

```

//HASOWNPROPERTY():
//Esse método é usado quando queremos identificar que chaves realmente pertencem a um objetos e quais são herdadas, para isso, o
hasOwnProperty() retorna true para as chaves que pertencem ao objeto e false para aquelas que são herdadas...

//VERIFICANDO QUE CHAVES SÃO HERDADAS COM O MÉTODO HASOWNPROPERTY():
//O método "hasOwnProperty()" é usado para verificar que valores realmente pertencem a um objeto e quais são herdados de um outro
objeto, veja como podemos utilizá-lo:
const paiHasOwn = {nome: 'Pedro', cabelo: 'Preto'}
const filhaHasOwn = Object.create(paiHasOwn, { //Veja que filhaHasOwn pegou o atributo nome e o sobrescreveu, fazendo com que a
herança agora alterada a sua maneira
  nome: {value: 'Patrícia', writable: false, enumerable: true}
})
filhaHasOwn.altura = 1.75 //E criamos uma chave nova para o elemento "altura"
console.log('\n1)', filhaHasOwn.hasOwnProperty('nome')) //Perceba que todas as chaves alteradas ou criadas no próprio objetos
retornam valor true
console.log('1)', filhaHasOwn.hasOwnProperty('altura'))
console.log('1)', filhaHasOwn.hasOwnProperty('cabelo')) //Mas chaves que não foram criadas nem alteradas são retornam valor false...

//USANDO HASOWNPROPERTY DE FORMA MAIS INTELIGENTE:
console.log('\n2)')
for (let i in filhaHasOwn) { //Usando um laço for podemos iterar mais facilmente...
  filhaHasOwn.hasOwnProperty(i) ? console.log(`Pertence a mim ${i}`) : console.log(`Não pertence a mim ${i}`)
}

```

RESULTADO NO CONSOLE...

```

[Running] node "c:\Users\Almoxarifado\Documents\javascript\arquivos_das_aulas\089-
hasOwnProperty()__Mostrando_Quais_Elementos_Sao_Herdados_e_Quais_Nao.js"

```

```

1) true
1) true
1) false

2)

```

```
Pertence a mim nome  
Pertence a mim altura  
Não pertence a mim cabelo
```

```
[Done] exited with code=0 in 0.117 seconds
```