

## Qual é o resultado?

Temos a já conhecida classe `Pessoa` que usamos em exercícios anteriores. Inclusive criamos mais uma vez uma proxy para interceptarmos a chamada ao método `grita`:

```
class Pessoa {  
  
  constructor(nome) {  
    this._nome = nome;  
  }  
  
  get nome() {  
    return this._nome;  
  }  
  
  set nome(nome) {  
    this._nome = nome;  
  }  
  
  grita(frase) {  
    return `${this._nome} grita ${frase}`;  
  }  
}  
  
let pessoa = new Proxy(new Pessoa('Barney'), {  
  
  get(target, prop, receiver) {  
    if(prop == 'grita' && typeof(target[prop]) == typeof(Function)) {  
  
      return function() {  
        console.log(`Interceptei o método: ${prop}, por isso estou ext`  
        Reflect.apply(target[prop], target, arguments);  
      }  
    }  
    return Reflect.get(target, prop, receiver);  
  }  
});  
  
console.log(pessoa.grita('Olá'));
```

Qual das opções abaixo contém o resultado de `console.log(pessoa.grita('Olá'))` ?

**A** Barney grita Olá

**B** Barney grita undefined

.

C

undefined grita Olá



.



undefined



.

O resultado de `console.log(pessoa.grita('Olá'))` é `undefined`. Isso acontece porque estamos chamando `Reflect.apply(target[prop], target, arguments)` sem retornar seu valor. Uma solução é adicionar a cláusula `return`:

```
let pessoa = new Proxy(new Pessoa('Barney'), {
  get(target, prop, receiver) {
    if(prop == 'grita' && typeof(target[prop]) == typeof(Function)) {
      return function() {
        console.log(`Interceptei o método: ${prop}, por isso estou exibindo essa`
          // FALTAVA O RETURN AQUI!
          return Reflect.apply(target[prop], target, arguments); // retorna o va
      }
    }
    return Reflect.get(target, prop, receiver);
  }
});
```

Agora, quando fizemos `console.log(pessoa.grita('Olá'))` o retorno do método `grita` de `pessoa` será passado para `console.log` imprimindo seu resultado no console que é "Barney grita 'Olá'". E se quiséssemos exibir o valor que será retornado no `console.log` do nosso handler?

```
let pessoa = new Proxy(new Pessoa('Barney'), {
  get(target, prop, receiver) {
    if(prop == 'grita' && typeof(target[prop]) == typeof(Function)) {
      return function() {
        console.log(`Interceptei o método: ${prop}, por isso estou exibindo essa mensagem!`)
        // aguarda o retorno em uma variável
        let retorno = Reflect.apply(target[prop], target, arguments);
```

```
console.log(`0 valor retornado do método foi ${retorno}`);
```

```
    return retorno; // retornando o resultado do método
```

```
  }
```

```
}
```

```
return Reflect.get(target, prop, receiver);
```

```
}
```

```
});
```

PRÓXIMA ATIVIDADE