

源码实现

```
function myForEach(arr, callback) {
  let T, k;
  if(arr === null) {
    throw new TypeError('this is null or not defined');
  }
  // 用于处理若传入的arr为非数组的情况(string等)
  const o = Object(arr);
  // 无符号右移: 将十进制转化为二进制 右移
  const len = o.length >>> 0;
  if(typeof callback !== 'function') {
    throw new TypeError(`${callback} is not a function`);
  }
  if(arguments.length > 1) {
    T = callback;
  }
  k = 0;
  while (k < len) {
    // 如果指定的属性在指定的对象或其原型链中, 则in运算符返回true
    // 用于过滤未初始化的值
    if(k in o) {
      const kvalue = o[k];
      // kvalue, k, o 对应着forEach回调函数3个参数, 数组当前项的值 数组当前项的索引 数组
      // 对象本身
      // call: 将callback的this指向其自己的内部
      callback.call(T, kvalue, k, o);
    }
    k++;
  }
  return undefined;
}
const test = [1,2,,3];
myForEach(test, (item) => {
  console.log(item); // 1 2 3
})
```

技能点

1、无符号右移：将对应数转化为二进制，接着向右移位得到的数值。

```
const oldValue = 64; // 等于二进制的100000
const newValue = oldValue >>> 5; // 等于二进制的10，即十进制的2
```

2、使用技巧：转化数据（数值不变，其他的类型全部转化为0）

```
1 >>> 0 // 1
undefined >>> 0 // 0
null >>> 0 // 0
string' >>> 0 // 0
```

注意事项

- 1、`async await`的语法糖不起作用：由于其内部封装并调用了回调函数，因此就算用了`async await`也不起作用。
- 2、无法随时退出循环：不能使用`break/continue`的方式退出或中断循环，因为其内部使用`while`循环。