

## نمایش دسته‌جمعی داده‌ها



یکی از مهم‌ترین المان‌های نمایش اطلاعات در دستگاه‌های همراه ListView است. این المان به شما کمک می‌کند تا اطلاعات خود را فهرستوار نمایش دهید.

کارکردن با ListView در اندروید

نمایش دسته‌جمعی داده‌ها

جام جم آنلاین: یکی از مهم‌ترین المان‌های نمایش اطلاعات در دستگاه‌های همراه ListView است. این المان به شما کمک می‌کند تا اطلاعات خود را فهرستوار نمایش دهید.

برای شروع فایل Layout مربوط به پروژه‌تان را باز کنید، سپس از پنل سمت چپ به قسمت Composite بروید و یک ListView به فایل خود اضافه کنید. کد تولید شده برای ListView به صورت زیر است:

```

ListView;#171&
"android:id="@+id/mylist
"android:layout_width="match_parent
« "android:layout_height="wrap_content
«ListView;/;#171&

```

برای اضافه کردن یکسری از المان‌ها ابتدا باید توسط الگوی findViewById شیء متناظر با آن ListView را که در فایل Layout ایجاد شده بازبازی سپس یک آرایه از رشته‌های متنی به صورت زیر ایجاد می‌کنیم:

```

;(ListView listView = (ListView) findViewById(R.id.mylist
,"String[] values = new String[] { "Android", "iPhone", "WindowsMobile
,"Blackberry", "WebOS", "Ubuntu", "Windows7", "Max OS X"
}; {"Linux", "OS/2"

```

در این مرحله به ازای هر رشته متنی موجود در values یک سطر به ListView اضافه کنیم. برای این کار به یک ArrayAdapter نیاز داریم، که به شیوه زیر است:

```

,ArrayAdapter<String> adapter = new ArrayAdapter<String>(this
;(android.R.layout.simple_list_item_1, android.R.id.text1, values
پارامترهای ورودی سازنده کلاس ArrayAdapter را به این صورت توضیح می‌دهیم:

```

(ArrayAdapter(Context context, int resource, int textViewResourceId, String[] objects) یک کلاس به نام MySimpleArrayAdapter ایجاد کرده سپس سازنده آن را به صورت زیر مشخص می‌کنیم:

```

} (public MySimpleArrayAdapter(Context context, String[] values
;(super(context, R.layout.rowlayout, values
;this.context = context
{ ;this.values = values

```

ما یک Layout برای هر سطر تعریف کرده‌ایم که با کد منحصر به فرد R.layout.rowlayout مشخص شده است. ابتدا با فراخوانی متد super سازنده کلاس پدر را که همان ArrayAdapter است مشخص کرده و در آخر با بازنویسی متد getView، نحوه نمایش هر سطر را تعیین می‌کنیم.

Override@

```

public View getView(int position, View convertView, ViewGroup parent) {
LayoutInflator inflater =
(LayoutInflator) context
;(getService(Context.LAYOUT_INFLATER_SERVICE.
View rowView = inflater.inflate(R.layout.rowlayout, parent, false);
TextView textView = (TextView)
;(rowView.findViewById(R.id.label));
ImageView imageView = (ImageView) rowView.findViewById(R.id.icon

```

```

        ;[textView.setText(values[position
Change the icon for Windows and iPhone //
        ];String s = values[position
        } ((if (s.startsWith("iPhone
;imageView.setImageResource(R.drawable.no
        } else {
;imageView.setImageResource(R.drawable.ok
        {
        ;return rowView
        {

```

در خط اول ما یک شیء از کلاس LayoutInflater ایجاد می‌کنیم؛ این کلاس برای برابرکردن یک فایل XML که شامل یک Layout است با یک شیء View متناظر با آن مورد استفاده قرار می‌گیرد.

```

;View rowView = inflater.inflate(R.layout.rowlayout, parent, false
rowlayout.xml در واقع شیء View متناظر با فایل rowlayout.xml است و از طریق آن می‌توان به بقیه المان‌های rowlayout.xml دسترسی پیدا کرد.

```

در دو خط بعدی ما یک TextView و ImageView از View بازیابی می‌کنیم، سپس با استفاده از position برای سطر مورد نظر از یک فهرست یک View ایجاد کنیم سپس با بازیابی متن مورد نظر از آرایه آن را در TextView که بازیابی کرده بودیم، نمایش می‌دهیم.

امیربهاء‌الدین سبط‌الشیخ