


JavaScript 引用数据类型

一、什么是引用数据类型？

 通俗解释：

- 原始类型是“直接存放值”的，比如数字 100、字符串 "hello"。
- 引用类型是“存放地址”的，它的值是一个指向内存中某个对象的“引用”。
- **我们操作的是那个地址指向的内容**，而不是值本身。

 引用数据类型包括：

1. 函数 (Function)
2. 数组 (Array)
3. 对象 (Object)

✨ 一、函数 (Function)

1. 概念解释：

函数是把一段可以“重复使用”的代码打包起来，等我们需要时就可以“调用”它。

2. 函数的作用：

- 减少重复代码
- 使程序结构更清晰

- 可以接收输入、返回结果

3. 三种函数定义方式：

✅ 函数声明：

```
function sayHello(name) {  
  return "Hello, " + name;  
}  
console.log(sayHello("小明")); // 输出: Hello, 小明
```

✅ 函数表达式：

```
const greet = function(name) {  
  return "Hi, " + name;  
};  
console.log(greet("小红")); // 输出: Hi, 小红
```

✅ 箭头函数（ES6）：

```
const greetArrow = (name) => {  
  return "Hey, " + name;  
};  
console.log(greetArrow("小刚")); // 输出: Hey, 小刚
```

4. 函数的参数、返回值、调用方式：

项目	说明
参数	函数输入的变量，如 <code>name</code>
返回值	函数输出的结果，用 <code>return</code> 返回
调用函数	使用函数名+括号，如 <code>sayHello("小明")</code>

二、数组（Array）

1. 概念解释：

数组就是一串 **有顺序的数据集合**，可以用来装多个值。每个值都有编号（索引），从0开始。

2. 创建数组的方式：

```
let fruits = ["苹果", "香蕉", "橘子"];
```

3. 访问和修改元素：

```
console.log(fruits[0]); // 苹果
fruits[1] = "草莓";
console.log(fruits); // ["苹果", "草莓", "橘子"]
```

4. 常用方法：

方法	作用	示例
<code>push()</code>	从末尾添加一个元素	<code>fruits.push("梨子")</code>
<code>pop()</code>	删除最后一个元素	<code>fruits.pop()</code>
<code>length</code>	获取数组长度	<code>fruits.length</code>
<code>forEach()</code>	遍历数组，每个元素都执行一次函数	<code>fruits.forEach(f => console.log(f))</code>

```
fruits.forEach(function(item, index) {  
  console.log(index + ": " + item);  
});
```

三、对象（Object）

1. 概念解释：

对象是一种“**键值对结构**”，可以用来表示一个“东西”或者“实体”的多个属性和行为。

2. 定义对象的方式：



```
let student = {  
  name: "小明",  
  age: 18,  
  sayHi: function() {  
    console.log("你好, 我是" + this.name);  
  }  
};
```

3. 访问属性的方式:

点语法:



```
console.log(student.name); // 小明
```

中括号语法:



```
console.log(student["age"]); // 18
```

4. 修改和新增属性:



```
student.age = 19;           // 修改  
student.gender = "男";      // 新增
```

5. 嵌套结构：对象中嵌套数组 / 对象



```
let classroom = {
  name: "三年级1班",
  students: [
    { name: "小明", age: 10 },
    { name: "小红", age: 9 }
  ]
};

console.log(classroom.students[1].name); // 输出: 小红
```

小结板块

类型	特点	举例
函数	封装行为，可传入参数、返回值	<code>function greet(name) { ... }</code>
数组	有序的数据集合，下标从0开始	<code>["苹果", "香蕉"]</code>
对象	键值对结构，描述事物的属性和功能	<code>{ name: "小明", age: 18 }</code>