

# 74+戴一帆+211205102388

## 正则表达式3

戴一帆

2022年3月20号

## 结果

发什么快递？  
亲！你的快递是百世汇通。  
什么时候发货？  
你的快递估计在4.1发货。  
大概需要多久到？  
你的快递估计在4.3到达。  
有没有7天无条件退货？  
亲！我们绝对支持7天无理由退货的并回退两倍邮费。  
有没有优惠券？  
亲！我们会提供本店满1000元减300元的优惠券。  
发货地址？  
你的快递在深圳发货。  
包邮吗？  
亲！该商品为贵重物品，需要本人亲自查验签收并支付50元快递费。  
有没有赠品？  
亲！该商品为贵重物品，我们将随机发放赠品，赠品详细在商品介绍里有详细介绍。  
订单号是多少？  
亲！你的订单号是89757。  
开不开发票？  
亲！如果你需要发票，请联系人工客服并提交Account。  
原价格多少？  
亲！商品原价6000元。  
谢谢  
亲！给个五星好评哦，非常感谢！

```
import re # 调用模块
ID = '89757'
Name = '手机'
Price = '6000'
Company = "百世汇通"
State = "拣货"
Start_time = "4.1"
End_time = "4.3"
```

```
Start_local = "深圳"
End_local = "珠海"
Account = 'miss/20210307'
Complaints = '投诉' # 数据文本
while True: # 问答循环
    问题 = input() # 输入问题
    type1 = r"公司|发.+快递" # 用来表达对字符串的一种过滤逻辑
    guize1 = re.findall(type1, 问题) # 对问题进行匹配
    if guize1 == []: # 如果问题是空, 就跳过
        pass
    else: # 问题匹配, 就回答print的内容
        print("亲! 你的快递是%s。" % Company) # %s表示引用

    type2 = r"什么时候发货?|多久发货|. +时候发货"# 用来表达对字符串的一种过滤逻辑
    guize2 = re.findall(type2, 问题)# 对问题进行匹配
    if guize2 == []: # 如果问题是空, 就跳过
        pass
    else: # 问题匹配, 就回答print的内容
        print("你的快递估计在%s发货。" % Start_time) # %s表示引用

    type3 = r"多.+到|时候到"# 用来表达对字符串的一种过滤逻辑
    guize3 = re.findall(type3, 问题)# 对问题进行匹配
    if guize3 == []: # 如果问题是空, 就跳过
        pass
    else: # 问题匹配, 就回答print的内容
        print("你的快递估计在%s到达。" % End_time) # %s表示引用

    type4 = r"7天无理由|7天无条件"# 用来表达对字符串的一种过滤逻辑
    guize4 = re.findall(type4, 问题)# 对问题进行匹配
    if guize4 == []: # 如果问题是空, 就跳过
        pass
    else: # 问题匹配, 就回答print的内容
        print("亲! 我们绝对支持7天无理由退货的并回退两倍邮费。")

    type5 = r"优惠券"# 用来表达对字符串的一种过滤逻辑
    guize5 = re.findall(type5, 问题)# 对问题进行匹配
    if guize5 == []: # 如果问题是空, 就跳过
        pass
    else: # 问题匹配, 就回答print的内容
        print("亲! 我们会提供本店满1000元减300元的优惠券。")

    type6 = r"在.+发货|发货地址"# 用来表达对字符串的一种过滤逻辑
    guize6 = re.findall(type6, 问题)# 对问题进行匹配
```

```
if guize6 == []: # 如果问题是空, 就跳过
    pass
else: # 问题匹配, 就回答print的内容
    print("你的快递在%s发货。" % Start_local) # %s表示引用

type7 = r"包邮"# 用来表达对字符串的一种过滤逻辑
guize7 = re.findall(type7, 问题)# 对问题进行匹配
if guize7 == []: # 如果问题是空, 就跳过
    pass
else: # 问题匹配, 就回答print的内容
    print("亲! 该商品为贵重物品, 需要本人亲自查验签收并支付50元快递费。")

type8 = r"赠品"# 用来表达对字符串的一种过滤逻辑
guize8 = re.findall(type8, 问题)# 对问题进行匹配
if guize8 == []: # 如果问题是空, 就跳过
    pass
else: # 问题匹配, 就回答print的内容
    print("亲! 该商品为贵重物品, 我们将随机发放赠品, 赠品详细在商品介绍里有详细介绍。")

type9 = r"订单号"# 用来表达对字符串的一种过滤逻辑
guize9 = re.findall(type9, 问题)# 对问题进行匹配
if guize9 == []: # 如果问题是空, 就跳过
    pass
else: # 问题匹配, 就回答print的内容
    print("亲! 你的订单号是%s。" % ID) # %s表示引用

type10 = r"投诉"# 用来表达对字符串的一种过滤逻辑
guize10 = re.findall(type10, 问题)# 对问题进行匹配
if guize10 == []: # 如果问题是空, 就跳过
    pass
else: # 问题匹配, 就回答print的内容
    print("亲! 请千万不要投诉我, 我是小可怜!")

type11 = r"发票"# 用来表达对字符串的一种过滤逻辑
guize11 = re.findall(type11, 问题)# 对问题进行匹配
if guize11 == []: # 如果问题是空, 就跳过
    pass
else: # 问题匹配, 就回答print的内容
    print("亲! 如果你需要发票, 请联系人工客服并提交Account。")

type12 = r"价格"# 用来表达对字符串的一种过滤逻辑
guize12 = re.findall(type12, 问题)# 对问题进行匹配
```

```
if guize12 == []: # 如果问题是空，就跳过
    pass
else: # 问题匹配，就回答print的内容
    print("亲！商品原价%s元。"%Price) # %s表示引用

end = '谢谢' # 顾客输入“谢谢”，即退出机器人客服服务。
if 问题 == end:
    print("亲！给个五星好评哦，非常感谢！")
    break # 打断循环
```

