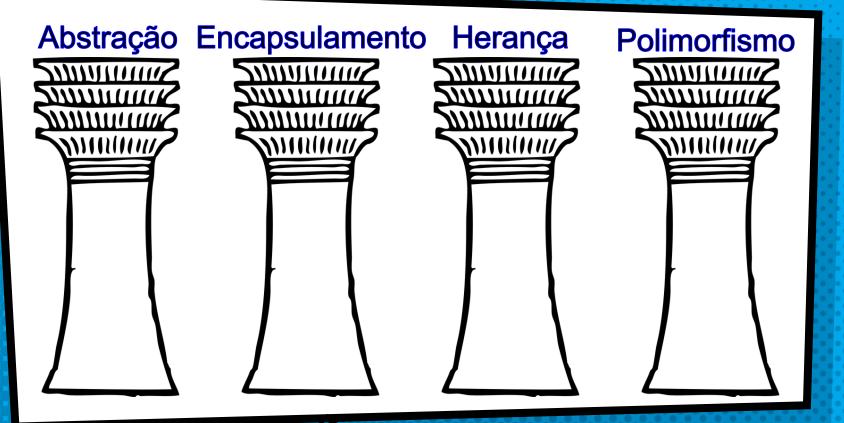
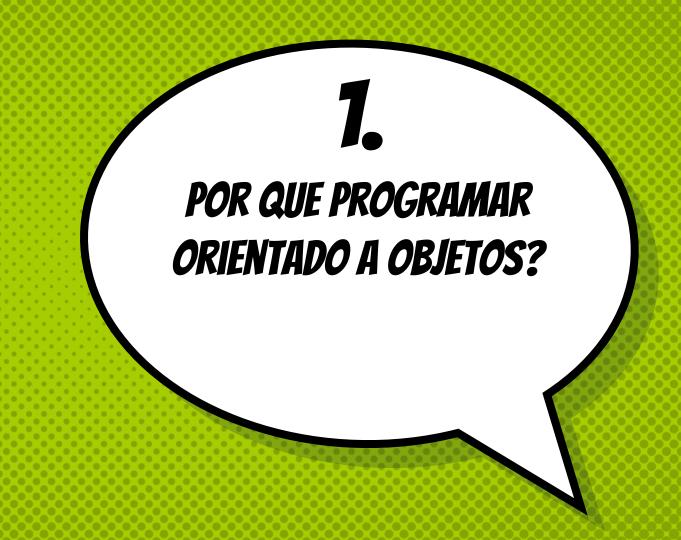
ORIENTAÇÃO A OBJETOS

4 PILARES

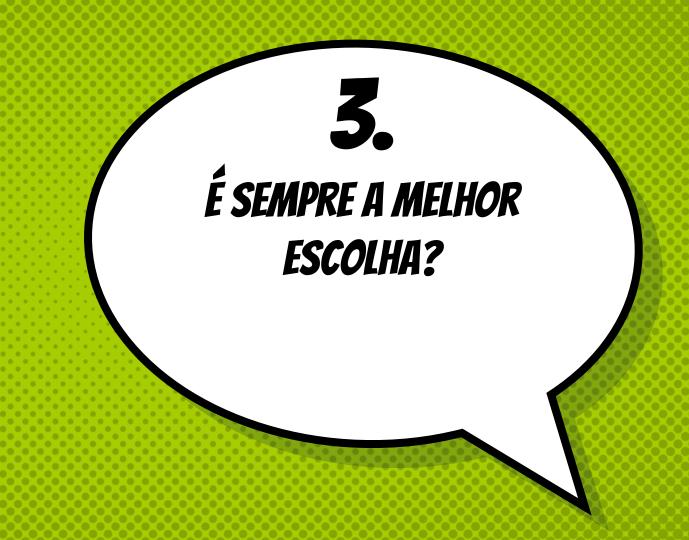


PROGRAMAÇÃO ORIENTADA A OBJETOS









VANTAGENS

- × Prática de programação
- × Segurança
- × Reutilização de código
- × Fácil manutenção

CLASSE

Abstração do que há em comum (em termos de características e comportamentos) a um conjunto de objetos.

atributos

métodos

CLASSES



Mamífero

data de nascimento peso Energia espécie

comer
mover
mamar



CLASSES





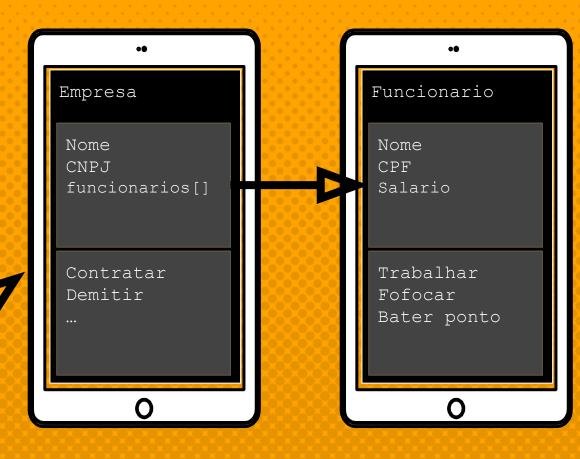
Nome
CNPJ
funcionarios[]

Contratar Demitir

O

CLASSES







OBJETO (1/2)

Combina a estrutura de dados (<u>atributos</u>) e o comportamento dos dados (<u>operações/métodos</u>) em uma única entidade.

Um objeto pode ser definido como um conceito, uma abstração, algo com limites nítidos e significado em relação ao mundo

OBJETO (2/2)

Objetos servem como uma base real para a elaboração de um sistema consistente com a realidade modelada.

Ex.: <u>Um</u> Livro, <u>um</u> Telefone, <u>uma</u> Faculdade, <u>um</u> Produto, <u>um</u> Aluno, <u>um</u> Professor.

CLASSE E OBJETO (1/2)

- Classe é o que projetamos
- × Objeto é o que criamos a partir da classe
- As classes estão para os objetos assim como as plantas arquitetônicas estão para as casas

CLASSE E OBJETO (2/2)

?

Carro

marca modelo ano potência

acelerar trocarMarcha



?



Fiat Bravo 2014 110cv

Ford Maverick 1975 199cv



Chevrolet Camaro 2014 406cv





Microsoft 2012



CARACTERÍSTICAS

pagem) le código jes ódigo



No fim, tudo é Javascript

TS — **JS**



Intellisense e verbosidade

```
export default class Product {
 constructor(code, description, buyPrice, sellPrice, tags,
    lovers = 0, id = uuid()) {
    this.code = code:
    this.description = description;
    this.buyPrice = buyPrice;
    this.sellPrice = sellPrice;
    this.lovers = lovers:
    this.tags = tags;
    this.id = id:
 doSomething(){
   this.description.
                      abc Product
                      abc buvPrice
                      abc code
                      abc description
                      abc doSomething
                      abc id
                      abc lovers
                      abc sellPrice
                      abc tags
                      abc uuid
```

```
export default class Product {
 code: number;
 description: string;
 buyPrice: number;
 sellPrice: number;
 lovers: number:
 tags: Array<string>;
 id: string;
 constructor({code, description, buyPrice, sellPrice, tags,
    lovers = 0, id = uuid()} : Product) {
    this.code = code:
    this.description = description;
    this.buyPrice = buyPrice;
    this.sellPrice = sellPrice:
    this.lovers = lovers;
    this.tags = tags;
    this.id = id;
 doSomething(){
    this.description.
                      * replace
                                                            (method) String.replace(searchVa
                      ⊕ * split
                                                            lue: string | RegExp, replaceVal

 ★ length

                                                            ue: string): string (+3 overload

   ★ toLowerCase

                                                            s)
                      ₩ match
                                                            Replaces text in a string, using a regular
                      m anchor
                                                            expression or search string.
                      1 big
                      19 blink
                                                            @param searchValue - A string to search
                      ⊕ bold
                                                            for.
                      charAt
                      charCodeAt
                                                            @param replaceValue — A string

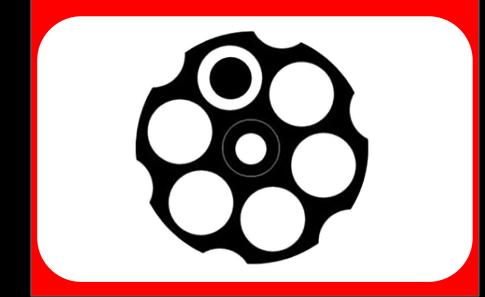
☆ codePointAt
```

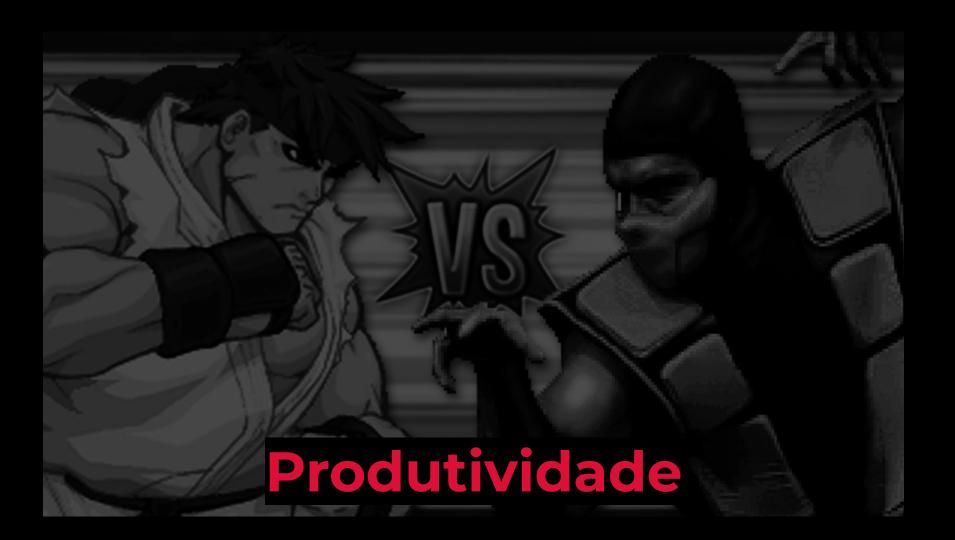
containing the text to replace for every

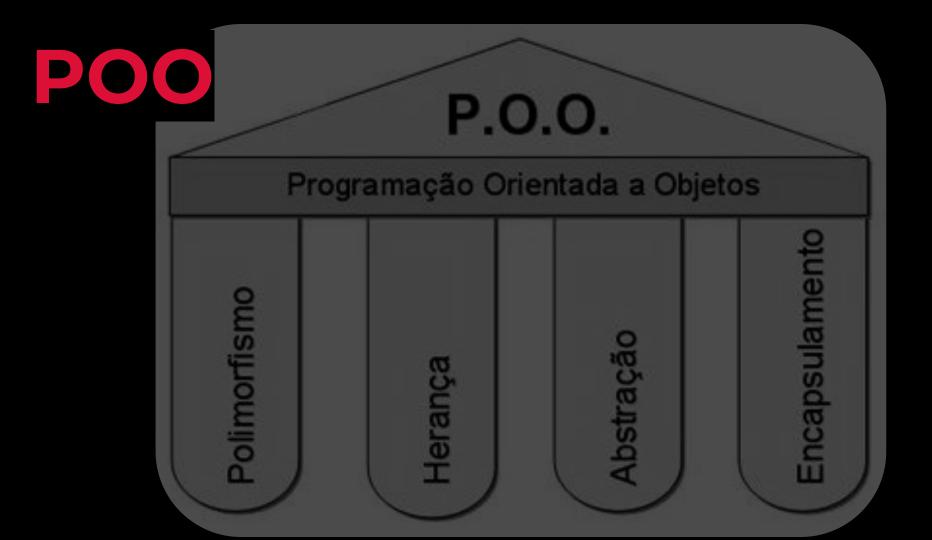
Recursos mais modernos do JS



Feedback em tempo de compilação







Let's code





A CLASSE

Analisando o simples diagrama, vamos desenvolver uma classe que representa o nosso Personagem



MAIS EXERCÍCIOS!



CACHORRO

Desenvolva um arquivo que crie um Cachorro e em seguida possibilita imprimir os valores de seus atributos



PERSONAGEM

Desenvolva um arquivo que crie um Personagem e em seguida possibilita imprimir os valores de seus atributos



CLIENTE

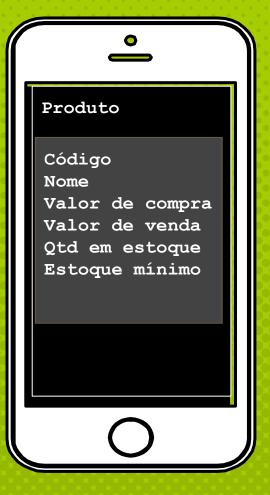
Desenvolva um arquivo que crie um Cliente e teste se ele tem dinheiro suficiente para fazer uma compra de um valor X



PRODUTO

Desenvolva um arquivo que crie um Produto e em um loop de menu permita:

Verificar valor de compra e venda, apresentar total investido em estoque e lucro que se terá ao vender todo estoque



FUNCIONÁRIO

Desenvolva um arquivo que crie um Funcionário, e em um loop se tenha um <u>simulador</u>, com possíveis ações do funcionário, tais como: trabalhar, tomar café, fofocar, solicitar aumento, dormir, etc...

Estas ações devem manipular os atributos da classe



IMPORTANTE!

Em uma boa orientação a objetos, as responsabilidades que dizem respeito a uma classe X, estão **na classe X**.





MÉTODOS!

