# 一、函数的定义和调用

学习目标：

1、掌握函数的定义调用

2、理解数组的定义和多维数组

3、理解数组的一些基本操作

## 1、函数的定义

函数就是重复使用的代码块，使用关键字function定义函数。

```
<script>
    // 定义函数
    function fnAlert(){
        alert('hello');
    }
    // 箭头函数
    let fnAlert = () =>{
        alert('hello')
    }
</script>
```

## 2、函数的调用

函数调用就是函数名加小括号，如：函数名（参数）

```
<script>
    // 函数定义
    function fnAlert(){
        alert('hello!')
    }
    // 函数的调用
    fnAlert();
</script>
```

## 3、定义有参数有返回值的函数

定义函数时，函数如果有参数，参数放到小括号里面，函数如果有返回值，返回值通过return关键字来返回。

```
<script>
    // 定义fnAdd返回两个值的和
    function fnAdd(iNum01, iNum02){
        let iRs = iNum01 + iNum02;
        return iRs;
        // 执行完return之后，后面的代码不会执行
        alert('哇哇哇');
    }
    let iCount = fnAdd(3,4);
    alert(iCount);
<script>
```

## 4、函数的不定参（可变参）

```
// arguments 代表所有实参的集合（类数组），可以通过下标获取各个实参，通过length获取集合长度
function args(){
    console.log(arguments);
    for (var i = 0; i < arguments.length; i++) {
        console.log("arguments的各个参数为："+arguments[i]);
    }
}

args(13,41,9995,1120.90,"你好","不定参");
```

## 5、示例代码

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <script>
        // 定义函数的关键字function
        function fnShow(){
            alert('我是一个没有参数没有返回值的函数')
        };
        // 调用函数fnShow
        fnShow();

        // 定义有参数和有返回值的函数
        function fnSum(iNum1, iNum2){
            let iResult = iNum1 + iNum2
            // return ： 1.可以为函数提供返回值 2.当执行return语句以后函数执行结束，后面的
代码不会再执行
            return iResult;
            alert('哇哇哇')
        }
        // 调用函数fnSum
        let iNum = fnSum(1,4);
        alert(iNum);
    </script>
</head>
<body>
```

```
</body>
</html>
```

# 二、数组及操作方法

## 1、数组的介绍

数组就是一组数据的集合，JavaScript中，数组里面的数据可以是不同类型的数据，好比python里面的列表。

## 2、数组的定义

```
// 实例化对象方式创建
let aList = new Array(1,2,3);

// 推荐使用
let aList2 = [1,2,3,'asd'];

// 创建空数组
let arr1 = new Array();
```

## 3、多维数组

多维数组指的是数组的成员就是数组，把这样的数组叫做多维数组。

```
let aList = [[1,2,3],['a','b','c']]
```

## 4、数组的操作

### 1、获取数组的长度（掌握）

```
let aList = [1,2,3,4]
alert(aList.length); //弹出4
```

### 2、根据下标取值（掌握）

```
let aList = [1,2,3,4]
alert(aList[0]); //弹出1
```

### 3、从数组最后添加和删除数据（掌握）

`push()` 方法用于向数组的末尾添加一个或多个元素

`pop()` 方法用于删除并返回数组的最后一个元素

```
let aList = [1,2,3,4]
aList.push(5);
alert(aList); // 弹出1,2,3,4,5
aList.pop();
alert(aList); //弹出1,2,3,4
```

## 4、copyWithin( ) — 批量复制方法（了解）

**copyWithin( )**会按照指定范围复制数组中的部分内容，然后将它们插入到指定索引开始的位置。

```
let ints = [0,1,2,3,4,5,6,7,8,9];

//从ints中复制索引0开始的内容，插入到索引5开始的位置
//在源索引或目标索引到达数组边界时停止
ints.copyWithin(5);
console.log(ints);//[0,1,2,3,4,0,1,2,3,4];

//从ints中复制索引5开始的内容，插入到索引0开始的位置
//ints.copyWithin(0,5);
//console.log(ints);//[5,6,7,8,9,5,6,7,8,9];

//从ints中复制索引0开始到索引3结束的内容
//插入到索引4开始的位置
//ints.copyWithin(4,0,3);
//console.log(ints);//[0,1,2,3,0,1,2,7,8,9];

//JavaScript引擎在插值前会完整复制范围内的值
//因此复制期间不存在重写的风险
//ints.copyWithin(2,0,6);
//console.log(ints);//[0,1,0,1,2,3,4,5,8,9];

//支持负索引值
//ints.copyWithin(-4,-7,-3);
//console.log(ints);//[0,1,2,3,4,5,3,4,5,6];
```

## 5、fill( ) — 填充数组方法(了解)

使用**fill( )方法**可以向一个已有的数组中插入全部或部分相同的值。

```
let zeroes = [0,0,0,0,0];

//用5填充整个数组
zeroes.fill(5);
console.log(zeroes);//[5,5,5,5,5];

//用6填充索引大于等于3的元素
zeroes.fill(6,3);
console.log(zeroes);//[0,0,0,6,6];

//用7填充索引大于等于1且小于3的元素
zeroes.fill(7,1,3)
console.log(zeroes);//[0,7,7,0,0];

//用8填充索引大于等于1且小于等于4的元素
zeroes.fill(8,-4,-1);
console.log(zeroes);//[0,8,8,8,0];
```

## 6、根据下标添加和删除元素（理解）

> arr.splice(start,num,element1,...,elementN)
>
> 参数解析：
>
> 1、start：必需，开始删除的索引
>
> 2、num：可选，删除数组元素的个数
>
> 3、elementN：可选，在start索引位置要插入的新元素
>
> 此方法会删除从start索引开始的num个元素，并将elementN参数插入到start索引位置

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <script>
        // 定义数组，new关键字用于创建对象的实例。
        let aArray1 = new Array(1,2,3);
        console.log(aArray1);

        // 直接赋值一个数组，数组的表现形式是一对中括号
        let aArray2 = [3,6,9]
        console.log(aArray2);
        alert(aArray2);

        // 多维数组
        let aArray3 = [[1,2,3],[3,6,9]];
        console.log(aArray3);
        // 根据下标获取数据
        alert(aArray3[0][1]);

        // 演示数组的相关操作
        let aArray4 = [3,6,9];
        // 获取数组的长度
        alert(aArray4.length);
        // 根据下标取值
        alert(aArray4[2]);
        // js里面不支持负数下标
        // alert(aArray4[-1]);
        // 根据下标修改数据
        aArray4[1] = 26
        console.log(aArray4);
        // 追加数组
        aArray4.push('hello');
        console.log(aArray4);
        // 删除最后一个元素
        let oValue = aArray4.pop()
        console.log(oValue);
        console.log(aArray4)

        // 插入数据
        // 1.开始删除的索引
        // 2.删除的个数
```

```
        // 3.插入的数据
        aArray4.splice(1, 0, '苹果');
        console.log(aArray4);
        // 从下标2删除2个数据
        aArray4.splice(2,2)
        console.log(aArray4);
        aArray4.splice(1, 0, '鸭梨', '香蕉')
        console.log(aArray4);
    </script>
</head>
<body>

</body>
</html>
```