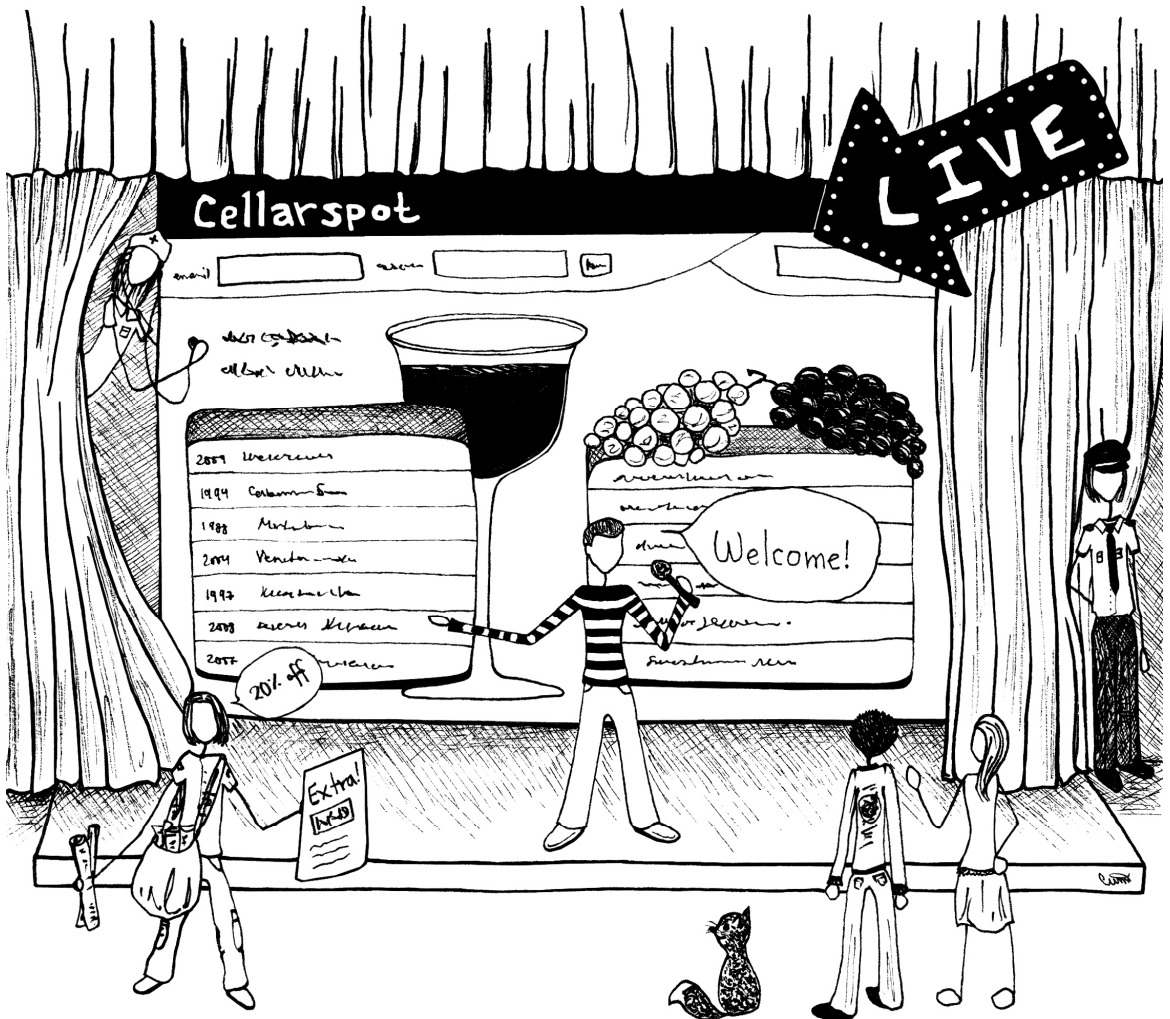


# Chapter 16    Going Live: Creating/Launching a Website

By guest author, Alex Loddengaard  
Professional software developer and  
University of Washington Computer Science graduate



## Introduction

This book has discussed the process one goes through to build a web site from the ground up. At this point you possess most of the skills needed to build a web site, but there is still much to be learned about how to put the web site in production on the Internet for the world to use. In web-speak this process is called "going live".

This chapter will discuss many different areas of knowledge that a site owner needs to be aware of when launching a site. These areas of knowledge include driving traffic to your web site, testing and maintaining your code, tracking your site's growth, and finding ways to make money.

The great thing about the internet is that the gap between personal projects and money-making, professional web sites is small. Many popular web sites, such as YouTube, Yahoo!, and Facebook, started with a young team trying to make something fun and useful. Those sites have grown to be some of the most visited on the entire web, employing hundreds or thousands of employees. This chapter is an attempt to show you the true beauty of the internet: right now you are in a great position to achieve the skills needed to make a useful, popular web site. This chapter also augments the skills you have learned so far with those you'll need to launch, grow, and maintain a web site.

### **13.1 Refine Your Idea**

#### **13.1.1 Talk to Your Users**

#### **13.1.2 Look at Your Competition**

#### **13.1.3 Decide on Technology**

#### **13.1.4 Encouragement**

### **13.2 Get Your Site on the Internet**

#### **13.2.1 Running a Web Server**

#### **13.2.2 Web Hosting**

#### **13.2.3 Debugging/Testing a Live Site**

### **13.3 Driving Traffic to Your Web Site**

#### **13.3.1 Google AdWords**

#### **13.3.2 Viral Marketing Features**

#### **13.3.3 Search Engine Optimization**

#### **13.3.4 PageRank**

#### **13.3.5 Google Analytics**

### **13.4 Making Money**

#### **13.4.1 Google AdSense**

#### **13.4.2 Referral Programs**

#### **13.4.3 Banner Advertisements**

#### **13.4.4 Other Options**

#### **13.4.5 Fund Your Site**

### **13.5 Legal Issues**

#### **13.5.1 Intellectual Property**

#### **13.5.2 Legal Liabilities**

#### **13.5.3 Security Issues**

#### **13.5.4 Conclusion**

Hi, I'm Alex Loddengaard, the guest author of this chapter. I created a web site called Cellarspot, a social network for wine lovers, in Spring 2007. I created, launched, and maintained the site with classmates while studying computer science at the University of Washington.

Throughout this chapter you will see boxes such as this one; these boxes tell my Cellarspot story. The intention is to give you a taste of what we did with Cellarspot and about the decision-making process we went through about technology, features, and everything else.

I was once in your shoes, learning about the web and the technologies that power it. I took what I had learned about PHP, MySQL, etc, and used it to build Cellarspot. This chapter is a result of my learning about what's involved with launching, maintaining, and growing a real web site. These boxes tell my Cellarspot story, a story that started where you are now. You are entirely capable of one day being able to tell perhaps an even more successful story, and this chapter will try its best to help you along the way. I hope this chapter motivates and teaches you how to create something real, something that others can use and benefit from. You're much closer to achieving this than you may think.

Thanks for listening.

## 16.1 Refine Your Idea

The purpose of a web site is to solve a problem. Take YouTube, for example. Without YouTube, none of us would be able to easily find that silly video of the guy dancing and lip-syncing from behind his computer. YouTube solves a problem: It fulfills people's desire to find videos. The purpose of refining your web site idea is to guarantee that your web site will solve a problem, which in turn will increase your chances of people using and benefiting from your web site. Let's face it, making a web site is always cool, but making web sites that people use is much more satisfying.

### 16.1.1 Talk to Your Users

The first step to refine a web site idea is always to talk to potential users, or users who at some point might use your web site. When ideas are first concocted, they are generally not well thought-out and lack the specifics necessary to create a web site that people will actually use. You, the inventor of an idea, are only one person. Unless you speak with others, you will have no way to validate that the idea you have will actually solve a problem.

The key to talking to users is asking good questions. By asking your potential users good questions, you are starting a specific dialog that, once synthesized, will serve as a basis to evaluate the problems your site will solve. Consider the following questions when speaking to potential users:

1. Do you think my web site would be useful?
2. How could I improve my web site to better cater to your needs?
3. Do you use any other web sites similar to my web site, and if so, would you prefer to use my web site or the other web site? Why or why not?
4. What factors would most motivate you to return to my web site? Good information? An aesthetic design? Desire to be part of a community? Lots of functionality?
5. Do you know of anyone that would use the site, and how do their needs differ from yours?

Make sure to speak to more than one potential user. The more conversations you have, the more diverse your understanding will be. It is also a good idea to approach potential users with a site prototype that you can demo. Prototypes can provide concreteness in a dialog, making it more effective.

My father is absolutely nuts over wine. He has been collecting wine for many years now; he even has a large refrigerated wine cellar in his house. I interviewed my father many times when refining my ideas around Cellarspot. I also had him organize a wine dinner with friends, which gave me a great opportunity to hear each of their individual opinions. I asked them questions about what mediums they use to discuss wine. I also asked them how willing they would be to insert information about their entire collections of wine into Cellarspot. From these interviews I learned more about the older wine-loving male demographic, which constitutes a large portion of Cellarspot's user base. Looking back, I should have also interviewed other demographics such as younger users.

Once you have spoken to a few different potential users, you must start to synthesize all of their feedback into one cohesive idea. A good approach to doing this is to define specific *use cases*, which are general actions that a user will perform while using your web site. With use cases defined, you can start proposing web site features to satisfy each use case. Let's look at YouTube, for example. YouTube's web site offers lots of features such as searching, categorizing, and browsing videos, to let users discover new videos and find familiar ones. These example features – searching, categorizing, and browsing – each satisfy a high-level use case. Table 16.1 lists a few YouTube use cases, along with some features that satisfy them.

#### use case

A description of a high-level action that a user might perform when using your site.

Use Case	Satisfying Feature
Find a familiar video	Search, video categorization
Discover new videos	Related videos, response videos, top-rated videos, most-viewed videos
Get involved with the community	Video uploading, friends, channel subscriptions

**Table 16.1 YouTube use cases and their satisfying features**

Use cases and features lay the groundwork for all web sites, whether they are formally defined or not. With well-thought-out, well-defined use cases and features created with the feedback from potential users in mind, the chances of web site success are much greater than they would be otherwise.

Some example use cases that we considered in Cellarspot are: users will want to share tasting experiences with their friends; users will want to remember what they have tasted; users will want to inventory their wine collections. I believe we were successful in creating a list of good use cases that, if executed correctly, would solve a true problem. However, I do not believe we built the right features to satisfy these use cases.

Defining good use cases is the first step. However, the next important step is creating features that satisfy those use cases. One of the features we created to facilitate taste experience sharing was a blog, or a pseudo-journal. Blogging is very prevalent on the internet these days, and at the time when we created the blog feature, we knew nothing about blogging. We ended up creating a terrible excuse for a blog, which nearly none of our users used. Prior to considering a blog, we should have done more research about blogs in order to better address the use case that it was attempting to solve.

### 16.1.2 Look at Your Competition

Many web developers regard competition as a potential threat, which is not necessarily wrong. After all, your competition is stealing your users and money! However, your competition is already one step ahead of you. They have already seen the ways in which their users use their web site, and they have already begun to think about how to improve their web site.

Competition can be a threat, but it should also be a source for ideas. While refining your idea, look closely at your competition to see what they do well and how they could improve. Try to speak to their users to get a better sense of what is good and what could be improved. The practice of comparing yourself to your competition will help you better understand your idea and how it fits with the users in the particular space, or industry, that you and your competition are involved in.

Cloning a competitor's web site is probably a bad idea (if not illegal), but implementing features that are similar is generally a good idea. Often users in a particular space, such as social networks, video web sites, etc., get comfortable with certain features and expect others' web sites to have them. For example, try to name a social network that does not have a messaging feature. All social networks have this feature because users expect it. If you are entering a space with at least one big competitor, then you should try and determine what the key features are that your site must also have.

You can also examine a competitor's web site to get ideas for design and code. It is easy to view the code of any given web page, simply by right-clicking on the page and clicking "View Source." Viewing the source of a web page can help you learn how certain features or tricks are done. But do not copy/paste source code from one page to yours. This may violate copyright laws, which could put you in legal trouble. We will discuss copyright law in more detail later.

Competitors should not be taken lightly. They should be monitored, analyzed, and thought about when creating the idea for a site. They can help you, but they should not be the sole purpose

behind a feature or focus of your site. They should be included in the debate that goes on either in your own head or amongst your teammates when deciding on features.

The two largest competitors to Cellarspot are Cellartracker and Corkd. Taking a quick glance at each of these two web sites tells you two things: Cellartracker is plain and Corkd is rich and beautiful. A deeper use of each of these web sites reveals other differences, such as the detail given to users to describe wines that they have tasted. We analyzed these web sites to understand why their users enjoyed using them. We were worried about Corkd and its elegant and useful design; we were confident we could outperform Cellartracker. In hindsight, we should have realized that the Cellartracker's large user base was a good indication that they were doing many things right.

### 16.1.3 Decide on Technology

Once you have refined your web site idea and have a better sense of the problem you're going to try to solve and the features you're going to build, the next logical step is to decide on the technology to use. Though this may seem like an obvious decision, deciding on technology can be more complex. Some types of technology allow you to develop a web site very rapidly or simply, while others may help you develop your web site in a very flexible or expandable way. All of these ideas – speed, simplicity, maintainability, and expandability – should be considered when deciding on technology. *Expandable code* is code that can be easily expanded to perform other tasks and features. Similarly, a *maintainable* web site is one that is easy to keep running, debug, and modify in minor ways as needed. Nearly all sites have small bugs here and there, and if yours is maintainable, then those bugs will be easier to discover and fix than they would be otherwise.

Many new web programmers begin by learning PHP, but PHP may not be the best language to use for every project. Other technologies called *rapid application development (RAD) frameworks* such as Django, Ruby on Rails, and CakePHP can automatically generate code for you, making your job easier and quicker. These frameworks allow you to develop maintainable, expandable code quickly, but they do not give you as much control over your code as you would have designing a site from scratch with PHP. They are worth learning in greater detail, but a review of them requires an entirely separate book. Rapid development frameworks exist to make your job easier, but hand-coded web sites can work just as well. Table 16.2 shows a comparison of several server-side web technologies.

#### **maintainable code**

Code that makes bug fixing, and feature changing easier at times after you have already launched the first version of your web site.

#### **expandable code**

Code that makes feature addition easy at times after you have already launched the first version of your web site.

#### **rapid development (RAD) framework**

A technology that does a lot of work for you by generating code and providing common features.

Cellarspot was implemented in JSP, using the Spring framework and Hibernate persistence layer. I personally regret using JSP, because too much of our programming time was spent writing code that a rapid development framework would have generated for us. We chose JSP because we were all proficient in Java. However, after working through two versions of Cellarspot, both written in JSP, I have felt the pain of not using a rapid development framework. My current preference for a rapid development framework is Django, because I love Python, and I find it intuitive and easy to use.

Web Technology	Pros	Cons
PHP	Fast and very widely used	Encourages non-maintainable and non-expandable code
CakePHP	Rapid development compared to plain PHP	Not as intuitive and clean as Ruby on Rails or Django
JSP (Java)	Based on Java	Usually requires lots of code
Ruby on Rails	Rapid development; widely used	Reinvents a lot of traditional web practices that do not necessarily need to be reinvented. Not as intuitive as Django
.NET	Any Microsoft languages (C++, C#, Visual Basic) can be used	Usually requires lots of code to be written; requires IIS web server
Django (Python)	Rapid development; intuitive	Somewhat immature, not as many features as Ruby on Rails

**Table 16.2 Server-side technologies and their tradeoffs**

There are other tradeoffs that one should consider when deciding on technology. For example, MySQL, the database that you have learned in this book, is great for most web sites. However, certain web sites, in particular web sites that use maps and geo-coded (latitude and longitude) coordinates, benefit much more by using PostgreSQL, a competitor of MySQL. Microsoft SQL Server, SQLite, and Oracle are also very widely used database implementations. Each database implementation does certain things better, and it is important to understand their pros and cons when deciding what to use. Table 16.3 discusses the pros and cons of various database implementations.

Database Software	Pros	Cons
Microsoft SQL Server	Good at optimizing large queries	Expensive; must run on Windows
MySQL	Free; widely used; fast for single-table queries	Slow for complex queries/operations
Oracle	Reliable, good for data analysis	Expensive and complicated
PostgreSQL	Free; good at optimizing complicated queries and other operations	Complicated to get running
SQLite	Free; simple; fast for reading data	Slow for writes; dangerous for sites with many insert/update operations

**Table 16.3 Database implementations and their tradeoffs**

We used MySQL for Cellarspot, because our database schema was relatively simple. All of us were also already familiar with MySQL. MySQL worked out fine for Cellarspot, because the site is relatively simple, and the amount of traffic we see is relatively small. In most cases, MySQL is a great database choice for its simplicity.

In addition to software tradeoffs, one should also consider the learning curve associated with learning new technologies. In order to give competing databases or rapid development frameworks a full evaluation, someone will have to install and try out each of the choices. This can be a long proc-



ess, and if your web site has to be developed very quickly, the best way to decide on new technology may be to choose technology that is already familiar.

You will also need to make decisions about what web browsers you want your site to support. Certain web sites (such as sites that target Mac users) do not support Internet Explorer because it does not exist for recent versions of Mac OS X. Similarly, some web sites such as Facebook need to support all browsers, including mobile browsers on devices such as the iPhone. Your browser compatibility should be based on your user base. If you are building a general web site, then you will most likely have to support Internet Explorer. As of this writing, most IE users use version 6, 7 or now 8. (Just be glad that Internet Explorer 5 is no longer widely used!) According to [hitslink.com](http://hitslink.com), Internet Explorer still constitutes 71% of all browser usage across the world. Firefox constitutes nearly 20%, Safari constitutes 6.5%, and the remaining percentage is split across a number of less common browsers such as Google Chrome and Opera.

Cellarspot chose to support both Internet Explorer 6 and 7, because an overwhelming majority of older men (the majority of the Cellarspot user base) use IE. Cellarspot also had to support Firefox and Safari, because these two browsers are most commonly used by younger, more internet-savvy wine lovers. We had to bear the brunt by supporting so many browsers, but our work paid off. About half of our users use IE, and none complained of browser-specific bugs.

In conclusion, there are many tradeoffs between frameworks and pieces of software, and these tradeoffs should be considered when determining the technology to use. Learning curves should be considered, along with expandability, maintainability, speed to develop, simplicity, and compatibility.

### 16.1.4 Encouragement

The truth is that the distance between a single person's personal project and a widely-used money-making web site is not large. This is the true beauty of the web: that anyone of any level of experience can create a web site that other people use and benefit from. You have seen this happen with large web sites such as Facebook and YouTube. They both started off small, perhaps as personal projects, and they have since developed into gigantic, famous, world-wide businesses. This truth is the driving reason behind many programmers' love for the web, and it is a truth that you should accept and exploit. You have the knowledge and ability to make something great, just as others in your position have done before you. Dream big, and execute effectively. Even if you create a failed web site, then you will make up for it with gained experience and knowledge.

Cellarspot failed to generate lots of traffic and money, but it is not a failed web site or a failed learning experience. I cannot begin to explain the significance that Cellarspot has played in my computer science career thus far. Cellarspot has been brought up in nearly all of my technical interviews, and it manages to impress nearly all of my interviewers, despite its inability to solve the problem that we hoped it would solve. Most importantly, Cellarspot taught me about the process one must go through when creating a web site. I would not be writing this chapter without the experience I gained from Cellarspot. I do not regret the many, many hours I spent working on Cellarspot; the experience I gained and the knowledge I acquired could not have been achieved from a book or lecture. Cellarspot was a real world experience, and it taught me how I can change my practices in the future to increase my probability of creating a successful web site. With each "failure" comes new perspectives and new insights that make all other future attempts more probable for success. As the ancient Chinese proverb goes, "The gem cannot be polished without friction, nor man perfected without trials."

## 16.2 Get Your Site on the Internet

Building your web site is the natural next step after refining your idea. However, given that this entire book thus far has been about building web sites, this chapter will continue with the step following building: launching your site and making it accessible to the world, or at least to the Internet.

### 16.2.1 Running a Web Server

Thus far you have learned how to accept information from the client in the form of query variables, and similarly you have written HTML, JavaScript, and CSS that the client's web browser understands. This simple interface -- accepting variables and writing text -- is made possible by web server software. Web server software is a program that waits for an HTTP request from a client and sends that request to your web application. Finally, the web server then takes your web page's HTML, JavaScript, and CSS output and sends it back to the client. Though this seems rather simple, web servers in practice, such as Apache HTTP and Microsoft IIS, are insanely complicated.

#### data center

A room or building that is filled with computers, power backups, and cooling supplies, with a very fast connection to the internet.

The most common place a web server is run is on a server in a distant *data center*, a building full of servers and computers and a fast internet connection. However, if while developing and testing your web site, you use a server to serve your web site, you will have to make code changes, upload these changes to the server via a transfer protocol such as FTP or SSH, and refresh the page to see your change. For this reason, developing and testing on a server is rather cumbersome. A much more efficient solution to developing and testing a web site is to run a web server on your own personal computer. Even though web servers are meant to be run on servers in datacenters, they can be run on your person computer, which makes applying code changes much more simple and quick.

When we were developing Cellarspot, we were all able to run the web site from our desktops and laptops. Once we all agreed that what was running on our local computers was sufficiently stable and ready for the internet, we deployed our software to a real web server in a data center.

Unfortunately, installing a web server is not as easy as just installing Apache. Apache by itself is capable of serving static content such as HTML, CSS, and image files, but in order to run PHP or query MySQL, special software packages need to be installed alongside the web server. Fortunately, hordes of web developers have installed Apache, PHP, and MySQL together, so installers exist that can install and configure everything automatically. Surprisingly enough, Mac OS 10.5 (Leopard) comes installed with Apache and PHP, only requiring MySQL to be installed. Refer to the following links to learn how to install Apache, PHP, and MySQL on either Windows or Mac:

Lean more about installing Apache, PHP, and MySQL from the links below:

- Windows: <http://www.wampserver.com/en/>
- Mac OS X: <http://www.mamp.info/en/index.php>
- Linux: [http://www.howtoforge.com/ubuntu\\_debian\\_lamp\\_server](http://www.howtoforge.com/ubuntu_debian_lamp_server)

### 16.2.2 Web Hosting

Running your own web server on your personal computer is great for developing and testing your web site. However, this setup will most likely not suffice for a web site that is publicly available. If you plan to use your personal computer to serve your public web site, then your computer must remain turned on at all times, which is impractical. Users would not be able to access your site if your computer were turned off.



Fortunately, many different web hosts offer hosting services that provide a web server for you on a server in a datacenter. Many of these web hosts are very cheap; rates can start as low as \$3.00 per month for access to a web server, PHP, and MySQL. Web hosts come in many different flavors, but they all accomplish the same goal: to allow web site owners without their own servers and datacenters to run public sites.

The first, simplest type of hosting is referred to as simply *web hosting*. A web host will host many different web sites on a single server, allowing the host to charge less per web site, but forcing the server to do more work. You connect to the server using SSH and upload your files into a particular directory on the server, which maps to some URL on the public internet. Using web hosts is relatively cheap, but if your web site traffic increases to a certain threshold, your host will not be able to reliably serve your web site. Given that many different web site owners are hosted from the same server, web hosts typically have to limit the technologies their customers are allowed to use. Typically certain web hosts will only provide one specific server-side web technology, such as PHP, Ruby on Rails, etc. When choosing a web host, make sure they provide the technology that you are interested in using. Certain technologies may cost more to host than others. For example, JSP hosting is typically more expensive than Python and PHP hosting.

Another form of hosting is known as *virtual private server* or *VPS* hosting. A VPS is essentially a virtual server, or a VPS instance, that is run alongside other VPS instances on the same physical server. A single physical server can have many VPS instances installed, where each VPS is essentially its own server. It follows that VPS hosting is more powerful in that web site owners have total control over what technology they can use, because they have total control over their VPS instance. However, because VPS hosts have many VPS instances on a single server, if your web site ever became very popular, your host would not be able to reliably serve your web site. VPS hosting is also on the order of four times as expensive as standard web hosting.

A much newer form of hosting called *cloud hosting* has recently become popular, where web applications are hosted from a "cloud" of computers. The two leaders in this type of hosting are Google's App Engine and Amazon's Elastic Compute Cloud (EC2). The idea behind this type of hosting is that you can host a web site for any number of users. Web sites with both large and small traffic can be hosted without drops in reliability, and the host will automatically handle large increases in traffic for you. The idea is that a "cloud" of computers is serving your web site. When your web site traffic is small, maybe only one computer is serving your web site, but when your web site traffic is large, maybe hundreds of computers are serving your web site. Though cloud and VPS hosting are very different, each solves the problem of web site reliability when an infrequently visited web site is visited much more frequently. The cost of cloud hosting is typically somewhere between a basic web host and a VPS host.

The last, most common type of hosting among professional businesses is known as collocation. *Collocation* is the service one purchases when they own their own rack-mounted servers. A rack-mounted server is a special server that is very thin and that can be installed on a rack, which houses many of these special servers. A photo of a rack-mounted server can be seen in Figure 16.1.

### web host

A company that provides its clients with a web server on a server in a data center, making it cheaper and easier for a site owner to host their web site.

### virtual private server

A virtual server that runs alongside other virtual servers on the same physical server.

### cloud hosting

A new type of hosting, which serves web sites from a cloud of computers, allowing web sites to be reliable with large amounts of traffic growth.

### collocation

A type of hosting which involves installing a rack-mounted server in a data center, where it has backup power, air conditioning, and a fast connection to the internet.



**Figure 16.1** Rack-mounted server that can be collocated

Collocating a rack-mounted server means installing a server in a data center that is managed by a collocation host. Once installed in the data center, the server is given backup power in case of power outages, along with a fast internet connection. Collocation is by far the most customizable solution, but it is also the most expensive. Collocation costs start at around \$100 per month, and can go as high as many tens of thousands of dollars per month depending on the number of servers.

You should choose hosting based on your site's amount of traffic and the technology your site uses. Table 16.4 does a side-by-side comparison of all the hosting options from this section.

Hosting Type	Pros	Cons
Standard Web Host (Dreamhost.com, InMotion.com)	Cheap; simple; easy	Cannot handle large traffic; only certain technologies
Virtual Private Server (VPS) (Slicehost.com, Rackspace.com)	Cheap; customizable; has any technology you need	Cannot handle large amounts of traffic
"Cloud" Hosting (Google App Engine, Amazon EC2)	Cheap; can handle small and large traffic	May support only limited choice of web technologies
Collocation Host (adtaq.com, Rackspace.com)	Customizable; can handle small and large traffic	Expensive; requires you to configure the server(s)

**Table 16.4** Web hosting options

Cellarspot was hosted on a powerful Dell rack-mount server and collocated in Seattle. We chose this hosting thinking that we would see lots of traffic right away, and at the time cloud hosting was not very popular. We did not recognize that most web sites grow slowly, and that choosing collocation right away is economically risky. We learned a lot about Linux server administration by collocating a rack-mounted server, but we spent much more money than we otherwise would have. The server cost upwards of \$2000, and the collocation service cost around \$100 per month. Given these figures, Cellarspot cost us around \$3200 for the first year. Fortunately for us, we managed to collect around \$100 in revenues for the first year, giving us an overall net loss of \$3100. (Ouch!) This section will later discuss means for making money. In hindsight, we should have chosen a basic web host to start, and upgraded to a collocated rack-mount as our traffic increased.

Web hosting provides you with space on a server and a connection to the internet, but web hosting does not always include a domain name, which is a web site name that ends with .com, .net, etc. In order to register a domain name to allow your web site to be accessed like all other web sites, you have to also sign up with a registrar and register a domain name. Nearly all web hosting companies, such as Dreamhost, Slicehost, etc, offer both web hosting services and domain registration services. In fact, they usually make it very easy to sign up for each of these services at the same time. When signing up for a web host, be sure to also register a domain name with them if you don't have one already.

### domain name

A name that is superseded by ".com," ".net," ".edu," etc. Domain names are registered with domain name registrars, which are often web hosts as well.

## 16.2.3 Debugging and Testing a Live Web Site

In the motion picture *Spider-man*, Peter Parker's uncle Ben said, "With great power comes great responsibility." This advice applies when running a web site. When making a web site for a homework assignment or just for fun, it's generally acceptable to have little problems with the site. But when you are running a real site, problems can be a drastic deterrent, depending on their severity. For this reason, special measures should always be put in place to help you understand when your web site has a problem. These measures should also make debugging and understanding the problem as easy as possible. This section discusses mechanisms for debugging and testing a live web site.

### Logging

The first and simplest approach that you should put in place to make debugging problems easier is *logging*. All widely used programming languages such as Python, Java, C++, PHP, etc., have tools and libraries to make logging easier for developers. A *log file* is a text file to which a web site, or any program for that matter, can append messages. These messages describe what the program is doing at a certain time. For example, log messages can contain information about when a certain database row was inserted, or when a certain operation was performed for a user. Log messages can be anything that you want them to be, so it's important to choose log messages that will be helpful for debugging purposes when a problem is discovered.

### log file

A file that contains an execution log for a given program, where each message in the log describes what the program is doing at a certain time.

Java's most common logging tool is the Apache Commons log4j library, and Python's most common logging tool is the built-in logging module. PHP, like Python, has built-in support for logging via a few functions such as `error_log`. On top of logging for your web site, your web server and database will also create their own logs, which may be useful to you for certain problems. Usually, in Linux systems, web server and database logs are found in `/var/log`. If this directory does not appear to have information about your web server or database, then take a look at their respective configuration files. Usually Linux configuration files can be found in `/etc`.

```
127.0.0.1 - - [25/Jan/2009:12:57:51 -0800] "GET /show_ads.js HTTP/1.1" 404 216
[Sun Jan 25 12:57:51 2009] [error] [client 127.0.0.1] File does not exist:
/pagead, referer: http://speeddemosarchive.com/KungFu.html
```

### 16.1 Apache log example output

### Error and Exception Handling

#### Security Note

Don't display error messages with code details on your site; attackers can use this information to better target your web server.



While developing your web site, you probably became familiar with errors and exceptions that were created because of bad code. When working on a site in progress, it's useful to have the site display any error messages directly on the page so you can see them and quickly fix them. But in a live web site, you want to hide these errors from users' eyes. Partly this is to maintain professionalism, because exception messages confuse or frustrate users. Another reason is because giving a malicious

user a lot of information about a site crash can create a security risk or give an attacker information to use against your site.

So you don't want every user to see detailed error messages when your site fails, but you still want to expose these errors to yourself (the developer) so you can find and fix them. All popular web technologies such as PHP, Python, JSP, etc, allow you to do this. To learn more, do a web search for error and exception handling for the technology that you are using. The most common implementation of error and exception handling is to display a plain error message to users and to send an email with the full error message to the maintainers of your web site.

### Automated Testing

#### cron job (service)

Cron jobs are Linux's way of automatically executing a program every so often, while services are Window's way of doing the exact same thing.

One last measure that can be put in place to help debug and guarantee web site nirvana (that is, zero problems) is to setup recurring testing programs to analyze your database and look for problems and inconsistencies. These recurring programs are called *cron jobs* in Linux and *services* in Windows. You can configure these programs to run as often as you would like. For example, you can run certain programs every minute, others every hour, and some every week, month, or year. Cron jobs and

services can be used to run a test program that analyzes your database. Below are some common things you can check in a data validation program:

1. Empty fields that should be required (for example, user emails, etc)
2. Special characters that may not display correctly on your site (<, >, &, ", etc.)
3. Bad relationships (for example, if your user table references a nonexistent second table)

Depending on your web site, you may even be able to have your cron job or service correct issues that it finds. For example, it would be relatively simple to create a script that looked for bad characters (<, >, etc.) and change them to HTML-safe equivalents (&lt;;, &gt;;, etc.).

We did not have any notifications or automated testing in place in Cellarspot, and for this reason we had a few bugs that persisted for a long time without notice. For example, one user was able to hack our web site by inserting custom JavaScript. This hack is known as an XSS attack. Another user was not able to register for an account because his last name contained a hyphen (-), which our web site considered a bad character. Another user wasn't able to register for an account because his zip code started with a zero (0), which made Cellarspot think their zip code was four characters long, and hence invalid. Had we not known these users personally, we would have never known that these issues existed. We needed to have put automated tests in place to verify our database, along with a notification framework that told us about errors that users came across while using our web site.

### Browser Compatibility

Another key piece of debugging and testing your web site is browser compatibility, and unfortunately, ensuring your web site works in different browsers can be a painstaking process. That said,

there are ways to distribute this manual process to make your life a little easier. Most web sites provide an easy way for their users to submit problems, issues, and feature requests. A great trick when accepting this information is to automatically include the user's browser information, which can be fetched by JavaScript and by most server-side languages such as JSP, PHP, etc. Providing this information will allow you to try and reproduce the problem that the user has encountered, which may not necessarily be a problem in other browsers. Browser compatibility is not a very intellectual aspect of web programming, but efforts must be made to ensure your web site works in all necessary browsers. You should not only do your own testing, but also allow your users to tell you about problems. Make sure you have a bug reporting mechanism on your web site and that you test extensively for cross-browser compatibility both with JavaScript and CSS.

Fortunately for Cellarspot, we did lots of testing in Internet Explorer 6 and 7, and we were never aware of an issue related to browser compatibility. However, we did not have a user-submitted issues tool in place, so there could exist a few small browser incompatibilities that are still around today.

### Site Environments: Staging and Production

Once a web site goes live and new changes are being introduced, the source code that powers that web site should be run in two different locations called *environments*. A *staging environment* is a version of your web site that is only accessible to the site developers; it gives the developers a place to test new changes to the web site. A *production environment* is the place storing the actual live web site. The reason for having two different environments is to ensure that certain changes work in staging before they are applied to the production web site.

#### environment

A context that a web site runs in. Typically most professional web sites have a staging environment for development purposes and a production environment, which houses their live web site.

It is never a good idea to make live changes to a production web site, because you will run the risk of making a mistake and causing your site to create an error or perform strangely for the user. Your staging and production environments will differ only slightly; one common difference between the two environments is database configuration. Staging environments can be configured to use an entirely different database than production: a testing database that can be tampered with. Your staging environment can be on your local computer, but generally you want your staging environment to be as similar to your production environment as possible. For this reason, most web sites have two different hosting plans with the same host: one for staging, and the other for production; typically the only difference between the staging and production environments are configuration settings. The important aspect of having two different hosting plans is to have two isolated environments that are mirror copies of one another.

Launching a web site is like freeing a caged animal into the wild. Without certain precautions in place, you may launch a web site with errors and problems that wouldn't look good in the eyes of your users. The most common way to avoid having problems with your web site is to create error email alerts, deploy testing programs that guarantee data validity, write comprehensive log messages, and allow for user-submitted feedback. Without monitoring, your web site will have a higher potential to be hacked or to appear to be lower quality, which can deter users. Testing and debugging efforts are an up-front cost that will make maintaining your web site and furthering its excellence much, much easier. Accept the cost up front; you will be glad you did when you are further down the road of web site maintenance.



## 16.3 Driving Traffic to Your Web Site

By this time you have seen how to build and launch a web site. Next is possibly the most difficult challenge in web site success: getting people to use it. This section will talk about strategies to drive traffic to your web site.

### 16.3.1 Google AdWords

#### Google AdWords

A service that allows you to purchase keywords, whereby an advertisement you specify is displayed when your keywords are searched for.

Though you may not have heard of Google AdWords, you have certainly seen them in action. Google AdWords are the sponsored advertisements you see on Google's search results page. AdWords allow web site owners to purchase specific keywords. By purchasing these keywords, Google will guarantee that your ad is displayed a certain percentage of the time when these keywords are searched. Figure 16.2 shows examples of AdWords displaying ads on Google search results.

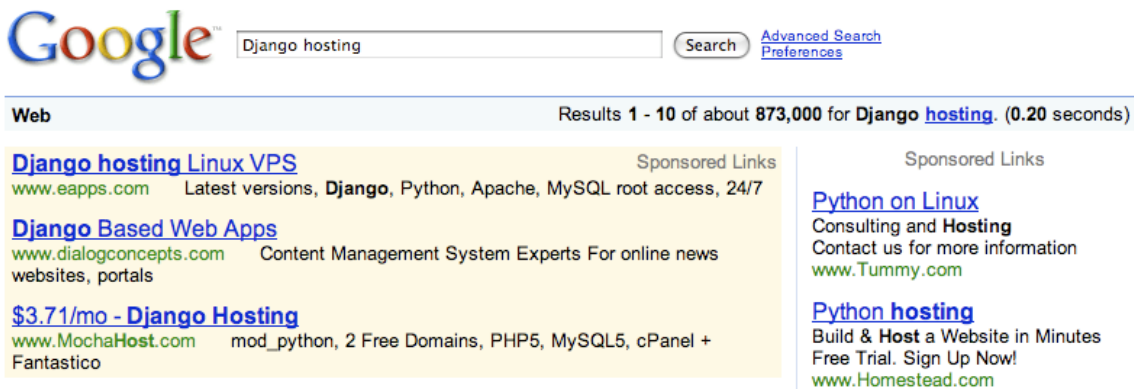


Figure 16.2 Google AdWords

#### Cost Per Click (CPC)

A form of advertising where the advertiser pays each time their ad is clicked.

In the figure, the web sites displayed in the "Sponsored Links" sections, both directly under the search box and in the right sidebar, purchased "Django hosting" AdWords. If a Google user clicks on any of these links, the web site whose link was clicked will be charged a small amount. This model of advertising is called *cost per click*, or CPC.

Google uses a complex algorithm to determine how much each ad click costs, but most clicks generally cost between \$.25 and \$1.50. For most keywords, there exist more advertisers than spots for advertisements, which means that your ad is not guaranteed to be displayed each time a certain keyword is searched. Google will cycle advertisements depending on the frequency at which each advertisement has been shown. Google also allows its customers to spread their clicks out as much as they would like. For example, customers can purchase \$100 worth of clicks for a keyword and request that the \$100 be spent as quickly as possible, or be spread out over a timeframe that the customer can specify.

Google AdWords advertisements can also be displayed on web sites that run AdSense (another Google product that we'll discuss in more detail later). This allows AdWords to display advertisements on non-Google web sites. Google first started selling ads on their search results page, but once they started selling ads on others' web sites by means of the AdSense product, their business took off and revenues skyrocketed. AdWords makes up most of Google's total revenues because it is such a useful product for advertisers.



Google AdWords are an easy way to get your web site name and description in front of people's eyes, but AdWords cost money and may not be as cost effective as other mediums of marketing discussed later in this section. However, AdWords is the most cost effective way to advertise on the internet. They allow an advertiser to display ads to users who are interested in them, and they are paid for on a pay-per-click basis, which means advertisers only pay when a user visits their web site.

We never purchased any keywords with Google AdWords for Cellarspot, because we didn't have much money to spend on advertising. AdWords may have driven traffic to our web site, but this new traffic may not have necessarily been helpful. I make this claim because we had a problem with converting new visitors to registered users. That is, most new Cellarspot visitors did not sign up for an account. Purchasing AdWords would have increased our traffic, but an increase in traffic probably wouldn't have meant an increase in account registration. For Cellarspot, like most social networks, account registration is more important than new visitor traffic, because registered users are the users who will be spending the most time on social networks. Cellarspot did, however, have Google AdSense, which will be discussed in more detail later.

As of this writing, Marty Stepp, one of the authors of this book, has had AdWords keywords purchased for his name. This means that if you search Google for "marty stepp," you'll notice a sponsored ad with some interesting text. Figure 16.3 shows the ad. You'll notice that the text was not written by Marty; it turns out that one of his friends purchased this ad for him as a joke.

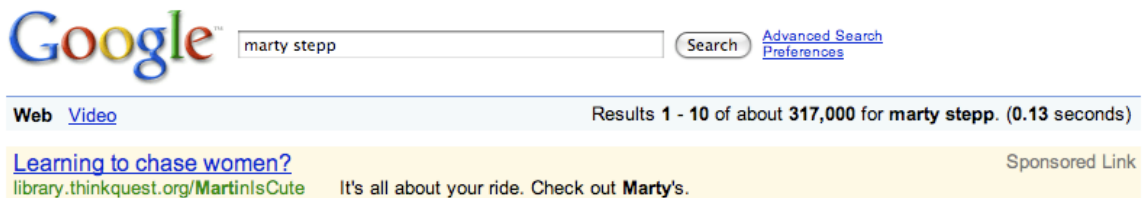


Figure 16.3 Marty Stepp's AdWords

To sign up for Google AdWords, visit <http://adwords.google.com/>. Other search engines such as Live Search and Yahoo! provide similar services as well.

### 16.3.2 Viral Marketing

A free and very effective mechanism for generating traffic is to create certain features that will help your web site spread virally. *Viral marketing* is marketing that spreads by word of mouth. Friends will naturally want to tell their friends about a web site that they have gotten involved with, and you should make that communication as easy and seamless as possible. Though the word "viral" usually has a negative connotation, it is a cost effective way to increase your web site's traffic. Included below are a few examples of viral features that are commonly found in modern web sites.

#### viral marketing

Allows a user to get friends involved in advertising a web site.

#### Invites

Invites usually come in the flavor of a single "Invite" link, which prompts the user for a list of their friends' email addresses. Upon submission, invitations are sent to each email address that was included. The email contains information about the web site, the user who created the invite, and a call to action to create an account. Invites can also help bootstrap a user's activity on the web site. For example, if an invitee signs up for a social network, the chances are good that they will be friends

## Chapter 16 Going Live: Creating/Launching a Website

with the inviter. For this reason, registrations from invites can create friend relationships automatically. Invites are a great way to get new users registered on your web site.

Invites can also be incentivized. That is, if your web site provides some sort of credit or score for its users, then users who invite their friends could receive free credits or increased scores. Giving users incentives to invite their friends will increase the number of invitations that your users send, which increases your user base and traffic. Netflix is a good example of an incentivized invite system. Users who invite their friends to use Netflix get discounts on their Netflix services, which encourages users to invite friends.

### Address Book Searching

When a user first signs up with a web site, the chances are good that some of their friends are already on the web site. For this reason, many web sites allow their users to upload an address book or even provide login information to their email accounts, which the web site will use to look for friends that are already signed up. This feature helps to bootstrap a new user's activity on the web site by automatically creating friend relationships. Address book searching is a great way to get your users started using your web site.

### Sharing

All web sites have content or material that can be consumed by a user. This consumable material can come in the form of funny images, videos, political analysis, etc. The chances are good that some consumers will want to inform their friends of the material that they have recently consumed. A sharing feature would allow users to send emails to their friends, where each sent email would contain the consumable material, along with some information about your web site. Sharing is a great way to drive traffic to your web site.

### RSS Feeds

#### RSS feed

A special type of content that usually represents recent activity and is subscribable by feed readers.

#### RSS feed reader

A program that syndicates a number of subscribed feeds into a single consumable list.

RSS feeds are a very effective way to make your web site more viral, but they are somewhat complicated. *RSS feeds* are a special type of content that are subscribable and that usually display some notion of recent activity. They allow users who are interested in seeing recent activity to subscribe to this recent activity with a feed reader. An *RSS feed reader* is a special type of program that allows users to add RSS feeds that they are interested in subscribing to. Every time one of their RSS feeds has a new item, that item is displayed in a list alongside other new items, allowing the user to effectively read new items from all of their subscriptions in a single consumable list. Some example feed readers are Google

Reader, FeedReader.com, and Netvibes.com. Figure 16.4 shows a screen shot of Google Reader.

RSS feeds are most commonly used in news web sites and blogs. However, they also allow other web sites to very easily fetch data from your web site, which means that other web sites can programmatically display your content, making your content visible to a broader audience. RSS feeds provide a "readers digest" version of a web site. They provide a recent snapshot of a web site, and feed readers combine recent snapshots from many different web sites together into a single digest.

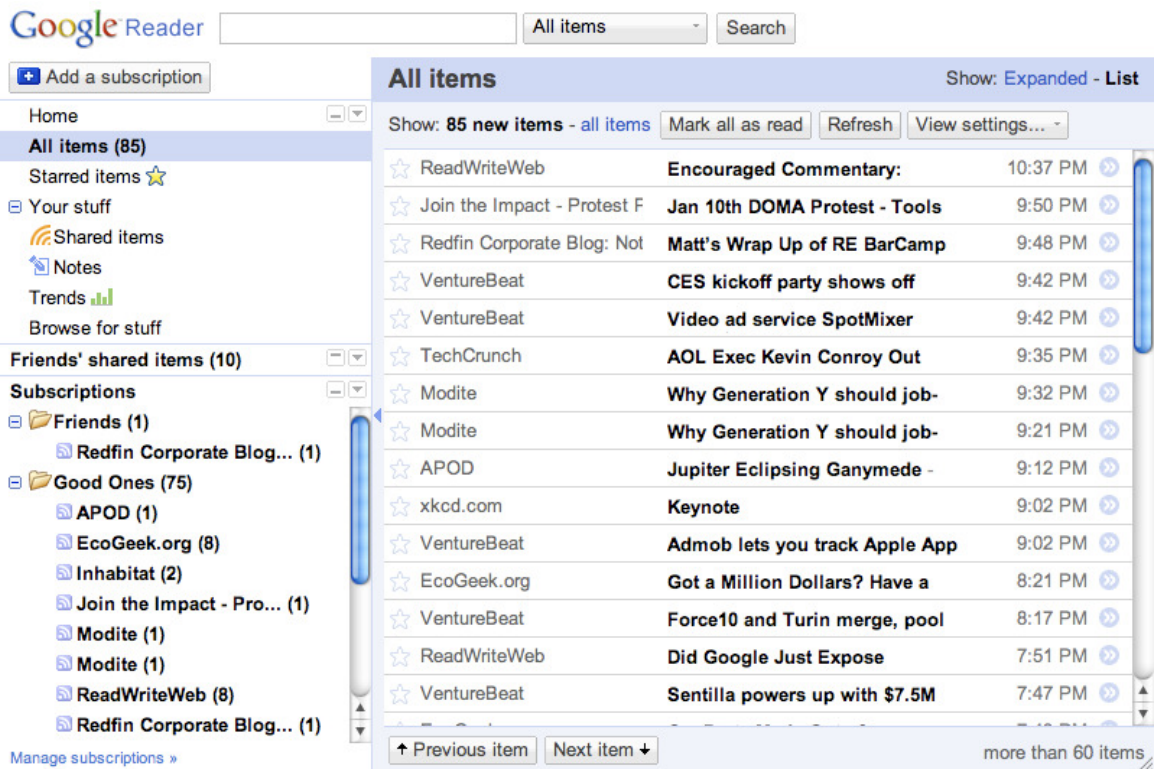


Figure 16.4 Google Reader (an RSS feed reader)

RSS feeds allow your users to subscribe to your web site, which effectively keeps them coming back for more. They also allow your data to be more portable, which broadens your audience. Most technologies such as Java, PHP, and Python also provide tools and libraries to create RSS feeds.

### Embedded Features

As was the case with AdWords, you may not have heard of embedded features before, but you have most certainly seen them. *Embedded features* are features of your site that can be included on other web sites. The most known use of this feature is YouTube's ability to let users embed YouTube videos on their web sites. YouTube provides a snippet of HTML code that can merely be copied and pasted into another web site's HTML source code; the copy-pasted HTML is all the user needs to get the video playing on their web site. Embedded features broaden your audience and generate incoming links to your web site, which can drive more traffic and also improve your search engine optimization. Search engine optimization will be covered in detail later in this chapter.

Specific viral features are not always relevant, depending on the goals of your web site. You should decide which of these features is most relevant and include them in your web site. You should also pay close attention to which features are most useful to you and bear this in mind when creating other features. This chapter will also cover ways to determine how users find your web site.

Cellarspot implemented one viral feature: invites. The invite feature noticeably increased the number of registered users we had. Cellarspot didn't launch with the invite feature, because at the time we didn't think it was a priority. However, we convinced ourselves that we needed the invite feature, and its usage proved that viral features can increase traffic to your web site.

### 16.3.3 Search Engine Optimization (SEO)

Nearly all Internet users use search engines as a starting point for finding something on the internet. Studies have also shown that most search engine users consider only the first page of results, which makes having a good search engine ranking very important.

#### search engine optimization (SEO)

The practice of optimizing your web site in such a way that it performs better in search engine rankings.

*Search engine optimization (SEO)* is the practice of designing your web site in such a way to make it perform better in search engines. A web site that has good SEO will appear high in search engines' search results for keywords relating to that web site. For example, if your web site is about cats, and a search for "cats" on a search engine displays your web site at the very top of the search results, your web site has great SEO. SEO is perhaps the most effective way to drive traffic to your web site.

Search engines succeed when they provide a useful, relevant link to a user who searches for a specific set of keywords. SEO is about understanding how search engines determine relevancy of each web site for a given set of keywords. The catch is that search engines don't always make these techniques publicly available, because they need to stay competitive with other search engines. However, given how important SEO is for a web site, a large amount of research has been done to determine the techniques one should follow to have good SEO.

A few years ago, one of Google's ranking algorithms was discovered and exploited by bloggers on the internet. Thousands of bloggers all linked to [whitehouse.gov](http://whitehouse.gov), which at the time was a biography of President George W. Bush. These bloggers linked with an anchor text of "miserable failure." As a result, searching for "miserable failure" on Google would display [whitehouse.gov](http://whitehouse.gov) as the first result. Google changed their search algorithm to not be susceptible to an exploit like this one, which is now known as a *Google bomb*.

Some techniques to improve SEO have to do with the way you design and code your web site, while others focus on how other web sites affect your web site's SEO. Let's first look at the former set of techniques: ways to design and code your web site to improve SEO.

#### Descriptive Titles and Headings

An easy way for a search engine to determine the essence of a web site is to analyze the site's `<title>` and `<h1>` tags. These tags are meant to describe the web site in a single sentence, so the keywords used in this sentence are often very similar to the general categorization of the web site itself. For example, if the `<title>` and `<h1>` tags have the words "online," "shopping," and "electronics," then the web site is probably an online technology store. Take a look at the `<title>` of Amazon.com's front page, which at the time of this writing was, "Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more." Search engines consider the `<title>` and `<h1>` tags to be great descriptors of a site, so make sure these tags describe yours in a concise way.

#### Meta Tags

The `<meta>` tag is another easy way for search engines to understand your web site. However, not all meta tags are totally relevant anymore. At the time the HTML specification that introduced meta tags was released, search engines were not capable of analyzing the entire internet and coming up with keyword descriptions of each web site. For this reason, the meta "keywords" tag was necessary to describe a web site according to a few keywords. In today's day and age, all reputable search engines such as Google, Yahoo!, and Live Search can determine the mapping of keywords to web sites by analyzing the actual content of each web site, not just its meta tags. That is, search engines no longer need (nor entirely trust) a site's meta "keywords" tag to determine a site's relevance. However, the meta "description" tag is important for SEO for two reasons. First, it gives a mechanism for a web site owner to describe their web site. And second, it's often used as the text that search

engines display on their search results page. You probably want control over this aspect of your site's appearance in the search rankings, so always make sure to use a meta "description" tag on your site's pages.

### Descriptive, Plain, Discoverable Text

More and more web sites these days are written entirely using technologies such as Flash, JavaScript, or images. These web sites, though fancy and fun to use, provide hardly any information to search engines, because search engines are poor at analyzing this kind of content. Search engines are most capable of analyzing plain text found in HTML source code. For this reason, try to avoid making web sites that rely too heavily on images, JavaScript, or Flash to display their core content. Try to have descriptive, plain text on every page. The more consumable content search engines have, the better job they'll do at categorizing your pages.

Search engines also aren't generally able to make Ajax queries. If your entire web site's content is served from an Ajax query, then search engines won't be able to see your content. The solution to using Ajax and providing content to search engines is to use a combination of Ajax and non-Ajax. For example, if your web site has a list that is sortable and searchable with Ajax, make sure that your server-side language populates this list on the first page load. Doing so will give search engines at least part of your list, along with allowing you to provide your users with a fancy Ajax interface. A good rule of thumb when determining what content search engines can and cannot see is to simply view the source of a web site in your browser. You'll notice that you will see the content that was created on the initial page load, and not content that was a result of Ajax queries. The source code that you see when viewing the source of a web site is the same source that a search engine will see.

Search engines aren't able to see content that requires logging in with a user name and/or password. If every page on your web site requires being logged in, search engines won't be able to learn anything about your web site. For this reason, a common practice is to offer two versions of each page: one that is viewable by users that are logged in, and another that is viewable by users that aren't logged in. Typically the version that is viewable by users who aren't logged in is more minimal and exposes only a small amount of the page. Again, offering two versions of every single page is excessive; only offer two versions for pages that you want to affect your SEO.

### Frequent Updates

Most search engines value web sites that are frequently updated more than web sites that are stagnant. For this reason, typically blogs and news sources perform better in search engines. However, update frequency is only a small part of the equation. If you have a really good cat web site, then the chances are good that you'll rank highly in search results even if your web site isn't updated very frequently. Update frequency isn't something you should worry about much when designing your web site; it has been included just for completeness.

The techniques described above have shown ways to improve SEO from a design and code point of view. These techniques are very important, but they aren't the largest factors that affect SEO. The most influencing factor of SEO is the way that other web sites link to your web site. The reason search engines rely on other web sites to gauge your web site is because web site owners can lie about their content. That is, spoofing the `<title>`, `<h1>`, `<meta>` tags, and content of a web site is as easy as changing the HTML. If I run a web site that sells advertisements, I could create a web site that, when described by my tags and content, looks like a web site about the 2008 presidential election. If search engines didn't rely on other web sites to describe my web site, then my advertisement site that really just consists of a lot of meaningless content or ads might be the top search result for searches about the 2008 presidential election. Clearly this is problematic, so by using lots of other web sites to

characterize a single web site, search engines are able to create a very accurate categorization for a certain web site, without relying on liar web site owners.

### link analysis

The analysis of incoming and outgoing links for a certain web site.

Search engines consider two properties when analyzing a web site according to its incoming and outgoing links: PageRank and link anchor text. PageRank and link anchor text are both means of *link analysis*, the task of examining incoming and outgoing links on a page to make conclusions about its content and relevance. Both are discussed in detail in

the following sections.

### 16.3.4 PageRank

#### PageRank

An algorithm created by Google that assigns relevance to a page based upon the number and relevance of other sites that link to it.

*PageRank* ranks your web site relative to the rest of the internet, based on the links that point to your web site. PageRank is an algorithm created by Google that revolutionized the search engine industry and is a large reason why Google was significantly better than its competition in its early days. PageRank is one of the largest factors in a web site's search ranking. The algorithm is fairly complex, but the circumstances it creates are easy to understand. In general, web sites with lots of incoming

links have better PageRank than web sites that do not have many incoming links. Furthermore, if a web site has incoming links from web sites who themselves have high PageRank, then that web site will have a better PageRank.

To be even more general, your web site will have better ranking if it has lots of incoming links and if those incoming links are coming from web sites with high PageRank. Figure 16.5 shows a graphical representation of PageRank, where each circle represents a web site. The diameter of the circle represents the PageRank for that web site, and the arrows represent links. If an arrow travels from A to B, then this means that web site B has an incoming link from web site A. The diameters are meant to be general comparisons of ranking and are not necessarily drawn "to scale."

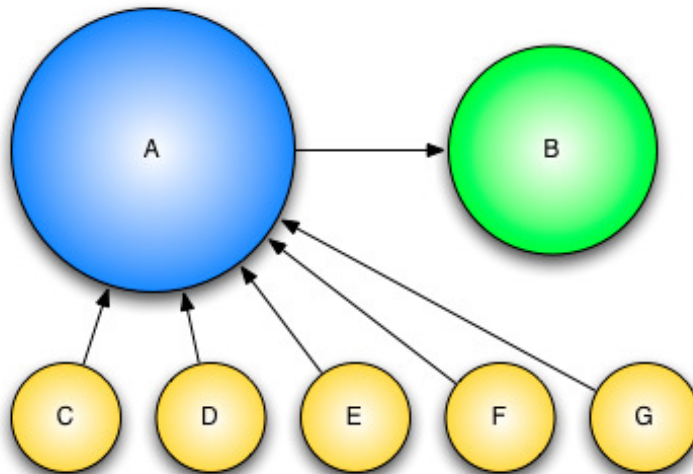


Figure 16.5 PageRank algorithm

Notice that web sites C through G have a relatively small PageRank. This is because they do not contain any incoming links. In contrast, web site A has a large PageRank because it contains many incoming links. Similarly, web site B has a large PageRank because its single incoming link comes



from web site A, which has a very large PageRank. The reason for web site A having a larger PageRank than web site B is complicated and requires a much deeper understanding of the PageRank algorithm. Consult Wikipedia if you are interested in understanding this difference.

The takeaway from PageRank is that your web site will have significantly better search ranking if it has incoming links and if those incoming links are from reputable web sites. Some companies whose businesses rely on good SEO employ teams of people who literally contact web sites and ask for incoming links from them. Some companies also hire PR firms who try and get news sources and high-profile blogs to write stories about these companies, which generate incoming links. Having a dedicated team or a PR firm getting incoming links for your web site may not be in the cards, but that doesn't mean you won't get incoming links.

If you make a good, useful web site, then you will naturally get links to your web site, either from bloggers who use your site or maybe even from news sites that are interested in your web site for a certain journalism piece. Furthermore, you can try to do the following to improve your PageRank:

- Get friends with blogs to write about your web site, which can help get you started.
- Create embedded features (like YouTube's embedding) and include a link to your web site in the embedded HTML, giving you an incoming link for each site that embeds your feature.
- Try to create incentives for people to link to your web site.

### Incoming Link Anchor Text

PageRank tells search engines the overall rank of a page relative to the rest of the internet, but it doesn't help search engines create keyword mappings for each web site on the internet. For keyword mapping, search engines look at the anchor text of incoming links, because the chances are good that this anchor text will describe the web site being linked to. For example, if I link to CNN with a link that says, "Great News Source," then I've essentially told search engines that my site thinks CNN is a "Great News Source." Search engines value incoming link anchor text very highly when determining the keywords that a certain web site is associated with. That is, the links to your web site from other web sites will largely impact your SEO.

Not only should you strive for incoming links, but you should also push for incoming links with anchor text that matches the keywords that you want your web site to be associated with. For example, if you have a cat web site, then you want incoming links to your web site to have "cat" in the anchor text. Similarly, if you're Amazon.com, then you want incoming links to have an anchor text of "online shopping," etc. Again, search engines value incoming link anchor text highly. Unfortunately you can't usually control incoming link anchor text; instead the web site linking to you controls the link anchor text. However, there are certain situations when you can control the incoming link anchor text. For example, if you create an embedded feature like YouTube videos, then you can include in the embedded HTML a link to your web site with whatever anchor text you want.

External links from other web sites impact SEO greater than incoming links from one of your pages to another. However, the anchor text used in your internal links also helps search engines create a keyword mapping for your web site. For this reason, be thoughtful when choosing the text for links that link to other pages within your web site. For example, try to avoid using anchor texts like "this," or "here," or "view more." Instead, use links like "More Cat Information," or "See more online shopping results," etc. You'll notice that keyworded anchor text is longer, requiring more space on your web site. Though keyworded anchor text may improve your search rankings, it might also hinder your user experience. You should never let SEO interfere with your web site's ability to be easy to use, because in most cases your users' love for your web site outweighs your SEO.

#### anchor text

The text that is displayed to the user in a hyperlink between the `<a>` and `</a>` tags.

### "nofollow" Tags

Not all incoming links give you an increase in PageRank. An anchor tag can have a `rel="nofollow"` attribute, which tells search engines to ignore the link entirely. Some places where "nofollow" tags are used are Wikipedia pages, blog comments, and Facebook profiles. If you have an incoming link from Wikipedia, the link will not help your PageRank because Wikipedia adds "nofollow" tags to all of their outgoing links. The "nofollow" tags are primarily in place to hinder spammers. For example, if Wikipedia didn't add "nofollow" tags, then spammers would have a huge incentive to edit Wikipedia by adding useless links to their web sites. These useless links would pollute Wikipedia, making it less useful, while improving SEO for malicious web sites. Generally speaking, incoming links are good, but it's important to realize that not all incoming links help SEO.

Services exist that claim they can increase your search engine performance as long as you pay them a certain price. Some of these services legitimately improve your ranking by the techniques described in this section, and other services are considered malicious and can possibly lower your SEO. It's generally a good idea to not invest money in services that claim to increase your SEO. Instead, pay attention to the techniques described in this section and grow your search ranking for free.

This section has discussed search engine optimization in great detail. SEO should always be in mind when building your web site, because decisions you make can make your web site perform much better in search engines. Just always remember that search engines try to give their users the best results they can, so if you have a good web site, you will probably perform well in search engines. However, following the advice given in this section will only help your search ranking. Always strive for a good ranking, because an unbelievable amount of your traffic can come from search engines. A good tool for evaluating your SEO is Google Webmaster tools; Webmaster tools offers information about your web site ranging from your PageRank to the number of incoming links your web site has, etc. Use the following SEO checklist as reference when building and improving your web site:

- Have a descriptive `<title>` and `<h1>`
- At least use the `<meta>` description tag
- The other `<meta>` tags are useful, but not as important as the description tag
- Make your content available to search engines
- Don't embed content in flash or images
- Be sure that at least some of your content is accessible by users that aren't logged in
- Be careful when using Ajax, Flash, etc.; make sure at least some of your content is loaded on the page load in plain HTML
- Always think about PageRank
- Always think about the anchor text other web sites are using to link to you
- Choose your link anchor text wisely
- Understand "nofollow" tags

At the time we were creating Cellarspot, we didn't know much about SEO. However, as we improved the site, we learned more about it. Later web sites that we built after Cellarspot used these techniques right off the bat, allowing them to grow in traffic much more rapidly than Cellarspot did.

Nearly half of Cellarspot's new visitors came from a search engine. This amazed us and made us realize how important SEO really is. SEO can literally make or break a web site, so thinking about SEO is absolutely essential when you are building a new site. The mechanism we used to measure how many users visited us from search engines will be discussed later in this chapter.

### 16.3.5 Google Analytics

Once you've explored ways to design and drive traffic to your web site, you'll want a way to evaluate your design and understand more about the traffic you're receiving. Being a web site owner and designer, you want to be able to measure traffic in an accurate way to see what features are helping, where users are coming from, and how users are using your web site. Fortunately for web site owners, an unbelievable, free tool is offered by Google called Google Analytics.

#### Google Analytics

A tool that provides traffic and usage statistics for web sites that have signed up and installed a JavaScript.

*Google Analytics* (GA) is a web site that allows you to track your user's behavior while they are using your web site. GA allows you to understand your user by providing a dashboard of useful statistics, graphs, and data points. GA records usage information about your web site only under two conditions: first, you must sign up for a GA account at <http://www.google.com/analytics/>, and second, you must insert a small JavaScript at the bottom of every page of your web site. Upon signing up for a GA account, you'll be given the specific JavaScript to include on every page. With an account created and the JavaScript installed, Google manages to generate an incredibly verbose report of your web site. GA gives you the following statistics:

- Number of unique visits per day
- Number of total page views per day
- Breakdown of how users came to your web site (direct, referrals, or search engines)
- Search keywords users searched for when they researched your web site
- Most visited pages
- Average time spent on your web site
- Screen resolutions
- Browsers used
- Operating systems used
- Geographical location, broken down by city, of your users
- Much more

GA allows you to look at the above statistics for any time period. That is, you can look at any of these statistics over any timeframe, such as a single day or a span of many years. Not only does GA give you numerical figures for these metrics, but they also provide attractive, comprehensive graphs. Take a look at the following screen shots of GA for Cellarspot.

Figure 16.6 shows the opening Cellarspot GA dashboard. The large graph you see shows how the number of unique visitors has varied from the time of April 1, 2007 through May 31, 2007. The information below the graph shows high-level, aggregate information about the total site. *Bounce rate* is the percentage of visitors that leave immediately after viewing the first page they see. If a user lands on your web site and leaves immediately without looking at any other pages, they have "bounced."

Figure 16.7 shows the breakdown of visits and how visitors reached Cellarspot. Direct Traffic is traffic that came from users typing "cellarspot.com" into their web browser's address bar. Search Engine traffic is traffic that came from search engines like Google or Yahoo. Referring Sites traffic is traffic that came from other web sites via a link.

Figure 16.8 shows the search keywords that users typed to lead themselves to Cellarspot. This information can be extremely helpful in determining what users were actually interested in that brought them to a site, or what they perceived as being the content of the site.

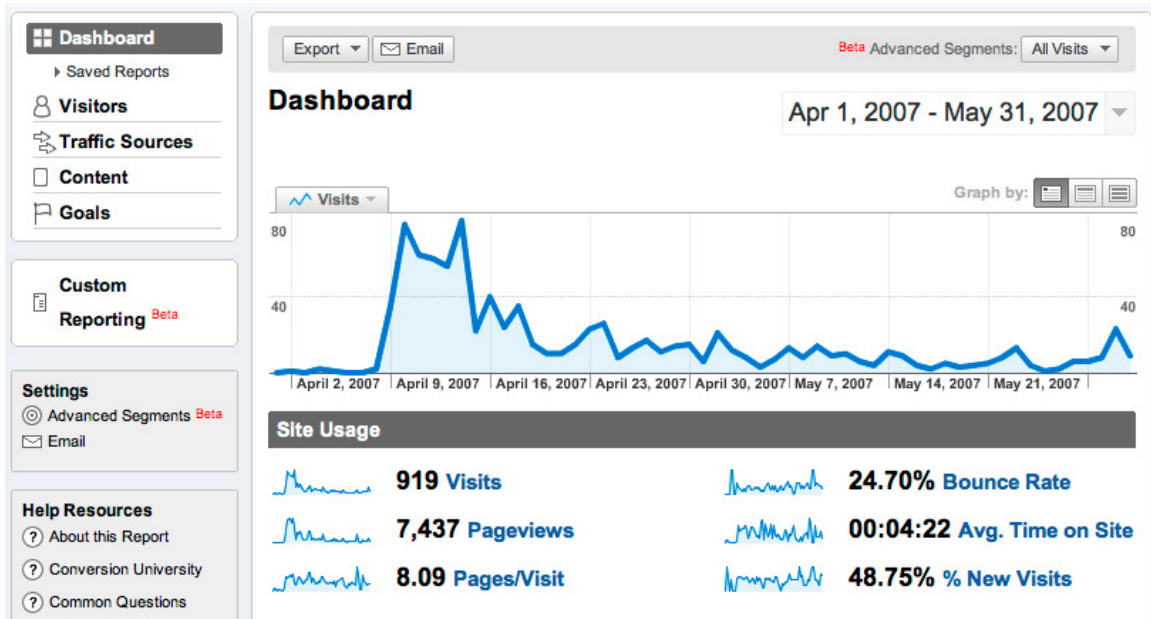


Figure 16.6 Google Analytics dashboard for Cellarspot

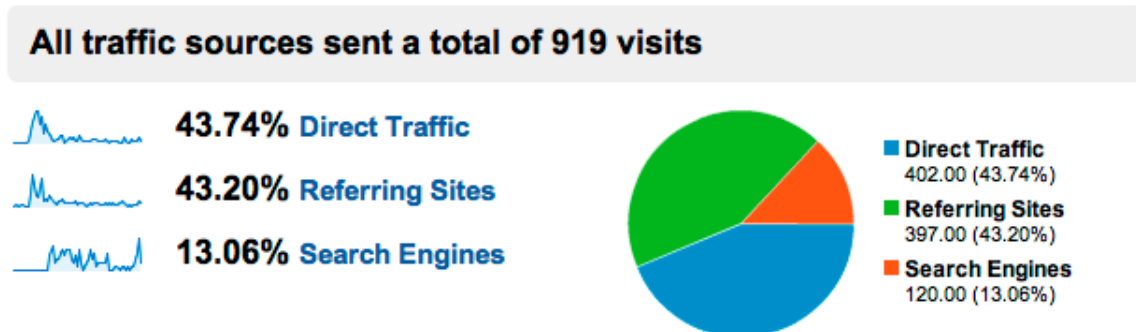


Figure 16.7 Google Analytics traffic sources for Cellarspot

Dimension:	Keyword	Visits	Pages/Visit	Avg. Time on Site	% New Visits	Bounce Rate
1.	cellarspot	84	8.31	00:04:06	13.10%	9.52%
2.	bossi siena	9	5.78	00:04:18	11.11%	33.33%
3.	cuvee napoleon	7	2.71	00:04:43	14.29%	57.14%
4.	les charmes chardonnay	4	1.50	00:00:59	25.00%	25.00%
5.	"alex loddengaard"	1	3.00	00:00:47	100.00%	0.00%

Figure 16.8 Keywords used to search for Cellarspot



**Figure 16.9** Geographic location and browsers used by Cellarspot visitors

The power that Google Analytics provides is impressive, especially because it's a completely free service. GA can be used to greatly refine your web site idea or focus. For example, you may find that your users spend most of their time on Feature A and very little of their time on Feature B. With this information, you can focus more of your time on improving Feature A. Or perhaps you would rather change your web site's design to put more focus on Feature B and less on Feature A. GA lets you understand your users to a point where you can see what pages they are visiting, how they are finding your web site, and what browsers they are using most.

Google doesn't charge any money for GA because they use its data to better understand the internet and its usage. Though Google doesn't allow others to see your web site's usage statistics, Google does use the data for internal purposes. The information is mutually beneficial: both you and Google can learn a lot about how your users use your web site. Without offering GA for free, Google wouldn't have access to all of this data about internet usage.

Google Analytics can help you understand how effective improvements to your web site are. You can measure the increase in traffic to your web site both before and after you launch a new feature. Similarly, you can measure the number of visitors who found your web site from a search engine before you implement a new SEO feature such as an embedded feature. You can then wait a few weeks and re-measure the number of visitors coming from search engines to see how that feature affected your SEO. Similar before and after tests can be performed on any metric that may be interesting to you: geographic visits, total visits, visits from a specific keyword, etc. GA is a perfect medium for understanding the impact certain parts of your web site have on your web site as a whole.

Google Analytics, given that it gives you specific keywords people searched for when they clicked on your web site, can also give you a good understanding of the keywords users are visiting your web site with. This can be helpful when deciding what your `<title>`, `<h1>`, and any link anchor text should be. Catering these elements to the keywords users search for to visit your site will only improve your web site's performance for these keywords, which will drive more traffic to your web site.

Google Analytics provided priceless information about Cellarspot. We learned a number of things ranging from what browsers we needed to support to the types of pages that were most visited from search engines. We also realized that users didn't often visit the Cellarspot front page first. Instead, they often landed on pages about wine when they came from search engines. This made us completely rethink the wine pages. We needed to provide information about Cellarspot on these pages, otherwise new visitors would most likely leave Cellarspot after they read the information they wanted to read on the wine page. Cellarspot was once linked to by a popular wine blog, which resulted in a huge traffic spike. Without GA, we wouldn't have known that we got this link.



## 16.4 Making Money

This section will outline a few simple, common ways for a web site to make money, some of which are available to any web site owner and free to install and get running.

### 16.4.1 Google AdSense

#### Google AdSense

Google's advertising service that allows you to place ads on your web page and make profits when users click the ads.

*Google AdSense* is an advertising service that displays ads purchased by AdWords customers on your web site. AdWords lets customers buy words for advertising purposes, while AdSense decides which participating web sites should show the ads. AdSense is clever about the ads it displays, only displaying ads that are similar to the content on your web site, but not displaying competitors of your web site. You are paid according to the number of clicks made on the ads displayed on your web site; as mentioned in the AdSense section, this type of advertisement program is known as cost per click, or CPC. The amount each click is worth is dependent on the popularity of your web site and of the web site providing the advertisement. Perhaps you've never heard of AdSense, but you've most certainly seen AdSense ads. Figure 16.10 shows an example of an ad served by Google AdSense.

According to the number of clicks made on the ads displayed on your web site; as mentioned in the AdSense section, this type of advertisement program is known as cost per click, or CPC. The amount each click is worth is dependent on the popularity of your web site and of the web site providing the advertisement. Perhaps you've never heard of AdSense, but you've most certainly seen AdSense ads. Figure 16.10 shows an example of an ad served by Google AdSense.

Figure 16.10 AdSense advertisements

AdSense, like Analytics, is free to sign up and easy to install. Upon creating an AdSense account, you design the look and feel of your ads and are given a JavaScript block to place in the page where you want the ads displayed. Once it is installed, you can view statistics about your ads on the AdSense web site such as click rates, dollars made, dollar amount made per month and per day, along with other statistics. To sign up and get started, visit <http://www.google.com/adsense/>.

The placement of AdSense ads is a key to their success. Ads placed in sections of a web site that users are most likely to notice have a higher chance of being clicked. For example, ads displayed at the top-left of a web site are much more likely to be noticed and clicked on than ads being displayed at the bottom of a page. However, ads that hinder the user's experience on your web site may result in those users becoming annoyed with your web site. There is a balance between placing ads to make money and placing ads not to be intrusive to the user. For this reason, AdSense ads are most commonly installed in the right sidebar of a web site, where users are likely to look but not be annoyed.

Cellarspot ran several AdSense advertisements on a few different pages. We saw a 0.7% click-through rate (CTR), which means that 0.7% of our displayed ads were clicked on. That probably sounds pretty small, but that's not far off from average CTR on the web. Over the course of about 18 months, we saw 71 clicks and made a whopping profit of \$35. The most significantly limiting factor of getting lots of clicks was our small amount of traffic. Generally speaking, with more traffic comes more advertisement clicks. However, having AdSense on Cellarspot was better than not having any ads at all. Installing AdSense was easy, and the ads were not annoying to the users.



### 16.4.2 Referral Programs

*Referral programs* are programs that let web site owners make money by directing users to other web sites. They work in the following way. First, a web site owner signs up to be a referrer for another web site, say Amazon.com. Next, the referred site (Amazon.com in this case) gives the web site owner a special URL that, when linked to, will track whether or not a user came from the web site owner's web site.

#### referral program

A program that lets web site owners gain a commission on sales that were sent to a given destination from their web site.

The special URL contains a series of query parameters (GET variables), which are what track where a user was referred from. If a referred user purchases something on Amazon.com, then the web site owner will receive a commission on the purchase, usually ranging between 5% and 10%. Many large web sites such as Best Buy, Amazon.com, and many others allow web site owners to sign up for a referral program.

Referral programs are often useful ways to make money. For example, if your web site is a book club social network, then providing a link that allowed your users to purchase the books being discussed would be a very useful service to your users; highly reviewed books will have a good probability of being purchased by your users. Referral programs let you provide a useful service to your users by giving them a clear path to purchasing something, while allowing you to gain a commission on sales. Your users usually won't even know that you're receiving a commission.

Signing up to be a referrer is free and easy, which makes it a low-risk program to be involved with. On the other hand, force-feeding your users links to purchase things can hinder their experience. You should use a referral program only when it's natural and useful for your users to be referred to a product on an e-commerce web site. Often referral programs may even provide the ability to embed product widgets on your web site, making purchasing even easier for your users. Amazon.com is particularly good at providing non-intrusive, useful embedded features to their referrers.

In order to sign up for a referral program, first you must find one that sells products that your users might be interested in. Once you have found a web site that you'd like to be a referrer for, take a look at the very bottom of their web site. Usually web sites offering referral programs will have a link that says something along the lines of, "Become a Referrer," or "Referral Program." For example, today, Amazon.com's referral link says, "Join Associates." Also note that simply having links to Amazon.com will not earn you a commission; you need to be sure to sign up with the referral program and use the special URL that they provide you.

We signed up to be a referrer for [wine.com](http://wine.com) through a third-party referral company called Link-Share. Our bottle information pages had a link to [wine.com](http://wine.com) to purchase the described bottle. Over the course of 18 months, one of our users purchased a \$130 bottle of wine, giving us a \$13 sales commission. Friends of ours have done very well with Amazon.com's referral program, some making on the order of hundreds of dollars each month.

### 16.4.3 Banner Advertisements

*Banner advertisements* are images, or possibly animated flash videos, that are displayed on a web site. Banner ads are similar to AdSense ads in that they take real estate on a web site and in return pay to be displayed there. Banner ads differ from AdSense in the way deals are negotiated and in the way the web site owner is compensated. As learned previously, AdSense works on a pay-per-click basis, where the web site owner is paid each time a user clicks on an ad. Banner ads, however, are paid on a cost-per-thousand (CPM) basis. That is, advertisers pay a dollar amount for each set of 1000 impressions, or page loads, that their ad is displayed on. If you were wondering, the "M" in CPM comes from the Roman numeral M, for 1000.

#### banner advertisements

Image or flash-video ads that are displayed on a web site.

### CPM

An advertisement pricing scheme based on thousands of impressions, or page loads.

CPMs can range anywhere from a few cents to tens or even hundreds of dollars, mostly depending on the popularity of the web site displaying the ads. For example, CNN.com's CPM will be significantly higher than a web site with significantly less reputation and traffic.

CPMs also vary depending on the size and placement of the ad, along with the actions of the ads. Popup ads actually cost a lot of money; advertisers pay more for popups than they do for ads that show up in the web site. The reasoning behind having CPMs vary is because some actions grab the user's attention more, hindering their experience. Advertisers, by purchasing popup ads, are essentially paying for the derogated user experience they incur.

Banner ads are most commonly found on large popular web sites. The main reason for this has to do with the way banner ads are bought and sold. In the case of AdSense, Google does all the work to match advertisers with web sites. Banner ads, however, require the web site running the ads to have a sales team to negotiate with advertisers on price, duration, and availability. Banner ads often yield higher revenues, but they also incur an operational cost to sell, get them installed, provide statistics to advertisers, etc. Banner ads require a lot more work, work that should be spent elsewhere in a new, unproven web site. Also worth mentioning is that many people have banner advertisement blockers enabled in their browsers, which means that some users don't even see banner ads.

Cellarspot never ran any banner ads, because we never had the bandwidth to negotiate, install, and coordinate with advertisers. However, if Cellarspot had grown to be a large, popular web site, then we would have hired a sales staff to manage the selling of banner ads. Again, banner ads generally yield more money, mostly because of their pay structure.

I recommend that you start simple with AdSense, and once your web site has proven itself, invest time and money in a program for banner ads. This is the common pattern that most sites follow.

### 16.4.4 Other Options

Though selling ads and participating in referral programs are easy ways to make a web site profitable, there exist plenty of other ways to make money. Some web sites offer simple services for free and charge a fee for their more advanced features. Remember The Milk is structured in this way: users get basic to-do lists for free, but they have to pay a fee for the more advanced features.

Some web sites survive by accepting donations, either by PayPal or some other donations mechanism. The best example of a donations-driven web site is Wikipedia, which is entirely funded by charitable contributions. Lots of businesses also use their web site as a medium for selling a product or service. Netflix is a good example here; they have created a powerful, easy-to-use web site that serves as a portal to their core business, completely rethinking the way movie rentals works.

Though this chapter has made an attempt at offering means for web sites to make money, its recommendations are not comprehensive. There exist plenty of other ways for web sites to make money; the mechanisms discussed in this chapter are the mainstream approaches.

We considered creating a subscription service for Cellarspot. Many of Cellarspot's users are wealthy, older males, who are familiar with subscription services. That is, people who grew up as late as the 1970s have been paying subscriptions for many of their services: magazines, cable TV, stock market news letters, etc. A subscription-only web site wouldn't be too unfamiliar to these users; many of them would be willing to pay, as long as the service they were paying for was worth the fee.

Generation Y, the generation who is growing up in the 1990s and 2000s, will be much less open to paying a subscription fee, mostly because so few web sites these days cost any money to use. Regardless, some web sites make sense to charge a subscription fee. It's up to you as the web site owner to determine what the best approach to making money is.

### 16.4.5 Funding Your Web Site

With any good idea come individuals and organizations interested in investing in that good idea. Web sites are no different than good ideas in this case. Startup companies, some of whose products are web sites, often spend time at their early stages to raise money through a set of investors. Money can come from many different sources, mostly dependent on the type of entity that is receiving the money. In academia, most fundraising comes from research grants. In industry, most fundraising comes from venture capital firms and angel investors. Typically startups use investment money to pay salaries, rent an office, buy office supplies, and pay for infrastructure required to build a solid product. Typically personal project web sites don't need investments, because they can be built in one's free time and require little money to operate. Companies such as Facebook, YouTube, Google, and others received money from venture capitalists when they were first getting started.

#### venture capital (VC)

Investment money that comes from a venture capital firm, who invests in a company because they see a prominent future in it.

*Venture capital* firms are firms whose job is to determine which startups are probable to succeed, and then to invest large sums of money into that company to help bootstrap their success story. In return for a venture capital's grant of money, the firm receives a significant ownership stake in the company that is being invested in. *Angel investors* are similar to venture capital firms in that they provide money and receive an ownership stake. However, angel investors are wealthy individuals, or groups of wealthy individuals.

#### angel investor

A wealthy individual or small group of individuals who invest money in companies.

*Research grants* typically come from colleges, non-profit organizations, or other academic entities. They do not usually involve an ownership stake. Instead they are merited on the furthering of an academic area. Generally, angels and venture capital firms invest in companies, while research grants are given to students, research groups, professors, and scientists.

Receiving funding involves a thorough application process. With all three of these mediums, the web site owner must prepare a plan as to how their product will succeed, and why it is important in today's world. Next, a presentation will be given to those interested in investing: a sales pitch. Once a group has agreed to fundraise, terms are agreed upon, and the web site owner receives the money.

Typically investors won't invest money in a company until that company has a comprehensive idea of the business they want to run. Typically investors are interested in seeing a prototype web site as well. Unfortunately, given that so many individuals want to start internet companies, getting access to angels and venture capital firms typically requires that you know someone close to these investors.

You should only take fundraising if you absolutely need to do so. In personal projects, fundraising probably isn't necessary or desired. With fundraising and investing comes pressure and expectation from those who gave you the money. Often with personal projects you don't want obligations; you want fun. In a business, fundraising is necessary to put food on the employees' tables.

Investors, like the rest of the economy, are subject to ups and downs. For example, towards the end of 2008, when the financial crisis seemed to be at its lowest point, venture capital firms decreased the amount of money they were investing by a significant amount. Though some startups were getting funded, the number of funded startups greatly decreased compared to the previous year.

Cellarspot didn't take any investing, because we always considered it more of a personal project than a business, at least while we were starting. We had hopes to one day become a business and at that time evaluate the need for investing, but our plan was to go as long as possible without looking for fundraising. We paid for our server and its hosting out of our pockets.

Perhaps fundraising would have bettered the financial burden of buying a server and hosting it, but we just wanted to have fun and make something cool. Fundraising can often take the fun out of things with the obligations that come along with receiving someone else's money.

## 16.5 Legal Issues

As if going live with a web site wasn't complicated enough, a few risks, dangers, and obligations should be understood in order to avoid potential legal offenses while you launch and operate a site.

### 16.5.1 Intellectual Property

*Intellectual property (IP)* is thoughts and ideas that are owned by a company or individual. Three of the most common types of IP are copyright, trademarks, and patents.

#### copyright

The implicit right of ownership and distribution one gets when creating original content.

*Copyright* allows the creator of a material entity to maintain ownership over the entity, prohibiting others from stealing or copying that entity. Trademarks allow companies and individuals to claim ownership of logos and phrases to prohibit others from using their branding. Copyright and trademarks are the most common forms of IP on the internet

Copyrights are the rights one has by being the creator of source code, images, or any other content. Copyrights are implicit, meaning that the creator of an entity naturally owns it, even without explicitly filing paperwork. This means that you cannot copy resources from other sites such as source code, images, or text. Doing so will be a violation of copyright law, and depending on the severity of your plagiarism, could result in serious legal action against you. On the other hand, you also do not need to do anything to establish ownership of your web site's source code.

Licenses can be created to accompany intellectual entities, which allow those entities to be used by other people. For example, the Prototype JavaScript framework comes with a license describing the terms under which anyone may use it. Along these same lines, many web sites provide free licensed downloadable icons and images known as *stock images* or stock icons. Examples of these stock icon and image web sites are [istockphoto.com](http://istockphoto.com) and [iconbuffet.com](http://iconbuffet.com). Make sure a license exists whenever you use an intellectual entity, such as source code or an icon, that you didn't create.

#### trademark

A legal record of owning a logo, name, or phrase.

A *trademark* establishes legal ownership over a name, logo, or phrase. Trademarks are like copyrights, except that trademarks have more to do with branding. By registering a trademark on your web site name, no one else is allowed to ever use the same web site name as you. The same fact is true with logos and phrases. For example, T-Mobile has a trademark on the marketing phrase, "Stick Together." This implies that no other company is allowed to use this phrase in marketing a similar product. Trademarks are registered through the United States Patent and Trademarks Office.

Though copyrights and trademarks are significant legal formalities that can impact you as a web site owner, you do not need to worry about them much while you're getting started. Registering a domain name is a good way of protecting your web site's name. Only companies usually apply for trademarks, because trademarks are fairly expensive to apply for, and are overkill in the case of a small, new web site. In general, you should try to choose a name and logo that are unique, and just build a cool web site. You can worry about the copyright and trademark obligations later.

A *patent* allows a company or individual to claim ownership of an idea, where others cannot use their idea, or at least have to pay to use it. Patents will not be discussed in this section, because they are the most complicated of IP. Filing for patents usually costs thousands of dollars in legal fees.

We never registered any trademarks for Cellarspot, because they're neither affordable nor necessary for small web sites. We also created all of our source code, images, and content, so we didn't have to worry about copyrights and licenses. Had we written more JavaScript, we would have probably used either the Prototype or jQuery library, which would have been legal. Each of these libraries comes with a license that allows anyone to use them for free.

## 16.5.2 Legal Liabilities

Believe it or not, but being a web site owner could make you legally liable for illegal activity that occurs on your site, such as copyright violations. Examples of copyright violation are using images, source code, and any other content in your web site that you got from different web site. If you violate copyright in a serious way, such as by creating an identical clone of another web site or by running an illegal software piracy service, then the owner of the copyright you are violating can take you to court. If you are an individual and not a registered company, then your personal assets, such as your laptop, car, or kitchen table can be taken from you if you're not able to pay for the legal penalties that you have accrued by violating copyright. Another interesting example of liability involves users' personal information. If you store your users' personal information and a hacker breaks into your web site and publishes that private data, you'll have a lot of angry customers. Some of them may even attempt to sue you for fraudulent use of that information, such as misuse of credit card numbers.

In order for a web site owner's individual assets to be protected, one must register a company. By registering a company, an individual is essentially putting all obligations on the company instead of on the individual. There exist many different types of companies, such as *Limited Liability Corporations (LLCs)* and corporations, and each of these has different legal requirements. Companies are registered through the state where the business is being run. Typically the process entails lawyers and lots of paperwork.

Don't worry; you're really not running a risk by operating a web site. The circumstances when a company would go after an individual's personal assets are very rare. In cases like these, typically the company would spend more money on legal and court fees than they would snagging an individual's assets. That is, a company will nearly always lose money if they go after an individual, depending on that individual's wealth and legal offenses. You, as a good, honest web site owner, are probably safe from legal persecution. In most cases you would get some sort of letter asking you to remove offending content before any legal action would be taken.

### Limited Liability Corporation (LLC)

A type of registered company that protects its owners' personal assets.

Cellarspot was never registered as a company, because the risk of our personal assets being seized was practically zero. We ran an honest web site, one that didn't use someone else's copyrighted material. Also, none of us had enough assets to make a court battle worth it for anyone.

## 16.5.3 Security Issues

You have an obligation to your users to secure their data as best you can. For this reason, a few measures can be taken to help tighten your web site's security. First, user passwords should never be stored unencrypted in your database. Instead, encrypted hashes should be stored, MD5 hashes being the most popular. An *MD5 hash* is an encrypted version of a normal string that is very difficult to decrypt. Storing a MD5 hash instead of a plain-text password means that when you perform checks to see if a user's login and password match, you'll have to compute a MD5 hash of their password before doing the comparison. Most web languages such as PHP and Python have built-in functions for computing MD5 hashes. Other techniques such as "salting" the MD5 hash can strengthen encrypting even more.

Wireless networks have grown to be very widespread these days, and most people don't realize how insecure they are. A malicious user can very easily see how other users on the same wireless network are accessing the internet. These malicious users can even see the form data that other wireless users submit to unencrypted web sites. For this reason, sensitive input fields in a form such as pass-

### MD5 hash

An encrypted version of a string that is hard to decrypt.





word fields should be encrypted on the client-side, in JavaScript, before they are submitted. In practice, this means using JavaScript to compute a MD5 hash of a user's password, and submitting the MD5 hash to the form, not the plain text password. Unfortunately, JavaScript does not have a built-in MD5 function. Fortunately, many JavaScript frameworks exist such as jQuery that do provide the MD5 function. Often web sites will use the Secure Sockets Layer (SSL) over HTTPS (Secure HTTP) to encrypt the entire communication between the client and the web site.

Security is a large and complicated topic, one that deserves its own book. You are encouraged to search Google and Wikipedia to learn more about securing a web site, because launching a web site is the real deal. Security issues with your web site can be a major deterrent for your users, making it important for you to do diligence on understanding how your web site may be vulnerable or insecure.

Cellarspot generated a plain MD5 hash of a user's password in JavaScript before any form was submitted. Then on the server side, we computed a second MD5 hash, first by "salting" the original hash, before storing the encrypted password in the database. We would perform the same operations when comparing emails and passwords for login purposes. We never had any issues with security, because we were careful to encrypt sensitive information.

Cellarspot, being a social network, also had to follow through in our promises to hide information according to a user's privacy settings. We conducted thorough testing to make sure that a user's privacy settings were accurately depicted in our web site.

Cellarspot was lucky for being a relatively low-traffic site or a small target for hackers. Small web sites often slip under the hacker radar. Hackers typically spend their time being malicious against big web sites such as Google, Amazon.com, etc. The reasoning behind hackers going after large web sites is that if they bring down a large web site, then they can brag that they did so. On the other hand, if a hacker hacked a small web site, then they wouldn't have much to brag about. With that said, we still took security very seriously with Cellarspot. Users trusted us with their data when they signed up, and we had all intentions of keeping our promise of security and privacy.

### 16.5.4 Conclusion

With the closing of this chapter, you're now ready to go live with your web site and show the world what you are capable of and what you worked so hard to build. You have learned about many things, ranging from maintaining a web site with log messages to building features that will improve your web site's search engine optimization. The road to learning doesn't end here, though; there is always more to learn, and always more to do.

Just always remember, the true beauty of the internet is its low entry point, the fact that the gap between a personal project and a thriving, widely used web site is smaller than you think. Even if your web site doesn't become one of the internet hotspots, you'll still have gone through an informative, useful process that you will benefit from greatly. Your "startup," if you will, can be put on your resume, used for job interviews, fundraising proposals, and much more. Go live!

I wouldn't trade my Cellarspot experience for anything. We didn't make a lot of money or attract a lot of users, but we enjoyed the journey that Cellarspot took us through. I have used Cellarspot as a focal point in many of my software interviews, and most employers are eager to hear more about the work I did. I measure Cellarspot's success on the ways in which it has made me a better programmer, a better project manager, and a better technical person altogether. In this regard, going live with Cellarspot was an invaluable experience, not to mention extremely fun and exciting.

I have told my story. Now it's time for you to tell yours!



## Chapter Summary

- The gap between being a hobbyist web programmer and running a full-fledged, real web site is very small. With the right knowledge, it is very doable to "go live" with your own site.
- Do research by talking to users and examining competitors to ensure your web site will solve a real problem.
- Be thoughtful when deciding on the technologies you use for your web site; some technologies such as Ruby on Rails or Django allow you to write significantly less code. Some web hosts only support certain technologies.
- With regard to hosting, start small and cheap and upgrade as your web site's traffic grows.
- Design your web site in a way that will allow you to more easily debug and test it once you launch it.
- Google AdWords, viral marketing features, and search engine optimization are all good ways to drive more traffic to your web site.
- Google Analytics is a great tool for measuring your web site's traffic and usage.
- Google AdSense, referral programs, and banner advertisements are just a few ways for web sites to make money.
- Fundraising, though not necessary, can help startup web sites get kick started.
- Creating a web site means you have to abide by intellectual property laws and watch out for various legal obligations related to your site's content.

## References

- Wikipedia – Use Case: [http://en.wikipedia.org/wiki/Use\\_case](http://en.wikipedia.org/wiki/Use_case)
- Apache Web Server: <http://httpd.apache.org/>
- Google AdWords: <https://adwords.google.com/>
- Wikipedia – Search Engine Optimization:
  - [http://en.wikipedia.org/wiki/Search\\_engine\\_optimization](http://en.wikipedia.org/wiki/Search_engine_optimization)
- Wikipedia – PageRank: <http://en.wikipedia.org/wiki/PageRank>
- Wikipedia – nofollow: <http://en.wikipedia.org/wiki/Nofollow>
- Google Analytics: <http://www.google.com/analytics/>
- Google Webmaster Tools: <https://www.google.com/webmasters/tools/>
- Google AdSense: <https://www.google.com/adsense/>
- Wikipedia – Intellectual Property:
  - [http://en.wikipedia.org/wiki/Intellectual\\_property](http://en.wikipedia.org/wiki/Intellectual_property)
- Wikipedia – Copyright: <http://en.wikipedia.org/wiki/Copyrights>
- Wikipedia – Trademark: <http://en.wikipedia.org/wiki/Trademark>