


1. Apresentar a coleção Map
2. Explicar sua aplicação
3. Explicar a diferença entre Map e Objeto

# Estrutura

```
> const myMap = new Map()
< undefined
> myMap
< Map(0) {}
  ▾ [[Entries]]
    No properties
    size: 0
  ▾ __proto__: Map
    ▶ clear: f clear()
    ▶ constructor: f Map()
    ▶ delete: f delete()
    ▶ entries: f entries()
    ▶ forEach: f forEach()
    ▶ get: f ()
    ▶ has: f has()
    ▶ keys: f keys()
    ▶ set: f ()
    size: (...)
    ▶ values: f values()
    ▶ Symbol(Symbol.iterator): f entries()
    Symbol(Symbol.toStringTag): "Map"
    ▶ get size: f size()
    ▶ __proto__: Object
```



```
const myMap = new Map()
```

## Características:

- Uma coleção de arrays no formato [chave, valor];
- Pode iterado por um loop for...of

# Métodos

Adicionar, ler e deletar



```
const myMap = new Map()  
  
myMap.set('apple', 'fruit');  
// Map(1) {"apple" => "fruit"}  
  
MyMap.get(apple);  
// "fruit"  
  
myMap.delete("apple")  
// true  
  
myMap.get("apple")  
// undefined
```

# Map vs Objeto



- Maps podem ter chaves de qualquer tipo;
- Maps possuem a propriedade length;
- Maps são mais fáceis de iterar;
- Utilizado quando o valor das chaves é desconhecido;
- Os valores tem o mesmo tipo.