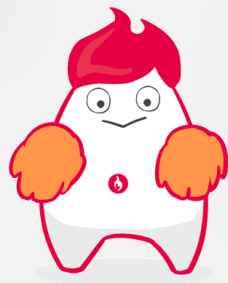


文件的创建与写入



本节课内容

- ◆ 利用内置函数open获取文件对象
- ◆ 文件操作的模式之写入
- ◆ 文件对象的操作方法之写入保存



利用内置函数open获取文件对象

功能:

生成文件对象，进行创建，读写操作

用法:

`open(path, mode)`

参数说明:

path: 文件路径

mode: 操作模式

举例:

```
f = open( 'd://a.txt' , 'w' )
```

返回值:

文件对象

文件操作的模式之写入

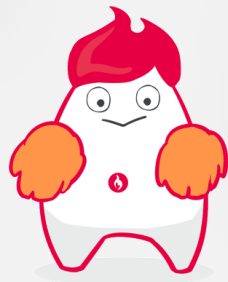
模式	介绍
w	创建文件
w+	创建或追加内容
wb	二进制形式创建文件
wb+	二进制形式创建或追加内容
a	追加内容
a+	读写模式的追加
ab+	二进制形式读写追加

文件对象的操作方法

方法名	参数	介绍	举例
write	Message	写入信息	f.write('hello\n')
writelines	Message_list	批量写入	f.writelines(['hello\n' , 'world\n'])
close	无	关闭并保存文件	f.close()

操作完成后，必须使用close方法！

文件的读取



本节课内容

- ◆ 文件操作的模式之读
- ◆ 文件对象的操作方法之读取



文件操作的模式之写入

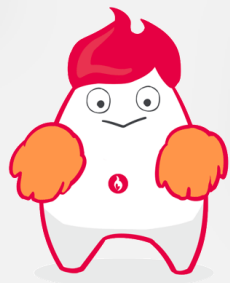
模式	介绍
r	读取文件
rb	二进制形式读取文件

文件对象的操作方法之读

方法名	参数	介绍	举例
read	无	返回整个文件字符串	f.read()
readlines	无	返回文件列表	f.readlines()
readline	无	返回文件中的一行	f.readline()
mode	无	文件模式	f.mode
name	无	返回文件名称	f.name
closed	无	文件是否关闭	f.closed

操作完成后，必须使用close方法！

序列化



本节课内容

- ◆ 初识序列化与反序列化
- ◆ 可序列化的数据类型
- ◆ Python中的json
- ◆ Python中的pickle



初识序列化

对象
信息

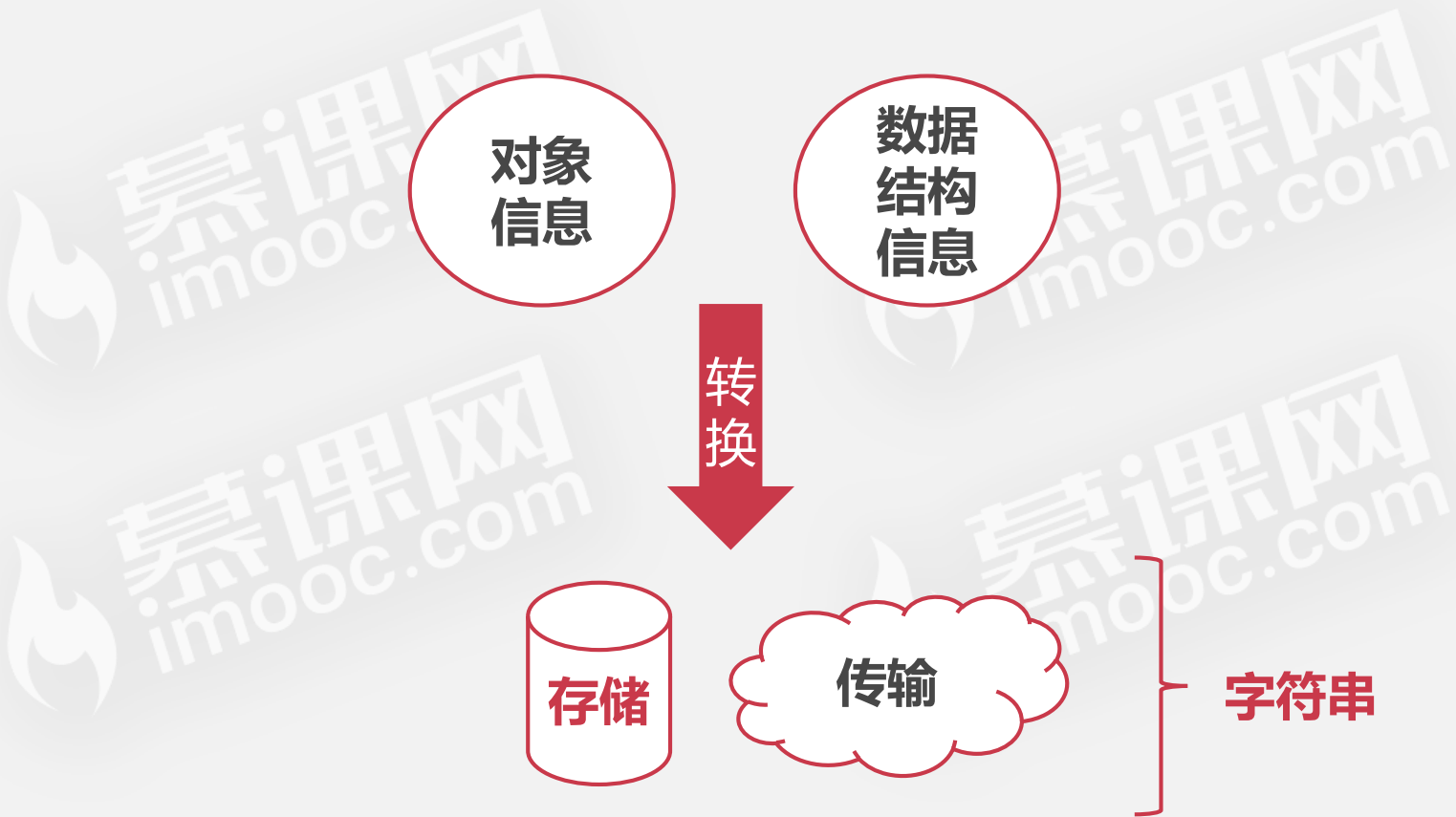
数据
结构
信息

转换

存储

传输

字符串



可序列化的数据类型

◆ number

◆ str

◆ list

◆ tuple

◆ dict



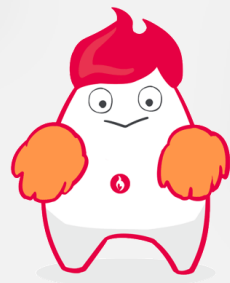
Python的json模块

方法名	参数	介绍	举例	返回值
dumps	obj	对象序列化	json.dumps([1,2])	字符串
loads	str	反序列化	Json.loads('[1, 2, 3]')	原始数据类型

Python的pickle模块

方法名	参数	介绍	举例	返回值
dumps	obj	对象序列化	<code>pickle.dumps([1,2])</code>	比特
loads	byte	反序列化	<code>pickle.loads('[1, 2, 3]')</code>	原始数据类型

Yaml文件的读取



本节课内容

- ◆ yaml格式的介绍
- ◆ Python的第三方包----pyyaml
- ◆ 读取yaml文件的方法



yaml格式的介绍

xxx.yaml

name:

xiaomu

age:

10

xinqing:

- haha
- heihei

new:

a: b

c: 1

key
value

列表

dict

Python的第三方包----pyyaml

◆ pip install pyyaml

◆ import yaml

读取yaml的方法

用法:

```
1: f = open(yaml_file, 'r' )  
2: data = yaml.load(f.read())  
3: f.close()
```

返回值:

字典类型

```
{  
    'name': 'xiaomu',  
    'age': 10,  
    'xingqing': ['haha', 'heihei'],  
    'new': {'a': 'b', 'c': 1}  
}
```