

```
//BIND():
//O bind() é um método built-in do Javascript que pode ser utilizado para amarrar o "this" a uma determinada execução, função, dado e etc que pertença a um determinado elemento. Isso possibilita que qualquer atributo pertencente a um determinado elemento possa ser chamado mesmo que externamente e ainda assim fazer com que o "this" do atributo referencie ao seu elemento gerador ou a qualquer outro elemento ao qual o bind amarrar o "this"...

//THIS NÃO PODE SER USADO PARA CHAMAR UM ATRIBUTO QUE PERTENCE A OUTRO OBJETO:
const pessoa = {
  saudacao: 'Bom dia!',
  falar() { //Perceba que temos uma função dentro de um objeto que retorna um outro atributo daquele objeto por através da palavra reservada "this"...
    return console.log(this.saudacao); //como this foi criado dentro de um object ele referencia ao object, perceba que trocamos o nome do objeto pelo this naturalmente...
  }
}

const outroObject = pessoa.falar; //Perceba que passamos para a constante a função falar do objeto pessoa...
console.log("Falhando em capturar o valor do atributo de outro objeto usando o this referenciado em outro objeto:")
outroObject(); //Porém quando chamamos a função ela executa e não consegue encontrar o valor da variável "saudacao", pois a chamada da função é um "this.saudacao" que está no contexto do objeto "pessoa", e não no contexto da constante "outroObject", por isso a chamada da função gera um valor "undefined"...

//AMARRANDO A REFERÊNCIA DE UM ELEMENTO POR ATRAVÉS DO bind():
const usandoBindNoutroObject = pessoa.falar.bind(pessoa); //Perceba que fizemos a mesma operação de cima, tentando capturar o valor do atributo "saudacao" por através da chave "falar" do objeto "pessoa", porém, dessa vez usamos o bind() para amarrar a chamada da função "falar" ao objeto "pessoa"...
console.log("\nConseguindo pegar o valor do atributo de outro objeto que usa this referenciado a ele mesmo com a ajuda do bind():");
usandoBindNoutroObject(); //Veja como o resultado é diferente quando usamos "bind()"...

//AMARRANDO A REFERÊNCIA DE OUTRO ELEMENTO QUE TAMBÉM TENHA O MESMO ATRIBUTO POR ATRAVÉS DO bind():
const et = { //Perceba que criamos um outro objeto "et" que contém o mesmo atributo "saudacao" com um valor diferente...
  saudacao: 'Bom dia! Eu não sou humano, eu sou um ET! Rsr rsrsrsrsrsrs'
}
```

```

const usandoBindParaET = pessoa.falar.bind(et); //Perceba que fizemos a mesma operação de cima, usando a mesma função do objeto pessoa, mas no bind() agora estamos amarrando o "this" ao objeto "et" que por acaso tem o mesmo atributo "saudacao"...
console.log("\nConseguindo pegar o valor do elemento somente trocando a referencia do objeto no bind():");
usandoBindParaET(); //Veja a resposta será segundo o "this" do objeto pelo qual o "bind()" foi amarrado...

//USANDO BIND PARA AMARRAR O THIS DE UMA FUNÇÃO EXECUTADA DENTRO DO SETINTERVAL:
//setInterval também é uma função, por isso, devemos ter cuidado quando executamos um elemento que pertence a outra função com o uso do this
function idadeDaPessoa() { //Perceba que temos aqui uma função construtora que possui um atributo chamado "idade" de valor "0"...
    this.idade = 0;
}

//Agora, fora do contexto da função construtora, chamamos um setInterval globalmente...
setInterval(function() { //Dentro do setInterval criamos uma função anônima...
    this.idade++; //A função tem o papel de chamar um atributo idade referenciando ao objeto que lhe chamou e incrementar 1 ao valor desse objeto, porém, quem está chamando o objeto é contexto global que não possui nenhum atributo "idade"
    console.log(this.idade); //Também queremos que os valores incrementados sejam impressos a cada espaçamento de tempo do setInterval...
}.bind(idadeDaPessoa()), 1000) //Como queremos que o setInterval chame o atributo idade a partir do escopo da função construtora "idadeDaPessoa", nós utilizamos um bind que chama essa função (como ela é uma função construtora poderíamos chamá-la também simplesmente instanciando a função, assim "new idadeDaPessoa")

```

## RESULTADO...

```

[Running] node "c:\Users\User\Documents\JAVASCRIPT\ARQUIVOS_DAS_AULAS\065-Bind()__Amarrando_o_This.js"
Falhando em capturar o valor do atributo de outro objeto usando o this referenciado em outro objeto:
undefined

```

```

Conseguindo pegar o valor do atributo de outro objeto que usa this referenciado a ele mesmo com a ajuda do bind():
Bom dia!

```

```

Conseguindo pegar o valor do elemento somente trocando a referencia do objeto no bind():
Bom dia! Eu não sou humano, eu sou um ET! Rsrrsrsrsrsrsrs
1

```

```
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13
```

```
[Done] exited with code=1 in 13.772 seconds
```