

JS Performances VS Common Good Practices

Andrea Giammarchi
(aka @WebReflection)

Who am I

- **WebReflection** (<http://webreflection.blogspot.com>)
- **Ajaxian** (<http://ajaxian.com>)
- **TE: JavaScript Patterns**
(@stoyanstefanov <http://amazon.com> Bestseller)
- **TE: Test-Driven JavaScript Development**
(@cjno <http://amazon.com> Bestseller)
- **Mobile Platform R&D SW Developer**
(NOKIA)

Moore's law

- The number of transistors that can be placed inexpensively on an integrated circuit has doubled approximately every two years
- Exponential processor transistor growth does not mean exponential performances

computing to go

THE MOBILE ERA

Scripting

- beautiful
- who cares about performances
- micro optimizations suck
- if it leaks it's a GC fault
- the “compiler” optimizes for me

Programming

- beautiful
- best algorithms means best performances
- hacks may be good to reach a goal
- performances practices do matter
- too much “magic” never happens here

JavaScript

- *“the good looking girl that comes at the party with an ugly boyfriend called DOM”*
- the most misunderstood language (Mr D)
- the less studied language on academic level
- the *“don’t need to learn it since I know Java, Ruby, C#, PHP, Whatever++”*
- ... aka *“if JSLint says I am wrong, I must be!”*

Still the most (ab)used
programming language
in the world!

JS “Good Practices”

- simulate classic Object Oriented Programming style
- being religious about whatever Lint SW
- “classic OOP style or readability” matters
- if it’s difficult to read, it’s not me being noob it’s obviously the code written badly ...
- JavaDoc Style comments ... Oh Yeah!

Performances

- WEB related, bottlenecks are **everywhere**
- hard to achieve cross platform speaking, engines are too often **that** different!
- abstraction **costs**

... rewind ...

PERFORMANCES!

- **define your target browser/engine** (lazily or separating specific files loading only what you need *when* you need)
- **avoid “exponential scaling” architecture**
(KISS/YAGNI Approach VS Over Engineering)
- **improve your language knowledge**

which means ...

.. and just fews ...

- ignore Lint warnings or “errors” if meant
- avoid Classic OOP emulation
(prototypal inheritance could be much better!)
- avoid repeated property access
- avoid closures when possible
- understand what’s going on!
(jQuery, frameworks folks, newcomers, please improve!)

“Take Care Of” List

- download size matters
- roundtrip matters
- responsiveness matters
- dynamic data and scaling matters
- JS engine matters

A Worthy Compromise

- more performances oriented practices
- more “compilers” help, when possible
- define best “ad hoc” practice, don’t trap yourself behind some “silly convention”
- “think deflate/gzip” if you can!

TESTS!

- 1000X up to 10000X+ performances boost!
- slower is the device, more “real cases” gap we gain!
- <http://3site.eu/ft>
- ... please don't freak out, I gonna explain those results ...

How Browsers Help

- **JIT** (V8, SquirrelFish+Extreme, Carakan, Tamarin)
- **Typed Arrays** (<https://cvs.khronos.org/svn/repos/registry/trunk/public/webgl/doc/spec/TypedArray-spec.html>)
- **2D HW Acceleration** (canvas, CSS3, FF4 http://weblogs.mozillazine.org/roc/archives/2010/09/full_hardware_a.html, IE9)
- **3D HW Acceleration** (CSS3, WebGL)

HW Acceleration IS NOT ENOUGH!

- Get a Netbook, possibly with a GPU
- Wow, Look at those fishes in IE9! (<http://ie.microsoft.com/testdrive/Performance/FishIE%20tank/Default.html>)
- Now ... Look at that Zooming in IE9! (<http://ie.microsoft.com/testdrive/Performance/I0MapZooming/Default.html>)
- Now, after zooming, simply PAN THE MAP!
- ... and now let's see if we can go faster or slower ... then I'll answer your questions :-)