salesforce如果简单的说可以大概分成两个部分：Apex,VisualForce Page.

其中Apex语言和java很多的语法类似，今天总结的是一些简单的Apex的变量等知识。

有如下几种常用的基本变量Integer,String,Decimal,Double,Long,Boolean,ID。

集合常用的对象：List<T>,Set<T>,Map<T>。

时间日期常用对象：Datetime,Time,Date。

其他：Object，sObject(与数据库相关，以后篇会讲)

与JAVA一个最大的区别是：Apex中基本对象的初始值均为null。

eg：

|  |  |
| --- | --- |
| 1  2  3 | Integer i;  i += 1;  System.debug(i); |

1

在java中此种写法是可以的，因为int类型初始值为0，i+=1以后则i变成1.但是在Apex中因为i初始值为null。

所以i+=1在运行时会抛出NullPointerException

当然，比较有意思的事情是这样，直接上代码：

|  |  |
| --- | --- |
| 1  2 | Integer i;  System.debug(i+'1'); |

此种方法输出的结果则为null1。起始这也不奇怪，因为Apex也是基于java拓展的，如果看java编程思想了解底层的null的toString()方法处理也就知道了,当执行Print操作时，一个变量为null时，他的toString方法则返回'null'字符串。当然，这个只是一个拓展，不多展开，如果感兴趣，可以查看一下java的api或者看一下java编程思想一书。

**一)基本变量：**

**1）Integer**

Integer表示一个32位整数的对象，取值范围为-2^31 -- 2^31.

Integer主要有两个方法：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | /\*      public String format()      //译：将Integer值转换成String类型的值   \*/  Integer goodsCount = 12;  System.debug('将Integer值转成String： ' + goodsCount.format());  /\*     public static Integer valueOf(String stringToObject)     //译：将String类型转成Integer类型  \*/  Integer goodsCountI = Integer.valueOf('12'); |

**2)Long**

Long类型表示一个64位整数的对象，取值范围为-2^63--2^63-1.

Integer类型可以直接转换成Long类型，Long类型在不超过范围情况下可以通过intValue()方法转成Integer类型。

以下为Long类型部分主要方法：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | Integer transferSource = 12345;<br>Long code = transferSource;//Integer类型可以直接转成Long类型  /\*     public String format()     //译：将Long类型转换成String类型  \*/  System.debug('Long类型转成String类型：' + code.format());  /\*     public Integer intValue()     //译：将Long类型转成Integer类型  \*/     System.debug('将Long类型转成Integer类型：' + code.intValue());  /\*     public static Long valueOf(String stringToLong)     //译：将String类型转成Long类型  \*/  Long codeLong = Long.valueOf('123'); |

**3）ID**

ID类型可以用任何一个符合规则的18位字符表示，如果你设置ID字符为15位，则将字符自动扩展成18位。不符合规则的ID字符在运行时则运行时异常。

以下为ID的主要方法：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | /\*      public static ID valueOf(String toID)      //译：将toId转换成ID  \*/  String idStr = '111111111111111111';  ID id = ID.valueOf(idStr);  /\*     public Boolean equals(String id)     //译：判断两个ID是否相同  \*/  Boolean isEquals = id.equals(idStr); |

**4）Decimal**

简单的来说，Decimal变量的意思为包含小数点的32位数就是Decimal，很像java中的float类型变量。

以下为Decimal的部分主要方法用来了解Decimal的功能：

C:\Users\SS67FD~1\AppData\Local\Temp\msohtmlclip1\02\clip_image001.gif

Decimal function

**5)Double**

Double变量为包含小数点的64位数，很像 java中的Double类型变量。

Decimal类型变量可以直接转换成Double类型变量，Double类型在不超过范围情况下可以通过

以下为Double的部分主要方法：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | Double price = 34.5678;  /\*     public static Double valueOf(String stringToDouble)     //译：将String类型转换成Double  \*/  String doubleString = '3.89';  System.debug('将字符串转换成Double' + Double.valueOf(doubleString));    /\*     public Long round()     //译：返回double最接近Long的值,四舍五入  \*/  Long priceLong = price.round();  System.debug('通过round方法将double转换成Long类型值为：' + priceLong);    /\*     public Integer intValue()     //译：将double值转换成int类型值  \*/  Integer priceInteger = price.intValue();  System.debug('将double转换成Integer类型值为：' + priceInteger);  Long priceLongByLongValue = price.longValue();  System.debug('将double转换成Long类型值为：' + priceLongByLongValue); |

**6）String**

String类型和Java中的String类型很类似，在这里不做过多解释，代码中主要需要看一下String类型对象和上述变量如何相互转换，这在项目中是经常用到的，也是必须需要知道的。以下为String类型主要方法：

String Function

**7）Boolean**

Boolean类型声明一个布尔类型，和java区别为：Boolean类型变量有三个取值：true,false,null(default)，所以使用Boolean类型声明的时候必须赋予初始值，否则初始值为null

**二）时间日期类型**

**1)Datetime**

Datetime类型声明一个日期时间的对象，包含两部分：日期，时间。因为salesforce一般制作global项目，所以日期时间一般取格林时间。Datetime无构造函数，如果实例化只能通过其静态方法初始化。以下为Datetime的部分主要方法：

C:\Users\SS67FD~1\AppData\Local\Temp\msohtmlclip1\02\clip_image001.gif

Datetime function

**2)Date**

Date类型声明一个日期的对象，Date可以和Datetime相互转换，主要需要掌握二者关系以及相互转换。

以下为Date部分主要方法：

Date function

**3)Time**

Time类型声明一个时间的对象，对于时间需要考虑的是：因为中国时间和格林时间相差8小时，所以具体项目时如果是global项目需要考虑使用格林时间，即GMT时间。

**三）集合类型**

集合类型主要有三种，List，Set以及Map。其中三种均为泛型方式，所以声明变量时，直接带上泛型。

**1）List<T>**

List代表一类的有序数据列表。数据序号从0开始。与JAVA不同的是：List是一个类，并且不存在ArrayList等子类。即实例化

eg:List<String> list1 = new List<String>();

List可以通过自身构造函数实例化，也可以通过数组进行实例化。

以下为List主要方法：

注：set()方法在设置插入位置以前应确保长度大于需要插入的位置，否则将抛出异常。

C:\Users\SS67FD~1\AppData\Local\Temp\msohtmlclip1\02\clip_image001.gif

List Function

**2)Set<T>**

Set代表一类数据的无序列表。与JAVA不同的是：Set是一个类，不存在HashSet等子类。即实例化

eg:Set<String> set1 = new Set<String>();

Set主要方法如下：

C:\Users\SS67FD~1\AppData\Local\Temp\msohtmlclip1\02\clip_image001.gif

Set Function

**3)Map<K,V>**

Map代表着键值对，与JAVA用法类似，区别为Map是一个类，不是接口，不存在HashMap<K,V>等子类

Map主要方法如下：

C:\Users\SS67FD~1\AppData\Local\Temp\msohtmlclip1\02\clip_image001.gif

Map Function

**四）运算及控制语句**

运算与控制语句和JAVA基本类似，所以在这里只是简单介绍一下增强for循环

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | List<String> goodsList = new String[] {'衣服','裤子'};    for(String goods : goodsList) {        System.debug(goods);    } |

详细了解这些基本类型变量，集合以及相关方法请自己查看API。