

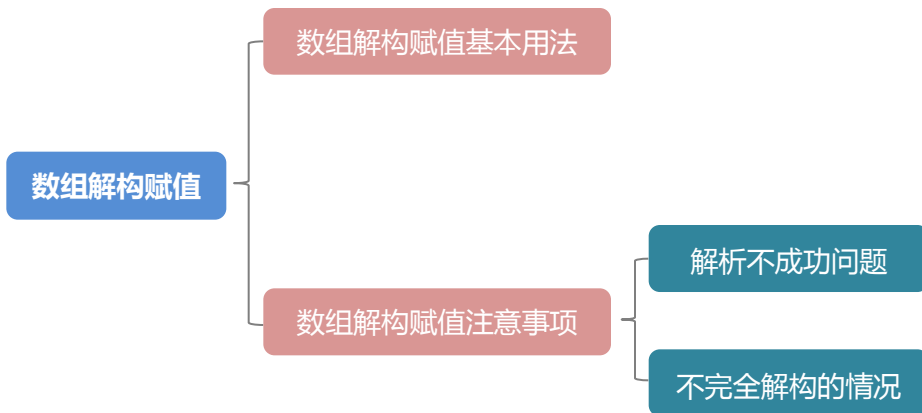
## 第二章 解构赋值

一样的在线教育，不一样的教学品质



# 目录 Contents

- ◆ 数组解构赋值
- ◆ 对象解构赋值
- ◆ 字符串的解构赋值
- ◆ 函数参数的解构赋值
- ◆ 解构赋值的优势



# 1. 数组解构赋值基本用法

将数组中的值通过解构赋值的方式，取出来赋给变量：

```
let arr = [1, 2, 3];  
let [num1, num2, num3] = arr;  
console.log(num1, num2, num3);
```

**注意：**解构赋值等号两侧的结构是类似的。

## ■ 2.数组解构赋值注意事项

### ◆ 解析不成功的问题

```
let [num1, num2] = [6]  
console.log(num1, num2);
```

### ◆ 不完全解构的情况

所谓的不完全解构，表示**等号左边只匹配右边数组的一部分**；

```
let [num1, num2] = [1, 2, 3];  
console.log(num1, num2);
```



## 总结

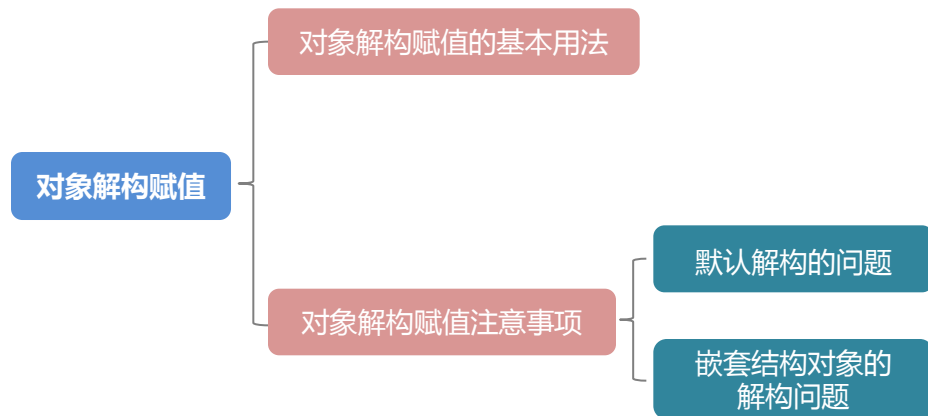
### 重难点

1. 数组解构赋值基本用法
2. 数组解构赋值注意事项



# 目录 Contents

- ◆ 数组解构赋值
- ◆ 对象解构赋值
- ◆ 字符串的解构赋值
- ◆ 函数参数的解构赋值
- ◆ 解构赋值的优势





# 1.对象解构赋值基本用法

解构不仅可以用于**数组**，还可以用于**对象**。

```
let {
  userName,
  userAge
} = {
  userName: 'ls',
  userAge: 20
};
console.log(userName, userAge);
```

**注意：**变量名必须与属性的名称一致，才能够取到正确的值。

## ■ 2.对象解构赋值注意事项

◆ 默认解构问题

◆ 嵌套结构对象的解构问题



# 目录 Contents

- ◆ 数组解构赋值
- ◆ 对象解构赋值
- ◆ 字符串的解构赋值
- ◆ 函数参数的解构赋值
- ◆ 解构赋值的优势

字符串也可以进行解构赋值，因为**字符串被转换成了一个类似于数组的对象**。

```
let [a, b, c, d, e, f] = 'itcast';  
console.log(a, b, c, d, e, f);
```

类似于数组的对象都有length属性，因此,也可以**对length属性进行解构赋值**。

```
let {  
    length: len  
} = 'itcast';  
console.log('len=', len);
```



# 目录 Contents

- ◆ 数组解构赋值
- ◆ 对象解构赋值
- ◆ 字符串的解构赋值
- ◆ 函数参数的解构赋值
- ◆ 解构赋值的优势

函数的参数也能够进行解构的赋值，如下所示：

```
function test([x, y]) {  
    return x + y;  
}  
console.log(test([3, 6]));
```

**说明：**函数test的参数不是一个数组，而是通过解构得到的变量x和y。



# 目录 Contents

- ◆ 数组解构赋值
- ◆ 对象解构赋值
- ◆ 字符串的解构赋值
- ◆ 函数参数的解构赋值
- ◆ 解构赋值的优势

- ◆ 交换变量的值
- ◆ 函数可以返回多个值
- ◆ 函数返回一个对象
- ◆ 提取JSON对象中的数据





一样的在线教育，不一样的教学品质