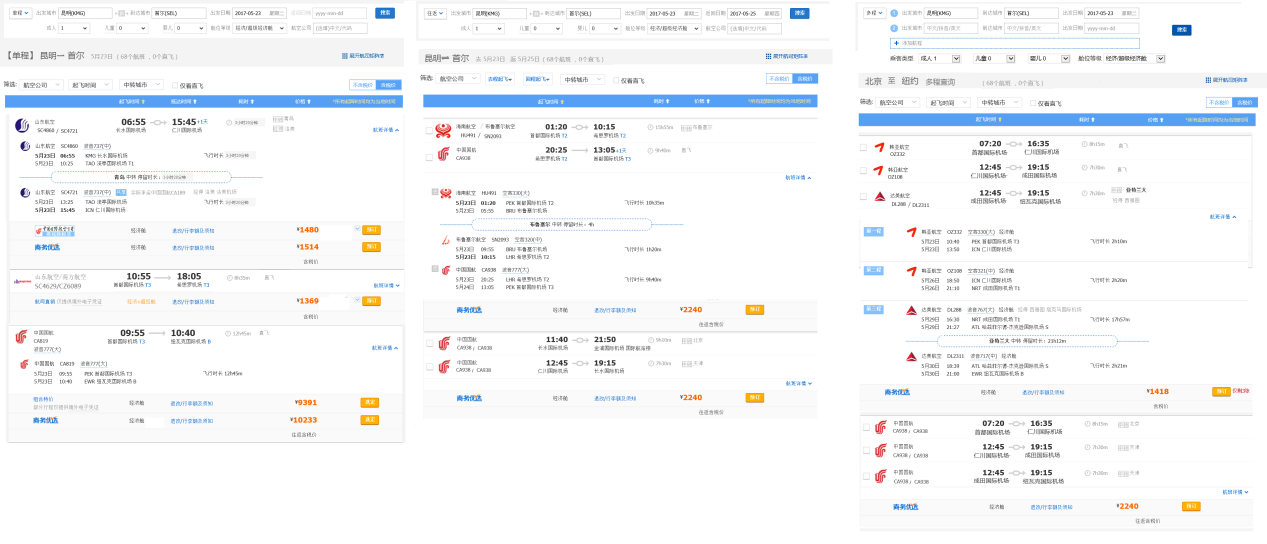
# 单程、往返、多程查询结果图示



航班结果列表

工具条

搜索输入框

为了方便对比将 单程、往返、多程的查询结果页放在一起。

查询结果页由三部分构成：搜索输入框、工具条、航变结果列表

# 搜索输入框

搜索输入框的动态效果可参考携程（[www.ctrip.com](http://www.ctrip.com)）。

### 单程和往返





单程、往返的唯一差别是 单程的“返回日期”项是disable的。

### 多程



多程跟单程、往返的差别比较大。 点“添加航程”会出来一个新的航段条件（红色圈起来的部分）默认放两个航段条件即可。

### 各输入项的说明

出发城市、到达城市、出发日期、返回日期、航空公司 这几个输入框 需要加控件。 控件用51book现有控件即可。

另外，希望几个关键的输入控件可以在当前控件输入完成后自动跳到下一个。（效果可参携程，今日天下通的响应更快，感觉更好 www.jinri.net）

单程的顺序是： 出发城市、到达城市、出发日期

往返的顺序：出发城市、到达城市、出发日期、返回日期

多程的顺序：逐个航段的 出发城市、到达城市、出发日期，且下一航段的出发城市默认上一航段的到达城市。

**各输入框需加校验：**

单程：出发、到达、出发日期，为必填项。

往返：出发、到达、出发日期、到达日期，为必填项。

往返：各航程的 出发、到达、出发日期 均为必填项（添加出来的航程也不例外）。

出发城市、到达城市， 这两个输入框 最终传給后台的是三字码（表示机场或城市的三字代码，例：BJS）。无效的三字码不允许提交表单。

出发日期、返回日期，均是日期格式。

单程、往返、多程： 所有情况下出发日期不得早于当天。

往返：出发日期不得早于当天。返回日期不得早于出发日期。

多程：后一程的出发日期不得早于前一程的出发日期。

希望通过日期控件做限制日期的选择

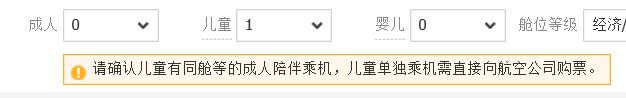
**乘客人数的校验：**

成人的下拉框可选值为（0-9）。儿童、婴儿下拉框可选值为（0-9）

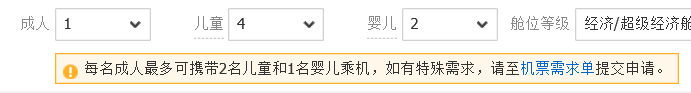
1、当婴儿、儿童、成人 人数都是0时。 提示，并禁用“搜索”按钮



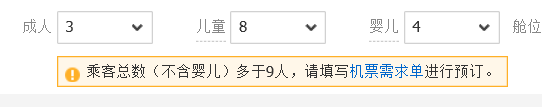
2、当（成人=0）且（婴儿+儿童）>0时。提示，并禁用“搜索”按钮



3、当 （儿童>成人\*2） 或者 （婴儿>成人）。提示，并禁用“搜索”按钮



4、当（成人+儿童+婴儿）>9。 提示，并禁用“搜索”按钮



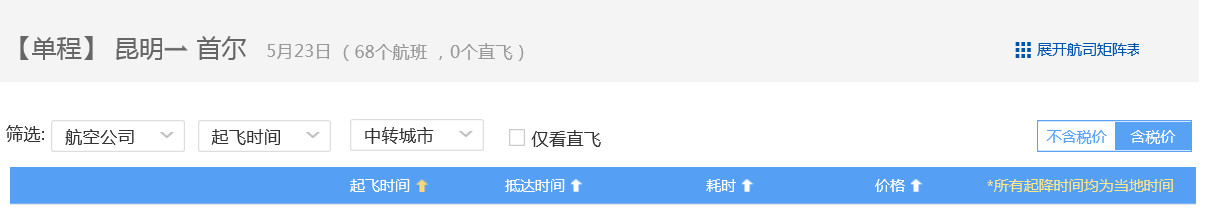
以上四个校验的提示文本回头给出，文本里含超链接。

（以上所有校验的效果均是用户完成输入后即时校验，效果可参看携程）

# 工具条

单程

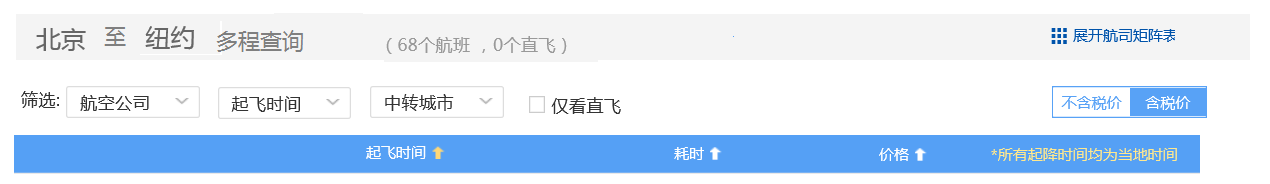
Title



往返



多程



注意各种行程类别下，这部分的区别：

在Title部分，单程、往返、多程显示各不相同。 （已经考虑过实现的可行性，就按图示显示）

筛选部分：往返程 多了“去程起飞时间”和“回程起飞时间”的筛选。 单程和往返只有“起飞时间”筛选，是指第一程的起飞时间。

排序部分：单程多了“抵达时间”的排序。 往返和多程只有“起飞时间”“耗时”“价格”的排序。

在往返和多程上，“耗时”的排序是指行程的总耗时。 “价格”的排序指某个航班方案的最低价排序（一个航班方案可能有多个可用价格）。

 点击这个图标，会展开如下的航班矩阵：



列出所有航司及不同中转次数下的价格。（对应返回json串中的priceTalbe）。 点击其中某个航司的列可以对搜索结果做按此航司筛选。

# 航班结果列表

### 一个航班方案

查询结果由多个航班方案组成。

返回的json串中有一个 avFlightList，avFlightList中的每个成员表示一个完整的航班方案。

每个航班方案由“航程信息”“航班详情”“可用价格列表”三部分构成。

“航程信息” 对应 avFlight.odList, “航班详情”由所有od的flightDetail构成，“可用价格列表”对应avFlight.fareList。

航程信息

下图是单程查询结果中一个中转航班方案的案例。



可用价格列表

航班详情

### 航程信息：

航程信息由 1到n个OD信息构成。（O——出发地，D——抵达地）

单程由1个OD构成。

往返由2个OD构成，一去一回。

多程由多个OD构成，取决于查询时传入了几段航程。

单程：



往返：



**② ⑨ ⑩ ⑾ ⑿**

1. **③ ④ ⑤ ⑥ ⑦ ⑧**

多程（案例是三段OD，最少2段、最多6段）：



不管是单程、往返、多程， 展示航程信息时， 只要将avFlight.odList 里的所有od遍历做展示即可。

每个od由上图红框中的 10个信息构成：

1. 航司：对应 od.airline
2. 航班号：od.flightNo
3. 出发时间：od.flightDetail[0].departureTime 此OD第一个航班的出发时间
4. 直飞中转标志：直飞标志是 中转标志是 
5. 到达时间：od.flightDetail[flightDetail.lenght-1].arriveTime 此OD最后一个航班的到达时间
6. 到达时间是否跨日期：

od.flightDetail[0].departureTime的日期 和od.flightDetail[flightDetail.lenght-1].arriveTime的日期 之间跨越的天数。第一个航班的起飞日期和最后一个航班的起飞日期跨越的天数。

比较便捷的计算方法是(Math.floor(D1/86400000)-Math.floor(D2/86400000))

D1=od.flightDetail[flightDetail.lenght-1].arriveTime

D2=od.flightDetail[0].departureTime

1. 航程时长：od.duration
2. 经停or中转信息：
   1. 如果od.stopCity 和od.transferCity 都是空，显示“直飞”。
   2. 如果 od.transferCity不为空，显示 ，中转图标后面的“布鲁塞尔”取od.transferCity。
   3. 如果 od.stopCity不为空，显示，经停图标后面的“西雅图”取od.stopCity
3. 出发机场：od.dPortName
4. 出发机场航站楼：如果od.flightDetail[0].depTerm 不为空，若此字段不为空，若为数字，显示 “T”+此值，若为字符,直接显示此值。为空不显示
5. 到达机场:od.aPortName
6. 到达机场航站楼：od.flightDetail[flightDetail.length-1].arrTerm不为空，若此字段不为空，若为数字，显示 “T”+此值，若为字符,直接显示此值。为空不显示

### 航班详情

单程的 中转 案例



单程的直飞案例



往返的航班详情案例



多程的航班详情案例：



单程的每个航班方案只有OD

往返的每个航班方案由两个OD构成，第一个OD前显示 ，第二个OD前显示 

多程的每个航班方案由多个od（2~6个）构成，每个od的航班前显示图标 或者 第二程、第三程 ……

每个OD都有 flightDetail，是个list，包含一到多个成员。

如果flightDetail有多个成员，表示这个OD由多个航班构成，各航班间用连接。

以下面这个多程的查询结果为例：

OD1

OD2

OD3

OD1.flightDetail

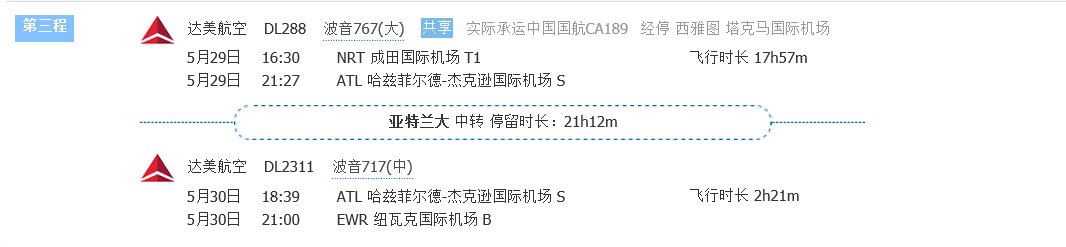
OD2.flightDetail

OD3.flightDetail



航程信息里的每个OD，对应到 下方“第一程”“第二程”“第三程”的航班详情。

其中“第一程”“第二程”都只由一个航班构成，“第三程”由2个航班构成。



1. **② ③ ④**

**⑤ ⑥ ⑦ ⑧ ⑨ ⑩**

**⑾ ⑿ ⒀ ⒁ ⒂**

**⒃ ⒄**

用flightDetail[0]代表 od.flightDetail这个list中的第一个成员

1. 航司：flightDetail[0].airlineName
2. 航班号：flightDetail[0].airline + flightDetail[0].airFlightNo
3. 机型：flightDetail[0].plane
4. 共享航班和经停信息：
   1. flightDetail[0].codeshare 如果是 true，表示此航班是共享航班，显示：+“实际承运”+ flightDetail[0].airCarrierName + flightDetail[0].airCarrierAline + flightDetail[0].airCarrierFlight
   2. flightDetail[0].stopInfoList 这是个list，如果这个list不为空。遍历list，显示 “经停”+cityName+airportName。cityName表示经停点城市名，airportName表示经停点机场名。
5. 此航班的出发日期：flightDetail[0].departureTme 转化为日期格式后 取 月、日，MM-dd
6. 此航班的出发时间：flightDetail[0].departureTme 转化为日期后去 时、分，hh-mm
7. 出发机场三字码：flightDetail[0].departurAirport
8. 出发机场名称: flightDetail[0].dPortName
9. 出发机场航站楼：flightDetail[0].depTerm，若此字段不为空，若数字，显示 “T”+此值，若为字符,直接显示此值。为空不显示
10. 此航班的飞行时长：flightDetail[0].duration
11. 此航班的抵达日期：flightDetail[0].arriveTIme转化为日期格式后 取 月、日，MM-dd
12. 此航班的抵达时间：flightDetail[0].arriveTme 转化为日期后取 时、分，hh-mm
13. 抵达机场三字码：flightDetail[0].arriveAirport
14. 抵达机场名称: flightDetail[0].aPortName
15. 抵达机场航站楼：flightDetail[0].arrTerm，若此字段不为空，若数字，显示 “T”+此值，若为字符,直接显示此值。为空不显示
16. 中转点：连接前后两个航班



技术上的处理方式是 在处理完当前这航班之后，如果后续还有航班，取当前航班的抵达城市 flightDetail[index].aCityName

和当前航班的抵达时间和下一航班的出发时间的时间差。timeDiff(flightDetail[index+1].departureTime, flightDetail[index].arriveTime)

**function** timeDiff(d1,d2){

**var** mm= ((d1-d2)/60000)%60 ;

**var** hh=Math.floor((d1-d2)/3600000);

**return** hh+'小时'+mm+'分钟'

} }

### 可用运价列表

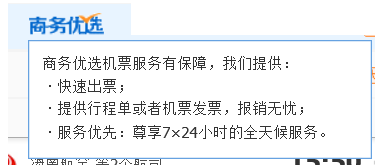
1. **② ③ ④ ⑤**



表示这个航班的可预订的价格有哪几个。每个avFlight都有一个 fareList，每个成员表示一个可预订价格。

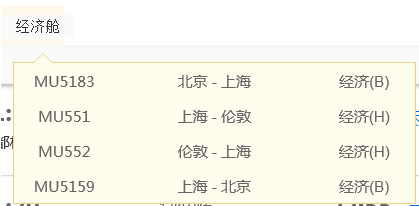
这部分有比较多的tips。

**1、运价的title**，会存在几种不同定义的运价。不同的Title有不同的tips内容。



**2、服务等级信息：fareList[i].physicalClassName**

**Tips里的内容取fareList[i].classInfo 是一个list，遍历。**



**3、退改及行李额须知。**

这个内容中的一部分信息需要再向服务器端发一次请求。以后详述。



**4、价格：**

价格这里有几种不同的情况。

如果工具条上的选项是“往返含税价” 

则价格这里显示 **fareList[i].avgSettlementPriceWithTax**

**且当乘客类型是几种乘客类型混合查询时，需要显示 tips**

**由于在搜索输入时对不同乘客类型的人数做了限制。**

**可查询的只有这几种情况：**

**a、成人：**



B、成人+儿童+婴儿 （人数按实际情况，下图仅为示例）

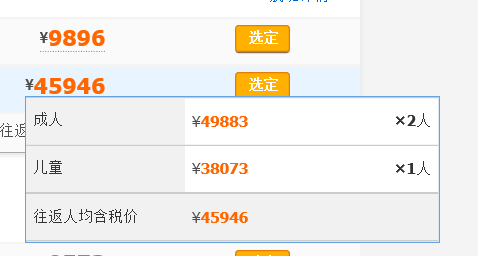
Tips里的信息分别取自**fareList[i].**adultPrice、**fareList[i].**childPrice、**fareList[i].**infPrice

价格= settlementPriceWithTax

人数=passengerCount



C、成人+儿童



D、成人+婴儿



另一种情况是工具条上选择了“不含税价格”

则价格处显示如下：



价格处是 **fareList[i].avgSettlementPrice ,**

**“税费”是fareList[i].avgTax**

Tips里显示如下（只给出了一种混合查询的案例，其他案例类推）：

Tips里的信息分别取自**fareList[i].**adultPrice、**fareList[i].**childPrice、**fareList[i].**infPrice

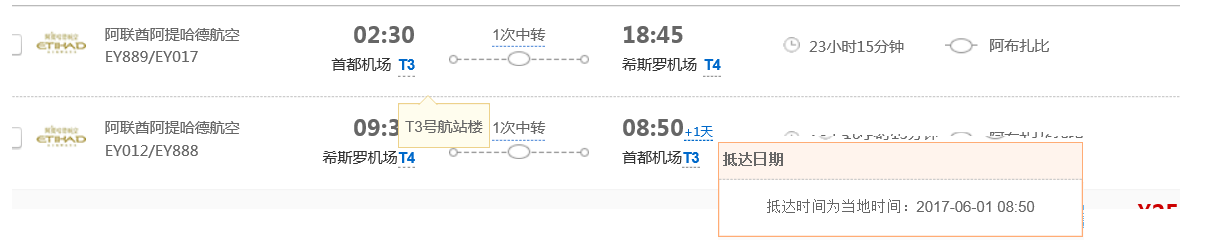
价格=settlementPrice

税=tax

人数=passengerCount



# 其他tips



航程信息上的 航站楼 和 抵达日期的跨天数， 也需要加tips。

未来，在航班详情的 机型信息上，需要加tips。