<layer>

or the ingrained habits of web development

Peter-Paul Koch
http://quirksmode.org
http://twitter.com/ppk
HTML Special, 16 June 2016

Who here remembers the <layer> tag?

<layer>

- Essentially what we call now an absolutely positioned element
- except that you can't change it, except for its coordinates
- and Netscape 4's bugs made it very hard to work with (reflow, anyone?)
- Fortunately, absolutely positioned elements in IE worked a lot better.

<layer>

- If you remember this, you also remember most of the history of web development.
- Today I'm going to talk about that history, and about the lessons we web developers learned - for good or for ill.
- And <layer> and its ilk stand at the start of that history.

Warning:
work in
progress

The Browser Wars

Browser Wars

- Incompatibilities during the Browser Wars were deliberate: Netscape and Microsoft introduced them in order to lure web developers to their platform.
- Result: web developers learned it was OK to test in only one browser
- and to use the dreaded “best viewed in”

The Dynamic Duo - Cross-

Peter Paul

https://grox.net/doc/web/javascript/dynduo/

grox.net says:

You must have Netscape 4.0 or Internet Explorer 4.0 to view the examples in this site.

OK

Habit acquired

Test in only one browser. The rest is not good enough anyway.

Browser Wars

- But before you laugh at those silly web developers from 15 years ago, remember that “best viewed in” is still a thing.

Finalists 2014 (best viewed in G...

https://picnet.ca/cleanshots/fina

Search

PICNet
PROVINCIAL INFECTION CONTROL
NETWORK OF BRITISH COLUMBIA
A program of the Provincial Health Services Authority

Finalists 2014 (best viewed in Google Chrome)

This page is best viewed in Google Chrome; if you still encounter problems, visit the Flickr site.
You can view **all 1000 entries** here.

Looking for the 2015 contest? Visit this page.

Browser Wars

- But before you laugh at those silly web developers from 15 years ago, remember that "best viewed in" is still a thing.
- To their credit, Google is pushing back aggressively.

Browser Peace

- As we all know, IE won the Browser Wars of 1998-2001. What we commonly forget is that that was deserved: IE was a way better browser than Netscape
- Who remembers first setting eyes on IE6? It was an insanely good browser when it came out.
- Disadvantage: “best viewed in” reinforced.
- Also: yesterday’s wonderful browser may be today’s piece of junk.

2 Tools - the beginning

Tools - the first wave

- Anyway, in order to combat incompatibilities you need tools.
- … right?

Dynamic Drive
Dynamicdrive.com

New | Hot | Forum | FAQ | Submit

Content Areas

Features

Dynamic Drive

"The future of JavaScript, Today."

Welcome to Dynamic Drive, the premier place on the net to obtain free, **original** DHTML scripts and components to enhance your web site! All scripts on this site take advantage of the latest in JavaScript and DHTML technology, with emphasis on practicality and backwards compatibility, to bring true power and interactivity to your site. Enjoy! **Last updated:** 02/29/00

Important: Please read Terms Of Use and FAQs before using any of the scripts on this site.

-Need a Web host? Search the Internet's LARGEST Web hosting directory, at HostIndex.com

DHTML Script categories

- Menus and Navigation Systems
- Special document effects
- Scrollers
- Image effects
- Links and buttons
- Dynamic Clocks and dates ▲
- Text animations
- Browser Window
- DHTML Games
- Other

Link to us! If you find this site helpful, please help us spread the word by linking to us. Click here for more information.

FREE newsletter!

Get the latest updates to Dynamic Drive, news and tips on DHTML, and more, for FREE. Subscribe now!

Email address here
Submit

Contact info

Contact Us!
Submit a script!
Search Dynamic Drive!
Privacy statement
Link to Us

Do Now!

News: Netscape releases NS 5 Alpha! See screenshot.

Learn about the **new** DHTML features of IE 5.5 beta, how to add multiple scripts to your page, and more, in the latest issue of the Dynamic Drive newsletter.

For all of you who have yet to **subscribe** to this information-packed newsletter, isn't it about time **you did**?

Subscribe and Win!

Habit acquired

Use tools to work around incompatibilities

Tools - the first wave

- But I think there's a second factor in our tool use.
- "Real" programmers use tools. And they follow rules and things.
- And they look down on us. Because we're just a bunch of script kiddies dealing with a browser and a markup language. Easy stuff.

Habit acquired

Feel inferior to “real” programmers.

Compensate.

Compensation

```
function getDocumentBody() {
    return document.body
}
```

3 The standards revolution

Standards revolution

- From 1998 on the WaSP (Web Standards Project) took aim at browsers that didn't implement W3C standards. Back then, that meant Netscape and IE both.
- Successful, as we all know
- But the real point is that web developers took action in their own right

Standards revolution

- This forced us to create our own ecosystem, since nobody else cared.
- Good and bad consequences.
  - Good: conferences. All good web dev conferences are organized by web devs themselves and are different from other software conferences.
  - Bad: impenetrable jargon such as progressive enhancement. Difference between pseudo-class and pseudo-element.

Habit acquired

Look inward. No one understands us anyway.

Has good and bad consequences.

4 The app revolution

The app revolution

The app revolution actually started before the mobile revolution.

In 2006-8, several successful web apps were built that emulated native desktop apps; most importantly Google Docs took on Microsoft Office.

Quality was generally good (enough), so this was rightfully seen as a victory for the web.

The app revolution

After those successes, web developers thought they could do better than native mobile apps as well.

This, generally speaking, has turned out not to be the case

but our feature priorities and the general direction of web development still point towards ever more complicated apps

Habit acquired

Compare browsers to things that are not browsers

The app revolution

Technically, it's simple.

Native apps communicate directly with the OS.

Web apps communicate with the browser, which communicates with the OS.

Therefore web apps will always be a bit slower and coarser than native apps.

5 The mobile revolution

The mobile revolution

- Many more screen resolutions than ever. Besides, they were important: iPhone!
- Many more browsers than ever. UC? BlackBerry? Samsung Chromium? Opera Mini?
- Native apps, and the desire to emulate them.

The mobile revolution

- Screens: responsive design. Solved because the solution is technically simple.
- Many browsers: ignore. BlackWHAT? Who uses Opera Mini? UC is for China only. There is but one Chrome.
- Native apps: emulate! OMG emulate! We must emulate!

Habit acquired

Responsive design

-> caring about

screen sizes

6 The tool revolution

The tool revolution

- Every *week* there are new libraries and frameworks that promise to solve slight problems in existing ones
- Sure, there are the big ones, such as Angular, React, Ember, etc.
- but they're pretty new as well
- and they have their share of problems

The tool revolution

- Polyfills
- MV* frameworks
- UX libraries
- Dependency thingies
- Other thingies with weird names
- etc.

The tool revolution

- Polyfills
- MV* frameworks
- UX libraries

Why so many?

- Dependency thingies
- Other thingies with weird names
- etc.

The tool revolution

I think we're using this many tools because we want to show web app development is a Serious Thing

and Serious Developers use long toolchains

but these long toolchains run on a server

except on the web, where we force *all of our users* to run them

even when they're on an old mobile phone on a crappy network

Habit reinforced

Use tools.

Because Serious

Programmers do it.

Modularization encourages over-design

John Daggett

The true JavaScripter

- uses libraries and frameworks when he needs
- but studies them in detail before doing so
- and prefers to use a single one per project
- is able to write a medium-complex application without any libraries or frameworks
- which gives him the technical background to change a library or framework if necessary

If you can't do without tools you're not a web developer

7 Platforms

Platforms

- Web developers too often look at "real" programmers for guidance.
- In general that's a good idea.
- But in some cases their guidance is totally wrong because not all software engineering precepts work on the web.
- Cause: a fundamental misunderstanding of the nature of the web.

Browsers are the most hostile development platforms in the world

Douglas Crockford

Browsers are the most ~~hostile~~ misunderstood development platforms in the world

not quite Douglas Crockford

Web platforms

I feel back-end developers underestimate the web platform, and thus front-end development

because they misunderstand one crucial aspect.

The web is not one platform.

It is a multitude of platforms, most of which you'll never test on.

Platforms

- Why do back-enders expect the web to be one platform?
- They usually work for a manageable number of known environments, where languages, libraries, power and memory, and tools are pre-defined.
- They expect front-end to be one environment that they have to learn, but that's not fundamentally different
- But it *is* fundamentally different.

DRY

- DRY: DO repeat yourself!
- Web development requires you to repeat yourself. If you have an Ajax script that adds data to the page, make sure there's also a simple link somewhere.
- You write the same functionality twice.
- Not all software engineering principles make sense on the web
- because the web is not one platform

8 Experience focus

Software market maturity

Users = **web developers**, and not visitors!

1. Technology focus. Concentrate on the fact that it works at all.
2. Feature focus. Concentrate on new features users may need.
3. Experience focus. Concentrate on the overall experience users get.

by Jared Spool

Software market maturity

We've been stuck in the feature focus phase for far too long.

I'd say it's time to move to the experience focus stage. I'd say we want to improve the overall experience of creating websites.

What does that mean? I have no clue.

9 Habits

Habits

- Test in only one browser
- Use tools. Everybody does it.
- Feel inferior to “real” programmers
- Look inward into our own ecosystem.
- Compare browsers to things that are not browsers
- Responsive design

Habits

- ~~Test in only one browser~~
- Use tools. Everybody does it.
- ~~Feel inferior to “real” programmers~~
- Look inward into our own ecosystem.
- ~~Compare browsers to things that are not browsers~~
- Responsive design

Thank you

I'll put these slides online

Questions?

Peter-Paul Koch
http://quirksmode.org
http://twitter.com/ppk
HTML Special, 16 June 2016