

# 11、对象的扩展

## 属性的简洁表示法

- ES6 允许在大括号里面，直接写入变量和函数，作为对象的属性和方法。这样的书写更加简洁。

```
const foo = 'bar';
const baz = {foo};
baz // {foo: "bar"}

// 等同于
const baz = {foo: foo};
```

- 上面代码中，变量foo直接写在大括号里面。这时，属性名就是变量名，属性值就是变量值。下面是另一个例子。

```
function f(x, y) {
  return {x, y};
}

// 等同于
function f(x, y) {
  return {x: x, y: y};
}
f(1, 2) // Object {x: 1, y: 2}
```

- 除了属性简写，方法也可以简写。

```
const o = {
  method() {
    return "Hello!";
  }
};

// 等同于
const o = {
  method: function() {
    return "Hello!";
  }
};
```

- 下面是一个实际的例子。

```
let birth = '2000/01/01';
```

```

const Person = {
  name: '张三',
  //等同于birth: birth
  birth,
  // 等同于hello: function ()...
  hello() { console.log('我的名字是', this.name); }
};

//这种写法用于函数的返回值，将会非常方便。
function getPoint() {
  const x = 1;
  const y = 10;
  return {x, y};
}

getPoint()
// {x:1, y:10}

```

- CommonJS模块输出一组变量，就非常合适使用简洁写法。

```

let ms = {};

function getItem (key) {
  return key in ms ? ms[key] : null;
}

function setItem (key, value) {
  ms[key] = value;
}

function clear () {
  ms = {};
}

module.exports = { getItem, setItem, clear };
// 等同于
module.exports = {
  getItem: getItem,
  setItem: setItem,
  clear: clear
};

```

- 简洁写法在打印对象时也很有用

```
let user = {  
  name: 'test'  
};  
  
let foo = {  
  bar: 'baz'  
};  
  
console.log(user, foo)  
// {name: "test"} {bar: "baz"}  
console.log({user, foo})  
// {user: {name: "test"}, foo: {bar: "baz"}}
```

- 上面代码中，console.log直接输出user和foo两个对象时，就是两组键值对，可能会混淆。
- 把它们放在大括号里面输出，就变成了对象的简洁表示法，每组键值对前面会打印对象名，这样就比较清晰了。