[TOC]

1

- const 左结合 左边为空 再右结合
- 所以 const char * s = char const * s
- char * const

2

2.1

- 指向const对象的指针
- 修改的是指针的指向

```
const char * p1 = "aaa";
```

```
p1 = "bbb";
```

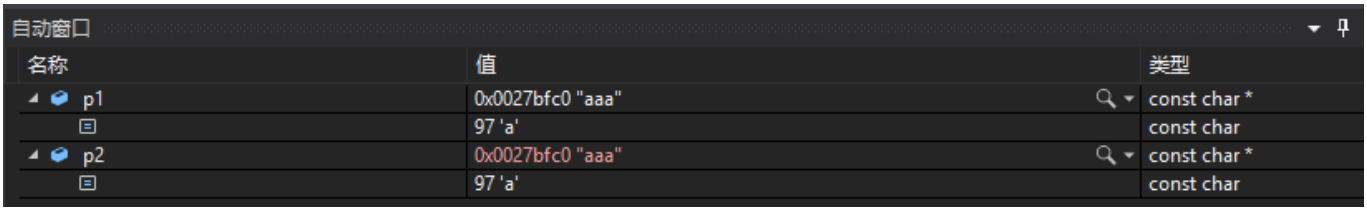
```
cout << p1 << endl;
```

- p1 = bbb

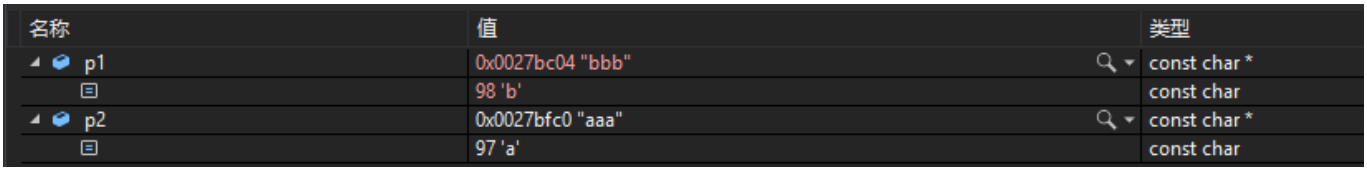
2.2

```
const char * p1 = "aaa";
```

```
const char * p2 = p1;
```




```
p1 = "bbb";
```




```
cout << p1 << endl;
```

```
cout << p2 << endl;
```

- p1 = bbb p2 = aaa

2.3

- 上面的字符串是常量，但是下面的不是常量
- t[]本身不是常量字符串，但是从s的角度来看，s认为自己指向了一个常量字符串，实际上t可以修改

```
char t[] = "sss";
```

```
const char * s = t;
```

```
s[0] = 'q';//表达式必须是可修改的左值
```

```
t[0] = 'q';
```

- s = qss

2.4

```
char const * p2 = "aaa";
```

```
p2 = "bbb";
```

```
cout<< p2 << endl;
```

- p2 = bbb

3

- 指针是const类型的 指针指向的对象不能修改，但是可以修改指向对象的内容

3.1

```
char * const s = "aaaaaaaa";
```

- "aaaaaaaaaaa":const char * 类型的

3.2

```
char t[] = "aaaaaaaa";
```

```
char * const p2 = t;
```

```
p2[0] = 's';
```

```
cout << p2 << endl;
```

- p2 = saaaaaaa

3.3

```
char t[] = "aaaaaaaa";
```

```
char * const p2 = t;
```

```
char * const p3 = p2;
```

名称	值	类型
p2	0x008ffdf0 "aaaaaaaa"	char * const
p3	0x008ffdf0 "aaaaaaaa"	char * const
t	0x008ffdf0 "aaaaaaaa"	char[9]

```
p2[0] = 's';
```

自动窗口

名称	值	类型
p2	0x008ffdf0 "saaaaaaa"	char * const
p2[0]	115 's'	char
p3	0x008ffdf0 "saaaaaaa"	char * const

```
cout << p2 << endl;
```

```
cout << p3 << endl;
```