

C 语言数组灵活多变的访问形式

C 语言中的指针使得代码的编写非常灵活，如果指针能够和数组结合，那将会有更多的“花招”，请看下面的代码：

```
1. #include <stdio.h>
2.
3. int main() {
4.     char str[20] = "c.biancheng.net";
5.
6.     char *s1 = str;
7.     char *s2 = str+2;
8.
9.     char c1 = str[4];
10.    char c2 = *str;
11.    char c3 = *(str+4);
12.    char c4 = *str+2;
13.    char c5 = (str+1)[5];
14.
15.    int num1 = *str+2;
16.    long num2 = (long)str;
17.    long num3 = (long)(str+2);
18.
19.    printf(" s1 = %s\n", s1);
20.    printf(" s2 = %s\n", s2);
21.
22.    printf(" c1 = %c\n", c1);
23.    printf(" c2 = %c\n", c2);
24.    printf(" c3 = %c\n", c3);
25.    printf(" c4 = %c\n", c4);
26.    printf(" c5 = %c\n", c5);
27.
28.    printf("num1 = %d\n", num1);
29.    printf("num2 = %ld\n", num2);
30.    printf("num3 = %ld\n", num3);
31.
32.    return 0;
33. }
```

运行结果：

```
s1 = c.biancheng.net
s2 = biancheng.net
c1 = a
c2 = c
c3 = a
c4 = e
c5 = c
num1 = 101
num2 = 2686736
num3 = 2686738
```

怎么样，够晕吧，如果你嗤之以鼻，那么恭喜你，你的 C 语言基础很扎实。

1) str 既是数组名称，也是一个指向字符串的指针；指针可以参加运算，加 1 相当于数组下标加 1。

printf() 输出字符串时，要求给出一个起始地址，并从这个地址开始输出，直到遇见字符串结束标志 `\0`。s1 为字符串 str 第 0 个字符的地址，s2 为第 2 个字符的地址，所以 printf() 的结果分别为 c.biancheng.net 和 biancheng.net。

2) 指针可以参加运算，str+4 表示第 4 个字符的地址，c3 = *(str+4) 表示第 4 个字符，即 'a'。

3) 其实，数组元素的访问形式可以看做 address[offset]，address 为起始地址，offset 为偏移量：`c1 = str[4]` 表示以地址 str 为起点，向后偏移 4 个字符，为 'a'；`c5 = (str+1)[5]` 表示以地址 str+1 为起点，向后偏移 5 个字符，等价于 str[6]，为 'c'。

4) 字符与整数运算时，先转换为整数（字符对应的 ASCII 码）。num1 与 c4 右边的

达式相同，对于 num1，*str+2 == 'c'+2 == 99+2 == 101，即 num1 的值为 101，

对于 c4，101 对应的字符为 'e'，所以 c4 的输出值为 'e'。

5) num2 和 num3 分别为字符串 str 的首地址和第 2 个元素的地址。

为了加深大家的理解，请继续阅读下面的代码：

```
1. #include <stdio.h>
2. #include <stdlib.h>
3.
4. int main() {
5.     char str[20] = {0};
6.     int i;
7.
8.     for(i=0; i<10; i++) {
9.         *(str+i) = 97+i; // 97 为字符 a 的 ASCII 码值
10.    }
11.
12.    printf("%s\n", str);
13.    printf("%s\n", str+2);
14.    printf("%c\n", str[2]);
15.    printf("%c\n", (str+2)[2]);
16.
17.    return 0;
18. }
```

运行结果：

```
abcdefghij
cdefghij
c
e
```

第 5 行代码用来将字符数组中的所有元素都初始化为\0，这样在循环结束时就无需添加字符串结束标志。

前面三个 printf() 比较容易理解，第四个 printf() 可以参照上面的说明 3)，str+2 表

示指向第 2 个元素， $(str+2)[2]$ 相当于 $*(str+2+2)$ ，也就是取得第 4 个元素的值。