剩余参数　语法允许我们将一个不定数量的参数表示为一个数组

```
function sum(...theArgs) {
  return theArgs.reduce((previous, current) => {
    return previous + current;
  });
}

console.log(sum(1, 2, 3));
// expected output: 6

console.log(sum(1, 2, 3, 4));
// expected output: 10
```

如果函数的最后一个命名参数以...为前缀，则它将成为一个数组，其中从0（包括）到theArgs.length（排除）的元素由传递给函数的实际参数提供。

剩余参数和 arguments对象的区别

剩余参数和 arguments对象之间的区别主要有三个：

- 剩余参数只包含那些没有对应形参的实参，而 arguments 对象包含了传给函数的所有实参。
- arguments对象不是一个真正的数组，而剩余参数是真正的 Array实例，也就是说你能够在它上面直接使用所有的数组方法，比如 sort，map，forEach或pop。
- arguments对象还有一些附加的属性 （如callee属性）。

解构剩余参数

剩余参数可以被解构，这意味着他们的数据可以被解包到不同的变量中。请参阅解构赋值。

```
function f(...[a, b, c]) {
  return a + b + c;
}

f(1)          // NaN (b and c are undefined)
f(1, 2, 3)    // 6
f(1, 2, 3, 4) // 6 (the fourth parameter is not destructured)
```