

1、优化了 json.stringify()

2、arr.flat() 方法会按照一个可指定的深度递归遍历数组，并将所有元素与遍历到的子数组中的元素合并为一个新数组返回。默认值为 1

```
const numbers = [1, 2, [3, 4, [5, 6]]]
console.log(numbers.flat())
// [1, 2, 3, 4, [5, 6]]
```

3、flatMap() 方法首先使用映射函数映射每个元素，然后将结果压缩成一个新数组。从方法的名字上也可以看出来它包含两部分功能一个是 map，一个是 flat（深度为1）。

语法

```
const new_array = arr.flatMap(function callback(currentValue[, index[,
array]]) {
  // 返回新数组的元素
  }, thisArg))

const numbers = [1, 2, 3]
numbers.map(x => [x * 2]) // [[2], [4], [6]]
numbers.flatMap(x => [x * 2]) // [2, 4, 6]
```

4、trimStart() 从字符串的开头删除空格 trimEnd() 方法从一个字符串的右端移除空白字符

5、Object.fromEntries()

方法 Object.fromEntries() 把键值对列表转换为一个对象，这个方法是和 Object.entries() 相对的。

```
Object.fromEntries([['foo', 1], ['bar', 2]])
// {foo: 1, bar: 2}
```

6、Symbol.prototype.description

Symbol 的描述只被存储在内部的 [[Description]]，没有直接对外暴露，我们只有调用 Symbol 的 toString() 时才可以读取这个属性

现在可以通过 description 方法获取 Symbol 的描述：

```
const name = Symbol('My name is axuebin')
console.log(name.description) // My name is axuebin
console.log(name.description === 'My name is axuebin') // My name
is axuebin
```

7、Function.prototype.toString()

`toString()` 方法返回一个表示当前函数源代码的字符串