18.迭代器和生成器

学习要点：

1. 迭代器和生成器
2. 默认迭代接口

本节课我们来开始学习 ES6 新增的迭代器和生成器的用法。

一. 迭代器和生成器

1. 迭代器(Iterator)，用于给数据结构提供统一的访问遍历的机制；
2. ES6 之前的迭代器比较麻烦，而现在引入了生成器对象，让迭代器更加容易；
3. 首先创建一个生成器方法，方法名前面加上*号，迭代的内容之前使用 yield；

```
//生成器
function *cit() {
    yield 1;
    yield 2;
    yield 3;
}
```

PS：1,2,3 是我们要遍历的值；下面我们要创建迭代器；

4. 迭代器对象的.next()方法，类似指针，每次执行将下移一行；

```
//迭代器
let it = cit();
//每执行一次.next()将下移一行
console.log(it.next());     //1, false
console.log(it.next());     //2, false
console.log(it.next());     //3, false
console.log(it.next());     //undefined, true
```

PS：属性 value 得到值，没有返回 undefined，当没有值了，done 则返回 true；

5. 生成器结合循环语句，并且进行传递数组进行迭代；

```
function *cit(items) {
    for (let i = 0 ; items.length; i++) {
        yield items[i]
    }
}

let it = cit([1,2,3,4,5]);
console.log(it.next().value);
```

PS：如果作为匿名函数使用：let cit = function *(items);

二．默认迭代接口

1. 很多数据结构类型拥有默认迭代接口，比如：Array、Map、Set 等等；
2. 对于原生就支持迭代器的数据结构，我们不用自己编写生成器迭代器；
3. 最简单的迭代方式，就是使用 for...of 迭代语句去遍历即可；
4. 对于 Array 数组类型，它提供了有关三个方法：keys()、values()和 entries()；

```
let items = [1, 2, 3, 4, 5];

console.log(items.keys());   //key, Object [Array Iterator]
console.log(items.values()); //value, Object [Array Iterator]
console.log(items.entries());//key+value, Object [Array Iterator]
```

5. 最简单的迭代方式，就是使用 for...of 迭代语句去遍历即可；

```
//for..of 遍历得到 value 值
for (let i of items.values()) {
    console.log(i);
```

6. 虽然 for...of 特别方便，不过你想要用.next()语法也是支持的；

```
let values = items.values();
console.log(values.next());
```

PS：下节课，我们把其它几种数据类型的默认迭代都演示一遍；