

04. 数组和对象的解构

学习要点：

1. 数组解构
2. 对象解构

本节课我们来开始学习 ES6 中数组和对象解构赋值的方法。

一. 数组解构

1. ES6 提供了对数组和对象的字面量提取相关数据的方法：解构操作；
2. 为何要使用解构操作？是因为 JSON 格式的普及，导致大量数据提取工作；
3. 而这种提取过程，在 ES6 的解构语法中，带来了极大的便捷性；
4. 数组解构赋值，有两种基本的写法：1. 分行解构；2 单行解构；

```
let info = ['Mr.Lee', 100, '男']; //数组赋值
let [name, age, gender] = info; //数组解构赋值
```

```
let info = ['Mr.Lee', 100, '男'],
    [name, age, gender] = info; //同上
```

//单行解构，以上均可输出 name 值

```
let [name, age, gender] = ['Mr.Lee', 100, '男'];
console.log(name);
```

5. 从上面的例子分行或单行，都可以确定必须一一完美匹配才可以正确赋值；

//数组层次也需要匹配

```
let [name, [age, gender]] = ['Mr.Lee', [100, '男']];
```

//用逗号作为占位符不赋值

```
let [, gender] = ['Mr.Lee', 100, '男'];
```

6. 在变量解构时，可以在数组的元素中设置一个默认值；

//当 gender 没有赋值时，采用默认值

```
let [name, age, gender = '男'] = ['Mr.Lee', 100];
```

7. 还有一种...var 的语法，可以将没有赋值的内容都赋值给这个变量；

//不定元素，将其余都赋值给 other

```
let [name, ...other] = ['Mr.Lee', 100, '男'];
```

二. 对象解构

1. 对象的解构方式和数组大同小异，定义一个对象字面量，然后解构赋值；

//定义对象字面量

```
let obj = {
    name : 'Mr.Lee',
```

```
    age : 100,
  };

//解构对象至变量
let {name, age} = obj;      //或({name, age} = obj);

//直接输出变量
console.log(name);
console.log(age);
```

2. 如果说，解构的变量名是已经存在的变量，那会导致如何？

```
let obj = {
  name : 'Mr.Lee',
  age : 100,
}, name = 'Mr.Wang';      //被替代

({name, age} = obj);
console.log(name);
```

3. 对象变量解构也可以设置一个默认值，在没有赋值时输出默认值；

```
let obj = {...}
let {name, age, gender = '女'} = obj;
console.log(gender)      //如果没有默认值则 undefined
```

4. 如果不想要对象属性名作为解构变量，可以通过键值对的方式更改变量名；

```
let {name : myName, age : myage} = obj;      //name 将失效
```

5. 在对象字面量里，还嵌套了对象，解构时也用相同的方法是解开即可；

```
let obj = {
  name : 'Mr.Lee',
  age : 100,
  info : {
    id : 1,
    gender : '男'
  }
};

let {info : {id, gender}} = obj;
console.log(gender);
```

6. 对象的解构也支持单行的简写模式，具体如下：

```
let {name, age} = {name : 'Mr.Lee', age : 100};
```