# X 语言程序设计

# 本节内容

一维数组

二维数组

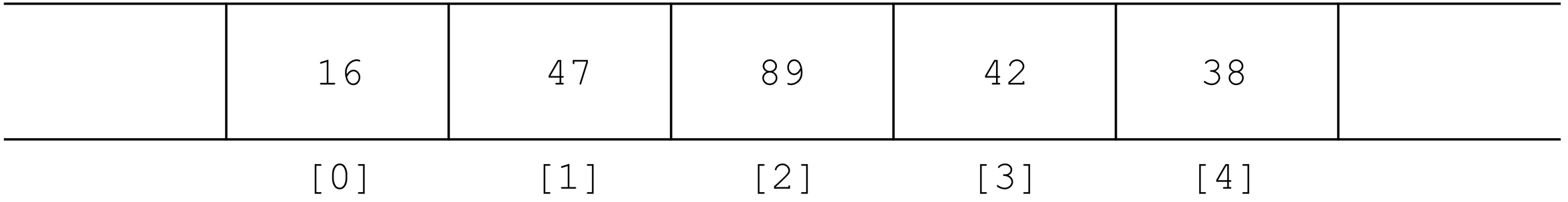
字符数组与字符串

# 数组

相同数据类型的集合。

数组的长度一旦定义就不能改变。

数组中的每一个元素可以用下标表示位置，如果一个数组中有 $\nu$ 个元素，
那么下标的取值范围是 $0 \sim \nu - 1$ 。



# 一维数组的声明与初始化

数据类型 数组名 [ 数组长度 ];

```
int a[5];
char ch[10];
float f[3];
```

# 一维数组的调用

使用“数组名 [ 下标 ]” 的形式可以访问数组中的某个元素。

```
int a[5];
a[0] = 6;
a[1] = 78;
printf("%d", a[1]);
```

# 一维数组的遍历

遍历：按顺序获取查看数组中的每一个元素

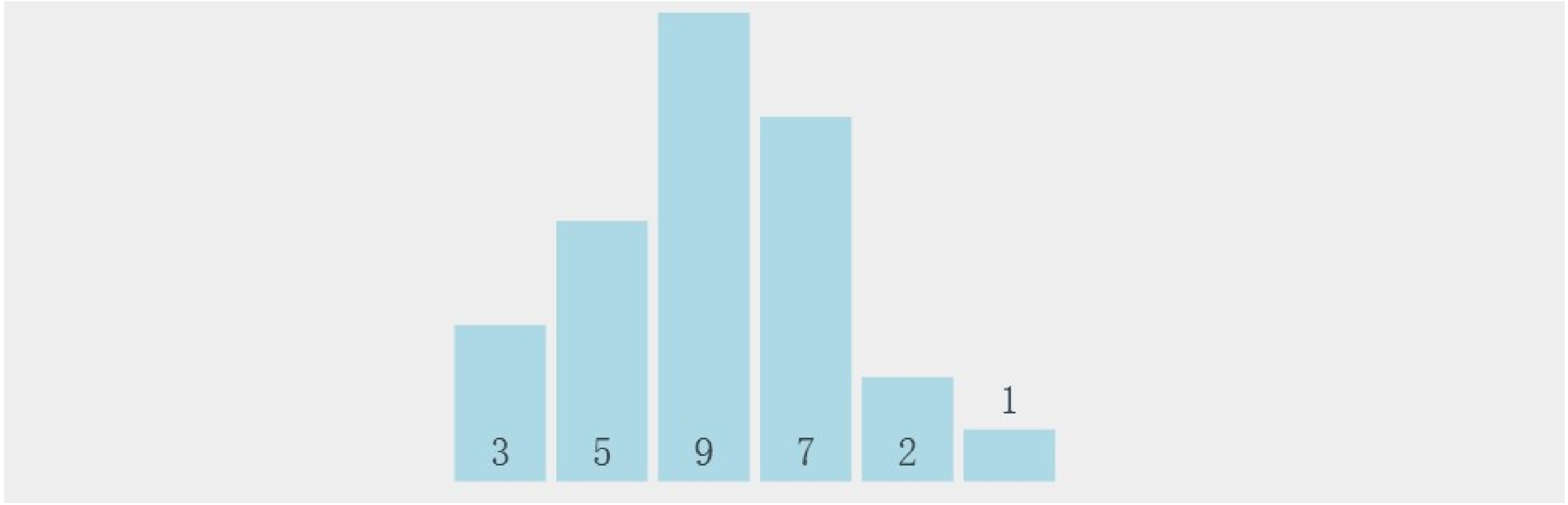
```
int a[5];
a[0] = 6;
a[1] = 78;
a[2] = 45;
a[3] = 55;
a[4] = 23;
for(int i = 0;i < 5;i++)
{
    printf("%d\n", a[i]);
}
```

# 一维数组的声明与初始化

数据类型 数组名 [ 数组长度 ] = { 元素 1, 元素 2, ..., 元素 v}
数据类型 数组名 [] = { 元素 1, 元素 2, ..., 元素 v}

```
int a[5] = {1, 2, 3, 4, 5};
int b[5] = {1, 2};
int c[] = {1, 2, 3, 4, 5, 6};
```

# 冒泡排序



# 二维数组的声明与初始化

数据类型 数组名 [ 数组长度 ][ 数组长度 ];

```
int a[5][6];
char ch[3][4];
float f[3][5];
```

# 二维数组的声明与初始化

数据类型 数组名 [ 数组长度 ][ 数组长度 ] = { 元素 1, 元素 2, ..., 元素 v}

数据类型 数组名 [ 数组长度 ][ 数组长度 ] = {{...}, {...}, {...}...}

数据类型 数组名 [][ 数组长度 ] = { 元素 1, 元素 2, ..., 元素 v}

```
int a[2][3] = {1, 2, 3, 4, 5, 6};
int b[2][3] = {{1, 2, 3}, {4, 5, 6}};
int c[][3] = {1, 2, 3, 4, 5, 6};
```

# 字符数组与字符串

X 语言中没有字符串数据类型，使用字符数组表示，字符串末尾加 ‘ ∴0′

```
char ch[] = {‘a’,’b’,’c’,’\0’};
char ch[10] = {‘a’,’b’,’c’,’\0’};
char ch[] = “HelloWorld”;
```

# 字符串输入输出 γετσ 与 πυτσ

γετσ(στρ)：接收字符串存储到变量 στρ 中。

πυτσ(στρ)：输出 στρ 。