

Introdução ao NodeJS



HelloJS

\$ who

- Instrutores
 - Leonardo Silveira
 - Yuri Oliveira
- Alunos do curso de Engenharia de Computação do IFCE
- Experiência de mercado
- Experiência acadêmica

Agenda

- Editor de código
- Revisão JavaScript
- Introdução ao node
- Entendendo o `module.exports` / `require()`
- Usando o npm para criar um projeto node com `express.js`

Editor

- Rápido
- Prático
- Você tem que confiar e gostar
- Não é o eclipse (é piada, mas é sério)
- Mínimo de destaque de cor
- Atom, Code, Notepad++ Geany, Kate, vi
- Testar snippets na linha de comando ou no interpretador do navegador

Sobre JavaScript

- Criado em maio de 1995 por Brendan Eich na Netscape
- Feito em 10 ou 15 dias, sobrevive há mais de 20 anos
- Servia para o applet java se comunicar com o browser
- Linguagem funcional
- Certas falhas de design
 - Inteiros
 - Contexto global
 - Falta de módulos
- Entre as 10 linguagens mais populares (TIOBE)

Bloco

- Da esquerda para a direita, de cima para baixo
- scripts adicionados à página são avaliados na sequência
- Uso de chaves (‘{‘ e ‘}’) para blocos de função, condição ou bloco de laço

Variáveis

- Sensíveis a contexto
- for/if/while/switch não alteram escopo
- ES6 promete o 'let', o 'var' feito direito

Condicionais

- `if(teste) {/**/}`
- Valores `0` ou `""` são avaliados como falso
- `null` e `undefined` são avaliados como falso
- `false` is `false`
- `switch(valor){ case 1: /* */ break;}`

Laços

- `for(inicialização; teste; incremento) { /* */ }`
- `while(teste) { /* */ }`

Listas

- `var x = [1, 2, "bola", 3]`
- `x[2] // "bola"`
- `x[0] = 4`
- `x.push(11)`
- `x.unshift(22)`
- `x.pop()`
- `x.shift()`
- `x.sort(function(a,b){ /* */ })`
- `x.map(function(e){ /* */ })`
- `x.filter(function(e){ /* */ })`

Mapas

- `var x = {a:1,b:2}`
- `x.a // 1`
- `x["b"] // 2`
- `x.c = 3`
- `for(var i in x) { x[i] = x[i] + 1 ; }`

Funções

- `function foo(x,y) {} // função nomeada`
- `var bar = function(a,b,c){} // atenção: a função é anônima, acessível em 'bar'`
- `var xib = function sib (k) {} // quase ninguém faz isso`
- `xib(x,y); // chamada`
- `foo.call(ctx,x,y) // ctx -> chamada mudando o contexto de execução`
- `bar.apply(ctx,[a,b,c]) // ctx -> mudando contexto, parâmetros em array`
- mais sobre a utilidade/problemas sobre o contexto adiante

Classes/Objetos

- 'Prototipal'
- Funções são construtores
- `function Cls (){} ; Cls.prototype.attr1 = 10;`
- A linguagem é dinâmica, não force nada diferente disso
- `Object.defineProperty`
- ES6 tem classes!

Sobre o node.js

- Escrito em 2009 por Ryan Dahl (joyent), baseado no V8, engine JS do chrome
- npm (node package manager) introduzido em 2011
- Orientado a eventos (e não a requests)
- Muitos problemas nas primeiras versões
- Força crescente de mercado
- Será a regra em 5 anos ou menos (daí este curso agora!)

Interpretador node

- Preferencialmente acessível via linha de comando
- Gerenciador de pacotes npm também
- Voltado para aplicativos web

require()/module.exports

- Arquitetura do CommonJS
- Melhor modularização dos scripts
- `module.exports = “feijoadá” // script.js`
- `var mymodule = require(“./script”) // index.js`

Express.js

- O padrão eleito pelo mercado (até então)
- Mantida pela Strongloop, recentemente adquirida pela IBM
- Fácil de usar, Extensível, Bom suporte da comunidade

Referências

1. https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript
2. <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
3. <http://es6-features.org/#BlockScopedVariables>
4. <http://mzl.la/1Hu3yDX>
5. <https://nodejs.org/>
6. <https://www.npmjs.com/>
7. <http://expressjs.com/>
- 8.