C 语言字符串指针（指向字符串的指针）

C 语言中没有特定的字符串类型，我们通常是将字符串放在一个字符数组中，这里不妨再来演示一下：

```
#include <stdio.h>
int main(){
    char str[] = "http://c.biancheng.net";
    int len = strlen(str), i;
    //直接输出字符串
    printf("%s\n", str);
    //每次输出一个字符
    for(i=0; i<len; i++){
        printf("%c", str[i]);
    }
    printf("\n");
    return 0;
}
```

运行结果：

http://c.biancheng.net

http://c.biancheng.net

字符数组归根结底还是一个数组，上节讲到的关于指针和数组的规则同样也适用于字符数组。更改上面的代码，使用指针的方式来输出字符串：

```
#include <stdio.h>
int main(){
    char str[] = "http://c.biancheng.net";
    char *pstr = str;
    int len = strlen(str), i;

    //使用*(pstr+i)
    for(i=0; i<len; i++){
        printf("%c", *(pstr+i));
    }
    printf("\n");
```

```
    //使用 pstr[i]
    for(i=0; i<len; i++){
        printf("%c", pstr[i]);
    }
    printf("\n");
    //使用*(str+i)
    for(i=0; i<len; i++){
        printf("%c", *(str+i));
    }
    printf("\n");

    return 0;
}
```

运行结果：

http://c.biancheng.net

http://c.biancheng.net

http://c.biancheng.net

除了字符数组，C 语言还支持另外一种表示字符串的方法，就是直接使用一个指针指向字符串，例如：

```
char *str = "http://c.biancheng.net";
```

或者：

```
char *str;
str = "http://c.biancheng.net";
```

字符串中的所有字符在内存中是连续排列的，str 指向的是字符串的第 0 个字符；我们通常将第 0 个字符的地址称为字符串的首地址。字符串中每个字符的类型都是 `char`，所以 str 的类型也必须是 `char *`。

下面的例子演示了如何输出这种字符串：

```
#include <stdio.h>
int main(){
    char *str = "http://c.biancheng.net";
    int len = strlen(str), i;

    //直接输出字符串
    printf("%s\n", str);
    //使用*(str+i)
    for(i=0; i<len; i++){
        printf("%c", *(str+i));
    }
    printf("\n");
    //使用 str[i]
    for(i=0; i<len; i++){
        printf("%c", str[i]);
    }
    printf("\n");

    return 0;
}
```

运行结果：

http://c.biancheng.net

http://c.biancheng.net

http://c.biancheng.net

这一切看起来和字符数组是多么地相似，它们都可以使用%s输出整个字符串，都可以使用*或[]获取单个字符，这两种表示字符串的方式是不是就没有区别了呢？

有！它们最根本的区别是在内存中的存储区域不一样，字符数组存储在全局数据区或栈区，第二种形式的字符串存储在常量区。全局数据区和栈区的字符串（也包括其他数据）有读取和写入的权限，而常量区的字符串（也包括其他数据）只有读取权限，没有写入权限。

内存权限的不同导致的一个明显结果就是，字符数组在定义后可以读取和修改每个字符，而对于第二种形式的字符串，一旦被定义后就只能读取不能修改，任何对它的赋值都是错误的。

我们将第二种形式的字符串称为字符串常量，意思很明显，常量只能读取不能写入。请看下面的演示：

```
1. #include <stdio.h>
2. int main(){
3.     char *str = "Hello World!";
4.     str = "I love C!";  //正确
5.     str[3] = 'P';  //错误
6. 
7.     return 0;
8. }
```

这段代码能够正常编译和链接，但在运行时会出现段错误（Segment Fault）或者写入位置错误。

第 4 行代码是正确的，可以更改指针变量本身的指向；第 3 行代码是错误的，不能修改字符串中的字符。

到底使用字符数组还是字符串常量

在编程过程中如果只涉及到对字符串的读取，那么字符数组和字符串常量都能够满足要求；如果有写入（修改）操作，那么只能使用字符数组，不能使用字符串常量。

获取用户输入的字符串就是一个典型的写入操作，只能使用字符数组，不能使用字符串常量，请看下面的代码：

```
1. #include <stdio.h>
2. int main(){
3.     char str[30];
```

```
    gets(str);
    printf("%s\n", str);

    return 0;
}
```

运行结果：

C C++ Java Python JavaScript

C C++ Java Python JavaScript

最后我们来总结一下，C 语言有两种表示字符串的方法，一种是字符数组，另一种是字符串常量，它们在内存中的存储位置不同，使得字符数组可以读取和修改，而字符串常量只能读取不能修改。