JavaScript进阶(ES6)

---ES6中变量的解构赋值





- > 数组、对象的解构赋值
- > 字符串、数字的解构赋值
- > 函数参数的解构赋值
- 解构赋值的常见应用及注意事项



数组、对象的解构赋值

•什么是解构赋值(Destructuring)

- ES6允许按照一定模式,从数组和对象中提取值,对变量进行赋值,这被称为解构赋值
- 这种写法属于"模式匹配",只要等号两边的模式相同,左边的变量就会被赋予对应的值

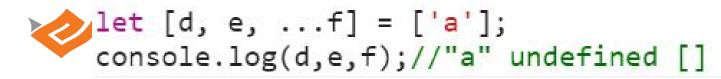
•数组的解构赋值

```
let [ , , third] = ["foo", "bar", "baz"];
console.log(third);// "baz"

let [x, , y] = [1, 2, 3];
console.log(x,y);//1 3

let [head, ...tail] = [1, 2, 3, 4];
console.log(head,tail);//1 [2, 3, 4]
```

参见实例demo09 数组的解构赋值



数组、对象的解构赋值

•对象的解构赋值

```
var { bar2, foo2 } = { foo2: "ccc", bar2: "ddd" };
console.log(foo2,bar2);
var { baz3 } = { foo3: "ccc", bar3: "ddd" };
console.log(baz3);
var { foo4: baz4 } = { foo4: 'aaa', bar4: 'bbb' };
console.log(baz4);// "aaa"
                           解构赋值时,左侧为键值对时
ccc ddd
                           要注意键值对赋值时的对应关系
undefined
                           如果是键值对的情况,键只用于
aaa
                           匹配,真正赋给的是对应的值
```



- > 数组、对象的解构赋值
- > 字符串、数字的解构赋值
- > 函数参数的解构赋值
- 解构赋值的常见应用及注意事项



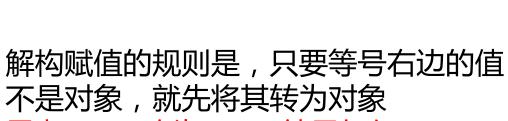
字符串、数字的解构赋值

•字符串的解构赋值

```
const [a, b, c, d, e] = 'hello';
console.log(a); // "h"
console.log(b); // "e"
console.log(c); // "l"
console.log(d); // "l"
console.log(e); // "o"
```

•数字的解构赋值

```
let {toString: s1} = 123;
console.log(s1); //
s1 === Number.prototype.toString // true
function toString() { [native code] }
true
```



思考:123改为true,结果如何?



- > 数组、对象的解构赋值
- > 字符串、数字的解构赋值
- > 函数参数的解构赋值
- 解构赋值的常见应用及注意事项



函数参数的解构赋值

•参数的解构赋值 案例一

```
function move1({x = 0, y = 0} = {}) {
    return [x, y];
}
console.log(move1({x: 3, y: 4})); // [3, 4]
console.log(move1({x: 3})); // [3, 0]
console.log(move1({})); // [0, 0]
console.log(move1()); // [0, 0]
```

•参数的解构赋值 案例二

```
function move2({x, y} = { x: 0, y: 0 }) {
    return [x, y];
}
console.log(move2({x: 3, y: 8})); // [3, 8]
console.log(move2({x: 3})); // [3, undefined]
console.log(move2({})); // [undefined, undefined]
console.log(move2()); // [0, 0] 参见实例demo12 函数参数解构赋值
```

- > 数组、对象的解构赋值
- > 字符串、数字的解构赋值
- > 函数参数的解构赋值
- 解构赋值的常见应用及注意事项



解构赋值的常见应用一

•交换变量(写法简洁,语义清晰)

```
var [x,y] = ["a","b"];
[x, y] = [y, x];
console.log(x,y);//b,a
```

•从函数中返回多个值

- 函数只能返回一个值,如果要返回多个值,只能将它们放在数组或对象里返回
- 有了解构赋值,取出这些值就非常方便

```
function example() {
    return [1, 2, 3];
}
var [a, b, c] = example();
console.log(a,b,c);//1,2,3
```



参见实例demo13 解构赋值的常见应用



解构赋值的常见应用二

•提取JSON数据

```
var jsonData = {id: 42,status: "OK",data: [867, 5309]};
let { id, status, data: number } = jsonData;
console.log(id, status, number);// 42, "OK", [867, 5309]
```

•通过解构赋值给函数传参并指定默认参数

```
jQuery.ajax = function (url, {
    async = true,
    beforeSend = function () {},
    cache = true,

// ... more config
}) {
// ... do stuff
};
```

解构赋值的注意事项(括号问题)

- •对于编译器来说,一个式子到底是模式,还是表达式 分 有办法从一开始就知道,必须解析到(或解析不到)等号 才能知道
- •由此带来的问题是,如果模式中出现圆括号怎么处理, ES6的规则是,只要有可能导致解构的歧义,就不得使用 圆括号
- •建议只要有可能,就不要在模式中放置圆括号





作业

- •复习本章的课件和练习
- •阅读ES6 标准入门的1-3章

