



# 内置数据类型2

北京理工大学计算机学院 高玉金

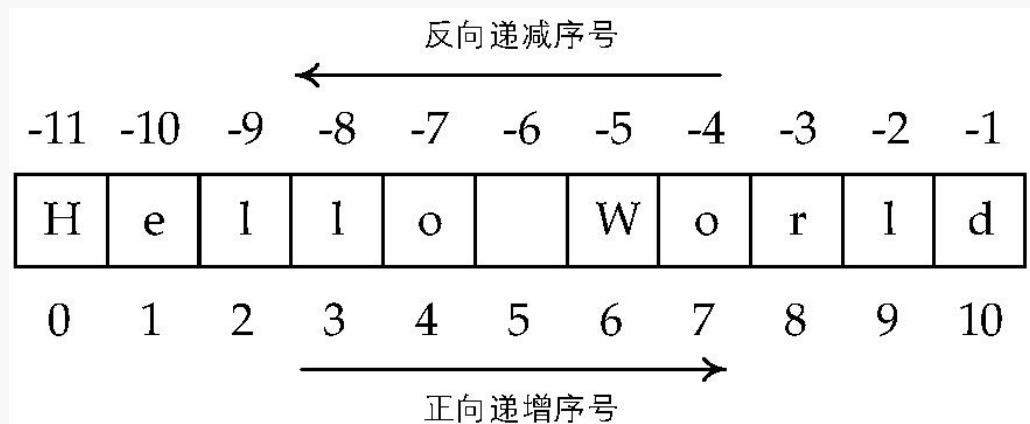
2019年3月



# Python内置类型对象str

- 字符串基本操作

- 连接:  $x+y$
- 复制:  $x*n$ 或 $n*x$
- 子串判断:  $c \text{ in } s$
- 索引:  $s[i]$
- 切片:  $s[i:j:m]$



```
>>> s="HelloWorld"
>>> s[2]
'l'
>>> s[-2]
'l'
>>> s[-1]
'd'
```

```
>>> s[::-2]
'drWle'
```

- 索引: 通过位置, 正向索引和反相索引, 如 $a[2]$ 和  $a[-2]$
- 切片slice:  $\langle \text{string} \rangle[\langle \text{start} \rangle:\langle \text{end} \rangle:\text{步长}]$   
(正负整数) 左闭右开



# 字符串对象的方法

- 常用方法
  - 切割split (sep=None)
  - 联合join(iterable)
  - 截取strip()
  - 计数count(sub)
  - 替换replace(old,new)
- 字符串中的元素是只读的
- 只能对变量重新赋值而不能修改其中的某些部分
- 字符串对象方法返回一个新的对象

```
1 line = "1, 2, 3, 4, 5"  
2 numbers = line.split(',')  
3 print(numbers)
```

```
['1', '2', '3', '4', '5']
```

```
In [37]: s="Helloworld"
```

```
In [38]: s[5]="W"
```

```
Traceback (most recent call last):
```

```
In [39]: s.replace("w", "W")
```

```
Out[39]: 'HelloWorld'
```

```
In [40]: s
```

```
Out[40]: 'Helloworld'
```





# 字符串的格式化

- Slot槽位: {<槽位标识>: <格式控制标记>}
- 槽位的位置序号或变量名 (包括词典、列表等)
- ‘公司包括: {0}, {1} 和 {other} ’ .format('google' , 'ali' , other='taobao')

"{ } : 计算机{ }的CPU占用率为{ }%。".format("2016-12-31","PYTHON",10)

↑  
0

↑  
1

↑  
2

↑  
0

↑  
1

↑  
2

字符串中槽{ }的顺序

format() 中参数的顺序

"{1}: 计算机{0}的CPU占用率为{2}%。".format("2016-12-31","PYTHON",10)



# 格式控制标记

- 格式控制标记用来控制参数显示时的格式
- 格式控制标记包括：<填充><对齐><宽度>,<.精度><类型>6个字段，这些字段都是可选的，可以组合使用

```
print('{:*^20,.2f}'.format(12345.14159))  
*****12,345.14*****
```

:	<填充>	<对齐>	<宽度>	,	<.精度>	<类型>
引导 符号	用于填充的 单个字符	< 左对齐 > 右对齐 ^ 居中对齐	槽的设定输 出宽度	数字的千位 分隔符 适用于整数 和浮点数	浮点数小数 部分的精度 或 字符串的最 大输出长度	整数类型 b, c, d, o, x, X 浮点数类型 e, E, f, %





# 字符串格式化示例

```
a = "i am {},age {}".format("seven",18,"alex")
b = "i am {},age {}, {}".format(*["seven", 18 , "alex"])
c = "i am {0}, age {1}, really {0}".format("seven", 18)
d = "i am {0}, age{1}, really {0}".format(*["seven", 18])
e = "i am {name}, age {age}, really {name}".format(name="seven", age = 18)
f = "i am {name}, age {age}, rally {name}".format(**{"name":"seven", "age":18})
g = "i am {0[0]},age{0[1]}, really{0[2]}".format([1,2,3],[11,22,33])
h = "i am {:s}, age {:d}, money {:f}".format("seven", 18, 888.1)
i = "i am {:s}, age {:d}".format(*["seven", 18])
j = "i am {name:s}, age {age:d}".format(name="seven",age=18)
k = "i am {name:s}, age {age:d}".format(**{"name":"seven","age":18})
l = "numers:{:b},{:o},{d},{:x},{:X},{:%}".format(15,15,15,15,15,15.32445,2)
m = "numbers:{0:b},{0:o},{0:d},{0:x},{0:%}".format(15)
tpl = "numbers: {num:b},{num:o},{num:d},{num:x},{num:X}, {num:%}".format(num=15)
```

更多格式化操作: <https://docs.python.org/3/library/string.html>



# 字符串处理示例：个人隐私保护

- 场景需求：对手机号的中间四位进行隐藏处理
- 输入：11位手机号
- 输出：135\*\*\*\*1234

参考：

```
s=input()  
print(s[:3]+"****"+s[-4:])
```