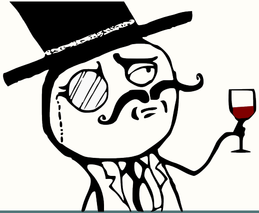
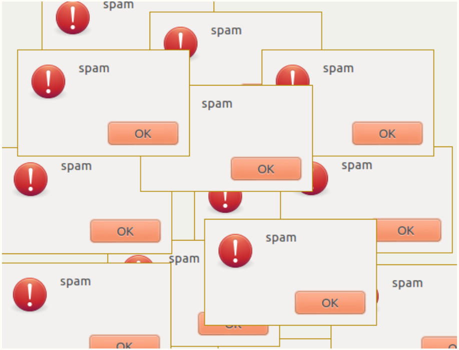
Making JavaScript respectable



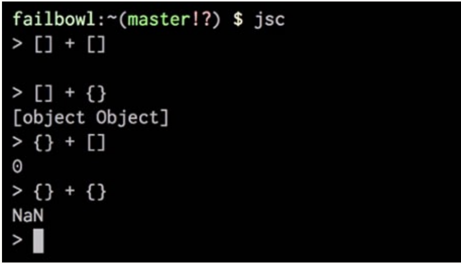
Florent Marchand de Kerchove
fmdkdd@mines-nantes.fr

ASCOLA Research Group
École des Mines de Nantes

JavaScript has issues ...



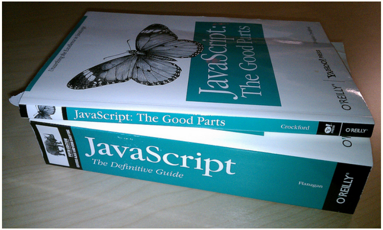
`alert(’spam’);`



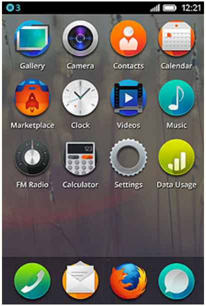
```
failbowl:~(master!?) $ jsc
> [] + []

> [] + {}
[object Object]
> {} + []
0
> {} + {}
NaN
> 
```

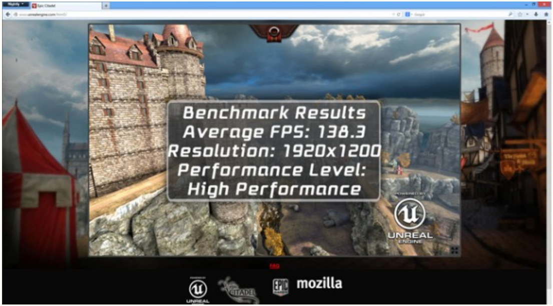
Wat @garybernhardt

...but JavaScript has good parts too



- prototypes
- dynamic typing
- first-class functions
- event loop
- *comprehensions*
- *lambdas*
- *destructuring assignments*

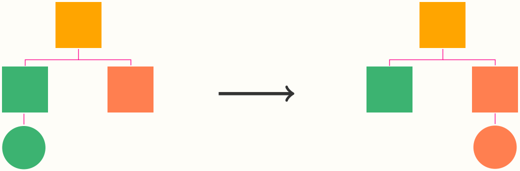




Benchmark Results
Average FPS: 138.3
Resolution: 1920x1200
Performance Level:
High Performance

Controlling JavaScript programs

How to maintain “arbitrarily structured” code ?



How to enforce dynamic code conventions ?

`[int, int, int]` ~~`eval(user_input)`~~

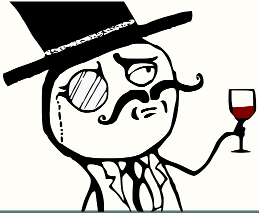
JavaScript control *sur mesure*:

- optional types,
- contracts

Tool support for practical:

- prototyping,
- debugging,
- dynamic analyses

Making JavaScript respectable



Florent Marchand de Kerchove
fmdkdd@mines-nantes.fr

ASCOLA Research Group
École des Mines de Nantes