

1. 一维数组

动态初始化:

```
int[] i = new int[4]; // 声明后即有值, 默认为0
```

```
i[0] = 1;
```

```
i[1] = 2;
```

```
i[2] = 3;
```

```
i[3] = 4;
```

静态初始化:

```
int[] ii = new int[] {1, 2, 3, 4};
```

所有数组都有一个属性, length, 表示其长度
ii.length

2. 多维数组

二维数组:

```
int[][] ii = new int[][] {  
    {1, 2},  
    {3, 4}  
};
```

```
int[][] jio = new int[][][3];
```

```
int[][] ii1 = new int[2][2];
```

```
int[][] ii2 = new int[2][2];
```

这写法比较特殊, 和C中截然相反, 我个人猜测可能是因为Java中的数组是对象的原因; C中的一维数组本质上是二级指针

特殊写法 `int[] x, y[];` // x是一维, y是二维

```
int[][] arr = new int[][] {
```

```
    {3, 8, 2},
```

```
    {2, 7},
```

```
    {9, 0, 1, 6}
```

```
}; // 多维数组的维度可以不一致
```

```

int len = arr.length
int res = 0
for (int i = 0; i < len; i++) {
    int[] arr0 = arr[i];
    int len0 = arr0.length
    for (int j = 0; j < len0; j++) {
        res += arr0[j];
    }
}

```

3. 数组中的常见算法

4. 数组操作常见问题

ArrayIndexOutOfBoundsException 下标越界

NullPointerException 空指针异常

↓
 int[] arr = null;
 System.out.println(arr[0]);