

```
//THIS NO NODE:
```

//O elemento this no Node pode ser usado para importação de objetos, pois o "this" sempre apontará para o "module.exports" do Node. //Nesse quesito, temos que tomar cuidado para não pensar que o "this" é semelhante ao objeto global do Node, pois o "this" aponta para sempre para o módulo de exportação de um arquivo, o objeto global já se refere ao objeto que compõe todo o espaço global do arquivo e ele não está encapsulado para outros arquivos, assim como os módulos estão.

```
//THIS É IGUAL AO MODULE.EXPORTS:
```

```
console.log('1)', this === module.exports)
```

```
//THIS É DIFERENTE DO OBJETO GLOBAL:
```

```
console.log('\n2)', this === global)
```

```
//THIS É DIFERENTE DE MODULE:
```

```
console.log('\n3)', this === module) //Note que o this não é igual ao módulo, pois o module não aponta para o módulo de exportação de um arquivo...
```

```
//THIS É IGUAL A EXPORTS:
```

```
console.log('\n4)', this === exports) //This é igual a "exports" por que o exports aponta sempre para o module.exports de um arquivo...
```

```
//THIS DENTRO DE UMA FUNÇÃO NÃO APONTA PARA O MODULE.EXPORTS:
```

```
function logThis() {  
  console.log('\n5) A função foi executada e o this é...')
```

```
    console.log('this é igual a module exports?:', this === module.exports) //Note que dentro de uma função o this não irá apontar
para o module.exports, ele sempre vai apontar para o contexto onde foi chamado, que no caso é contexto global. E como module.exports
é um objeto que pode armazenar funções mas não pode chamá-las, o this nunca apontará para o module.exports...
    console.log('this é igual a global?:', this === global)
}

logThis()
```

RESULTADO NO CONSOLE...

```
[Running] node "c:\Users\Almoxarifado\Documents\javascript\arquivos_das_aulas\139-This_no_Node.js"
1) true

2) false

3) false

4) true

5) A função foi executada e o this é...
this é igual a module exports?: false
this é igual a global?: true

[Done] exited with code=0 in 0.234 seconds
```