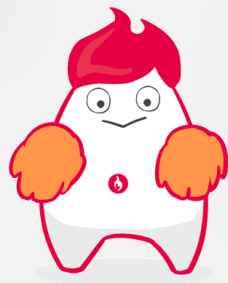


字符串与数字间的转换



本节课内容

- ◆ 什么是类型转换，为什么做类型转换
- ◆ 字符串与数字之间转换的要求
- ◆ 字符串与数字之间转换的函数



什么是类型转换，为何做类型转换

- ◆ 将自身数据类型变成新的数据类型，并拥有新的数据类型的所有功能的过程即为类型转换

a = '1' # 无法做数字操作

- ◆ 为方便更好的帮助处理业务，将类型变更为更适合业务场景的类型

字符串与数字之间转换的要求

- ◆ str → number : 数字组成的字符串

```
In [61]: int_str = '102983475'
```

```
In [62]: float_str = '3.14159'
```

- ◆ number → str : 无要求

字符串与数字之间转换的函数

原始类型	目标类型	函数	距离
整型	字符串	str	new_str = str(123456)
浮点型	字符串	str	new_str = str(3.14)
字符串	整型	int	new_int = int(' 12')
字符串	浮点型	float	new_float =float('1.2')

```
In [63]: int_str = '102983475'
```

```
In [64]: new_int = int(int_str)
```

```
In [65]: new_int
```

```
Out[65]: 102983475
```

```
In [66]: int_str = '10298347a5'
```

```
In [67]: int(int_str)
```

```
ValueError
```

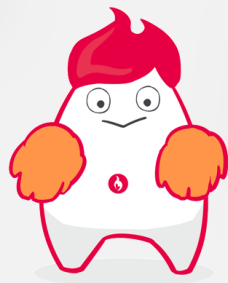
```
Traceback (most recent call last)
```

```
<ipython-input-67-a25a7528c9bf> in <module>()
```

```
----> 1 int(int_str)
```

```
ValueError: invalid literal for int() with base 10: '10298347a5'
```

字符串与列表间的转换



本节课内容

- ◆ 字符串转列表的函数--split
- ◆ 列表转字符串的函数--join



字符串转列表之split 功能

◆ 将字符串以一定规则切割转成列表

字符串转列表之split 用法

用法:

`string.split(sep=None, maxsplit=-1)`

参数:

sep: 切割的规则符号，不填写，默认空格，如字符串无空格则不分割生成 列表

maxsplit: 根据切割符号切割的次数，默认-1无限制

返回值:

返回一个列表

字符串转列表之split 用法

```
In [68]: info = 'my name is dewei'
```

```
In [69]: info_list = info.split()
```

```
In [70]: info_list
```

```
Out[70]: ['my', 'name', 'is', 'dewei']
```

列表转字符串之join 功能

- ◆ 将列表以一定规则转成字符串

列表转字符串之join 用法

用法:

'sep'.join(iterable)

参数:

sep: 生成字符串用来分割列表每个元素的符号

iterable: 非数字类型的列表或元组或集合

返回值:

返回一个字符串

列表转字符串之join 用法

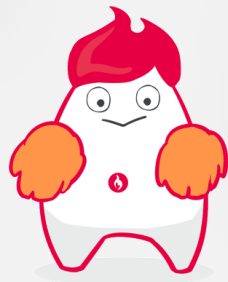
```
In [72]: test = ['a', 'b', 'c']
```

```
In [73]: new_str = '.'.join(test)
```

```
In [74]: new_str
```

```
Out[74]: 'a.b.c'
```

字符串与bytes的转换



本节课内容

- ◆ 什么是bytes（比特类型）
- ◆ 字符串转bytes的函数--encode
- ◆ bytes转字符串的函数--decode



什么是比特类型

- ◆ 二进制的數據流---byte
- ◆ 一种特殊的字符串
- ◆ 字符串前 + b 标记

```
In [4]: bt = b'my name is dewei'
```

```
In [5]: type(bt)
```

```
Out[5]: bytes
```


字符串转bytes的函数—encode 功能

◆ 将字符串转成比特 (bytes) 类型

字符串转bytes的函数—encode 用法

用法:

```
string.encode(encoding='utf-8', errors='strict')
```

参数:

encoding: 转换成的编码格式, 如ascii, gbk, 默认 utf-8

errors: 出错时的处理方法, 默认strict,

直接抛错误, 也可以选择 ignore忽略错误

返回值:

返回一个比特 (bytes) 类型

字符串转bytes的函数—encode 用法

```
In [11]: str_data = 'my name is dewei'
```

```
In [12]: byte_data = str_data.encode('utf-8')
```

```
In [13]: byte_data
```

```
Out[13]: b'my name is dewei'
```

bytes转字符串的函数--decode

◆ 将比特 (bytes) 类型转成字符串

bytes转字符串的函数--decode

用法:

```
bytes.decode(encoding='utf-8', errors='strict')
```

参数:

encoding: 转换成的编码格式, 如ascii, gbk, 默认 utf-8

errors: 出错时的处理方法, 默认strict, 直接抛错误, 也可

以选择ignore忽略错误

返回值:

返回一个字符串类型

bytes转字符串的函数--decode

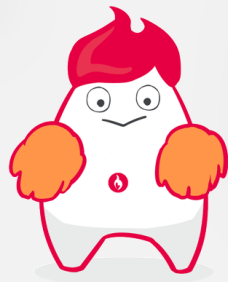
```
In [15]: byte_data = b'python is a good code'
```

```
In [16]: str_data = byte_data.decode('utf-8')
```

```
In [17]: str_data
```

```
Out[17]: 'python is a good code'
```

元组，列表，集合间的转换



本节课内容

列表元组集合之间的转换函数



列表元组集合之间的转换函数

原始类型	目标类型	函数	举例
列表	集合	set	<code>new_set = set([1,2,3,4,5])</code>
列表	元组	tuple	<code>new_tuple = tuple([1,2,3,4,5])</code>
元组	集合	set	<code>new_set = set((1,2,3,4,5))</code>
元组	列表	list	<code>new_list = list((1,2,3,4,5))</code>
集合	列表	list	<code>new_list = list({1,2,3,4,5})</code>
集合	元组	tuple	<code>new_tuple = tuple({1,2,3,4,5})</code>