you. If an affiliate's marketing efforts result in a sale, they'll receive a portion of the money from that sale, giving them a strong incentive to market your product in ways that will be useful to you.

The real benefit of affiliate networks is their cost-effectiveness for the business owner: you don't have to pay to publish your link to an extremely wide audience unless you actually make money as a result of your affiliates' efforts. If you don't sell any additional products through affiliate marketing, you don't lose any additional money. Depending on the commission you offer, however, this could be either very useful or very problematic--a 22% commission, for example, means that you'd be paying out 30% of your product's price for every sale and only keeping 70%. (This also means that much of your careful site design work isn't being used.) But used in conjunction with other advertising methods, you can sell your product to a much wider audience, still make a profit, and increase the overall visibility and reputation of your business for any eventual expansion.

We've given you a number of potential strategies for promoting your product in this chapter. Don't think, however, that the key to successful promotion is simply to choose the most effective of these strategies. The key to successful promotion is to use all of these strategies as often as possible: to distribute your link to friends and to directly promote your link on sites and forums while still paying for some wider-range advertising or affiliate networking. The more strategies you use, the more people you reach--and the more sales you'll get.

And if you follow our advice about reinvesting your profits whenever you can, you'll eventually achieve one of the most enjoyable signs of success: the ability to expand your business.
We'll talk briefly about this in our next and final chapter.

10

EXPANDING YOUR BUSINESS

If you've followed our advice to this point: congratulations. You've mastered site design, managing the bridge between online business and offline accounting and shipping, and promoting your site. At this point, your business is either successful--or it's well on the way to being successful. And you've earned it.

The big question that this chapter seeks to address is: what's next? How can you expand your site or expand your business to take advantage of any other products you might have? How might you improve your site to do more business in the future with your existing product? We'll look briefly at both of these possibilities in order to give you some ideas.

EXPANDING AND IMPROVING YOUR SITE

If your business is already successful enough to warrant expansion, then you probably did something right with the initial design and layout of your website. There are any number of reasons that you might be unhappy with your current website, however: maybe you ran out of money for design costs and weren't able to implement some of your favorite ideas. Maybe you've noticed a significant problem in the basic layout of your site that you wish you could fix. Or maybe you're tired of paying service charges to your commerce system and you want to design your own and integrate it into your site.

These are changes that you know for certain that you need to make, and now that you have the money and the time, you should make them. The problem, however, is to identify changes that you don't know that you should make, but that would immensely improve the efficiency, look, and conversion rate of your site.

In order to identify these changes and decide whether or not to make them, you'll want to have some feedback from your customers. One improvement that you should always make in order to get that feedback is to provide your customers with some means of talking to you about your site and suggesting changes. An easy way to do that is to include a "comments" section in your commerce system or in some unobtrusive part of your site, allowing customers to leave their thoughts about your design and overall site. You might also email some of your satisfied customers and solicit their ideas for site improvements: they've used the site, after all, and they know from experience what you might need to change.

Either of these options is workable as long as you keep the basic principles of direct response in mind before making any changes--or before implementing a feedback system that ultimately distracts new customers from buying your product.

Whatever improvements you choose to make, try to use the same coder that you used before (unless there were significant problems with that person.) They'll know your site from the ground up, which means that they know what changes to make and how to make those changes--and they'll be more than willing to work with you.

EXPANDING YOUR PRODUCT LINE

In a traditional online commerce model, expanding your product line is simple. All you have to do is develop all of the materials you need to effectively market your product, expand your website to include those materials, and promote your new product in much the same way as you promoted the old one.

But adding a new product when your website model is based on direct response is more complicated. Remember our basic rules about direct response marketing: don't distract customers from your product with unnecessary information or external links, don't attract unnecessary traffic to your page, and keep your website simple by putting no more than three clicks between your front page and the start of the purchasing process. A new product on your existing site makes it much more difficult to follow these basic rules: your new product will necessarily distract people from your old product, will raise the bandwidth cost for your existing product without converting that higher traffic into sales, and will make your site more complicated to navigate.

So in order to expand your product line, you need to get creative and think about investing a bit more money. (Fortunately, the successful marketing of your first product should mean that you have that money to invest.) There are two basic options:

- Build a new direct response site for your new product.
- Build a new front page (at a separate URL) for your business and link to both of your products from that front page.

Whichever of these you choose, you'll need to build a new direct response site for the new product in order to achieve the same results. If you choose the latter option, you may need to do some work on your existing site as well in order to preserve the three click rule: a front page, whatever its merits, adds an extra click to your total before the customer can purchase a product. What all of this gives you, however, is an easy way to promote all your products while promoting your overall company as well. You can promote the link to your front page to promote your entire product line and raise awareness of your company, while you can still promote your existing products directly to targeted markets. If you are willing to do the work, it can be a win-win situation.

Ultimately, how you expand your business is up to you. And that is the beauty of direct response website marketing. The ability to convert viewers into sales gives you an excellent revenue stream-which gives you a decent stake of money--which gives you the power to decide whatever you want to do in your life and with your business. And if you follow our advice, you'll have that power--and you'll be a success at direct response website marketing.