**How browsers work**

**Preface**

This comprehensive primer on the internal operations of WebKit and Gecko is the result of much research done by Israeli developer Tali Garsiel. Over a few years, she reviewed all the published data about browser internals and spent a lot of time reading web browser source code. She wrote:

In the years of IE 90% dominance there was nothing much to do but regard the browser as a "black box", but now, with open source browsers having [more than half of the usage share](http://techcrunch.com/2011/08/01/open-web-browsers/), it's a good time to take a peek under the engine's hood and see what's inside a web browser. Well, what's inside are millions of C++ lines…

Tali published her research on [her site](http://taligarsiel.com/), but we knew it deserved a larger audience, so we've cleaned it up and republished it here.

As a web developer, **learning the internals of browser operations helps you make better decisions and know the justifications behind development best practices**. While this is a rather lengthy document, we recommend you spend some time digging in; we guarantee you'll be glad you did.

*Paul Irish, Chrome Developer Relations*

**Introduction**[**#**](https://web.dev/howbrowserswork/#introduction)

Web browsers are the most widely used software. In this primer, I will explain how they work behind the scenes. We will see what happens when you type google.com in the address bar until you see the Google page on the browser screen.

**The browsers we will talk about**[**#**](https://web.dev/howbrowserswork/#the-browsers-we-will-talk-about)

There are five major browsers used on desktop today: Chrome, Internet Explorer, Firefox, Safari and Opera. On mobile, the main browsers are Android Browser, iPhone, Opera Mini and Opera Mobile, UC Browser, the Nokia S40/S60 browsers and Chrome, all of which, except for the Opera browsers, are based on WebKit. I will give examples from the open-source browsers Firefox and Chrome, and Safari (which is partly open source). According to [StatCounter statistics](http://gs.statcounter.com/) (as of June 2013) Chrome, Firefox and Safari make up around 71% of global desktop browser usage. On mobile, Android Browser, iPhone and Chrome constitute around 54% of usage.

**The browser's main functionality**[**#**](https://web.dev/howbrowserswork/#the-browsers-main-functionality)

The main function of a browser is to present the web resource you choose, by requesting it from the server and displaying it in the browser window. The resource is usually an HTML document, but may also be a PDF, image, or some other type of content. The location of the resource is specified by the user using a URI (Uniform Resource Identifier).

The way the browser interprets and displays HTML files is specified in the HTML and CSS specifications. These specifications are maintained by the W3C (World Wide Web Consortium) organization, which is the standards organization for the web. For years browsers conformed to only a part of the specifications and developed their own extensions. That caused serious compatibility issues for web authors. Today most of the browsers more or less conform to the specifications.

Browser user interfaces have a lot in common with each other. Among the common user interface elements are:

1. Address bar for inserting a URI
2. Back and forward buttons
3. Bookmarking options
4. Refresh and stop buttons for refreshing or stopping the loading of current documents
5. Home button that takes you to your home page

Strangely enough, the browser's user interface is not specified in any formal specification, it just comes from good practices shaped over years of experience and by browsers imitating each other. The HTML5 specification doesn't define UI elements a browser must have, but lists some common elements. Among those are the address bar, status bar and tool bar. There are, of course, features unique to a specific browser like Firefox's downloads manager.