Chapter 1 Resources

Perhaps the most familiar part of the web is the HTTP address. When I want to find a recipe for a dish featuring broccoli, which is almost never, then I might open my web browser and enter `http://food.com` in the address bar to go to the food.com website and search for recipes. My web browser understands this syntax and knows it needs to make an HTTP request to a server named food.com. We'll talk later about what it means to "make an HTTP request" and all the networking details involved. For now, we just want to focus on the address: `http://food.com`.

Resource Locators

The address `http://food.com` is what we call a URL—a uniform resource locator. It represents a specific resource on the web. In this case, the resource is the home page of the food.com website. Resources are things I want to interact with on the web. Images, pages, files, and videos are all resources.

There are billions, if not trillions, of places to go on the Internet—in other words, there are trillions of resources. Each resource will have a URL I can use to find it. `http://news.google.com` is a different place than `http://news.yahoo.com`. These are two different names, two different companies, two different websites, and therefore two different URLs. Of course, there will also be different URLs inside the same website. `http://food.com/recipe/broccoli-salad-10733/` is the URL for a page with a broccoli salad recipe, while `http://food.com/recipe/grilled-cauliflower-19710/` is still at food.com, but is a different resource describing a cauliflower recipe.

We can break the last URL into three parts:

1. `http`, the part before the `://`, is what we call the **URL scheme**. The scheme describes *how* to access a particular resource, and in this case it tells the browser to use the hypertext transfer protocol. Later we'll also look at a different scheme, HTTPS, which is the secure HTTP protocol. You might run into other schemes too, like FTP for the file transfer protocol, and mailto for email addresses.

   Everything after the `://` will be specific to a particular scheme. So, a legal HTTP URL may not be a legal mailto URL—those two aren't really interchangeable (which makes sense because they describe different types of resources).

2. `food.com` is the **host**. This host name tells the browser the name of the computer hosting the resource. The computer will use the Domain Name System (DNS) to translate `food.com` into a network address, and then it will know exactly where to send the request for the resource. You can also specify the host portion of a URL using an IP address.

3. `/recipe/grilled-cauliflower-19710/` is the **URL path**. The food.com host should recognize the specific resource being requested by this path and respond appropriately.

Sometimes a URL will point to a file on the host's file system or hard drive. For example, the URL `http://food.com/logo.jpg` might point to a picture that really does exist on the