

## 13. 对象的简写改进

学习要点：

1. 简写方案
2. 表达式方案

本节课我们来开始学习 ES6 新增的对象的扩展和改进功能。

### 一. 简写方案

1. ES6 可以让对象字面量中属性初始值实现简写，一定程度降低了代码量；

```
function fn(name, age) {  
  //ES5 的写法  
  return {  
    name : name,  
    age : age  
  }  
}  
  
function fn(name, age) {  
  //ES6 的写法  
  return {  
    name, age  
  }  
}  
  
console.log(fn('Mr.Lee', 'age'));
```

2. ES6 还提供了对象字面量中方法的简写方式，也降低了一定的代码量；

```
let obj = {  
  //ES5 的写法  
  fn : function () {  
    return 'ES5 fn';  
  }  
};  
  
let obj = {  
  //ES6 的写法  
  fn() {  
    return 'ES6 fn';  
  }  
};  
  
console.log(obj.fn());
```

## 二. 表达式方案

1. ES6 允许对象字面量中，使用表达式进行属性名称的拼装操作；

```
//拼装组合属性
let obj = {
  ['user' + 'Name'] : 'Mr.Lee',
};
console.log(obj.userName);
console.log(obj['userName']);

//有空格的情况下
let obj = {
  ['user' + ' Age'] : 100,
};
console.log(obj['user Age']);

//字符串属性
let obj = {
  'user Gender' : '男',
};
console.log(obj['user Gender']);
```

2. ES6 提供了在对象字面量中使用可计算(动态)属性名称，具体如下；

```
//通过变量 myName 动态更改 obj 属性名
//问题是当变量值更改了，属性名就失效了
let myName = 'name';
let obj = {
  name : 'Mr.Lee'
};
console.log(obj[myName]);

//使用[myName]可计算属性名
//实现了真正的动态计算
let myName = 'name';
let obj = {
  [myName] : 'Mr.Lee'
};
console.log(obj[myName]);
```

3. ES6 在对象字面量方法上，也可以使用拼装名称；

```
let obj = {
  ['f' + 'n']() {
    return 'fn';
  }
};
console.log(obj.fn());
```