

用 for-in 还是 for-of 遍历数组？

日期

2017-7-18 星期二

小组成员

姓名
李绍康
李银帅
周雪楠

for-in 和 for-of 有啥区别？

- for-in循环会遍历一个object所有属性名，因为数组也是一个对象，for-in也可以用来遍历数组.
- for-of是ES6引进的新语法，MDN的解释是：for-of语句在可迭代对象(包括 Array, Map, Set, String, TypedArray, arguments 对象等等)上创建一个迭代循环，对每个不同属性的属性值,调用一个自定义的有执行语句的迭代钩子.

举一个栗子🌰先说明问题:

```
Object.prototype.objFunc = function () {};  
Array.prototype.arrFunc = function () {};  
  
let arr = [0, 1, 2];  
arr.child = "hello";  
  
for (let item in arr) {  
  console.log(item); // 0, 1, 2, child, objFunc, arrFunc  
}  
  
for (let item of arr) {  
  console.log(item); // 0, 1, 2  
}
```

从上面的例子可以看出，for-in 遍历（当前对象及其原型上的）每一个属性名称，而 for-of 遍历（当前对象上的）每一个元素值。

但是for-of的iterator方法存在兼容性问题(低版本safari再次拖后腿)，使用for-in如果不想遍历原型方法和属性的话，可以在循环内部判断一下，通过`_.hasOwnProperty()`方法可以判断某属性是否是该对象的实例属性，然而并过滤不掉该数组对象的实例化属性`child`，具体可以查看[李国帅组](#)本期的讨论。

```
for (let key in arr) {
  if (arr.hasOwnProperty(key)){
    console.log(key); // 0, 1, 2, child
  }
}
```

所以在低版本里面，如果数组真的没有被污染的话，可以尝试用for-in加上`_.hasOwnProperty()`方法。

为啥不用 `forEach()` 可以用

还是前面那个arr

```
Object.prototype.objFunc = function () {};  
Array.prototype.arrFunc = function () {};  
  
let arr = [0, 1, 2];  
arr.child = "hello";  
  
arr.forEach(  
  function(item, index, array){  
    if(item == 1)return;  
    console.log(item);  
  }  
)  
//0,2  
  
arr.map(  
  function(item, index, array){  
    if(item == 1)return;  
    return item*2;  
  }  
)  
//[0,undefined,4]
```

可以看出 `forEach()` 和 `map()` 并没有`break`，不会跳出循环。如果一定要在一定条件下中断循环，可以尝试用`try...catch`抛出异常阻断。

结论

比较上述遍历方式：

- **for** 最常见的遍历方式，浏览器都支持；

- **for-in** 一直都支持，但是不推荐，两个原因：遍历键值不能保证数组的顺序；遍历可能会带出原型链上的可枚举属性，需要加上hasOwnProperty()方法过滤；
- **for-of** ES6+，适用于全部元素的遍历，会过滤掉原型链上的属性，但是不能返回元素的index，兼容性不能保证；
- **forEach()** 非常好用的遍历方式，ES5+. 可以拿到元素的index，缺陷是不能中断，但可以用 try...catch 来 hack；
- **map()** 非常好用的遍历方式，ES5+. 跟forEach()类似，但是会return一个新数组。