



第8章 指针

——字符数组与字符指针： 字符串的表示与存储

哈尔滨工业大学

苏小红

sxh@hit.edu.cn

字符串常量

```
printf("How are you");
```

用双引号括起的一串字符是**字符串常量**，系统自动为其添加**空字符 '\0'**

H	o	w		a	r	e		y	o	u	\0
---	---	---	--	---	---	---	--	---	---	---	----

用**空字符 '\0'**（ASCII码值为0）标志字符串的末尾——**字符串结束标志**

字符串常量

```
printf("How are you.\n");
```

```
How are you.
```

```
—
```

```
printf("\\"How are you.\\\"\n");
```

```
"How are you."
```

```
—
```

- 问题：如果字符串**太长**，怎么表示？

```
printf("How are you. Press \\  
a key and then press Enter:\n");
```

```
How are you. Press a key and then press Enter:
```

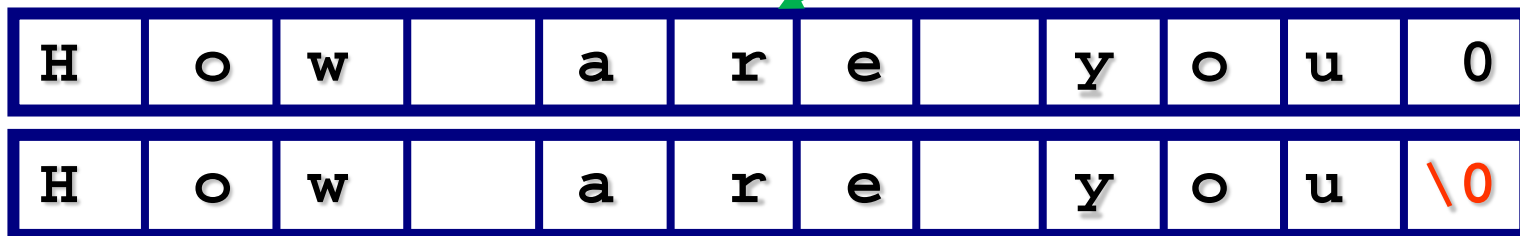
```
—
```

```
printf("How are you. Press "  
      "a key and then press Enter:\n");
```

字符串变量？

- C语言没有提供专门的字符串数据类型
- 字符数组——每个元素都是字符类型的数组

是字符数组，但不一定代表字符串



H	o	w		a	r	e		y	o	u	0
H	o	w		a	r	e		y	o	u	\0

数组的最后一个元素必须是 `'\0'` 才表示字符串

不要混淆 `'\0'` (ASCII码为0) 与 `'0'` (ASCII码为48)

字符数组的定义和初始化

■ 字符数组的定义

- `#define STR_LEN 80`
- `char str[STR_LEN+1];`

■ 字符数组的初始化

* 用字符常量的初始化列表对数组初始化

* `char str[6] = {'C', 'h', 'i', 'n', 'a', '\0'};`

* 用字符串常量直接对数组初始化

* `char str[6] = {"China"};`

* `char str[6] = "China";`

* `char str[] = "China";`

* `char str[10] = "China";`

* `char str[4]  "China";`

字符指针的定义和初始化

- 字符指针就是指向字符串首地址的指针
- 定义一个字符指针，使其指向一个字符串常量

```
char *pStr = "Hello China";
```



pStr

A brown arrow points from the label 'pStr' to the first cell of the string array, which contains the character 'H'.

```
char str[] = "Hello China";
```


可修改

不可修改

为什么?

将字符指针指向一个字符串



 **pStr** `char *pStr = "Hello China";`

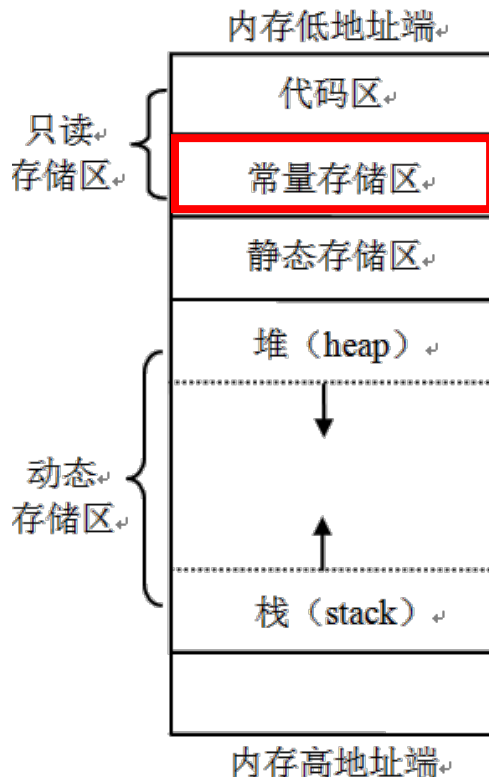
`char *pStr;`

`pStr = "Hello China";`

pStr是一个指向常量存储区中的字符串的指针**变量**

可修改**pStr**的值（指向），但不可以对它所指向的存储单元进行写操作

***pStr**  `'W';`



用字符数组保存一个字符串


str

H	e	l	l	o		C	h	i	n	a	\0
---	---	---	---	---	--	---	---	---	---	---	----

char str[] = "Hello China";

str  "Hello China";

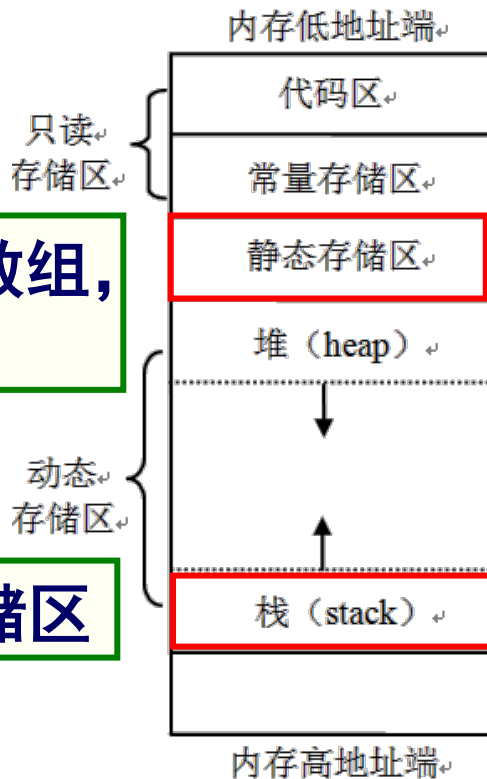
数组名**str**的值不可修改，
是地址常量

str[0]  **'w' ;**

数组中存储的字符可被修改

函数外定义，或定义为静态数组，
字符串保存在静态存储区

函数内定义，保存在动态存储区



将字符指针指向一个字符数组

str

H	e	l	l	o		C	h	i	n	a	\0
---	---	---	---	---	--	---	---	---	---	---	----

↑
pStr `char str[] = "Hello China"`

`char *pStr;`

`pStr = str;`

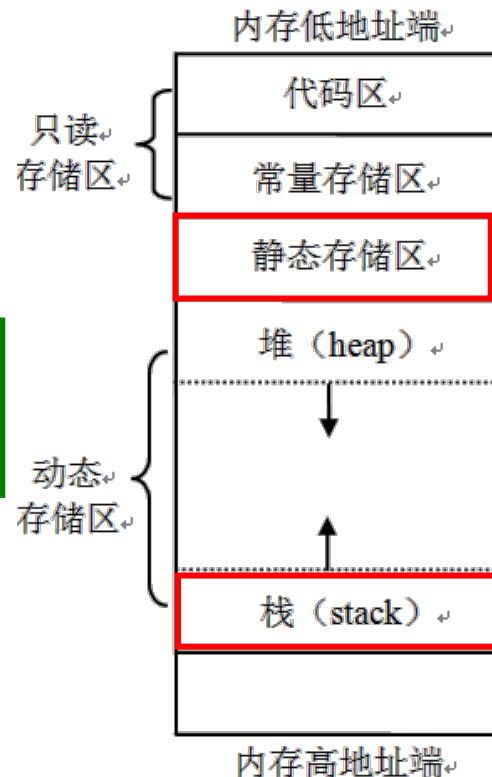
`pStr = &str[0];`

pStr的值（指向）可被修改，
它所指向的字符串也可被修改

`*pStr = 'W';`

`str[0] = 'W';`

`pStr[0] = 'W';`



使用字符指针的基本原则

■ 正确使用字符指针须牢记以下基本原则：

- * 明确字符串被保存到了哪里
- * 明确字符指针指向了哪里

```
char *pStr;
```

```
pStr[0]  'a';
```

```
*pStr  'a';
```

```
scanf("%s", pStr); 
```



使用指针的基本原则

■ 指针使用原则

- * 永远清楚指针指向了哪块内存
- * 永远清楚指针指向的内存中的内容是什么
- * 一个x型的指针指向x型变量的地址

■ 总纲

- * 永远清楚你正在操作哪块内存
- * 永远清楚这种操作是否合理、合法



