1.

**Python的可变类型和不可变类型**

* 数字、字符串、元组是不可变的，列表、字典是可变的。

2.

**类型转换**

* int(str) : 转换为int型。int('1') >>> 1
* float(int/str) : 将int型或字符型转换为浮点型。float('1') >>> 1.0
* str(int) : 转换为字符型。str(1) >>> '1'
* bool(int) : 转换为布尔类型。 str(0) >>> False str(None) >>> False

3.

**Python写9\*9乘法表的两种简单方法**

1 for i in range(1,10):

2     for j in range(1,i+1):

3         print("%s \* %s = %s" %(j,i,i\*j),end="")

4     print("")

print "\n".join("\t".join(["%s\*%s=%s" %(x,y,x\*y) for y in range(1, x+1)]) for x in range(1, 10))

4.**python是否区分大小写？**

是。Python是一种区分大小写的语言。

**5.Python数组和列表有什么区别？**

Python中的数组和列表具有相同的存储数据方式。但是，数组只能包含单个数据类型元素，而列表可以包含任何数据类型元素。

Python中的内置数据类型称为字典。它定义了键和值之间的一对一关系。字典包含一对键及其对应的值。字典由键索引。

**列表list []**

列表是处理一组有序的数据结构，可以读写，添加和删除，或者搜索列表里的元素。因为可以添加和删除，所以称为可变的数据类型，即这种类型是可以被改变的，并且列表可以嵌套。

**元组tuple（）**

元组跟列表非常相似，用（）来表示，但是元组是不可变的，不能修改元组。元组可以嵌套。

字典dict

字典是通过键值对的方式就数据存储下来，键必须是唯一的

键值对在字典中以这样的方式标记：d = {key1 : value1, key2 : value2 }。注意它们的键/值对用冒号分割，而各个对用逗号分割，所有这些都包括在花括号中。另外，记住字典中的键/值对是没有顺序的。如果你想要一个特定的顺 序，那么你应该在使用前自己对它们排序

集合(set)

特性：与字典类似，但只包含键，而没有对应的值，包含的数据不重复。

创建：s = set(list or tuple or string)。重复的值在集合中只存在一个。如：

　　　　列表list：s = set([1,2,3,3]) ->s = set([1,2,3])

　　　　元组tuple：s = set((1,2,3)) ->s = set([1,2,3])

　　　　字符串string：s = set(“abc”) ->s = set([“a”,“b”,“c”])

4。== 和 is 区别。

* == 是比较两对象的值，is 是比较在内存中的地址(id)， is 相当于 id(objx) == id(objy)。
* *说说你知道的 HTTP 方法和 状态码。*
  + [状态码](https://zh.wikipedia.org/wiki/HTTP%E7%8A%B6%E6%80%81%E7%A0%81)，这里只需要大概说说，以 1××，2××，3×× 这样的层面说，没有必要细到每一个状态码。
  + [HTTP 请求方法](https://zh.wikipedia.org/wiki/%E8%B6%85%E6%96%87%E6%9C%AC%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE" \l ".E8.AF.B7.E6.B1.82.E6.96.B9.E6.B3.95)
* *HTTP 的 GET 和 POST 有什么区别？*
  + 本质上，GET 和 POST 只不过是 发送机制不同 。
* HTTP 和 HTTPS 的区别？
  + HTTPS = HTTP + SSL
  + HTTP 默认使用 80 端口，HTTPS 使用 443 端口。
  + [更详细](https://zh.wikipedia.org/wiki/%E8%B6%85%E6%96%87%E6%9C%AC%E4%BC%A0%E8%BE%93%E5%AE%89%E5%85%A8%E5%8D%8F%E8%AE%AE" \l ".E4.B8.8EHTTP.E7.9A.84.E5.B7.AE.E5.BC.82)
* 说说你知道的 HTTP 包头部信息里都有哪些字段。
  + 这个随便抓下包就知道了，就不说了～
* *HTTP 包头部信息里面的 Host 字段是什么作用？*
  + 表示当前请求服务器的主机名

**22、如何测试一个纸杯？**

功能度：用水杯装水看漏不漏；水能不能被喝到

安全性：杯子有没有毒或细菌

可靠性：杯子从不同高度落下的损坏程度

可移植性：杯子在不同的地方、温度等环境下是否都可以正常使用

兼容性：杯子是否能够容纳果汁、白水、酒精、汽油等

易用性：杯子是否烫手、是否有防滑措施、是否方便饮用

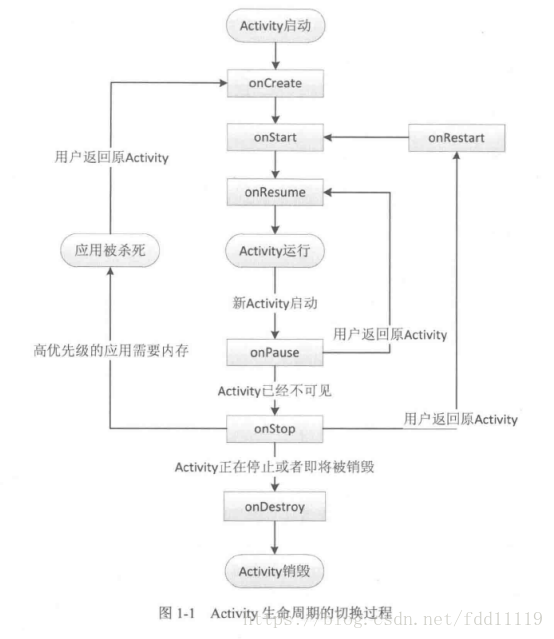
用户文档：使用手册是否对杯子的用法、限制、使用条件等有详细描述

疲劳测试：将杯子盛上水（案例一）放24小时检查泄漏时间和情况；盛上汽油（案例二）放24小时检查泄漏时间和情况等

压力测试：用根针并在针上面不断加重量，看压强多大时会穿透

**假设有一个文本框要求输入10个字符的邮政编码，对于该文本框应该怎样划分等价类？**

特殊字符，如10个\*或￥；英文字母，如ABCDefghik；小于十个字符，如123；大于十个字符，如11111111111；数字和其他混合，如123AAAAAAA；空字符；保留字符



二、正常情况下，Activity的生命周期调用过程

然后我们来梳理一下几种情况下，生命周期的方法调用顺序是怎么样的。

启动一个特定的Activity时，第一次启动，生命周期回调如下：

onCreate –> onStart –> onResume

当用户打开新的Activity或者切换到桌面的时候，回调如下：

onPause –> onStop

但是有一种特殊的情况，如果新的Activity采用了透明的主题，那么当前Activity不会回调onStop

当用户再次回到原Activity时，回调如下：

onRestart –> onStart –> onResume

当用户点击Back键回退时，回调：

onPause –> onStop –> onDestory

当Activity被系统回收后再次打开时，回调onCreate –> onStart –> onResume（和过程1一样，但是只是生命周期的回调过程一样，这里还是有一些其他过程的区别的，比如回收时onSaveInstanceState的方法的调用，以及打开时onRestoreInstanceState的调用）

在生命周期中，onCreate和onDestroy是对应的，创建和销毁，并且在生命周期过程中，只被调用一次。从Activity是否可见来说，onStart和onStop是配对的，这两个方法有可能会被多次调用。从Activity是否在前台来说，onResume和onPause是配对的，也有可能会被多次调用。

其中有两个问题：

第一：onStart和onResume，onPause和onStop差不对，但是又有什么本质的不同呢？

第二：当从A 窗口打开B窗口时，B窗口的onResume和A的onPause方法哪个先执行呢？

首先第一个问题：

从上面的描述中就知道了，虽然差不多，但是角度还是不通的。onStart和onStop从Activity是否可见来说的，onResume和onPause从Activity是否在前台来说的。其他的就没有什么区别了。

第二问题：

自己实践一下就知道了，先是A的onPause方法调用结束之后，才会执行B窗口的onResume方法。

[Android](http://lib.csdn.net/base/15" \o "Android知识库)四大组件分别为activity、service、content provider、broadcast receiver。

   Android 开发的四大组件分别是：活动（activity），用于表现功能；服务（service），后台运行服务，不提供界面呈现；广播接受者（Broadcast Receive），勇于接收广播；内容提供者（Content Provider），支持多个应用中存储和读取数据，相当于数据库。