

≡ 20

## Congelando um objeto

Temos a seguinte classe:

```
class Negociacao {
    constructor(data, quantidade, valor) {
        this._data = data;
        this._quantidade = quantidade;
        this._valor = valor;
    }

    get volume() {
        return this._quantidade * this._valor;
    }

    get data() {
        return this._data;
    }

    get quantidade() {
        return this._quantidade;
    }

    get valor() {
        return this._valor;
    }
}
```

Sabemos que uma alteração como essa não tem efeito porque é um *getter*:

```
var negociacao = new Negociacao(new Date(), 1, 100);
negociacao.quantidade = 100; // não tem efeito
```

Contudo, nada impede do programador não seguir a convenção de que propriedades com `_` só podem ser acessadas através de métodos da classe:

```
var negociacao = new Negociacao(new Date(), 1, 100);
negociacao._quantidade = 100; // é possível, apesar de ferir a convenção
```

Aprendemos que podemos congelar objetos, tornando suas propriedades imutáveis. Qual das sintaxes abaixo realiza o "congelamento do objeto `negociacao`"?



`Object.freeze(negociacao);`



Podemos solicitar à `Object.freeze` o congelamento de um objeto com todas as suas propriedades. Qualquer alteração de suas propriedades serão ignoradas. No exemplo anterior, podemos aplicar `Object.freeze` no construtor da classe, tendo como alvo `this`, mas claro, só depois de termos atribuído os valores recebidos no construtor às propriedades.

**B** `Object.Immutable(negociacao);`



**C** `Object.isFrozen(negociacao);`



PRÓXIMA ATIVIDADE