

```
//SETPROTOTYPEOF:
```

//Como sabemos, podemos gerar heranças em Javascript por através do prototype, antigamente usávamos o método `__proto__` para fazer isso, porém, alguns browsers mais antigos não aderem a esse método. Por mais que os browser atuais utilizem esse método, não sabemos que browser nossos clientes podem estar usando, por isso seria melhor usarmos o método de herança que se encaixa em todos os browsers, esse método é a função `"setPrototypeOf()"` da função `Object...`

```
//COMO PROTOTIPAVAMOS ANTIGAMENTE:
```

```
const avo = {attr1: 'A'}
```

```
const pai = {__proto__: avo, attr2: 'B'} //Perceba que atribuímos todos os atributos de avo em pai literalmente usando o __proto__
```

```
const filho = {__proto__: pai, attr3: 'C'} //Fizemos o mesmo em filho, onde filho herda os atributos de pai, e ainda por cima herda todos os atributos que o pai herdou de avo.
```

```
//VENDO HERANÇA ACONTECER DE ELEMENTO FILHO PARA ANCESTRAL:
```

```
console.log('\n1)', filho.attr1) //Perceba que existe uma herança onde filho procura pelo atributo attr1, ao não encontrar procura no protótipo dele que é pai, ao não encontrar procura no protótipo de pai que é "avô", por ter encontrado no avô ele retorna a busca com o resultado do valor do atributo...
```

```
//ATRIBUINDO HERANÇA DE PROTOTYPE NO FORMATO MAIS MODERNO DO ECMASCRIPT 2015:
```

//O formato de herança mais moderno do ECMA Script 2015 trouxe a função `"setPrototypeOf()"`, usar essa função é mais indicado do que utilizar o método literal utilizando o método `__proto__`, um método que não é suportado se o usuário estiver usando um browser muito antigo...

```
const carro = { //Perceba que temos um objeto carro que possui atributo para...
```

```
  velMax: 200, //Velocidade máxima, com o padrão de 200Km/h...
```

```
  velAtual: 0, //Velocidade atual comb o padrão e 0...
```

```
  acceleraMais(delta) { //Uma função que pega o this atual dela e compara com a velocidade máxima para não deixar ultrapassar
```

```
    if (this.velAtual + delta <= this.velMax) {
```

```
      this.velAtual += delta
```

```
    } else {
```

```
      this.velAtual = this.velMax
```

```
    }
```

```
  },
```

```
  status() {//E uma função de status que mostra qual a velocidade em que o carro está e quanto ele pode atingir...
```

```

        return `${this.velAtual}Km/h de ${this.velMax}Km/h`
    }
}

//Criamos um objeto ferrari que irá herdar os atributos de métodos de carro, porém, irá sobrescrever o atributo "velMax" assim que a
prototipação acontecer...
const ferrari = {
    modelo: 'F40',
    velMax: 324
}

//Temos também um objeto volvo que vai sobrescrever somente o método status do objeto carro quando a prototipação acontecer...
const volvo = {
    modelo: 'V40',
    status() {
        return `${this.modelo}: ${super.status()}`//Perceba 2 coisas aqui:
        //1º: lembre-se que o objetivo da criação desse método é sobrescrever o já existente no protótipo de carro, então, assim que
a herança acontecer, o this irá referenciar ao próprio objeto volvo, que será o this da vez.
        //2º: estamos usando "super" para chamar o próprio status do protótipo, que é o método status do objeto carro. Sempre que
usarmos super, vamos referenciar ao elemento pai em uma herança, e não ao objeto chamador. Se colocássemos "this" aqui teríamos um
estouro de pilha, pois ele iria chamar ao próprio status do volvo infinitamente...
    }
}

Object.setPrototypeOf(ferrari, carro) //Aqui temos a prototipação de fato com a função "setPrototypeOf()", onde colocamos sempre o
objeto que desejamos que ser o herdeiro, e depois o objeto que desejamos que tenha seus elementos herdados...
Object.setPrototypeOf(volvo, carro)

//Veja como a herança ocorreu corretamente entre os objetos e o objeto pai "carro"
ferrari.aceleraMais(300)
console.log('\n2)', ferrari.status())

volvo.aceleraMais(180)
console.log('\n3)', volvo.status()) //Veja como o método status causou uma sobrescrita sobre o status do objeto pai "carro"

```

RESULTADO NO CONSOLE...

```
[Running] node "c:\Users\Almoxarifado\Documents\javascript\arquivos_das_aulas\086-  
setPrototypeOf().__Gerando_Herancas_de_Forma_Mais_Confiavel.js"
```

1) A

2) 300Km/h de 324Km/h

3) V40: 180Km/h de 200Km/h

```
[Done] exited with code=0 in 0.114 seconds
```