



# آموزش برنامه نویسی اندروید در محیط اندروید استودیو

متریال دیزاین

بخش یازدهم: کار با Bottom Navigation

مدرس : سید مهدی مطهری

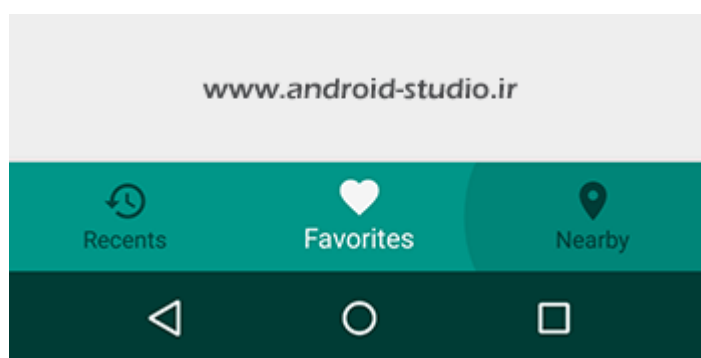
[www.android-studio.ir](http://www.android-studio.ir)



## به نام خدا

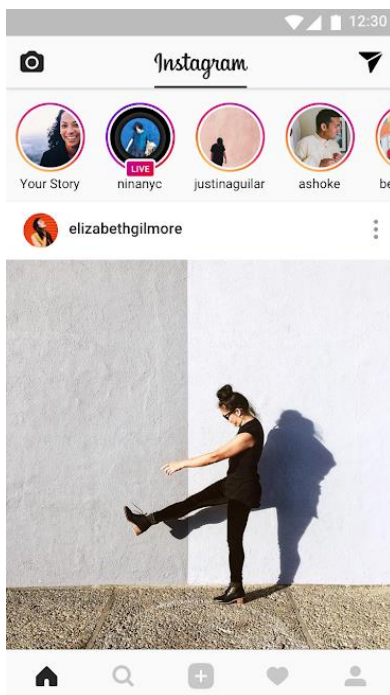
### معرفی Bottom Navigation:

Bottom Navigation یکی دیگر از کامپوننت‌های متریالی اندروید است که کاری مشابه Tabs انجام می‌دهد. این کامپوننت در پایین صفحه قرار می‌گیرد و برای نمایش ۳ الی ۵ گزینه مناسب است.



در استفاده از این کامپوننت باید یکسری موارد را مدنظر قرار داد. از جمله:

- از Bottom Navigation عموماً برای نمایش صفحات و قسمت‌های مهم اپلیکیشن استفاده می‌شود. صفحاتی که به لحاظ اهمیت دارای سطح و تراز یکسانی هستند و کاربر بیشترین سروکار را با آنها دارد. اینستاگرام را در نظر بگیرید:



در نوار پایین صفحه ۵ آیتم قرار گرفته که جزء پرکاربردترین صفحات این اپ است و کاربر به راحتی به آنها دسترسی دارد.

- برخلاف Tabs در اینجا تعداد آیتمها باید محدود بوده (بین ۳ تا ۵) تا نیاز به اسکرول بین آیتمها نباشد. همچنین استفاده از آن برای کمتر از ۳ آیتم هم پیشنهاد نمی‌شود. بنابراین اگر اپ شما فقط دو صفحه مهم دارد، استفاده از این ویجت راه حل مناسب نیست (بجای آن از Tabs و یا Navigation Drawer استفاده کنید).
- در صورت استفاده از text label (متن) به همراه آیکون، تا حد امکان متن کوتاه انتخاب شود تا در یک خط قرار گیرد.
- استفاده همزمان از Tabs در بالای صفحه و Bottom Navigation در انتهای صفحه ممکن است باعث سردرگمی کاربر شود بنابراین توصیه نمی‌شود.
- اندازه متن و آیکون همه آیتمها یکسان تعیین شود.
- از بکار بردن رنگهای متفاوت برای آیکون یا متن آیتمها اجتناب کنید. رابط کاربری هرچقدر یکدست و ساده تر باشد برای کاربر تجربه بهتری به همراه دارد.

در گذشته برای پیاده سازی یک Bottom Navigation می‌بایست از کتابخانه‌های متفرقه موجود در سطح وب استفاده می‌شد. یا اینکه توسعه دهنده شخصا این قابلیت را از پایه بنویسد. اما در نسخه ۲۵ کتابخانه Support Design این کامپوننت نیز به صورت رسمی معرفی و اضافه شد.



یک پروژه جدید با نام BottomNavigation با یک Empty Activity ایجاد می‌کنم.

در اولین قدم کتابخانه Design را به پروژه اضافه می‌کنم:

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
    implementation 'com.android.support:appcompat-v7:27.1.1'  
    implementation 'com.android.support:design:27.1.1'  
}
```

سپس مطابق کد زیر یک BottomNavigationView به Layout اضافه می‌کنم:

:activity\_main.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context=".MainActivity">  
  
    <android.support.design.widget.BottomNavigationView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:id="@+id/bottom_nav"  
        android:layout_alignParentBottom="true" />  
  
</RelativeLayout>
```

با استفاده از خاصیت layout\_alignParentBottom ویجت در انتهای صفحه قرار می‌گیرد.

مانند **Navigation Drawer** که در مباحث گذشته آشنا شدیم، در اینجا هم برای نمایش آیتم‌ها از Menu استفاده می‌کنیم. در مسیر res > New > Android Resource File یک فایل با نام دلخواه bottom\_navigation.xml و از نوع Menu به پروژه اضافه می‌کنم که در نهایت یک فولدر با نام menu و حاوی این فایل به فولدر res اضافه می‌شود.



سه آیتم به صورت زیر داخل menu تعریف می‌کنم:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item
        android:title="Home"
        android:id="@+id/item_1"
        android:icon="@drawable/home" />

    <item
        android:title="Account"
        android:id="@+id/item_2"
        android:icon="@drawable/account"/>

    <item
        android:title="Settings"
        android:id="@+id/item_3"
        android:icon="@drawable/settings"/>

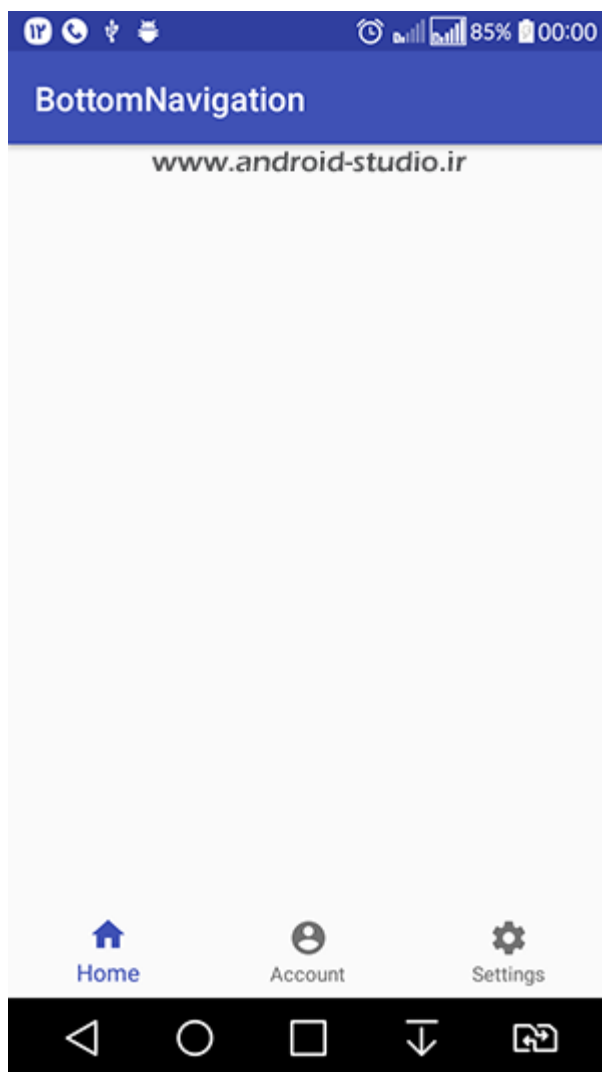
</menu>
```

هر آیتم شامل یک title، id و icon است که از آیکون‌های پیش فرض اندروید استودیو در مسیر  
res > New > Vector Asset استفاده کرده‌ام.

حالا منو را به BottomNavigationView اضافه می‌کنم:

```
app:menu="@menu/bottom_navigation"
```

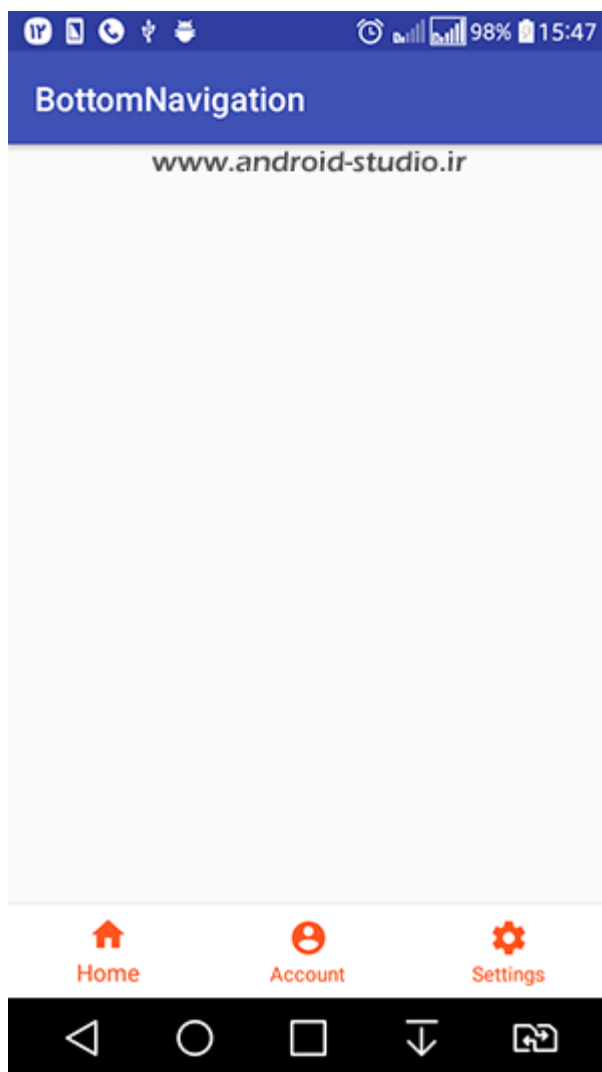
پروژه را اجرا می‌کنم:



چند خاصیت هم برای customize کردن Bottom Navigation در اختیار داریم:

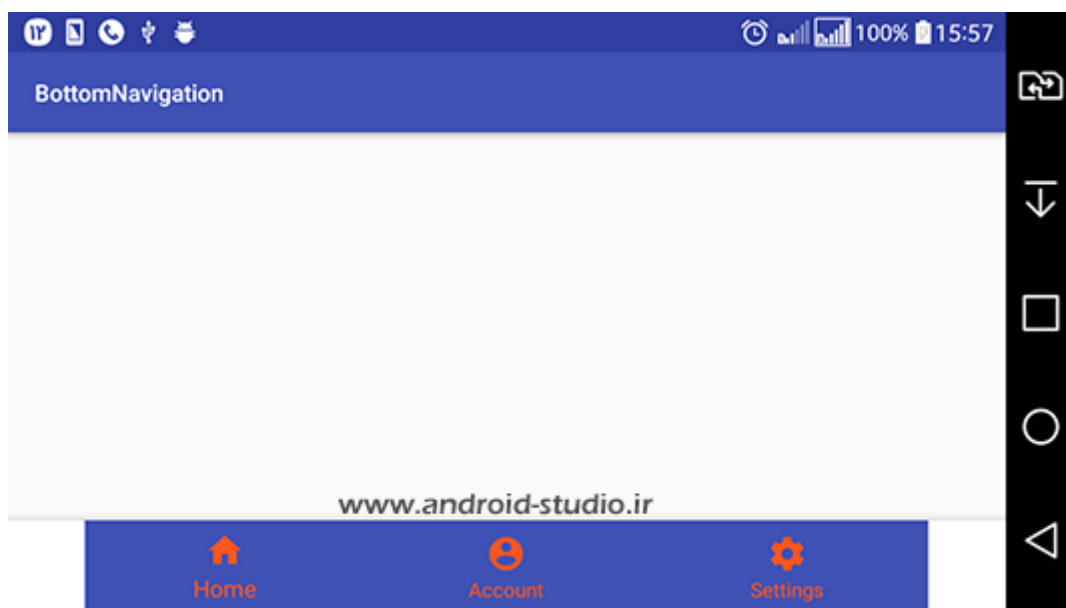
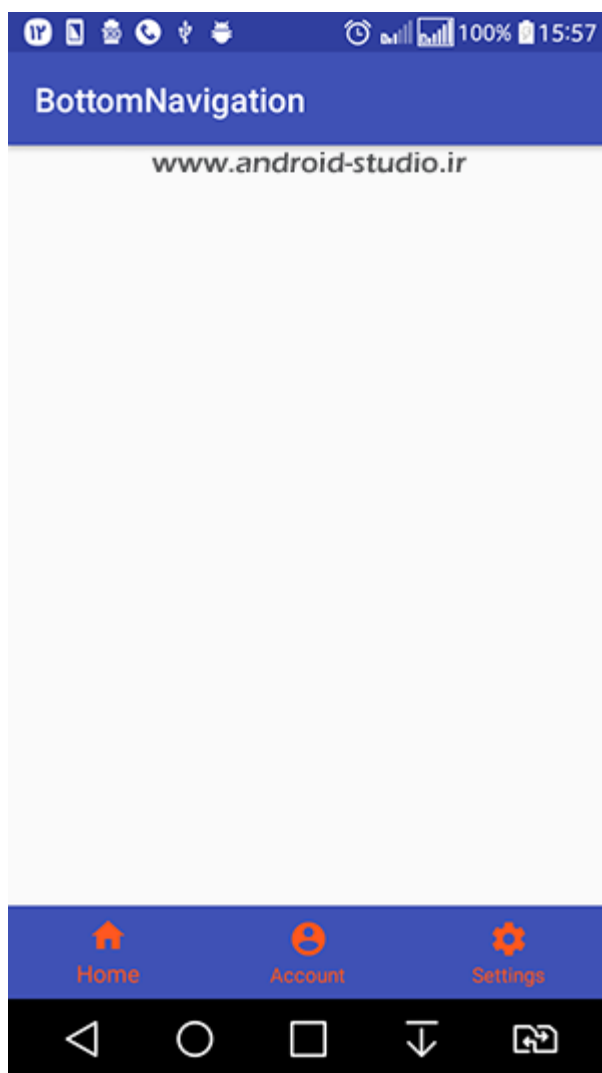
```
<android.support.design.widget.BottomNavigationView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/bottom_nav"
    android:layout_alignParentBottom="true"
    app:menu="@menu/bottom_navigation"
    android:background="#ffffff"
    app:itemTextColor="#FF5722"
    app:itemIconTint="#FF5722"/>
```

خاصیت background که کاربرد آن واضح است و رنگ پس زمینه کامپوننت را مشخص می کند. دو مورد بعد یعنی itemTextColor و itemIconTint برای تعیین رنگ متن و آیکون استفاده می شوند:



با تعیین رنگ پس زمینه، shadow نیز به بالای نوار اضافه می‌شود. علاوه بر موارد فوق، خاصیت `itemBackground` هم داریم که برای تعیین رنگ پس‌زمینه آیتم‌ها بکار می‌رود. برای فهم بهتر، این خاصیت را به کامپوننت اضافه کرده و در دو حالت قرارگیری دیوایس نمایش می‌دهم:

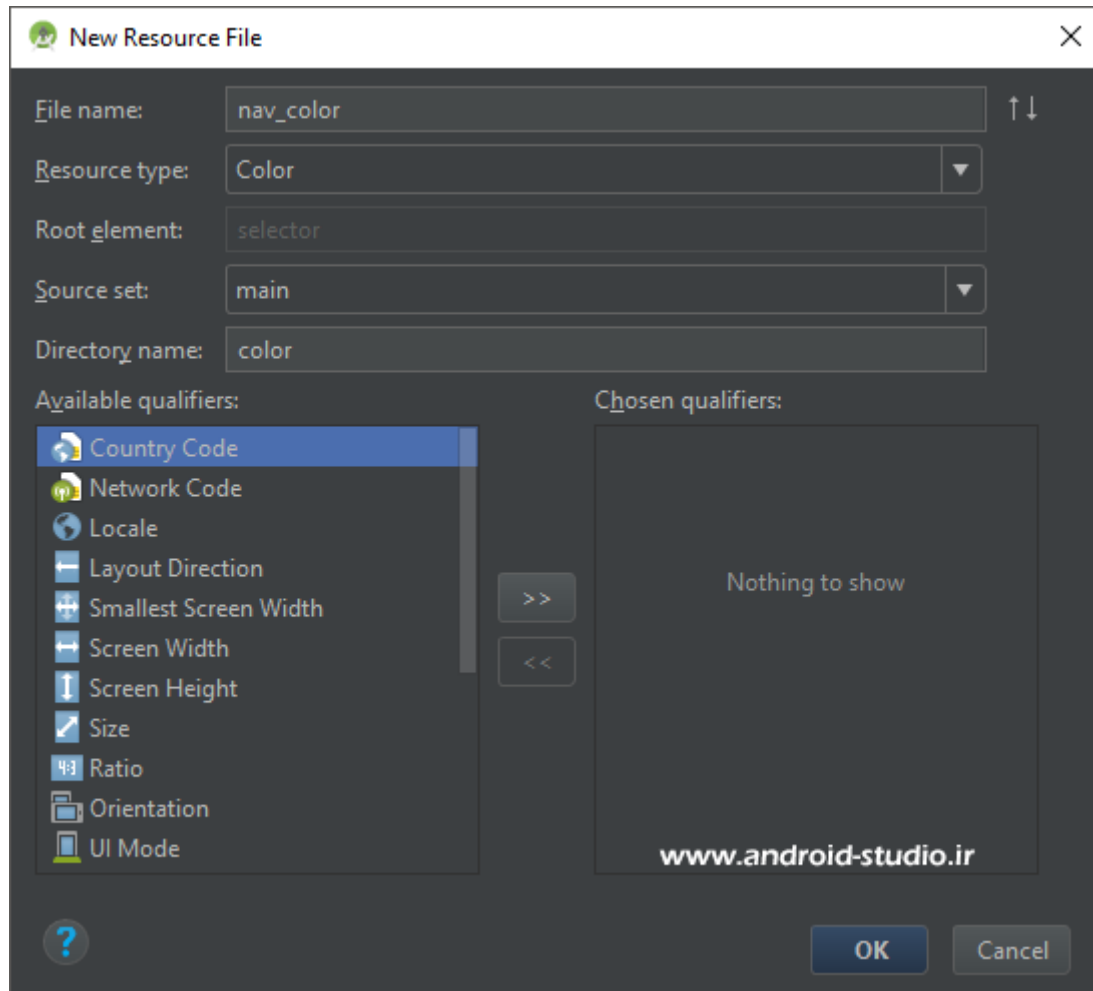
```
app:itemBackground="@color/colorPrimary"
```







با تعریف `itemIconTint` و `itemTextColor` رنگ متن و آیکون هر سه آیتم یکسان است. در صورتی که بخواهیم رنگ آیتم انتخاب شده (`checked`) از سایر آیتمها مجزا باشد، به صورت زیر عمل می‌کنیم: ابتدا روی `res` راست کلیک کرده، در مسیر `New > Android Resoure File` یک فایل `xml` از نوع `Color` به پروژه اضافه می‌کنم:



**تذکر:** با تایید پنجره فوق، یک فولدر جدید با نام `color` به پروژه اضافه می‌شود.

فایل را به اینصورت تکمیل می‌کنم:

`:nav_color.xml`

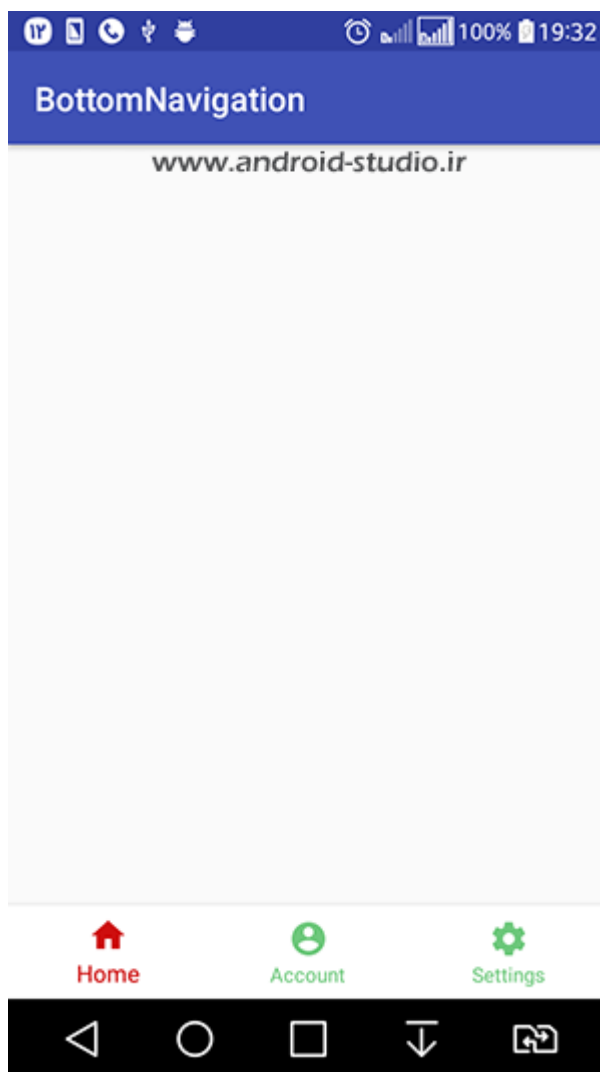
```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_checked="true" android:color="#ce0e0e" />
    <item android:state_checked="false" android:color="#6fc97d"/>
</selector>
```



در کد فوق دو آیتم با خاصیت state\_checked و مقادیر true و false تعریف شده که هرکدام یک رنگ را به خود اختصاص داده.

حالا برای دو خاصیت مربوط به رنگ متن و آیکون، فایل فوق را تعریف می‌کنم:

```
<android.support.design.widget.BottomNavigationView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/bottom_nav"
    android:layout_alignParentBottom="true"
    app:menu="@menu/bottom_navigation"
    android:background="#ffffff"
    app:itemTextColor="@color/nav_color"
    app:itemIconTint="@color/nav_color" />
```



ملاحظه می‌کنید آیتمی که در حالت انتخاب شده قرار دارد رنگ مربوط به مقدار true

```
<item android:state_checked="true" android:color="#ce0e0e" />
```



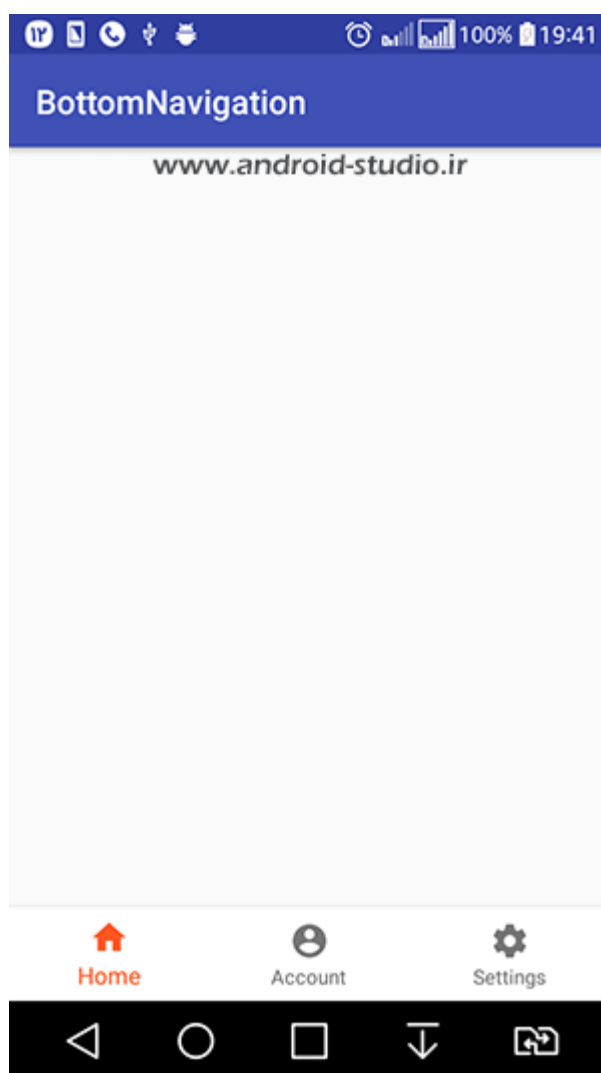
و سایر آیتم‌ها رنگ خط زیر (یعنی مقدار false) را نشان می‌دهند:

```
<item android:state_checked="false" android:color="#6fc97d"/>
```

آیتم انتخاب شده، رنگ پیش فرض خود را از colorPrimary تم متریال می‌گیرد بنابراین اگر فقط قصد تغییر رنگ آن را داشته باشیم، کافیهست به جای مراحل قبل، یک style جدید تعریف کرده، مقدار colorPrimary را جایگزین کنیم:

```
<style name="BottomNavTheme">
    <item name="colorPrimary">#FF5722</item>
</style>
```

```
<android.support.design.widget.BottomNavigationView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/bottom_nav"
    android:layout_alignParentBottom="true"
    app:menu="@menu/bottom_navigation"
    android:background="#ffffff"
    android:theme="@style/BottomNavTheme" />
```



مرحله نهایی، هندل کردن آیتم‌هاست که با انتخاب هر آیتم چه عملی انجام شود. من یک مثال ساده پیاده سازی می‌کنم که با انتخاب هر آیتم، یک text متناسب با آن در وسط صفحه نمایش داده شود.



یک TextView به لایه اضافه می‌کنم:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="TextView" />

    <android.support.design.widget.BottomNavigationView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/bottom_nav"
        android:layout_alignParentBottom="true"
        app:menu="@menu/bottom_navigation"
        android:background="#ffffff"
        app:itemTextColor="@color/nav_color"
        app:itemIconTint="@color/nav_color" />

</RelativeLayout>
```

در ادامه دو ویجت Bottom Navigation و TextView را داخل اکتیویتی تعریف کرده و سپس متد Listener آنرا می‌نویسم:

```
bottomNav.setOnNavigationItemSelectedListener(new Bc() ;
```

```
);
```



## :MainActivity.java

```
package ir.android_studio.bottomnavigation;

import android.support.annotation.NonNull;
import android.support.design.widget.BottomNavigationView;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.MenuItem;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    BottomNavigationView bottomNav;
    TextView itemText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        bottomNav = findViewById(R.id.bottom_nav);
        itemText = findViewById(R.id.text);

        bottomNav.setOnNavigationItemSelectedListener(new
        BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                return false;
            }
        });
    }
}
```

از متد `setOnNavigationItemSelectedListener` برای تعیین عملکرد آیتم‌ها استفاده می‌شود.

متد را با استفاده از یک `switch case` تکمیل می‌کنم:

```
bottomNav.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {

        switch (item.getItemId()) {

            case R.id.item_1:
                itemText.setText("Home Item");
                return true;

            case R.id.item_2:
                itemText.setText("Account Item");
                return true;

            case R.id.item_3:
                itemText.setText("Settings Item");
                return true;

        }
    }
})
```



```
        return false;
    }
});
```

در switch فوق، تعیین شده تا با دریافت id مربوط به هرکدام از آیتمها، چه متنی در TextView نمایش داده شود.

(تصویر bottom\_navigation.gif در پوشه Gif را مشاهده کنید)

یک آیتم دیگر به Menu اضافه می‌کنم:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

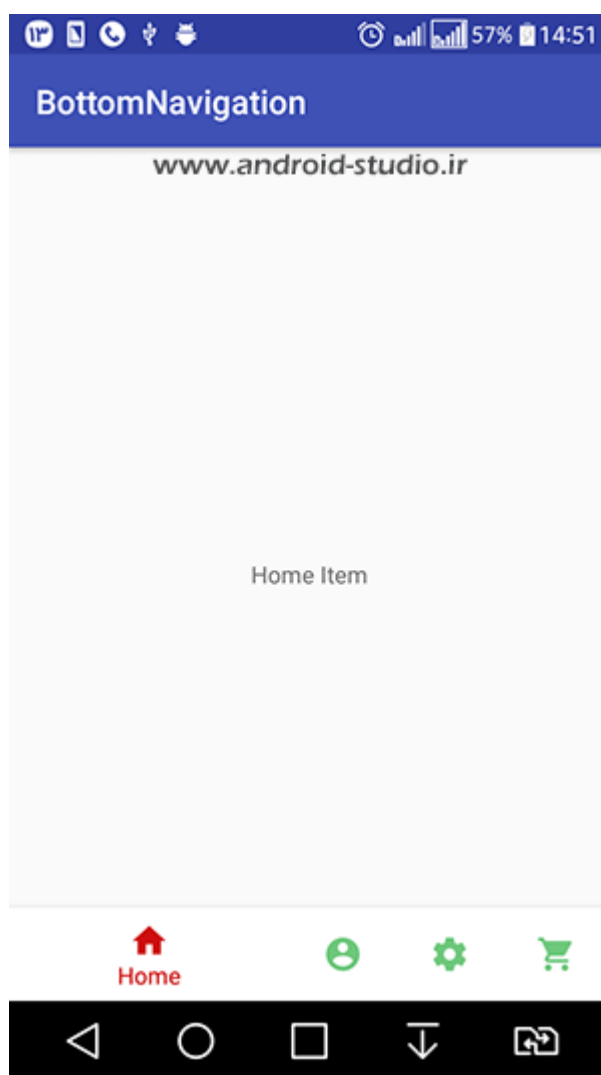
    <item
        android:title="Home"
        android:id="@+id/item_1"
        android:icon="@drawable/home" />

    <item
        android:title="Account"
        android:id="@+id/item_2"
        android:icon="@drawable/account"/>

    <item
        android:title="Settings"
        android:id="@+id/item_3"
        android:icon="@drawable/settings"/>

    <item
        android:title="Premium"
        android:id="@+id/item_4"
        android:icon="@drawable/shop"/>

</menu>
```



ملاحظه می‌کنید در صورتی که آیتم‌ها بیش از سه مورد باشد، فقط لیبل آیتم انتخاب شده نمایش داده می‌شود و مابقی فقط آیکون. در اکثر موارد توسعه دهنده تمایل دارد لیبل تمام آیتم‌ها نمایش داده شود نه فقط آیتم انتخاب شده.

برای یافتن راه حل، عبارت زیر را جستجو کردم:

How to show more than 3 items in bottomNavigationView

با مطالعه پرسش و پاسخ‌های موجود در [stackoverflow.com](https://stackoverflow.com) متوجه شدم باید قابلیت shiftMode را غیر فعال کنم که کد آن به صورت زیر است:





```
private void disableShiftMode(BottomNavigationView view) {
    BottomNavigationMenuView menuView = (BottomNavigationMenuView) view.getChildAt(0);
    try {
        Field shiftingMode = menuView.getClass().getDeclaredField("mShiftingMode");
        shiftingMode.setAccessible(true);
        shiftingMode.setBoolean(menuView, false);
        shiftingMode.setAccessible(false);
        for (int i = 0; i < menuView.getChildCount(); i++) {
            BottomNavigationView item = (BottomNavigationView) menuView.getChildAt(i);
            item.setShiftingMode(false);
            // set once again checked value, so view will be updated
            item.setChecked(item.getItemData().isChecked());
        }
    } catch (NoSuchFieldException e) {
        Log.e("BNVHelper", "Unable to get shift mode field", e);
    } catch (IllegalAccessException e) {
        Log.e("BNVHelper", "Unable to change value of shift mode", e);
    }
}
```

نیازی نیست وارد جزئیات متد بالا شوید. من هم درک کاملی از عملکرد کد ندارم و صرفاً از آن استفاده می‌کنم.

متد را به اکتیویته اضافه کرده سپس در متد onCreate آنرا فراخوانی می‌کنم:

```
package ir.android_studio.bottomnavigation;

import android.support.annotation.NonNull;
import android.support.design.internal.BottomNavigationView;
import android.support.design.internal.BottomNavigationMenuView;
import android.support.design.widget.BottomNavigationView;
import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.widget.TextView;

import java.lang.reflect.Field;

public class MainActivity extends AppCompatActivity {

    BottomNavigationView bottomNav;
    TextView itemText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        bottomNav = findViewById(R.id.bottom_nav);
        itemText = findViewById(R.id.text);

        disableShiftMode(bottomNav);

        bottomNav.setOnNavigationItemSelectedListener(new
        BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
```



```

        switch (item.getItemId()) {

            case R.id.item_1:
                itemText.setText("Home Item");
                return true;

            case R.id.item_2:
                itemText.setText("Account Item");
                return true;

            case R.id.item_3:
                itemText.setText("Settings Item");
                return true;

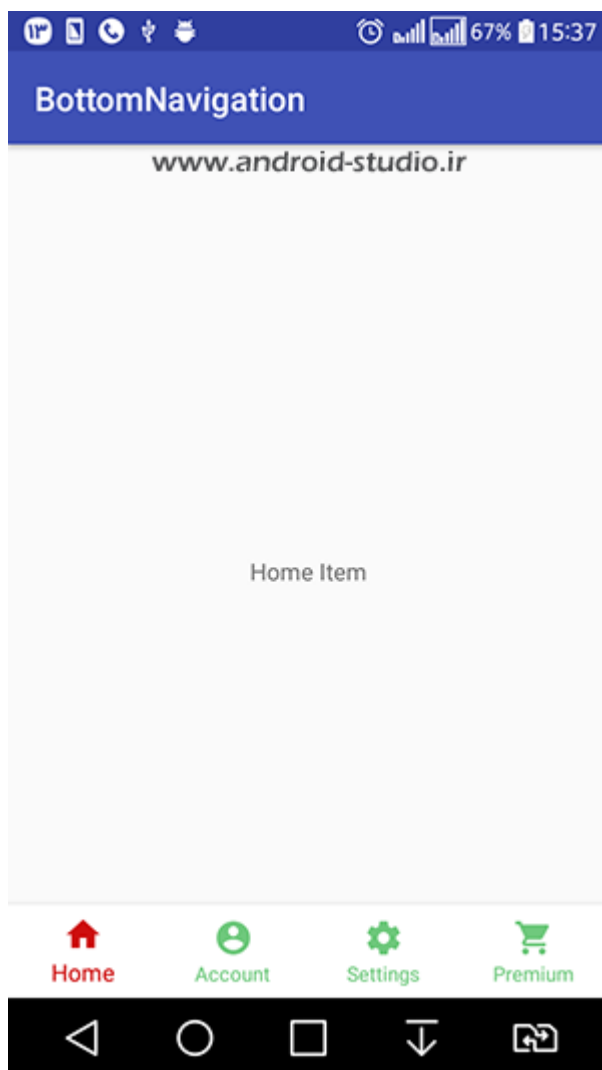
            case R.id.item_4:
                itemText.setText("Premium Item");
                return true;

        }

        return false;
    }
});
}

// Method for disabling ShiftMode of BottomNavigationView
private void disableShiftMode(BottomNavigationView view) {
    BottomNavigationMenuView menuView = (BottomNavigationMenuView) view.getChildAt(0);
    try {
        Field shiftingMode = menuView.getClass().getDeclaredField("mShiftingMode");
        shiftingMode.setAccessible(true);
        shiftingMode.setBoolean(menuView, false);
        shiftingMode.setAccessible(false);
        for (int i = 0; i < menuView.getChildCount(); i++) {
            BottomNavigationViewItem item = (BottomNavigationViewItem)
menuView.getChildAt(i);
            item.setShiftingMode(false);
            // set once again checked value, so view will be updated
            item.setChecked(item.getItemData().isChecked());
        }
    } catch (NoSuchFieldException e) {
        Log.e("BNVHelper", "Unable to get shift mode field", e);
    } catch (IllegalAccessException e) {
        Log.e("BNVHelper", "Unable to change value of shift mode", e);
    }
}
}
}

```



تقریباً در تمام مواردی مانند مسئله فوق که به یک راه حل نیاز داریم، با یک جستجوی ساده می‌توان به راه حل رسید و لازم نیست خودمان اقدام به پرسش کنیم و چند ساعت و چند روز منتظر پاسخ باشیم. مشکل و سوال ما قبلاً سوالی افراد دیگری نیز بوده.

**تذکر:** پروژه فوق، با نام BottomNavigation در پوشه Exercises قرار دارد.



## منابع تکمیلی:

<https://material.io/design/components/bottom-navigation.html>

<https://developer.android.com/reference/android/support/design/widget/BottomNavigationView>

با ارائه انتقادات و پیشنهادات خود، ما را در ارائه آموزش های بهتر یاری فرمائید.

[www.android-studio.ir](http://www.android-studio.ir)