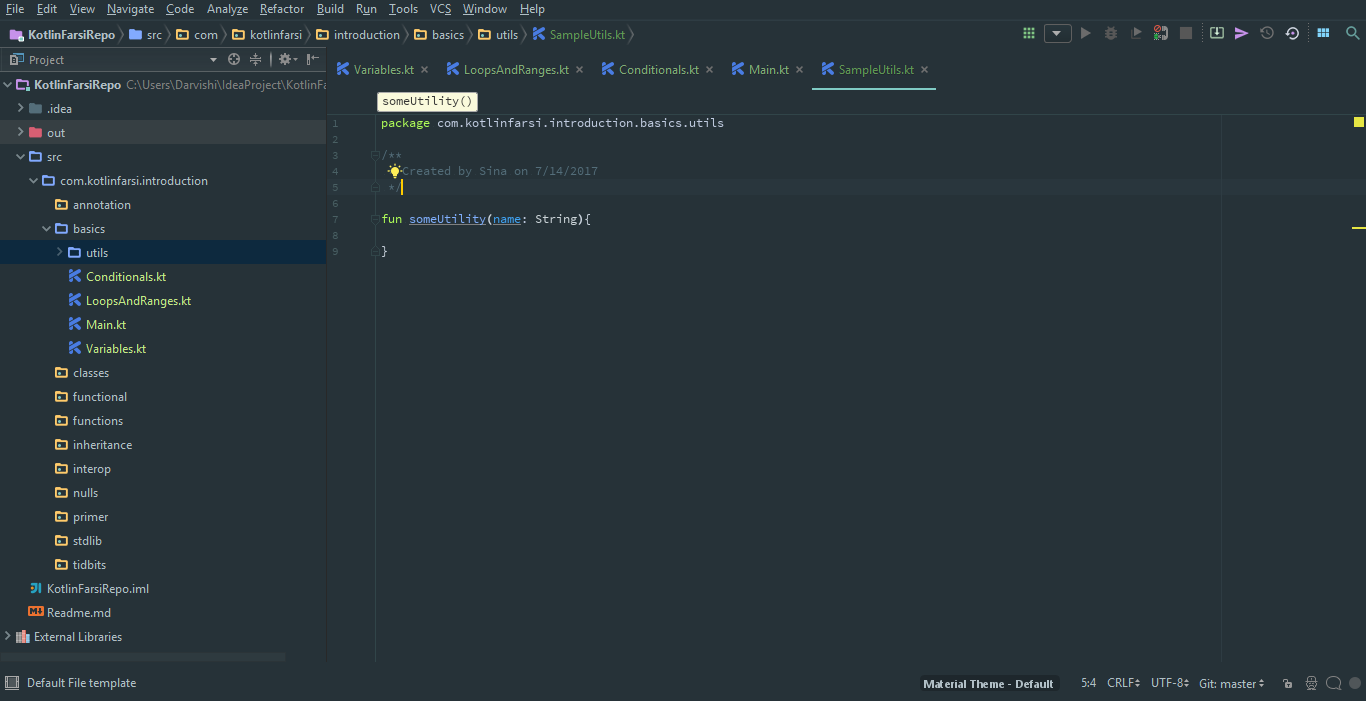
خب ­­­اگه تموم مثال هایی که زدیم رو دیده باشین متوجه شدین که ما تقریبا به یک پکیج محدود بودیم و ما داخل اون پکیج کار میکردیم.مثلا اکه همون فایل Main که دفعه اول استفاده کردیم رو نگاه کنیم متوجه خواهیم شد که ما از تابعی به نام println() استفاده کردیم . این تابع از کجا آمد؟ چجوری ازش استفاده کردیم ؟

اگه روی ویندوز هستید و دکمه Ctrl رو نگه دارید و روی تابع کلیک کنین به سمت تعریف اون تابع هدایت میشین. و اگر به ابتدای اون فایل برین متوجه میشین که این تابع داخل بسته ای به نام kotlin.ioهستش. پکیج ها به مانند جاوا و دیگر زبون ها میتونن import بشن ولی در واقع نیازی به import کردن توابعی مثل println() و به مانند این ها نیست، در واقع اونها به طور ضمنی import شدن ولی من میتونم پکیج های دیگه رو از جاهای دیگه ی برنامم import کنم. به عنوان مثال بیاین و داخل همون پکیج basic یک پکیج دیگه به نام utils باز کنیم. و داخل اون هم یک فایل به نام SampleUtils باز کنیم.

حالا بیایم و یک تابع به نام SomeUtility() در اون باز کنیم که یک متغییر name از جنس String میگیره.

خب حالا اگر به فایل Main برگردم میتونم با استفاده از کلیدواژه import ابتدا فایل SampleUtils رو اضافه کنم و بعدش از توابع داخلش استفاده کنم. توجه کنین که اگر هم ابتدا اون فایل رو import نکنم و مستقیم بنویسم SomeUtility خود intelliJ برام فایل رو import میکنه و یا از من سوال میپرسه که import اش کنم یا نه.

package com.kotlinfarsi.introduction.basics  
  
import com.kotlinfarsi.introduction.basics.utils.someUtility  
  
fun main**(**args: Array<String>**)** {  
 println**(**"Hello World!"**)** someUtility**(**"Some Name"**)**}

ما میتونیم به جای این که مشخص کنیم از چه فایلی میخوایم دقیقا استفاده کنیم بگیم از کل فایل های داخل این پکیج استفاده کن و اون رو با گذاشتن \* به جای someUtility انجام میدیم.

import com.kotlinfarsi.introduction.basics.utils.\*

همچنین اومدیم و فایل رو import کردیم و متوجه شدیم فایلی که import شده هم نام با یکی از فایل های دیگمونه،میتونیم به این روش جور دیگه ای ازش استفاده کنیم

package com.kotlinfarsi.introduction.basics  
  
import com.kotlinfarsi.introduction.basics.utils.someUtility as someAditionalFunctions  
  
fun main**(**args: Array<String>**)** {  
 println**(**"Hello World!"**)** someAditionalFunctions**(**"Some Name"**)**}

خلاصه بخش 3

1. تا اینجا یاد گرفتیم که چگونه یک متغیر رو تغییرپذیر یا تغییر ناپذیر کنیم
2. با حلقه for اشنا شدیم و فهمیدیم میتونه روی یک کالکشن هم راه بره
3. و همچنین یاد گرفتیم که if و when میتونن به صورت یک عبارت منطقی به عنوان نتیجه برای یک متغییر استفاده بشه