

# A History of Tcl in the Browser

Oh no, not again!



# The Motivation

- need a scripting language

Anyway I know only one programming language worse than C and that is Javascript ....the most horrible kluge in the history of computing

Robert Cailliau - CERN



**We don't compile**  
**Everything is a string**  
**Types are for wimps**  
**Eschew obfuscation!**  
**Speed is overrated**  
**We are the 0.1%**



# The Motivation

- need our scripting language
  - portability
  - productivity
  - deployment
  - relevancy







# The Motivation

- Android
  - no mainstream Tcl release
  - no Tk
- iOS
  - Objective C / Javascript only
  - Tcl - feasible and practical?
  - deployment

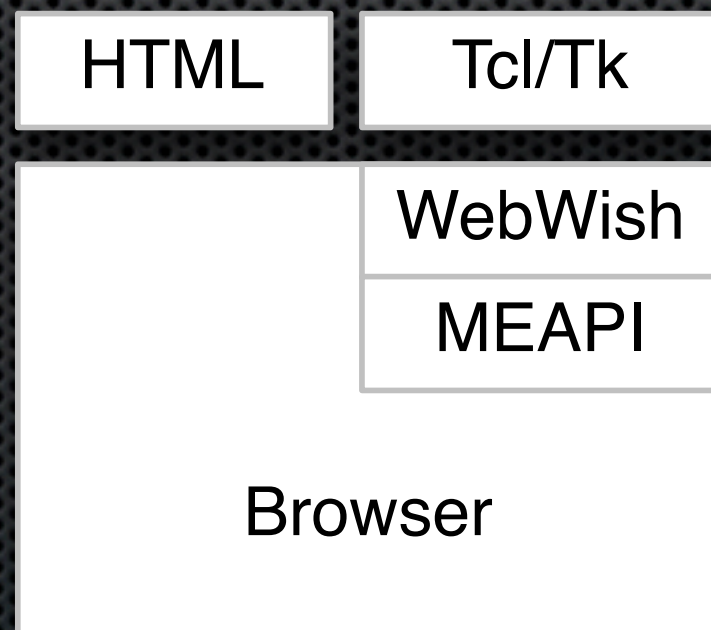


# The Survey



# 1995 - Eolas WebRouser

- the first Web Tcl
- one of the first plugins





# 1995 - Eolas WebRouser

- the first Web Tcl
- one of the first plugins

## Pros

- Tcl + Tk
- security model
- web apps

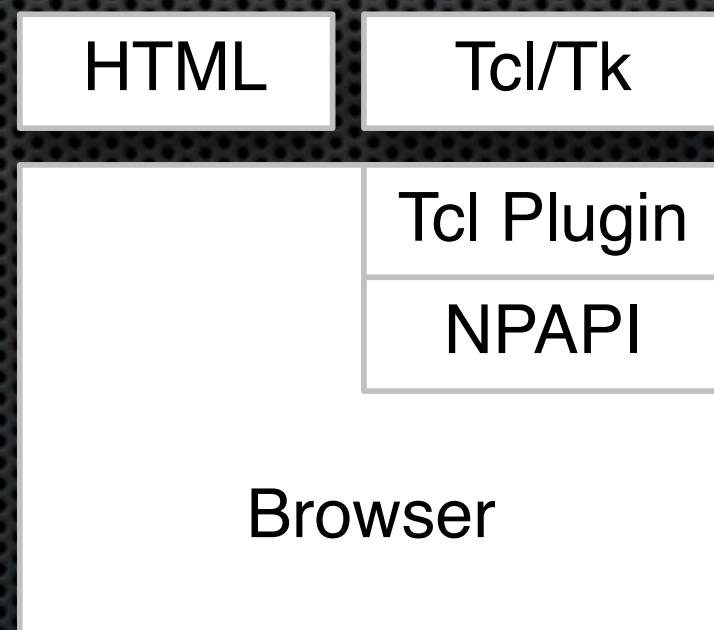
## Cons

- plugin
- no longer available



# 1996 - The Tcl Plugin

- SunLabs Tcl Group
- one of the first Netscape plugins





# 1996 - The Tcl Plugin

- SunLabs Tcl Group
- one of the first Netscape plugins

## Pros

- Tcl + Tk
- Safe-Tcl security
- still available FF + IE

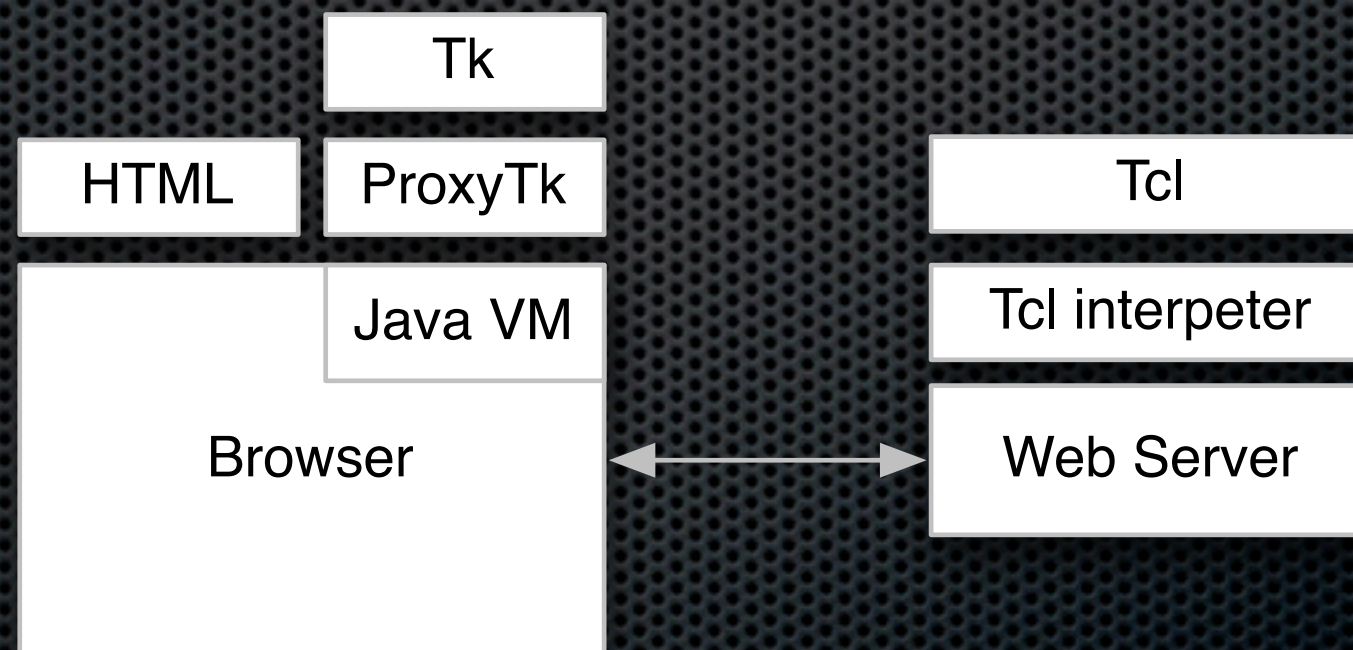
## Cons

- plugin
- not on mobile
- deployment



# 1998 - Proxy Tk

- Java applet + custom server
- efficient client/server protocol





# 1998 - Proxy Tk

- Java applet + custom server
- efficient protocol

## Pros

- Tcl + Tk
- client / server
- deployment

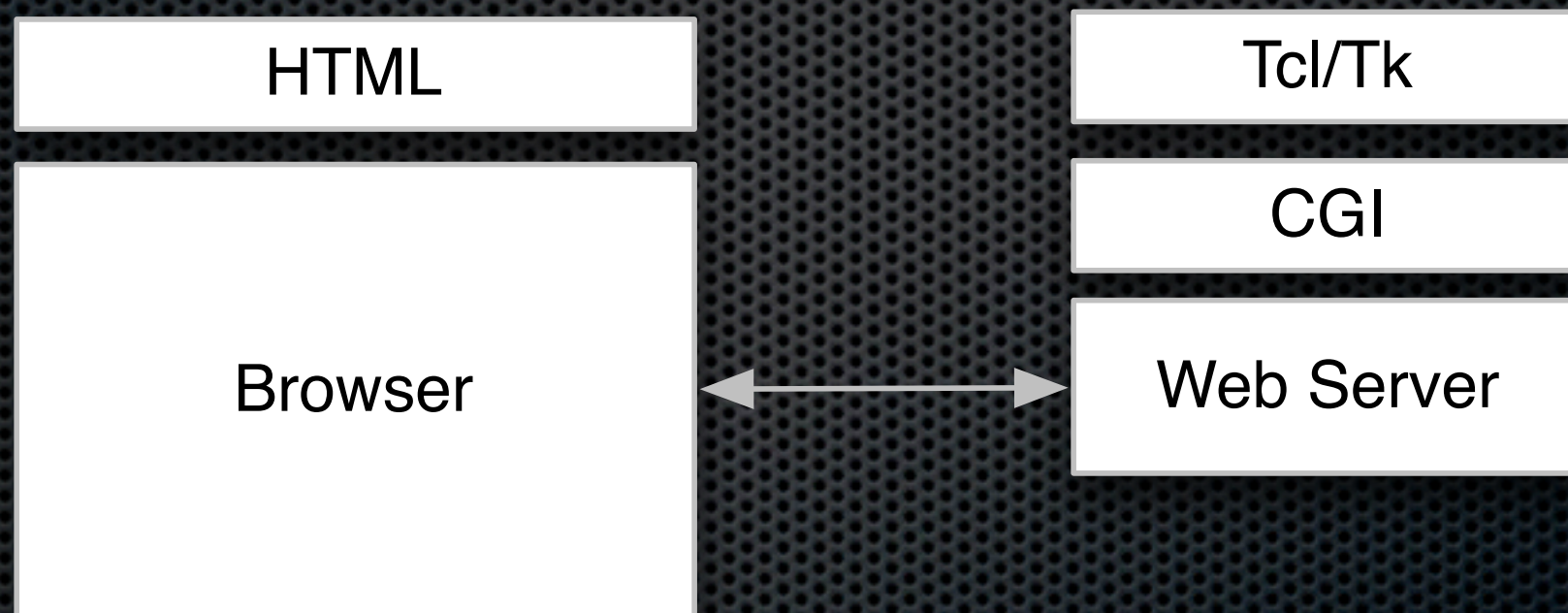
## Cons

- no offline support
- subset of Tk
- no longer available



# 2003 - TkWeb

- render Tcl/Tk using HTML + CGI





# 2003 - TkWeb

- render Tcl/Tk using HTML + CGI

## Pros

- Tcl + Tk
- Javascript
- no plugin

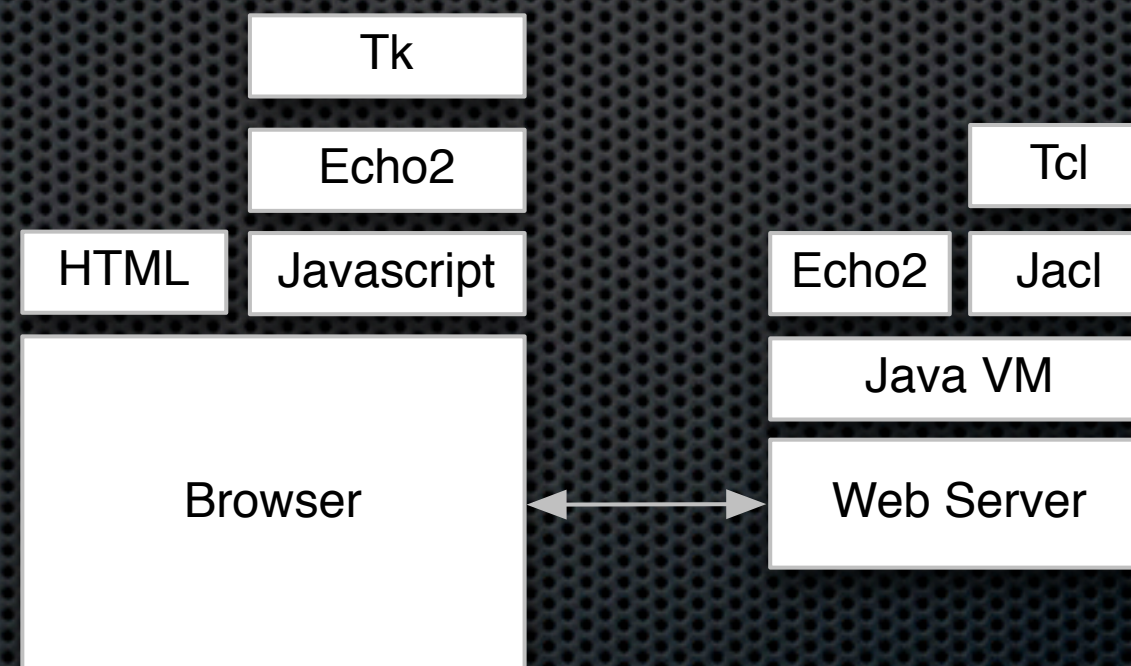
## Cons

- experimental
- incomplete
- no offline support



# 2006 - AEjaks

- Tcl in the server (via Jacl)
- Ajax-based windowing system





# 2006 - AEjaks

- Tcl in the server (via Jacl)
- Ajax-based windowing system

## Pros

- Tcl + Tk
- Javascript
- no plugin

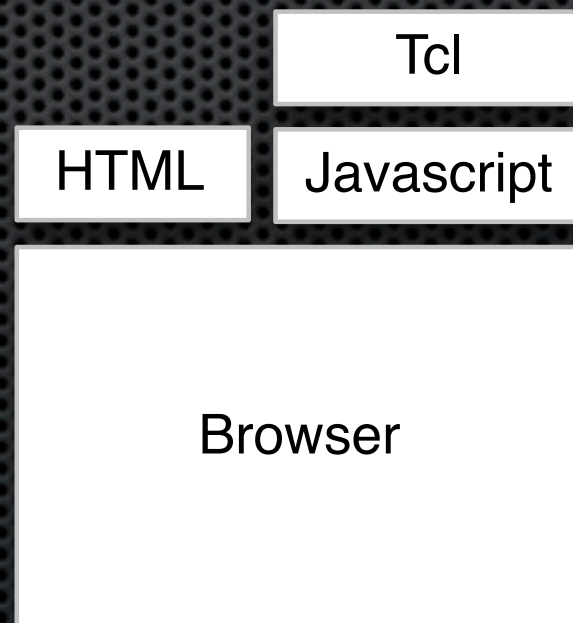
## Cons

- subset of features
- no offline support



# 2007 - JsTcl

- Tcl interpreter in Javascript
- transliteration of Picol





# 2007 - JsTcl

- Tcl interpreter in Javascript
- transliteration of PicoL

## Pros

- Javascript
- no plugin

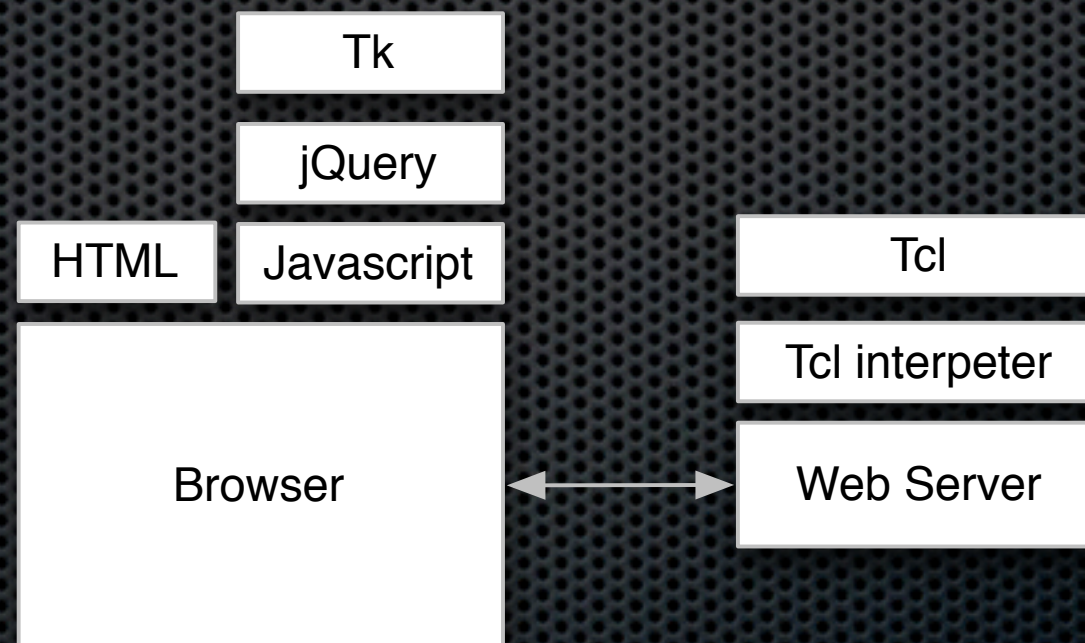
## Cons

- experimental
- incomplete



# 2010 - WubTk

- Tcl in server
- Tk over jQuery over Javascript in browser





# 2010 - WubTk

- Tcl in server
- Tk over jQuery over Javascript in browser

## Pros

- Tcl + Tk
- Javascript
- no plugin

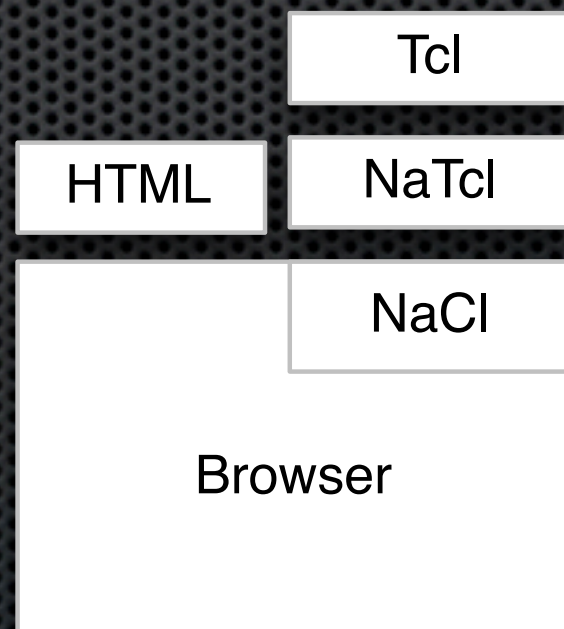
## Cons

- subset of Tk
- no client-side Tcl
- no offline use



# 2011 - NaTcl

- Tcl in Google Native Client (NaCl) sandbox
- real Tcl, native code





# 2011 - NaTcl

- Tcl in Google Native Client sandbox
- Tk over jQuery over Javascript in browser

## Pros

- speed
- full Tcl in the browser
- interface with the DOM

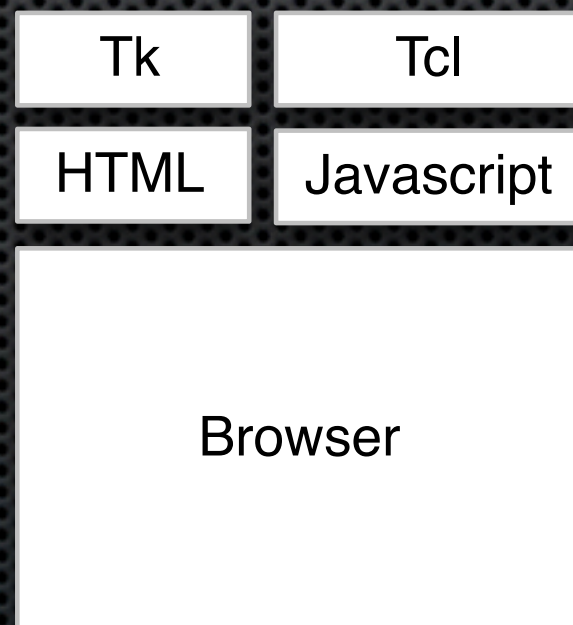
## Cons

- no Tk
- Google Chrome only
- plugin



# 2011 - IncrTcl in Javascript

- Tcl in Google Native Client sandbox
- Tk over HTML/CSS/Javascript in browser





# 2011 - IncrTcl in Javascript

- Tcl in Google Native Client sandbox
- Tk over jQuery over Javascript in browser

Pros

Cons



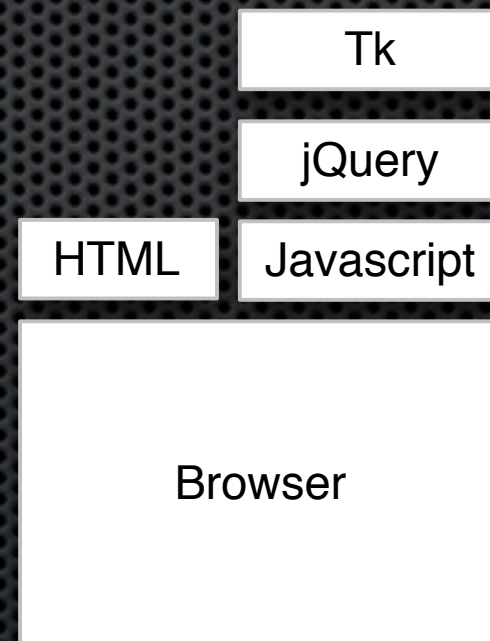
# 2011 - NaTk

- Tk over jQuery over Javascript
- client-side



# 2011 - NaTk

- Tk over jQuery over Javascript
- client-side Tk





# 2011 - NaTk

- Tk over jQuery over Javascript
- client-side Tk

## Pros

- Javascript
- HTML5/CSS3
- offline use

## Cons

- proof of concept
- subset of Tk



# Summary

- several options available
  - the Venerable Plugin
  - *Æjaks*
  - WubTk
  - NaTcl
  - incrTcl in Javascript



# Summary

- several options available
- arguably none ready for prime time



Oh no, not again!



# Three approaches


- translate application code to Javascript
- implement the TEBC engine in Javascript
- implement Tcl in Javascript



# Linux in a browser

- PC emulator Javascript
- small
- fast
- Linux boots in the browser



 + <http://bellard.org/jslinux/>

```
TCP bind hash table entries: 512 (order: -1, 2048 bytes)
TCP: Hash tables configured (established 1024 bind 512)
TCP reno registered
checking if image is initramfs...it isn't (bad gzip magic numbers); looks like a
n initrd
Freeing initrd memory: 2048k freed
Total HugeTLB memory allocated, 0
io scheduler noop registered
io scheduler anticipatory registered
io scheduler deadline registered
io scheduler cfq registered (default)
Real Time Clock Driver v1.12ac
JS clipboard: I/O at 0x03c0
Serial: 8250/16550 driver $Revision: 1.90 $ 4 ports, IRQ sharing disabled
serial8250: ttyS0 at I/O 0x3f8 (irq = 4) is a 16450
RAMDISK driver initialized: 16 RAM disks of 4096K size 1024 blocksize
loop: loaded (max 8 devices)
TCP cubic registered
NET: Registered protocol family 1
NET: Registered protocol family 17
Using IPI Shortcut mode
Time: pit clocksource has been installed.
RAMDISK: ext2 filesystem found at block 0
RAMDISK: Loading 2048KiB [1 disk] into ram disk... done.
EXT2-fs warning: maximal mount count reached, running e2fsck is recommended
VFS: Mounted root (ext2 filesystem).
Freeing unused kernel memory: 124k freed
Booted in 4.866 s
Welcome to JS/Linux
~ #
```

© 2011 Fabrice Bellard - [News](#) - [FAQ](#) - [Technical notes](#)



# Linux in a browser

- PC emulator Javascript
- small
- fast
- Linux boots in the browser
- hand-coded Javascript



# Emscripten

- translate C to Javascript





# Emscripten

- translate C to Javascript



- acceptable performance
- other languages + packages ported
- which Tcl codebase?



# Jim Tcl

- small footprint
- small codebase
- advanced features
- high degree of compatibility



# Jim JS

- build environment

- invoking Tcl

```
function execute(text) {  
    Module.run(text);  
}  
  
function print(text) {  
    console.log(text);  
}
```



# Jim JS

- build environment
- invoking Tcl
- malloc 0
- missing functions

```
command = expr 1
==== Tokens ====
[ 0]@1 ESC 'expr'
[ 1]@1 SEP ' '
[ 2]@1 ESC '1'
[ 3]@1 EOF ''
==== Script ====
[ 0] LIN
[ 1] ESC expr
[ 2] ESC 1
==== Expr Tokens ====
[ 0]@0 INT '1'
[ 1]@0 EOL ''
_strtoul is not a function
```



# Jim JS

- build environment
- invoking Tcl
- malloc 0
- missing functions
- performance

```
time {set a 10; set b $a}
ActiveTcl 8.6b1.2      0.43
Jim/Firefox           30
Jim/Safari            27
```



# Jim JS

- build environment
- invoking Tcl
- malloc 0
- missing functions
- performance
- tactical not strategic solution



# Deja vu all over again

- technoarchaeology ?
- archeotechnophilia ?
- technonecrophilia !







# Demo



# Where to now?

- Tcl - tactical
  - Jim JS
- Tcl - strategic
  - ubiquity - optimized Javascript
  - speed - native or NaTcl
- Tk over HTML5 / CSS3
  - desktop + browser



# Typple anyone?



Typeless Programming Language

[typple.net](http://typple.net)