# آموزش برنامه نویسی اندروید

آموزش ساختن برنامه برای اندروید به زبان فارسی

شنبه, ۲۱ دی ۱۳۹۲، ۱۲:۲۹ ق.ظ

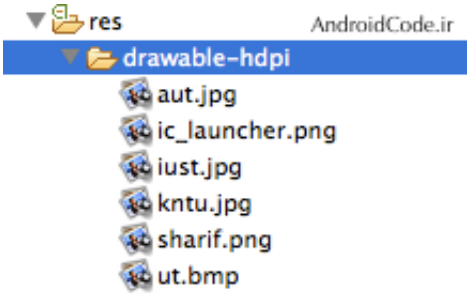
## فهرست‌های سفارشی (Customized Lists)

تماشای برخط [لینک مستقیم] فیلم آموزشی «فهرست‌های سفارشی»

زیر نویس انگلیسی [English Subtitle]

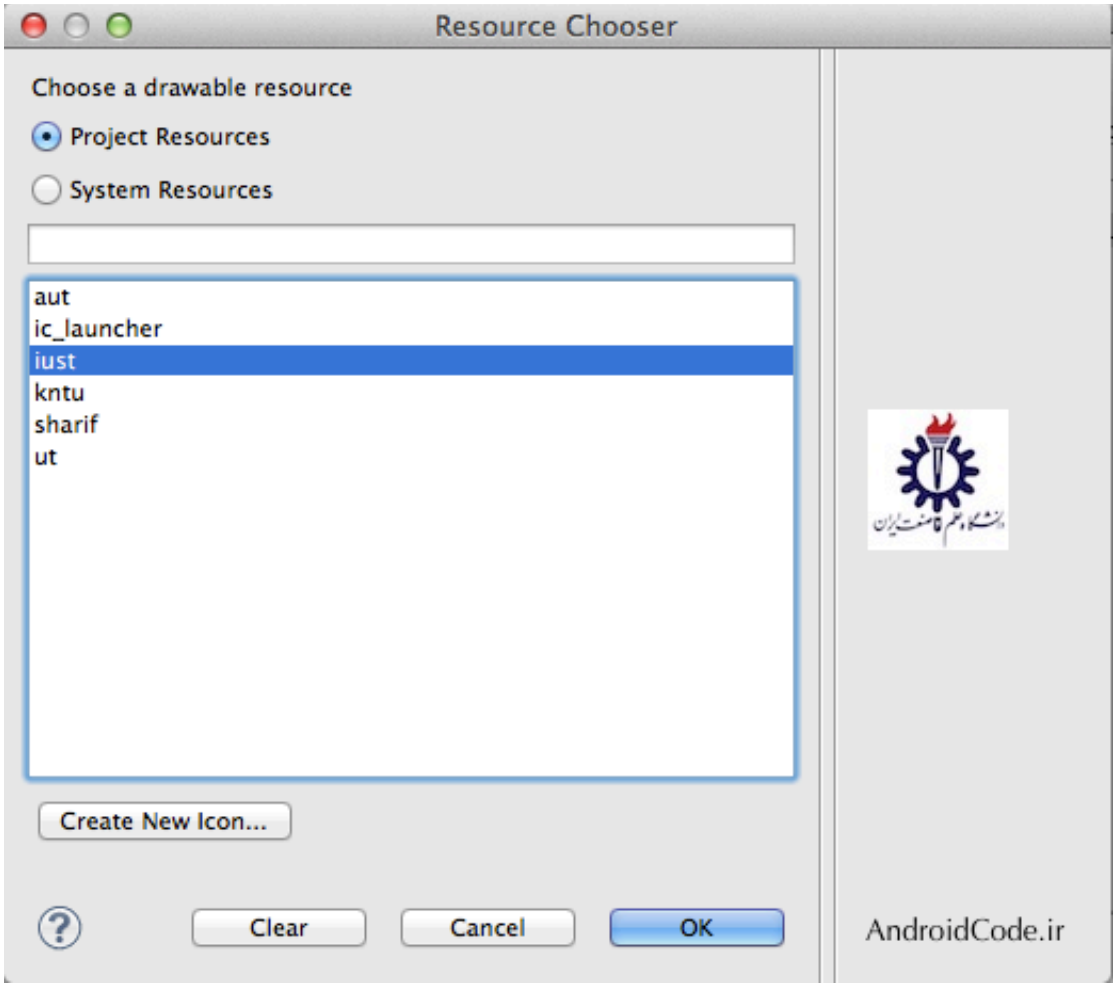
تو آموزش قبلی یاد گرفتیم که چطور یک فهرست ساده درست کنیم. حالا می‌خوایم یاد بگیریم که چطور اون فهرست رو سفارشی کنیم و از حالت سادگی درش بیاریم.

۱.مثل آموزش قبل یه پروژه با یک فهرست ساده بسازید در ادامه می‌خوایم برای هر ردیف از فهرستمون یه عکس نشان اون دانشگاه رو هم کنار اسمش بندازیم. پس اول از همه تصاویر مورد نظر رو به پروژه مون در زیرشاخه res و پوشه drawable-hdpi اضافه می‌کنیم، فایل عکس‌هایی که من استفاده کردم رو می‌تونید از اینجا دانلود کنید. وارد کردنش به پروژه هم کار آسونیه، هرجایی که عکسهاتون هست کپی کنید و بعد روی پوشه drawable-hdpi راست کلیک کنید و paste.



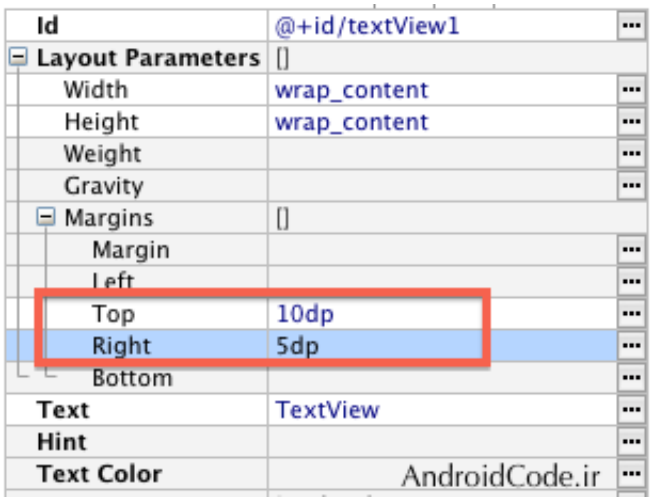

۲.اولین کاری که باید بکنیم اینه که یه طرح‌بندی برای هر ردیف از فهرست درست کنیم. تا با توجه به اون اندروید اطلاعتمون رو در هر ردیف بدونه چطور نمایش بده. پس احتیاج داریم تا یه فایل xml برای طرح‌بندی‌مون بسازیم. روی فولدر layout (در res) راست کلیک کنید. گزینه New و other رو انتخاب کنید. (این تصویر) بعد هم از زیرشاخه Android گزینه Android XML Layout File رو انتخاب کنید و براش اسم بذارید من اسمشو می‌ذارم redife_fehrest.

۳. خوب اولین چیزی که هر ردیف می‌خوایم داشته باشه یه تصویره، پس یه نمای‌تصویر (imageView) در فایل تازه ساخته شدمون می‌ذاریم. نمای‌تصویر رو می‌تونید از ستون سمت چپ از بخش Images & Media بکشید و روی صفحه رها کنید. همین که رهاش می‌کنید یه پنجره باز می‌شه که ازتون می‌خواد منبع عکس رو مشخص کنید، و اگه به درستی تصاویرتون رو وارد کرده باشید، می‌بینیدشون.




۴.هر ردیف قراره یه متن داشته باشه که نام دانشگاه درش نوشته بشه، پس یه نمای‌متنی (TextView) هم وارد صفحه می‌کنیم و کنار نمای‌تصویر می‌ذاریم. الان تصویرمون در کنار متن اینطوریه که به نظر خیلی مطلوب نمی‌یاد


می‌تونیم یه کم ویژگی‌های متن رو تغییر بدیم، مثه اندازه، فاصله از بالای صفحه و سمت راستش و ...




۵.به فایل MainActivity.java (در src) می‌ریم. در آموزش قبل یک آرایه‌ی‌وفق‌دهنده درست کرده بودیم:

```
setListAdapter (new ArrayAdapter <String> (this,
```

```
android.R.layout.simple_list_item_1,
getResources ().getStringArray(R.array.daneshgah)));
```

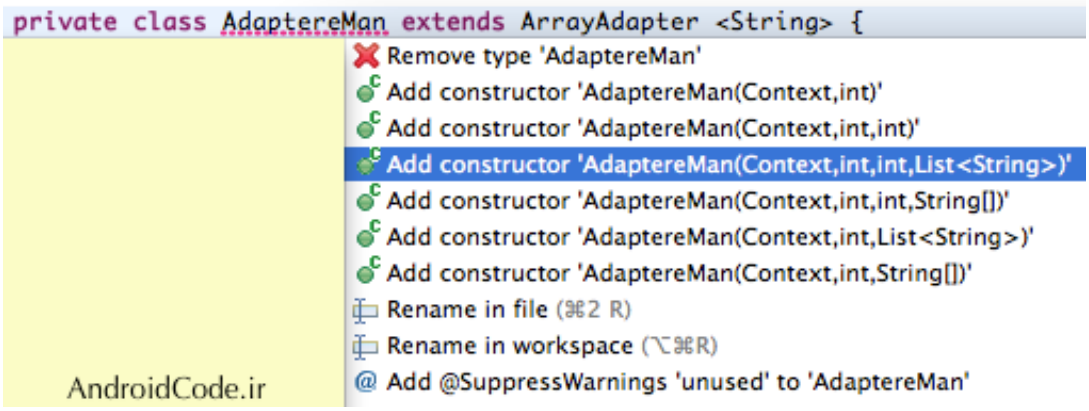
حالا باید برای این فهرست سفارشی‌مون خودمون کلاس وفق‌دهنده (Adapter) بسازیم. پس کلمه ArrayAdapter رو به AdaptereMan تغییر می‌دیم (می‌دونم که هنوز کلاسشو نساختیم). حالا زیر کلمه AdaptereMan خط قرمز می‌کشه و وقتی ctrl+1 رو بزنید بهتون پیشنهاد می‌ده که اون کلاس رو بسازید، ولی ما خودمون می خوایم اون کلاس رو بسازیم. اما همین کد قبلی رو یه تغییر دیگه هم باید بدیم و اون اینه که یه ورودی دیگه قبل از getResources اضافه کنیم و اون هم شناسه (Id) اون نمای‌متنی هست که قراره نوشته‌های هر ردیف درونش قرار بگیره. پس کد اینطوری می‌شه:

```
setListAdapter (new AdaptereMan <String> (this,
                                android.R.layout.simple_list_item_1, R.id.textView1,
                                getResources ().getStringArray(R.array.daneshgah)));
```

۶.می‌تونیم کلاس AdaptereMan رو توی همون کلاس اصلی‌مون بسازیم، البته اگه پروژه بزرگ باشه بهتره تو یه فایل جداگانه ساخته بشه ولی برای پروژه الان ما مشکلی ایجاد نمی‌کنه. در ضمن می‌خوایم که کلاسمون از ArrayAdapter توسعه پیدا کنه.

```
private class AdaptereMan extends ArrayAdapter <String> {
```

۷. وقتی کد بالا رو می‌نویسیم زیر نام کلاس خط قرمز می‌کشه و ازمون می‌خواد که یه سازنده برای کلاسمون ایجاد کنیم، اونی رو انتخاب می‌کنیم که ۴ تا ورودی به این ترتیب داشته باشه : محتوا، عدد صحیح (int)، عدد صحیح و فهرستی از نوع رشته (<list <string>)




وقتی روش می‌زنیم سازنده رو برامون ایجاد می‌کنه و ما تغییرش نمی‌دیم چون داره سازنده ArrayAdapter رو پیاده می‌کنه، کاری که ما می‌خوایم بکنیم اینه که تابع getView رو دوباره‌نویسی (override) کنیم.

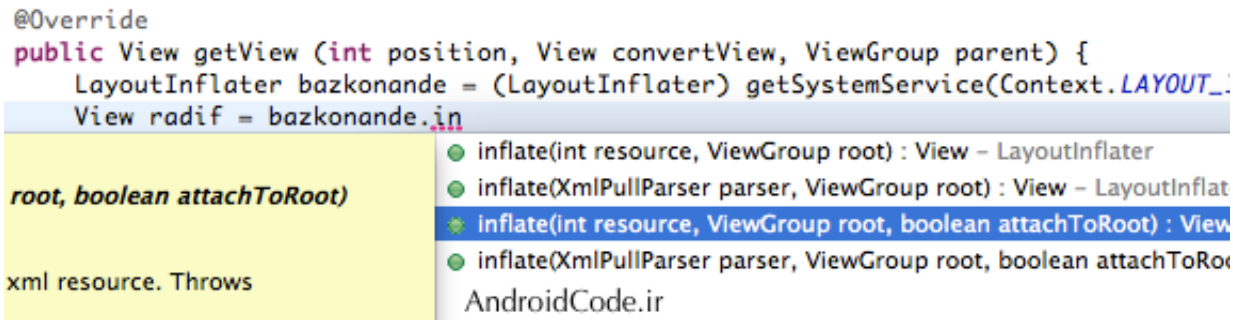
```
@Override
                public View getView (int makan, View convertView, ViewGroup parent)
                        // TODO Auto-generated method stub
                        return super.getView(makan, convertView, parent);
                }
```

۸.تابع getView برای هر ردیف از فهرست صدا زده می‌شه. و زمانی که صدا زده بشه ما این انتخاب رو داریم که برای هر ردیف در فهرستمون چه چیزی رو نشون بدیم. در این مثال ما می‌خوایم این تابع رو تغییر بدیم تا در کنار نام هر دانشگاه، نشان مربوط به همون دانشگاه قرار بگیره. باید به اون نمای‌تصویر و نمای متنی یه طوری دسترسی پیدا کنیم و راهش اینه که از کلاس بازکننده‌صفحه‌بندی (LayoutInflater) یه نمونه بگیریم و به عنوان ورودی بهش اون صفحه‌بندی که برای هر ردیف فهرست درست کردیم رو بدیم تا برامون بازش کنه و بتونیم تغییرش بدیم. پس اون خط توضیحات (comment) رو از کد بالا پاک می کنیم و به جاش

می‌نویسیم LayoutInflater bazkonande = حالا باید از یکی از خدمات سیستمی اندروید استفاده کنیم، به نام Context.LAYOUT_INFLATER_SERVICE.

```
LayoutInflater bazkonande = (LayoutInflater) getSystemService(Context.LAYOUT_INFLATE
```

۹.یه نمونه از کلاس نما (View) می‌سازیم تا اطلاعات بازشده از صفحه‌بندی بالا رو بهش بدیم و اون نمای‌تصویر و نمای‌متنی رو ازش بکشیم بیرون و مقداردهی‌اش کنیم. اسمش رو radif می‌ذاریم و مساوی با bazkonande.inflate.




این تابع سه ورودی می‌خواد، شماره منبع چیزی که می‌خوایم بازش کنیم (همون صفحه‌بندیه)، گروه‌نما (همون والد این کلاس) و یه متغیر درستی\نادرستی که الان می‌ذاریمش نادرست.

```
View radif = bazkonande.inflate(R.layout.radife_fehrest, parent, false);
```

۱۰.خوب برای اینکه خود فهرستمون رو داشته باشیم یه آرایه‌ی‌رشته‌ای محلی تو همین کلاس درست می کنیم و مقادیر اون فهرست رو می‌ریزیم توش. (برای راحتی کار با فهرست) اینطوری:

```
String [] radifha = getResources ().getStringArray(R.array.daneshgah);
```

حالا وقتشه که از اون نمای‌متنی و نمای‌تصویری مربوط به هر ردیف ارجاع بگیریم تا بتونیم تغییرشون بدیم. خیلی ساده مثل قبلنا:

```
ImageView tasvir = (ImageView) radif.findViewById(R.id.imageView1);
TextView matn = (TextView) radif.findViewById(R.id.textView1);
```

۱۱.متنی که نمای‌متنی باید نمایش بده رو باید از اون آرایه‌ی رشته‌ها در بیاریم. اینکه الان در مکان کدام عنصر از آرایه هستیم رو اولین ورودی تابع getView مشخص می‌کنه به اسم makan. پس:

```
matn.setText (radifha[makan]);
```

تصویری که نمای‌تصویر باید نمایش بده بر حسب نوشته است، یعنی یه سری عبارت شرطی می‌ذاریم و بررسی می‌کنیم نمای‌متنی چه عبارتی رو داره نشون می‌ده و با توجه به اون تصویر مناسب رو تو نمای‌تصویری بارگزاری می‌کنیم. اما چون متن ما فارسی بوده قبول نمی‌کنه که تو عبارت شرطی فارسی بنویسیم، یعنی این‌طوری:

```
if (radifha[makan].equals("دانشگاه صنعتی شریف"))
                tasvir.setImageResource(R.drawable.sharif);
```

به خاطر همین می‌تونیم با توجه به مکانی که اون عنصر آرایه داره تصویرش رو مشخص کنیم:

```
if (makan == 0)
        tasvir.setImageResource(R.drawable.kntu);
else if (makan == 1)
```

```
        tasvir.setImageResource(R.drawable.iust);
else if (makan == 2)
        tasvir.setImageResource(R.drawable.sharif);
else if (makan == 3)
        tasvir.setImageResource(R.drawable.aut);
else if (makan == 4)
        tasvir.setImageResource(R.drawable.ut);
```

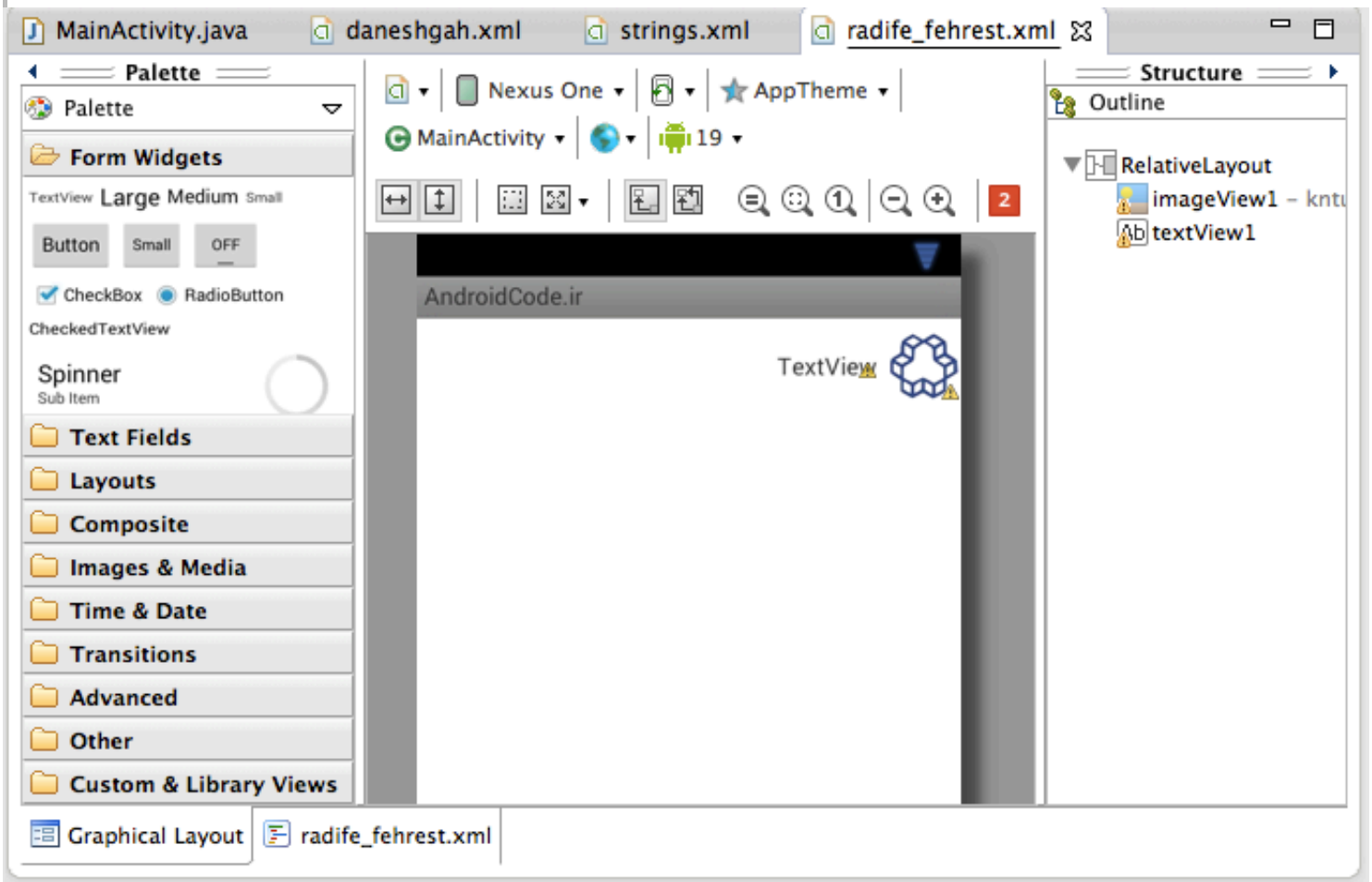
برای خروجی تابع هم فقط هم radif رو برگردونیم کافیه.

زیر AdaptereMan که اول استفاده کردیم یه خط قرمز کشیده، روش موس رو نگه می‌داریم و گزینه remove type argument رو می‌زنیم، بعد زیر کل ورودی‌ها خط قرمز می‌کشه، موس رو روش نگه می‌داریم و گزینه change constructor رو می‌زنیم. پس کلا کلاسمون این‌طوری می‌شه:

```
private class AdaptereMan extends ArrayAdapter <String> {

        public AdaptereMan(Context context, int resource,
                        int textViewResourceId, String[] strings) {
                super(context, resource, textViewResourceId, strings);
                // TODO Auto-generated constructor stub
        }

        @Override
        public View getView (int makan, View convertView, ViewGroup parent) {
        LayoutInflater bazkonande = (LayoutInflater) getSystemService(Context.LAYOUT
        View radif = bazkonande.inflate(R.layout.radife_fehrest, parent, false);
        String [] radifha = getResources ().getStringArray(R.array.daneshgah);

        ImageView tasvir = (ImageView) radif.findViewById(R.id.imageView1);
        TextView matn = (TextView) radif.findViewById(R.id.textView1);

        matn.setText (radifha[makan]);

        if (makan == 0)
                tasvir.setImageResource(R.drawable.kntu);
        else if (makan == 1)
                tasvir.setImageResource(R.drawable.iust);
        else if (makan == 2)
                tasvir.setImageResource(R.drawable.sharif);
        else if (makan == 3)
                tasvir.setImageResource(R.drawable.aut);
        else if (makan == 4)
                tasvir.setImageResource(R.drawable.ut);

        return radif;
        }
}
```

وقتی برنامه رو تو شبیه‌ساز دیدم این شکلی بود


به نظرم اومد بهتره برای radife_fehrest.xml از صفحه‌بندی نسبی استفاده کنم تا بتونم عکس و متن رو راست‌چین کنم به این صورت:




تغییراتی در ویژگی‌هاشون دادم:

نمای‌متنی: textSize:14sp - margin:top:20dp - margin:right:5dp

نمای‌تصویر: margin:top:5dp

که نتیجه این شد:


نوشته شده توسط وجدانی

بلاگ بیان، رسانه متخصصان و اهل قلم

# آموزش برنامه نویسی اندروید

آموزش ساختن برنامه برای اندروید به زبان فارسی

- سوالات متداول
- دانلود JDK و SDK برای ایرانیان
- جدول محتوا
- خانه