任务4-1字符串概述及简单操作

什么是字符串

用户登录

系统验证用户登录信息时，
登录信息是如何验证的？

什么是字符串

字符串是一种表示文本数据的类型。

使用单引号	使用双引号	使用三引号
‘a’、'123'	“a”、"123"	""" Hello """

转义字符

看下面的代码：

```
>>>'let's go! go'
  File "<input>", line 1
    'let's go! go'
        ^
SyntaxError: invalid syntax
```

对于单引号或者双引号这些特殊的符号，我们可以对他们进行转义。例如，对字符串中的单引号进行转义：

```
>>>'let\'s go! go'
"let's go! go"
```

转义字符

转义字符	代表含义
\(在行尾时)	反斜杠符号
\\	反斜杠符号
\"	双引号
\n	换行
\b	退格
\t	横向制表符

字符串的输出

比如有以下代码:

```
print(“我今年10岁”)
print(“我今年11岁”)
print ("我今年12岁")
    ...
```

大家试想一下，上述代码多次输出"我今年xx岁"，是否有一种简化程序的方式呢？

当然有。可以通过字符串格式化来完成。

字符串的输出

下面是字符串的格式化输出

```
name =“小明”
print("大家好，我叫%s"%name)
```

字符串的输出

常见的格式化符号

格式化符号	转换
%s	通过str()字符串转换来格式化
%d	有符号十进制整数
%f	浮点实数

字符串的输出

Python2.6 开始，新增了一种格式化字符串的函数 **str.format()**，它增强了字符串格式化的功能。
基本语法是通过 **{}** 和 **:** 来代替以前的 **%** 。
format 函数可以接受不限个参数，位置可以不按顺序。

```
print("{} {}".format("hello", "world"))    # 不设置指定位置，按默认顺序
print("{0} {1}".format("hello", "world"))  # 设置指定位置
print("{:.2f}".format(3.1415926)); #保留小数点后两位
```

```
hello world
hello world
3.14
```

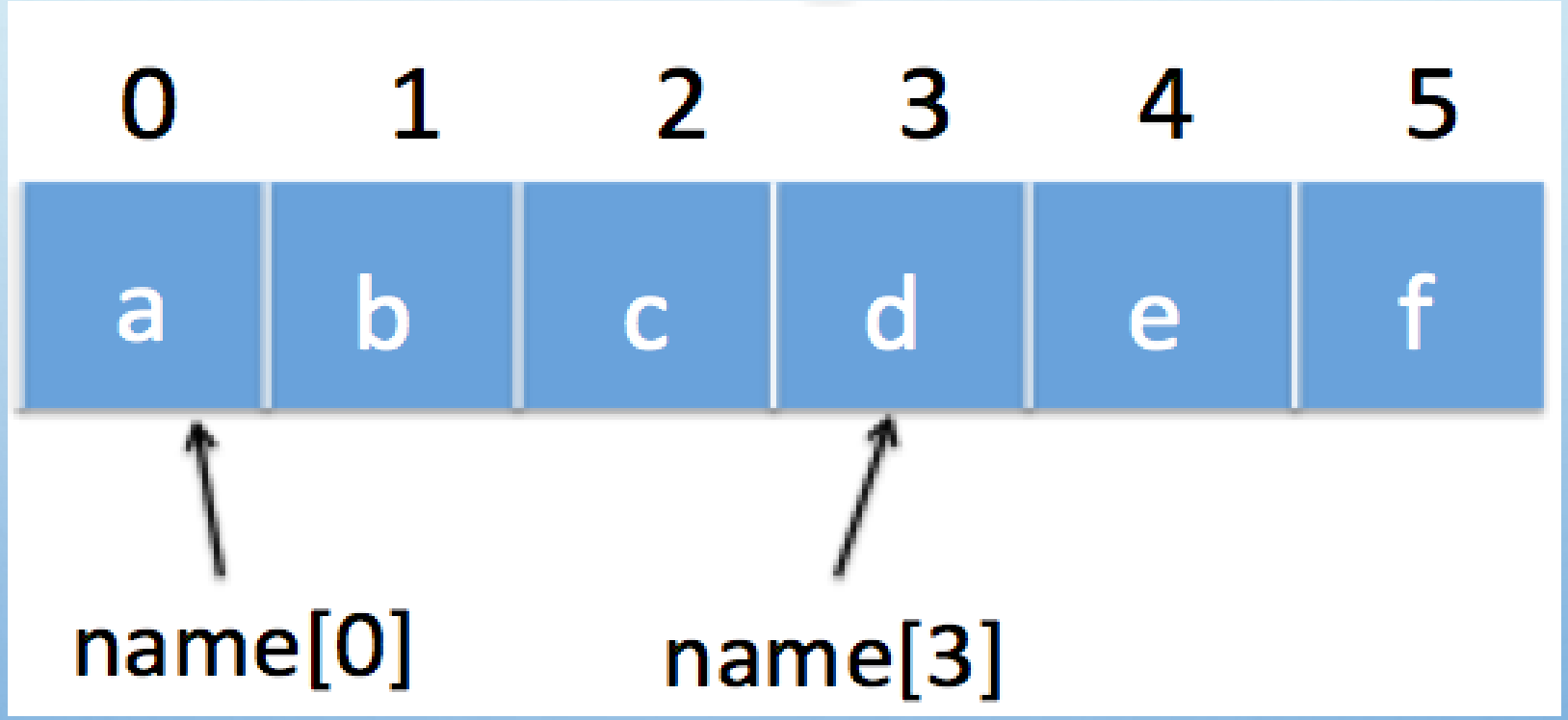
字符串的输入

Python3提供了 input() 函数从标准输入读取一行文本，默认的标准输入是键盘。input可以接收一个Python表达式作为输入，并将运算结果返回。

```
username=input("请输入用户名")
print(username)
```

字符串的存储方式

字符串中的每个字符都对应一个下标，下标编号是从0开始的。



什么是切片

切片的语法格式如下所示：

```
[起始:结束:步长]
```

切片选取的区间属于左闭右开型，即从"起始"位开始，到"结束"位的前一位结束（不包含结束位本身）

使用切片截取字符串

假设有字符串 name= “abcdef”，则:

name[0:3] → abc

name[3:5] → de

name[1:-1] → bcde

name[2:]) → cdef

name[::-2] → fdb

负索引表示从字符串右端索引

Python字符串操作

操作符	含义
+	连接
*	重复
<string>[ ]	索引
<string>[:]	切片
Len(<string>)	长度
For<var>in<string>	迭代遍历字符串