

下标和切片

一、下标/索引

所谓“下标”又叫“索引”，就是编号，就好比超市中的存储柜的编号，通过这个编号就能找到相应的存储空间

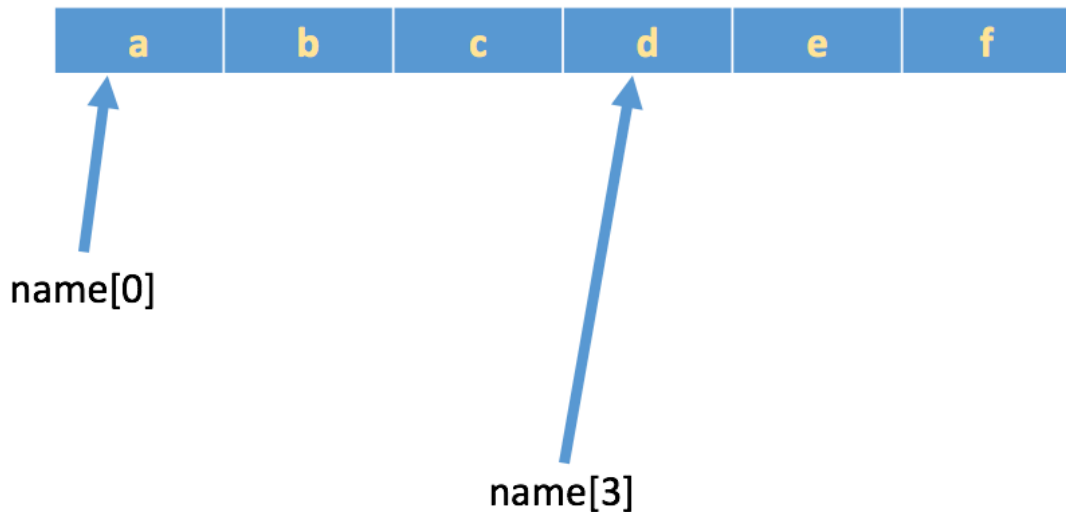
- 生活中的 "下标"

超市储物柜



- 字符串中"下标"的使用

如果有字符串: `name = 'abcdef'`，在内存中的实际存储如下:



如果想取出部分字符，那么可以通过 **下标** 的方法，（注意在计算机中，下标从 0 开始）

```
name = 'abcdef'

print(name[0])
print(name[1])
print(name[2])
```

运行结果:

```
a
b
c
```

2. 遍历

所谓遍历，可以理解为按照一定的规则(一般情况下就是数据的下标)，访问每一个数据。不是所有的数据都能够被遍历，字符串是可迭代对象，能够进行遍历。

可以使用while和for语句，对字符串里的元素进行遍历。

while语句遍历:

```
msg = 'hello world'
i = 0
while i < len(msg):
    print(msg[i])
    i += 1
```

for语句遍历:

```
msg = 'hello world'
for x in msg:
    print(x)
```

3. 切片

切片是指对操作的对象截取其中一部分的操作。字符串、列表、元组都支持切片操作。

切片的语法: [起始:结束:步长], 也可以简化使用 [起始:结束]

注意: 选取的区间从"起始"位开始, 到"结束"位的前一位结束 (不包含结束位本身), 步长表示选取间隔。

```
# 索引是通过下标取某一个元素
# 切片是通过下标去某一段元素

s = 'Hello World!'
print(s)

print(s[4]) # o 字符串里的第4个元素

print(s[3:7]) # lo w 包含下标 3, 不含下标 7

print(s[:]) # Hello World! 取出所有元素 (没有起始位和结束位之分), 默认步长为1

print(s[1:]) # ello World! 从下标为1开始, 取出 后面所有的元素 (没有结束位)

print(s[:4]) # Hell 从起始位置开始, 取到 下标为4的前一个元素 (不包括结束位本身)

print(s[:-1]) # Hello World 从起始位置开始, 取到 倒数第一个元素 (不包括结束位本身)

print(s[-4:-1]) # rld 从倒数第4个元素开始, 取到 倒数第1个元素 (不包括结束位本身)

print(s[1:5:2]) # el 从下标为1开始, 取到下标为5的前一个元素, 步长为2 (不包括结束位本身)

print(s[7:2:-1]) # ow ol 从下标为7的元素开始 (包含下标为7的元素), 倒着取到下标为2的元素 (不包括下标为2的元素)

# python 字符串快速逆置
print(s[::-1]) # !dlrow olleH 从后向前, 按步长为1进行取值
```