WebAssembly

E o que Python tem a ver com isso

Sumário

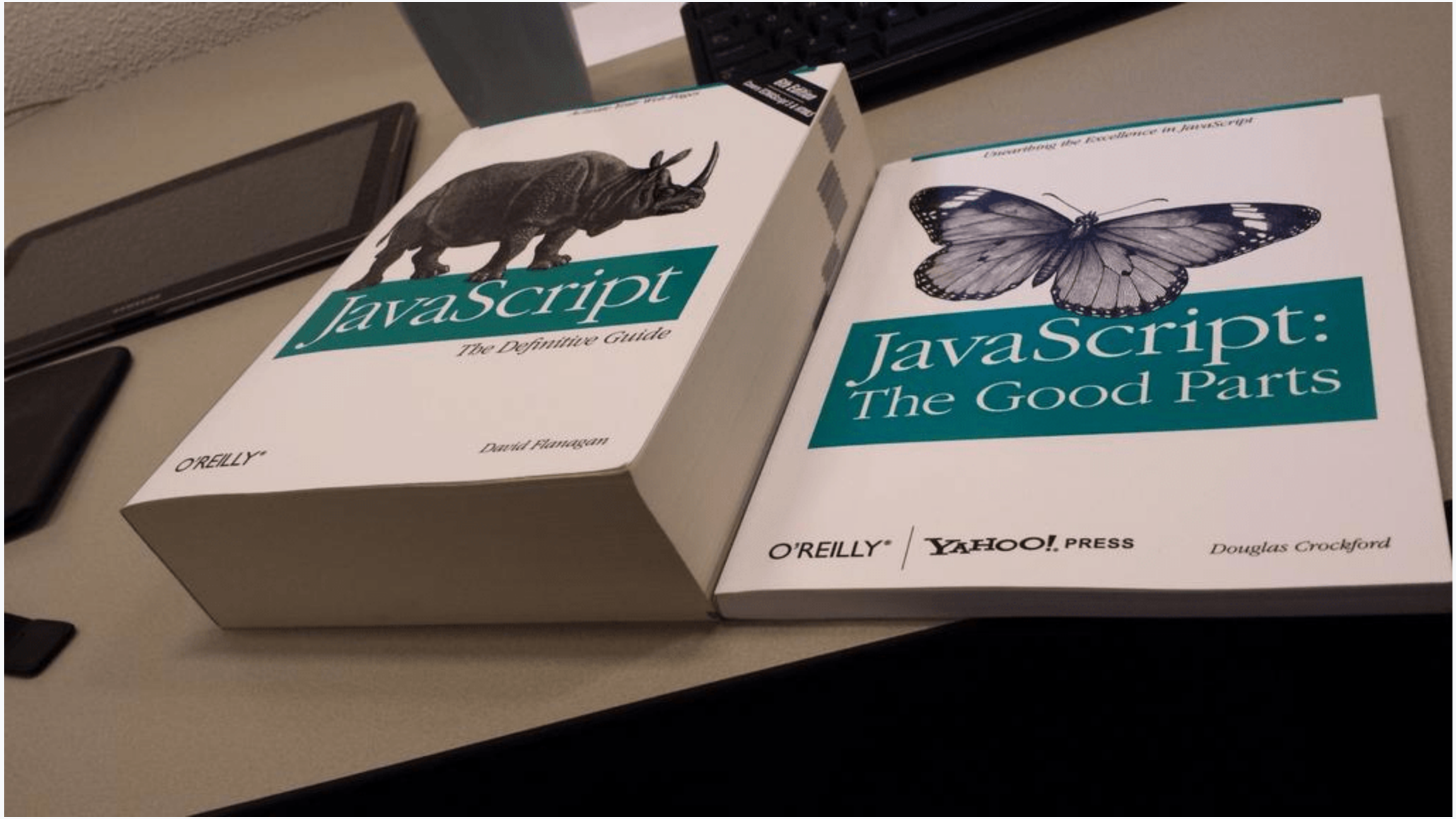
Quem sou

- Desenvolvedor de software desde 2008
- `['Recife'].append('Florianópolis')`
- 2008~15 Instituto Nokia de Tecnologia
- 2015~ Expertise Solutions
- Projetos:
  - PySide
  - Webkit
  - EFL (JS, C#)
- Contato:
  - Twitter: @lauromoura
  - Github: lauromoura
  - email: lauromoura at gmail.com

No começo, era o javascript

- Hack de 10 dias em 1995
- Que virou um dos pilares da web
  - Junto com HTML e CSS
- Alto nível, prototypes, sintaxe que lembra Java/C e afins
- Principais implementações
  - V8 - Blink (Chrome, Edge) e Node
  - JavaScriptCore - Webkit (Safari)
  - Spidermonkey - Gecko (Firefox)

Problemas...


JavaScript
The Definitive Guide
O'REILLY
David Flanagan
Unearthing the Excellence in JavaScript
JavaScript:
The Good Parts
O'REILLY
YAHOO! PRESS
Douglas Crockford

Problemas

- Array(16).join("LoL" - 2) + " Batman!"
- Ficar preso a uma única linguagem
  - Transpilers e afins
  - Ainda assim, é Javascript
- TOO MUCH JS
- Alto nível demais
  - Problema ao usar JS normal como "assembly"

Problemas

```
!function(e,t){"use strict";"object"==typeof
module&&"object"==typeofmodule.exports?
module.exports=e.document?t(e,!0):function(e){
if(!e.document)throw new Error("jQuery
requires a window with a document");
return t(e)}:t(e)}("undefined"!=typeof
window?window:this,function(e,t){"use strict";
var n=[],r=e.document,i=Object.getPrototypeOf,
o=n.slice,a=n.concat,s=n.push,u=n.indexOf,l={},
...
```

Porque é tão pesado?

- Javascript
  - Parsear
  - Gerar bytecode
  - Otimizar (JIT e afins)
  - Re-Otimizar
  - Rodar

"Assembly" na Web

asm.js

- Subset de Javascript
- Dicas para o interpretador do código que vai ser executado
- Permite otimizações mais certeiras e eficientes

asm.js - C

```
size_t strlen(char *ptr) {
  char *curr = ptr;
  while (*curr != 0) {
    curr++;
  }
  return (curr - ptr);
}
```

asm.js - js

```
function strlen(ptr) {
  ptr = ptr|0;
  var curr = 0;
  curr = ptr;
  while (MEM8[curr]|0 != 0) {
    curr = (curr + 1)|0;
  }
  return (curr - ptr)|0;
}
```

asm.js

- Ainda é javascript
- Ainda precisa ser parseado
- Pode ser melhorado

WebAssembly

Assembly de verdade na web

WebAssembly

"Se WASM+WASI existissem em 2008, nós não precisaríamos ter criado o Docker. Essa é a importância deles. Webassembly no servidor é o futuro da computação. Uma interface de sistemas padrão era o elo que faltava. Vamos torcer que WASI esteja a altura"

Solomon Hykes - Criador do Docker

WebAssembly

- Formato binário
  - Não precisa ser parseado, apenas decodado
- Tipos já determinados
- VM em pilha
- Suporte a uma versão MVP nos 4 principais navegadores
- Gerenciamento manual de memória

WebAssembly Text Format

```
(module
        (func $addTwo (param $lhs i32)
                      (param $rhs i32)
                      (result i32)
                get_local $lhs
                get_local $rhs
                i32.add)
        (export "addTwo" (func $addTwo))
)
```

Processador hipotético

- 8-bit bytes
- Memória byte a byte
  - Linear
- Suporte a acessos não-alinhados
- Inteiros de 32bits e opcionalmente 64bits
- IEEE 754-2008 para floats de 32 e 64bits
  - Mas sem exceções p/ NaN por enquanto
- Little endian
- Ponteiros de 32bits
  - wasm64 suporta ponteiros de 64bits

Syscalls e afins

- Não existem (ainda)
  - ABI em discussão
- Atualmente cada host expõe uma API
- WASI a caminho
  - Arquivos e sistemas de arquivos
  - Sockets
  - Relógios
  - Random
  - Etc

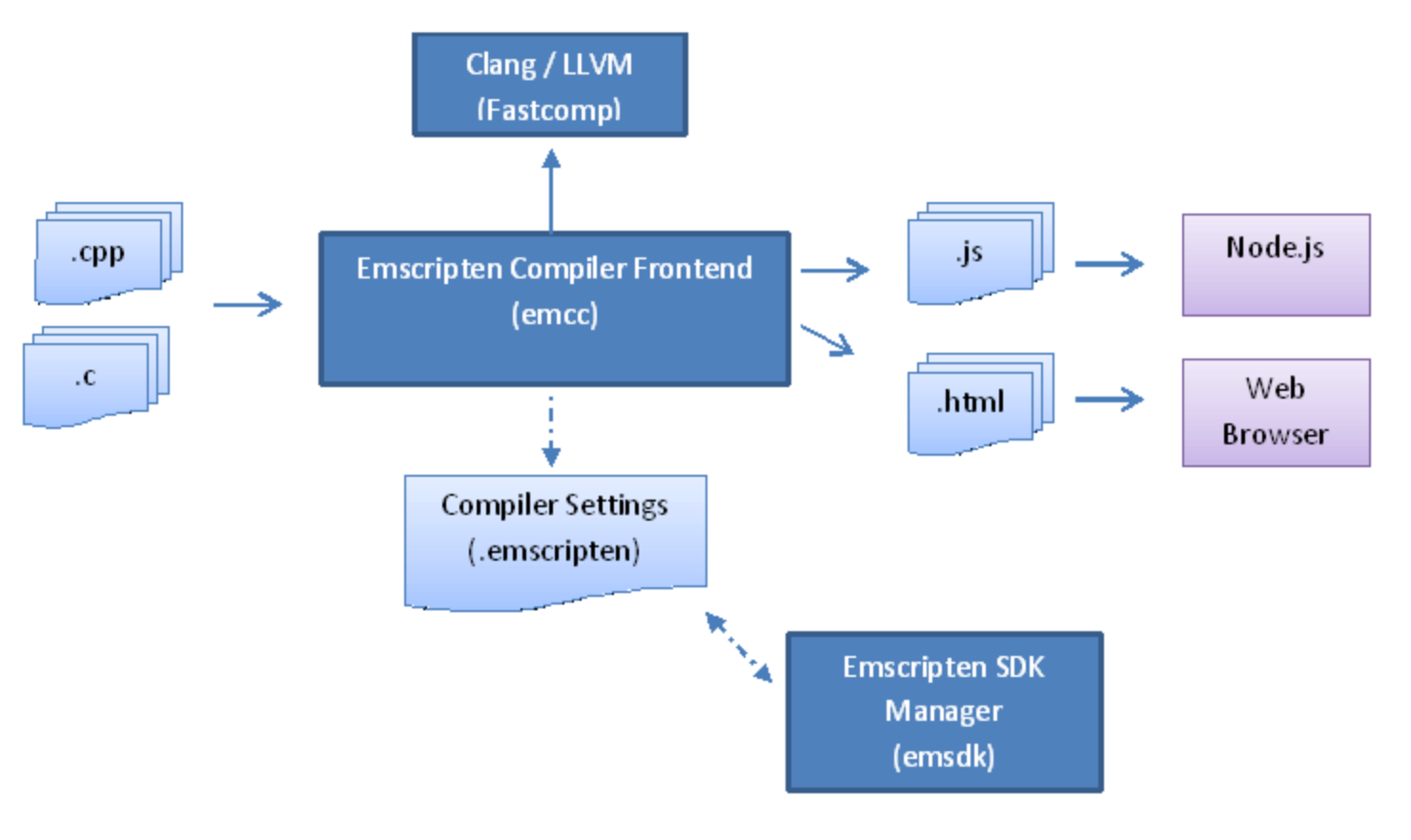
emscripten

Uma ponte entre JS e C/C++

emscripten

- Backend LLVM
- clang como frontend
- Gera código asm.js ou WebAssembly
- Usos
  - Unity
  - Unreal
  - Qt
  - DOSBox
  - **Pyodide**

emscripten


Clang / LLVM
(Fastcomp)
.cpp
.c
Emscripten Compiler Frontend
(emcc)
.js
Node.js
.html
Web
Browser
Compiler Settings
(.emscripten)
Emscripten SDK
Manager
(emsdk)

emscripten runtime environment

- Ports
  - SDL 2.0
  - ...
- OpenGL ES 2.0
- Sistema de arquivos virtual
  - MEMFS
  - IDBFS
  - NODEFS (node.js)
- Memória
  - Typed Array único
  - Várias views (int, float, etc)
- pthreads
  - asm.js
  - Experimental em WebAssembly

emscripten - main loop

- Não pode bloquear
- Função de iteração
- emscripten_set_main_loop
- emscripten_cancel_main_loop

Python no frontend - Transpilers

- Exemplo: Transcrypt (thanks, Berin)
- Compilam .py (ou parecido) para .js
- Pros:
  - Mais fácil integrar com libs JS
  - Resultado da compilação de tamanho aceitável
  - Execução relativamente rápida (comparado a JS)
- Cons:
  - Complicado utilizar extensões Python nativas
  - Nem sempre toda sintaxe de Python é suportada
  - Compatibilidade entre tipos JS e Python em todo código

Python no frontend - Webasm

- Exemplo: Pyodide
- Compilam um runtime em Webasm
- Pros:
  - Extensões nativas podem funcionar mais facilmente
  - Usar o próprio CPython
  - Compatibilidade entre tipos JS e Python apenas na API
- Cons:
  - Tamanho: .wasm ainda estão bem grandes (MB's)
    - Mas podem ser cacheados pelo browser
  - Pode ser lento demais para alguns casos

Pyodide

- Port do CPython e stack científica
- Inspirado no Iodide
  - Notebook-like project
- numpy, pandas, matplotlib
- Experimental e planos p/ futuro
  - scipy
  - scikit-image/learn
  - Wheels Python "puro sangue" no PyPI
- Tour

Pyodide com Javascript

```
languagePluginLoader.then(() => {
    result = pyodide.runPython('import sys\nsys.version');
    console.log(result);
});
```

Demos

Hoje e Amanhã

Hoje

- C/C++
- Rust
- MVP nos principais navegadores.
  - Módulo exportado
  - Instruções
  - Formato binário
  - Formato textual
  - Implementações nos browsers e outros ambientes
- Python
  - Pyodide
- Outras linguagens

Amanhã (e depois de amanhã)

- Roadmap WASM
  - Especificação
  - Threads
  - GC
  - Exceções
  - Host bindings (DOM/JS)
  - Ferramentas
  - Multi processos (fork & cia)
  - Mais controle de memória
  - ABI
  - Segurança

Amanhã (e depois de amanhã)

- Pyodide
  - Extensões FORTRAN (BLAS/LAPACK)
  - Diminuir tamanho dos downloads
  - Async/threading
  - PyGame? (emscripten já suporta SDL muito bem...)

Links úteis

- Página oficial do Webassembly
- Wasm Explorer
- Awesome Wasm - MUITOS links de projetos, ferramentas, etc
- WebAssembly and the Death of Javascripat - JS Monthy London - Março 2018
- Emscripten Github
- Post no blog da Mozilla sobre Wasm
- Post no blog da Mozilla sobre Pyodide
- Pyodide Github
- Apresentação sobre Pyodide na FOSDEM 2019

Perguntas ?

Obrigado