

Os objetos não são (por padrão) iterable. Tentando usar para/de um Objeto regular lança um TypeError em tempo de execução:

```
Seja o = {x: 1, y: 2, z: 3};
```

```
para (deixe o elemento de O) { // lança TypeError porque O não é iterável
```

```
  console.log (elemento);
```

```
}
```

Se você deseja iterar através das propriedades de um objeto, você pode usar o loop for/in (introduzido no §5.4.5), ou uso para/de com o

Método object.Keys ():

```
Seja o = {x: 1, y: 2, z: 3};
```

```
Let Keys = "";
```

```
para (deixe k de object.keys (o)) {
```

```
  chaves += k;
```

```
}
```

```
chaves // => "xyz"
```

Isso funciona porque object.keys () retorna uma variedade de propriedades

Nomes para um objeto, e as matrizes são iteráveis ??com/of. Nota também

que essa iteração das chaves de um objeto não é viva como a matriz

Exemplo acima foi - mudanças para o objeto O feito no corpo do loop

não terá efeito na iteração. Se você não se importa com as chaves de

um objeto, você também pode iterar através dos valores correspondentes como

esse:

```
deixe soma = 0;
```

```
para (deixe v de object.values ??(o)) {
```

```
  soma += v;
```

```
}
```

```
soma // => 6
```