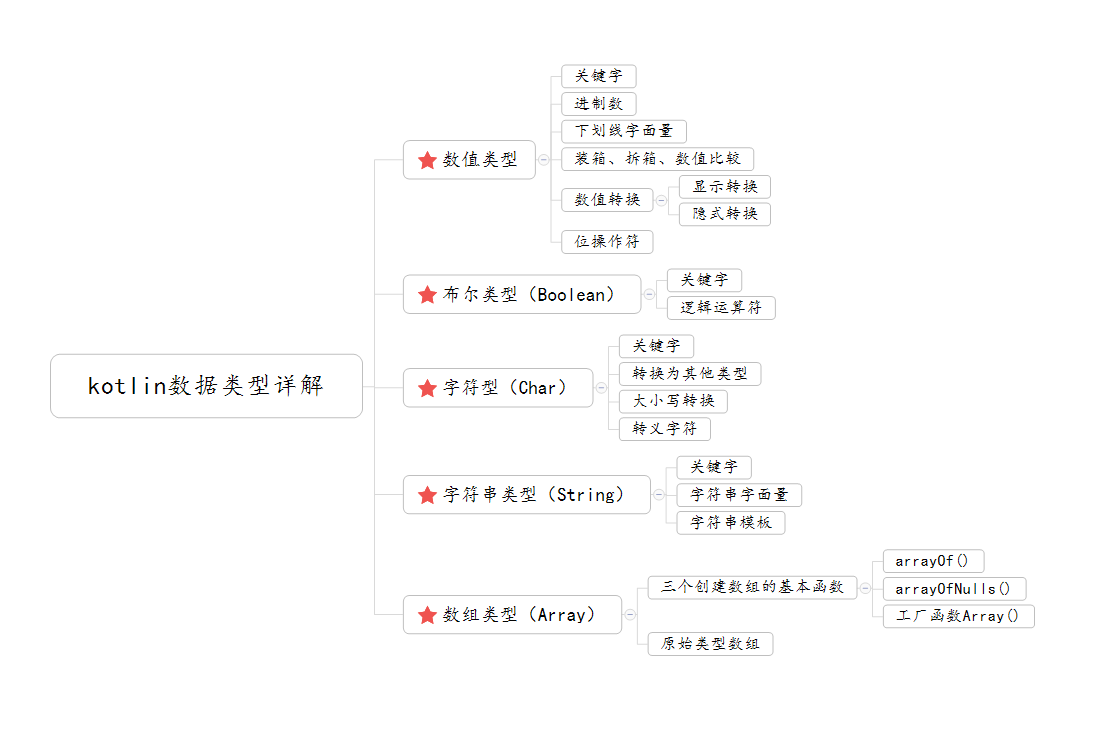
# Kotlin学习笔记（数据类型）

## 数据类型图解



## 数组

在Kotlin中数组用Array<T> 表示，创建数组有三种方式：

### arrayOf():

创建一个数组，参数是一个可变参数的泛型对象

var arr1 = *arrayOf*(1, 3, 5, 7, 9)  
for (i in arr1) {  
 *print*("i=> $i \t")  
}

最终打印结果：i=> 1 i=> 3 i=> 5 i=> 7 i=> 9

var arr2 = *arrayOf*(1, 'a', 5, "Kotlin", 9, 3.1f)  
for (i in arr2) {  
 *print*("i=> $i \t")  
}

最终打印结果：i=> 1 i=> a i=> 5 i=> Kotlin i=> 9 i=> 3.1

### arrayOfNulls()

用于创建一个指定数据类型且可以为空元素的给定元素个数的数组

var arr1 = *arrayOfNulls*<Int>(5);  
for (i in arr1) {  
 *print*("i=> $i \t");  
}  
*println*("arr1数组的大小 => ${arr1.size}")

最终打印结果：i=> null i=> null i=> null i=> null i=> null arr1数组的大小 => 5

可以看到若不赋值元素值都为空

var arr2 = *arrayOfNulls*<String>(3)  
arr2[0] = "第一个"  
arr2[1] = "Kotlin"  
arr2[2] = "项目"  
for (i in arr2) {  
 *print*("$i \t")  
}

最终打印结果：第一个 Kotlin 项目

### 工厂函数（Array()）

### 原始类型数组

Kotlin还有专门的类来表示原始类型的数组，没有装箱开销，它们分别是：

1. ByteArray => 表示字节型数组
2. ShortArray => 表示短整型数组
3. IntArray => 表示整型数组
4. LongArray => 表示长整型数组
5. BooleanArray => 表示布尔型数组
6. CharArray => 表示字符型数组
7. FloatArray => 表示浮点型数组
8. DoubleArray => 表示双精度浮点型数组

PS: Kotlin中不支持字符串类型这种原始类型数组，可以看源码Arrays.kt这个类中并没有字符串数组的声明。而源码中StringArray.kt这个类并不是声明字符串型数组的。

var intArr = *intArrayOf*(2,4,6,8)  
var longArr = *longArrayOf*(100L,300L,56L)  
var charArr = *charArrayOf*('a','b','c','h')