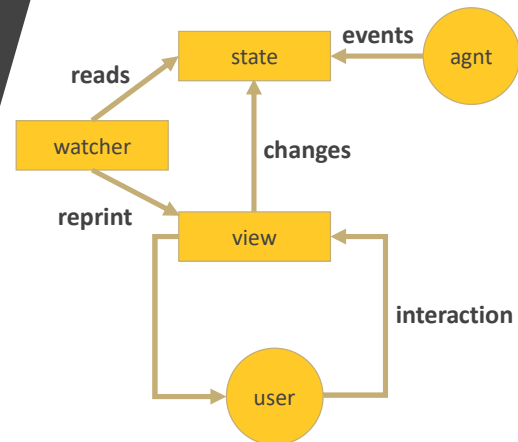
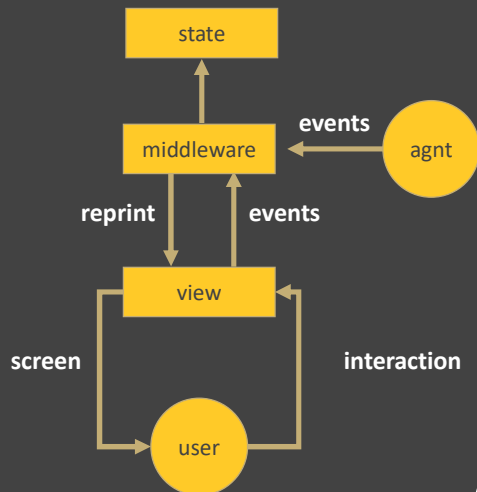
A large yellow square containing the letters 'JS' in a bold, dark grey sans-serif font. The background of the slide is split diagonally from the top-left to the bottom-right, with white on the left and dark grey on the right.

JS

@vtaquette
@rzuquim



One way

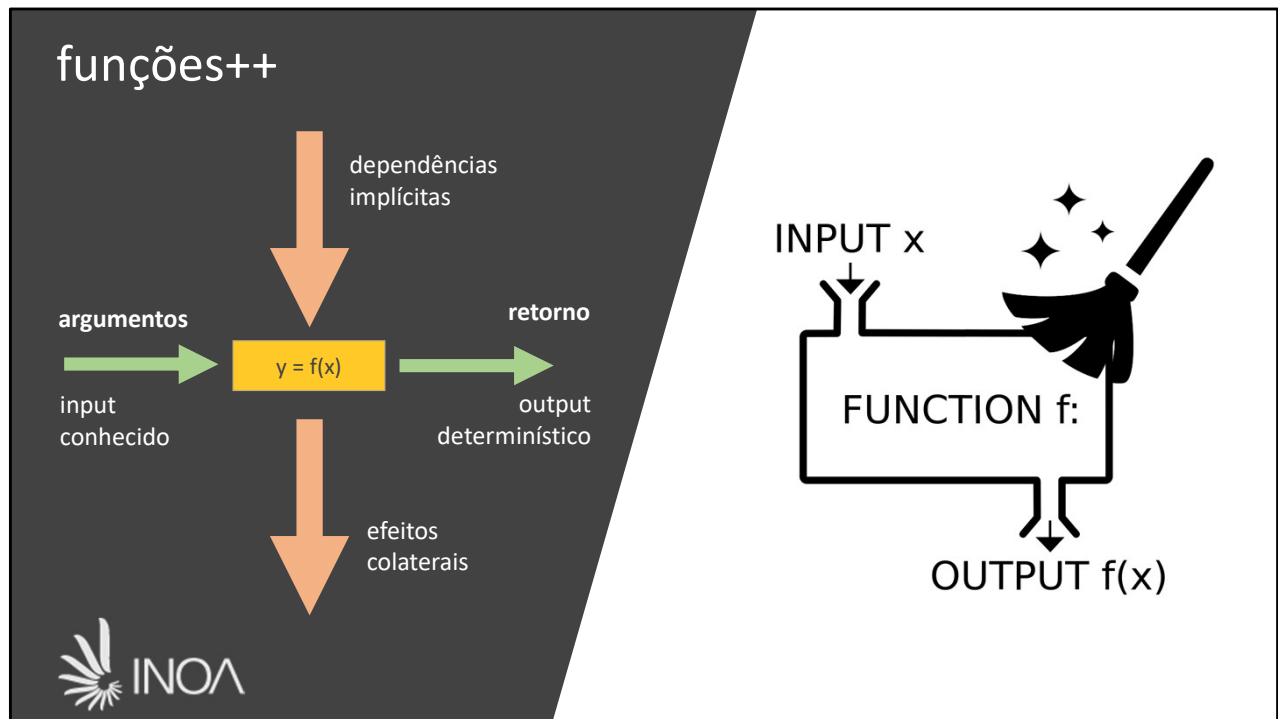


Two way

Referência

<https://staltz.com/unidirectional-user-interface-architectures.html>

<https://www.pluralsight.com/guides/one-and-two-way-data-binding-angular>

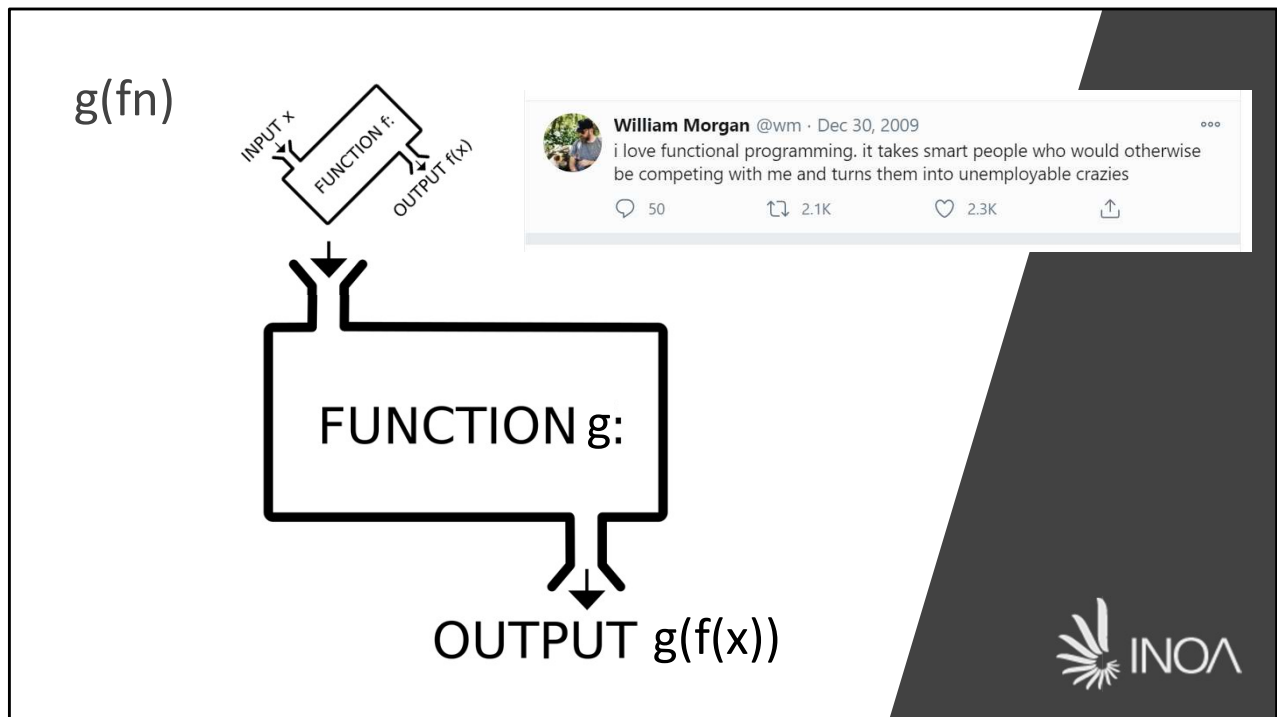


Outra forma de pensar funções é como uma máquina que tem parâmetros de entrada que são processados e retornam um novo valor. Essa é uma visão matemática. Um procedimento que transforma dados de entrada em dados de saída.

Uma função é chamada de **pura** quando para um mesmo input ela sempre retorna o mesmo valor sem nenhum efeito colateral.

Referência

<https://medium.com/javascript-scene/master-the-javascript-interview-what-is-a-pure-function-d1c076bec976>



Uma função é chamada **de ordem superior** se recebe como argumento uma função ou produz uma função no resultado.

Funções relevantes:

- map
- filter
- reduce

Referência:

<https://medium.com/javascript-scene/higher-order-functions-composing-software-5365cf2cbe99>

<https://medium.com/@zaid.naom/exploring-folds-a-powerful-pattern-of-functional-programming-3036974205c8>

~~__proto__~~

~~this~~

~~spread~~


~~arguments / gather~~

~~destructuring~~


~~iterators / generators~~

~~class~~

~~tag functions~~



ementa



Para priorizar as necessidades dos alunos com relação aos projetos vamos pular conceitos previstos na ementa. A gente perde “vocabulário” e capacidade de expressividade na linguagem, mas atacamos os problemas práticos.

Referências para estudo:

prototype chain

- <https://medium.com/@chamikakasun/javascript-prototype-and-prototype-chain-explained-fdc2ec17dd04>

this:

- https://www.w3schools.com/js/js_this.asp
- https://www.youtube.com/watch?v=Pi3QC_fVaD0&ab_channel=codeSTACKr

spread:

- <https://medium.com/@paulryan17/spread-operator-javascript-a77a890b96eb>

arguments / gather (ou ...rest):

- <https://codeburst.io/parameters-arguments-in-javascript-eb1d8bd0ef04>

destructuring:

- <https://hacks.mozilla.org/2015/05/es6-in-depth-destructuring/>

iterators / generators:

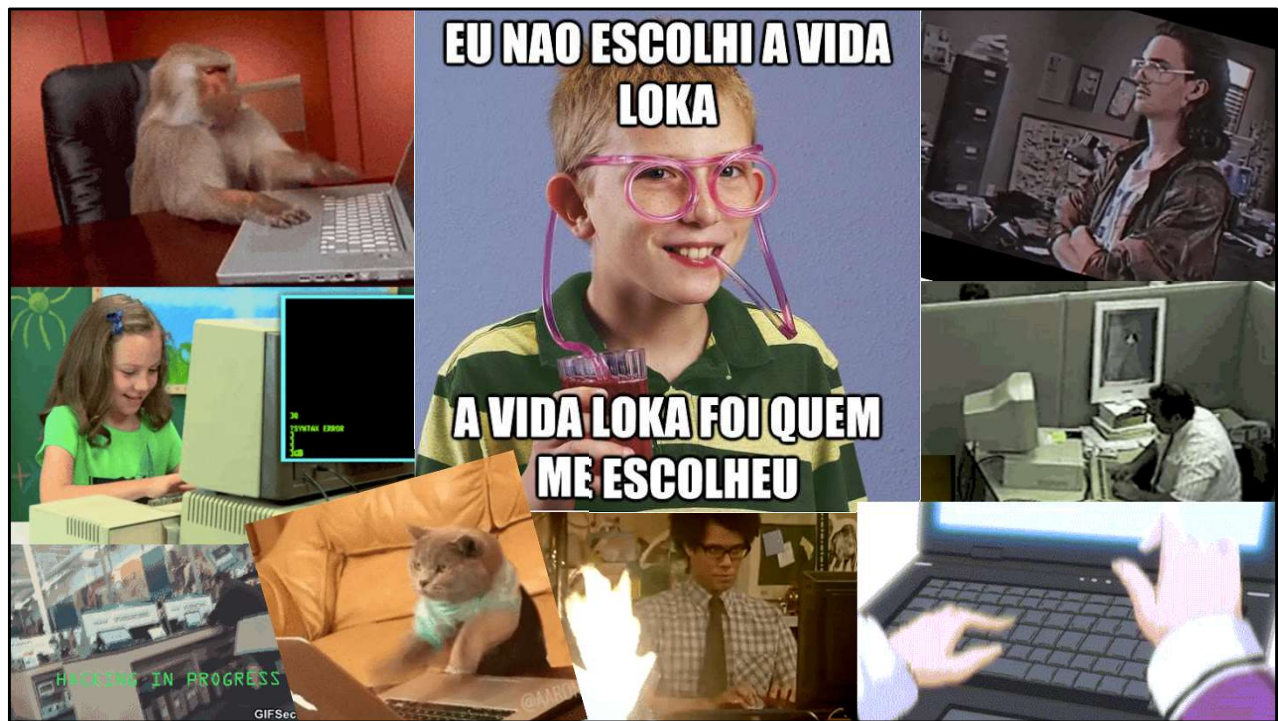
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Iterators_and_Generators

class:

- <https://medium.com/better-programming/object-oriented-programming-in-javascript-b3bda28d3e81>

tag functions

- <https://wesbos.com/tagged-template-literals>
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals



Vamos tentar ver ao máximo:

- componentes reusáveis
- http: comunicação client/server
- nodedjs + express
- persistência com um banco de dados
- hospedagem