Are static Websites the Future or the Past?

Martin Fenner, Gobbledygook

March 5, 2014

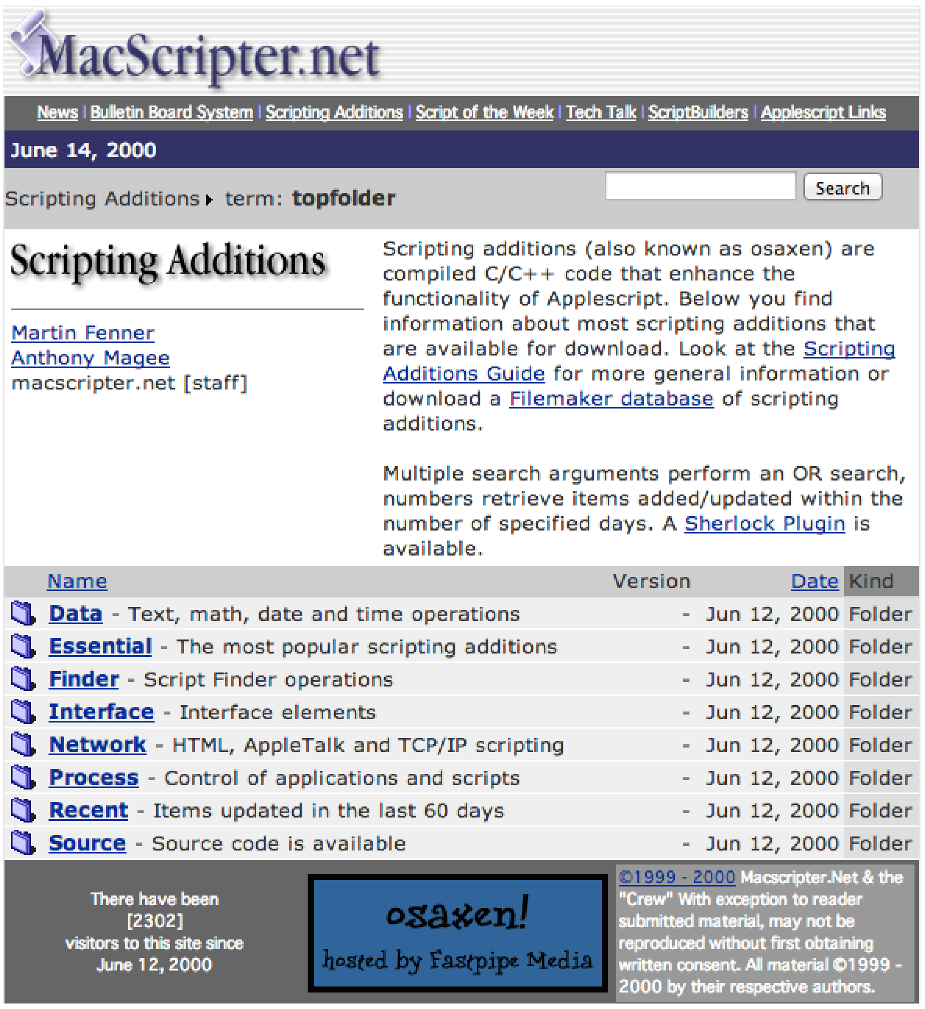
Last week I had a little discussion on Twitter about a great blog post by Zach Holman: Only 90s Web Developers Remember This. The post is not only fun to read, but also reminded me that it is now almost 20 years (1995) that I built my first website - of course using some of the techniques (the one pixel gif!, the `&nbsp;` tag!) described in the post.

Figure 1: ScriptWeb logo 1995

We started ScriptWeb back in 1995 as a central resource for scripting on the Mac (Applescript and Frontier). It was a nice collaborative effort and I was resposible for a directory of scripting additions (or osaxen), joining forces with MacScripter.net a few years later:

Since then I have built many other websites for fun and work, adapting to how technology changed over the years:

- 1995: website running on a Mac Quadra 610 using the WebSTAR HTTP server and server side includes (SSI)
- 1995: static site generation with outline navigation using Applescript. FTP to transfer files
- 1995: Visual HTML editors (Adobe PageMill 1.0)
- 1999: database server and application layer (too long ago to remember the technology)
- 2001: Open source database and application code with MySQL and PHP. CVS for version control
- 2001: web frameworks with PHP and MySQL: PostNuke and Xaraya
- 2005: more complex web application frameworks: Ruby on Rails. Subversion version control
- 2008: git for version control
- 2011: more complex frontend Javascript



MacScripter.net

News | Bulletin Board System | Scripting Additions | Script of the Week | Tech Talk | ScriptBuilders | Applescript Links

June 14, 2000

Scripting Additions ▸ term: **topfolder**

Search

Scripting Additions

Martin Fenner
Anthony Magee
macscripter.net [staff]

Scripting additions (also known as osaxen) are compiled C/C++ code that enhance the functionality of Applescript. Below you find information about most scripting additions that are available for download. Look at the Scripting Additions Guide for more general information or download a Filemaker database of scripting additions.

Multiple search arguments perform an OR search, numbers retrieve items added/updated within the number of specified days. A Sherlock Plugin is available.

Name	Version	Date	Kind
Data - Text, math, date and time operations	-	Jun 12, 2000	Folder
Essential - The most popular scripting additions	-	Jun 12, 2000	Folder
Finder - Script Finder operations	-	Jun 12, 2000	Folder
Interface - Interface elements	-	Jun 12, 2000	Folder
Network - HTML, AppleTalk and TCP/IP scripting	-	Jun 12, 2000	Folder
Process - Control of applications and scripts	-	Jun 12, 2000	Folder
Recent - Items updated in the last 60 days	-	Jun 12, 2000	Folder
Source - Source code is available	-	Jun 12, 2000	Folder

There have been [2302] visitors to this site since June 12, 2000

osaxen! hosted by Fastpipe Media

©1999 - 2000 Macscripter.Net & the "Crew" With exception to reader submitted material, may not be reproduced without first obtaining written consent. All material ©1999 - 2000 by their respective authors.

Figure 2: Scripting Additions at Macscripter.net 2000

- 2013: static site generator Jekyll

Since last June this blog is running on Github pages and the site is generated with Jekyll. Jekyll works really well to build static websites such as this blog, but I am increasingly using it for more complex projects, e.g. for the online version of a book on Open Science.

What I find interesting in this timeline is that with Jekyll there is a shift in focus. Rather than building even more complex web pages that are generated dynamically by the server, we are going back to a two-stage process where the HTML pages are built first and then served as HTML, CSS and Javascript without any database or server application layer. Doesn't sound too different from what we did in the 1990s. This approach obviously works well for content-heavy sites like this blog or book chapters, not so much for dynamically generated content that changes every few minutes, or where the page is put together from many different page fragments.

What I don't know, and I am really interested to find out, is how well this scales to larger sites, specifically publisher websites that host thousands of scholarly journal articles - again content that is very text-heavy and doesn't change that much. The potential benefits of replacing the paradigm of a database layer that holds all content with a paradigm that stores all content in files managed by git version control are clear: serving the content on the web becomes less complex, cheaper and faster. The tradeoff is of course that generating the static content becomes more complex and time-consuming, and it can become a challenge to mix the static content with dynamic content generated by servers as well as the user's browser. For a now infamous example using this technology, look no further than Heathcare.gov. I don't know enough details to understand what went wrong, and it might have more to do with the scale of the project and the tight timeline to launch. For scholarly journal articles this might be a reasonable approach, as even when there is no longer a printed version of the journal, articles are still published on a specific date, and changing the content is a very formal process.

Figure 3: Netscape Navigator 1. Flickr photo by bump