

数组

数组的定义

数组：得到一个容器，存储一批同种类型的数据。

静态初始化数组

特点：定义数组的时候，数据已经确定了，数据也存入进去了

```
// 完整格式
数据类型[] 数组名 = new 数据类型[]{元素1, 元素2 , 元素3... };
double[] scores = new double[]{89.9, 99.5, 59.5, 88.0};
int[] ages = new int[]{12, 24, 36}
```

```
// 简化格式
数据类型[] 数组名 = { 元素1, 元素2 , 元素3, ... };
int[] ages = {12, 24, 36};
```

格式

动态初始化数组

特点：定义的时候只确定数组的元素类型和长度，不存入具体的数据（先定义后赋值）

```
数据类型[] 数组名 = new 数据类型[长度];
int[] arr = new int[3];
```

```
// 后赋值
arr[0] = 10;
System.out.println(arr[0]); // 10
```

格式:

数组的访问

格式 — 数组名称[索引]

属性

- 索引：元素的编号，从0开始
- 长度：length 元素的个数

取值 — 数组名称[索引]

赋值 — 数组名称[索引] = 数据;

数组的遍历

遍历：一个一个的访问数据

```
int[] ages = {20, 30, 40, 50};
for (int i = 0; i < ages.length; i++) {
    System.out.println(ages[i]);
}
```

数组遍历方式

数组的案例

参见代码 — 自己练习

数组的内存图

方法区 — 放类信息class文件

栈内存 — 方法执行的区域，变量存储的区域

堆内存 — new出来的东西都在这里，数组对象

参见PPT

数组使用的常见问题

注意访问索引不要越界

如果数组变量里面没有地址，存储的是null,输出是没毛病，但是调用数组的功能就出现空指针异常

Debug工具的使用

作用：断点调试，看代码流程、看代码是否有bug,定位错误理解代码的重要手段

怎么使用 — 参见PPT