



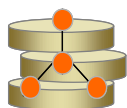
第七单元（3）：字符串指针

教学内容	字符串的指针和指向字符串的指针变量
教学目标	
应知	■ 字符串可以用字符数组和字符指针表示
应会	■ 用字符指针表示字符串，用字符串指针作函数参数实现关于字符串的常用操作
难点	■ 明确字符串指针的当前指向

教学方法

- 项目递进驱动教学法，通过数组的几种表示方法引入数组的指针

教学过程



1. 提问与复习

- 可以有几种方式引用数组元素？分别是什么？
- 采用指向数组的指针时，指针 $p++$ ($++p$) 的操作代表什么？
- 作函数参数时，数组名与指向该数组的指针变量可否替换？

2. 字符串的字符数组表示、引用法（引导学生编程实现，复习字符串结束标志）

- 定义字符串 `char str[]="Polytechnic"` 将一个字符串分别逐个输出和整体输出。

3. 字符串的指针表示、引用法

- 不定义字符数组，直接指针 `char *p = "polytechnic"` 对字符串进行整体操作。（也可以 `char *p; p = "polytechnic";`）

4. 字符串指针的编程练习

- 将字符串 `str1` 拷贝到字符串 `str2`.
- 注意：（1）定义时目的字符串长度应大于源字符串长度
- （2）使用字符串结束标志判断源字符串的结束
- （3）拷贝结束要给目的字符串添加字符串结束标志
- （4）输出目的字符串，查看是否正确拷贝



5. 将第 4 步字符串拷贝的功能放在函数里实现，以字符指针和数组名作函数参数

- (1) 形参为数组名（利用下标变量引用字符数组元素），实参为数组名或指针

```
while(s1[i] != '\0') {s2[i] = s1[i]; i++;}
```

- (2) 形参为指针（利用下标引用指针变量所指数组元素），实参为数组名或指针

```
while(ps1[i] != '\0') {ps2[i] = ps1[i]; i++;}
```

- (3) 形参为指针（利用指针变化引用数组元素），实参为数组名或指针

```
while(*ps1 != '\0') { *ps2=*ps1; ps2++; ps1++;}
```

6. 编程练习：编写一个合并字符串的函数



学生容易出错的地方

- 用字符串结束标志时误用 “\0”
- 字符指针移动后没有重新指向字符串首地址

问题与讨论

- 字符串指针的初始化与字符数组的初始赋值有何区别？

小结（可由问题与讨论方式给出）



- 字符数组和字符指针变量都能实现字符串的存储和运算
- 字符数组与字符串区别：
 - (1) 字符数组的每个数组元素存放一个字符，字符指针变量存放字符串的首地址
 - (2) 对字符数组不能有以下操作：

```
char str[20]; str = "hello";
```

但是字符指针变量可以：

```
char *pstr; pstr = "hello";
```



课后任务

- 实训项目单 3 后 2 项，整理笔记