

# Python语言基础与应用

数据类型 / 基本类型：字符串

陈斌 北京大学 [gischen@pku.edu.cn](mailto:gischen@pku.edu.cn)



# 基本类型：字符串

- › 文本的表示
- › 字符串和名字的区别
- › 常见的字符串操作
- › 字符串是一种序列

# 文本的表示

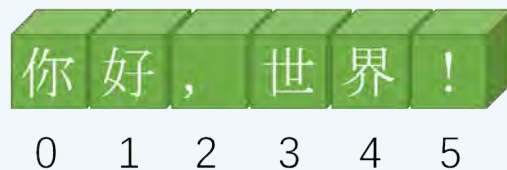
## › 字符串就是把一个个文字的字符“串起来”的数据

- 文字字符包含有拉丁字母、数字、标点符号、特殊符号，以及各种语言文字字符

"Hello World!"



“你好，世界！”





# 文本的表示

## › 表示字符串数值

- 用双引号或者单引号都可以表示字符串，但必须成对
- 多行字符串用三个连续单引号表示

```
>>> "abc"  
'abc'  
>>> 'abc'  
'abc'  
>>> '''abc def  
ghi jk'''  
'abc def\nghi jk'
```

# 文本的表示

## › 特殊字符用转义符号 “\” 表示

转义字符	描述
(在行尾时)	续行符
\\	反斜杠符号
\'	单引号
\"	双引号
\a	响铃
\b	退格(Backspace)
\e	转义
\000	空
\n	换行
\v	纵向制表符
\t	横向制表符
\r	回车
\f	换页
\oyy	八进制数 yy 代表的字符，例如：\o12 代表换行，数值参见 ASCII 码表
\xyy	十六进制数 yy 代表的字符，例如：\x0a 代表换行，数值参见 ASCII 码表
\other	其它的字符以普通格式输出

# 文本的表示

## › 字符的编号

- 第一个字符的编号是0，第二个字符编号是1，...
- 最后一个字符的编号是-1，倒数第二个字符编号是-2，...

## › 用这种整数编号可以从字符串中抽出任何一个字符



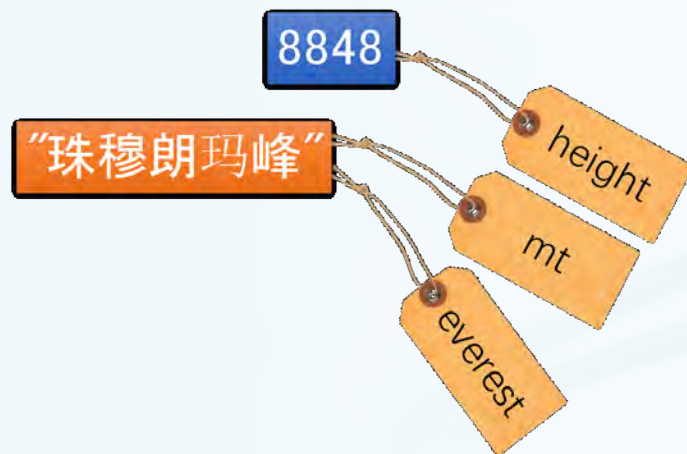
# 字符串与名字的区别

- › 字符串是数据本身
- › 名字是数据的标签
- › 名字和字符串是“名”和“值”之间的关系
  - 一个字符串数值可以关联多个名字  
一个名字在同一时刻只能关联一个字符串数值
  - 字符串数值只能是字符串类型  
名字则可以关联任意类型的数值。

# 字符串与名字的区别

## › 看看python怎么说

```
>>> height = 8848
>>> mt = everest = “珠穆朗玛峰”
>>> height
8848
>>> mt
'珠穆朗玛峰'
>>> everest
'珠穆朗玛峰'
>>> type(height)
<class 'int'>
>>> type(everest)
<class 'str'>
```





# 常见的字符串操作

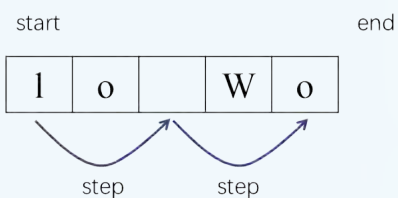
## › 获取字符串的长度

len函数

## › 切片(slice)操作

s[start:end:step]

```
s="Hello World!"  
s[3:8:2]
```



"l o"

# 常见的字符串操作

## › “加法” 和 “乘法”

**+**: 将两个字符串进行连接, 得到新的字符串。

**\***: 将字符串重复若干次, 生成新的字符串

## › 判断字符串内容是否相同(==)

```
>>> a = 'Hello'  
>>> a == 'Hello'  
True
```

## › 判断字符串中是否包含某个字符串(in)

```
>>> 'h' in a  
False  
>>> 'H' in a  
True
```

# 常见的字符串操作

## › 删除空格

- `str.strip`: 去掉字符串前后的所有空格，内部的空格不受影响
- `str.lstrip`: 去掉字符串前部（左部）的所有空格
- `str.rstrip`: 去掉字符串后部（右部）的所有空格

## › 判断字母数字

- `str.isalpha`: 判断字符串是否全部由字母构成
- `str.isdigit`: 判断字符串是否全部由数字构成
- `str.isalnum`: 判断字符串是否仅包含字母和数字，而不含特殊字符

# 字符串的高级操作

split: 分割; join: 合并

upper/lower/swapcase: 大小写相关

ljust/center/rjust: 排版左中右对齐

replace: 替换子串

```
>>> 'You are my sunshine.'.split(' ')
['You', 'are', 'my', 'sunshine.']
>>> '-'.join(["One", "for", "Two"])
'One-for-Two'
>>> 'abc'.upper()
'ABC'
>>> 'aBC'.lower()
'abc'
>>> 'Abc'.swapcase()
'aBC'
>>> 'Hello World!'.center(20)
'   Hello World!   '
>>> 'Tom smiled, Tom cried, Tom shouted'.replace('Tom', 'Jane')
'Jane smiled, Jane cried, Jane shouted'
```



# 字符串是一种序列

## › 序列 (sequence)

- 能够按照整数顺序排列的数据

## › 序列的内部结构:

- 可以通过从0开始的连续整数来索引单个对象;
- 可以执行切片, 获取序列的一部分;
- 可以通过len函数来获取序列中包含多少元素;
- 可以用加法“+”来连接为更长的序列;
- 可以用乘法“\*”来重复多次, 成为更长的序列;
- 可以用“in”来判断某个元素是否在序列中存在。