

## Singleton pattern

Singletons são classes que só podem ser instanciadas uma vez, ou seja, a classe retorna uma referência ao seu próprio objeto. Dessa forma toda vez que formos usar essa classe, estaremos usando o mesmo objeto.

Esse pattern possui a característica de inicialização preguiçosa, ou seja, o singleton somente estará disponível em memória a partir do momento de sua inicialização.

Ele pode ser útil quando queremos manter um único ponto de acesso a uma funcionalidade do sistema, por exemplo, o acesso a configuração do programa ou aos recursos de uma API RESTful.

Nesse exemplo é mostrado como criar um singleton usando o module pattern:

```
var singletonEstacionamento = (function () {

    // armazena uma referência ao singleton
    var instance;

    function init(){

        // métodos e propriedades do singleton vão aqui

        // objetos privados
        var patio = [];
        var deletaCarro = function (carro) {
            index = patio.indexOf(carro);
            if (index > -1) {
                patio.splice(index, 1);
            }
        };

        // objetos públicos
        return {
            lista: patio
            entrar: function (carro) {
                patio.push(carro);
            },
            sair: function (carro) {
                deletaCarro(carro);
            }
        };
    };

    // retorna sempre a mesma instância dessa classe
    return {

        // se a instância existir, retorna ela, senão cria uma nova
        getInstance: function () {

            if ( !instance ) {
                instance = init();
            }
        }
    };
})();
```

```
        }

        return instance;
    }
};
})();

est1 = singletonEstacionamento getInstance();
est2 = singletonEstacionamento getInstance();

// true!
console.log(est1 === est2)
```