

[首页](#) [ANDROID](#) [互联网](#) [杂乱无章](#) [科技资讯](#) [程序员人生](#) [程序员笑话](#) [编程技术](#) [网址导航](#)

6.2 数据存储与访问之—— SharedPreferences保存用户 偏好参数

分类 [Android 基础入门教程](#)

**Android 基础入门教程(Q群号 :
153836263)**

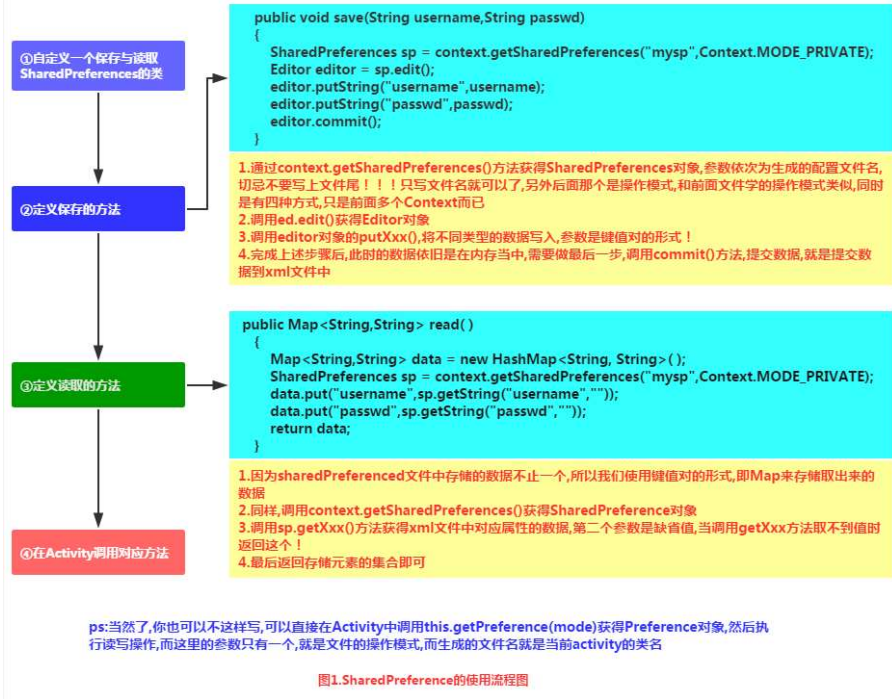
本节引言：

本节给大家介绍的是第二种存储用户数据的方式，使用SharedPreferences(保存用户偏好参数)保存数据，当我们的应用想要保存用户的一些偏好参数，比如是否自动登陆，是否记住账号密码，是否在Wifi下才能联网等相关信息，如果使用数据库的话，显得有点大材小用了！我们把上面这些配置信息称为用户的偏好设置，就是用户偏好的设置，而这些配置信息通常是保存在特定的文件中！比如windows使用ini文件，而J2SE中使用properties属性文件与xml文件来保存软件的配置信息；而在Android中我们通常使用一个轻量级的存储类——SharedPreferences来保存用户偏好的参数！SharedPreferences也是使用xml文件，然后类似于Map集合，使用键-值的形式来存储数据；我们只需要调用SharedPreferences的getXxx(name)，就可以根据键获得对应的值！使用起来很方便！

1.SharedPreferences使用示例：

使用流程图：

- 1.0 Android基础入门教程
- 1.0.1 2015年最新Android基...
- 1.1 背景相关与系统架构分析
- 1.2 开发环境搭建
- 1.2.1 使用Eclipse + ADT + S...
- 1.2.2 使用Android Studio开...
- 1.3 SDK更新不了问题解决
- 1.4 Genymotion模拟器安装
- 1.5.1 Git使用教程之本地仓...
- 1.5.2 Git之使用GitHub搭建...
- 1.6 .9(九妹)图片怎么玩
- 1.7 界面原型设计
- 1.8 工程相关解析(各种文件...
- 1.9 Android程序签名打包
- 1.11 反编译APK获取代码&...
- 2.1 View与ViewGroup的概念
- 2.2.1 LinearLayout(线性布局)
- 2.2.2 RelativeLayout(相对布...
- 2.2.3 TableLayout(表格布局)
- 2.2.4 FrameLayout(帧布局)
- 2.2.5 GridLayout(网格布局)
- 2.2.6 AbsoluteLayout(绝对...
- 2.3.1 TextView(文本框)详解
- 2.3.2 EditText(输入框)详解
- 2.3.3 Button(按钮)与ImageB...
- 2.3.4 ImageView(图像视图)
- 2.3.5.RadioButton(单选按钮...
- 2.3.6 开关按钮ToggleButton...
- 2.3.7 ProgressBar(进度条)
- 2.3.8 SeekBar(拖动条)
- 2.3.9 RatingBar(星级评分条)
- 2.4.1 ScrollView(滚动条)



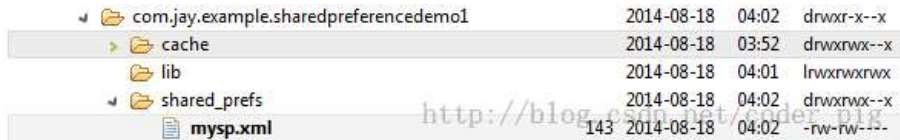
实现代码示例：

运行效果图：

流程是输入账号密码后点击登录,将信息保存到SharedPreference文件中, 然后重启app,看到数据已经显示在文本框中了



另外保存后，我们可以在File Expoler打开data/data/<包名>可以看到在shared_prefs目录下 生成了一个xml文件(因为N5没root，这里找了以前的效果图)：



点击导出到桌面可以看到里面的内容：

- 2.4.2 Date & Time组件(上)
- 2.4.3 Date & Time组件(下)
- 2.4.4 Adapter基础讲解
- 2.4.5 ListView简单实用
- 2.4.6 BaseAdapter优化
- 2.4.7ListView的焦点问题
- 2.4.8 ListView之checkbox错...
- 2.4.9 ListView的数据更新问题
- 2.5.0 构建一个可复用的自定...
- 2.5.1 ListView Item多布局的...
- 2.5.2 GridView(网格视图)的...
- 2.5.3 Spinner(列表选项框)...
- 2.5.4 AutoCompleteTextVie...
- 2.5.5 ExpandableListView(...
- 2.5.6 ViewFlipper(翻转视图)...
- 2.5.7 Toast(吐司)的基本使用
- 2.5.8 Notification(状态栏通...
- 2.5.9 AlertDialog(对话框)详解
- 2.6.0 其他几种常用对话框基...
- 2.6.1 PopupWindow(悬浮框...
- 2.6.2 菜单(Menu)
- 2.6.3 ViewPager的简单使用
- 2.6.4 DrawerLayout(官方侧...
- 3.1.1 基于监听的事件处理机制
- 3.2 基于回调的事件处理机制
- 3.3 Handler消息传递机制浅析
- 3.4 TouchListener PK OnTo...
- 3.5 监听EditText的内容变化
- 3.6 响应系统设置的事件(Co...
- 3.7 AsyncTask异步任务
- 3.8 Gestures(手势)
- 4.1.1 Activity初学乍练
- 4.1.2 Activity初窥门径
- 4.1.3 Activity登堂入室
- 4.2.1 Service初涉
- 4.2.2 Service进阶
- 4.2.3 Service精通
- 4.3.1 BroadcastReceiver牛...
- 4.3.2 BroadcastReceiver庖...
- 4.4.1 ContentProvider初探

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <string name="passwd">1234</string>
  <string name="username">Jay</string>
</map>
```

代码实现：

布局文件activity_main.xml的编写：

```
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MyActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="用户登陆" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="请输入用户名" />

    <EditText
        android:id="@+id/editname"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="用户名" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="请输入密码" />

    <EditText
        android:id="@+id/editpasswd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="密码"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/btnlogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="登录" />

</LinearLayout>
```

编写简单的SP工具类：SharedHelper.java：

- 4.4.2 ContentProvider再探...
- 4.5.1 Intent的基本使用
- 4.5.2 Intent之复杂数据的传递
- 5.1 Fragment基本概述
- 5.2.1 Fragment实例精讲—...
- 5.2.2 Fragment实例精讲—...
- 5.2.3 Fragment实例精讲—...
- 5.2.4 Fragment实例精讲—...
- 5.2.5 Fragment实例精讲—...
- 6.1 数据存储与访问之——文...
- 6.2 数据存储与访问之——S...
- 6.3.1 数据存储与访问之——...
- 6.3.2 数据存储与访问之——...
- 7.1.1 Android网络编程要学...
- 7.1.2 Android Http请求头与...
- 7.1.3 Android HTTP请求方...
- 7.1.4 Android HTTP请求方...
- 7.2.1 Android XML数据解析
- 7.2.2 Android JSON数据解析
- 7.3.1 Android 文件上传
- 7.3.2 Android 文件下载 (1)
- 7.3.3 Android 文件下载 (2)
- 7.4 Android 调用 WebService
- 7.5.1 WebView(网页视图)基...
- 7.5.2 WebView和JavaScrip...
- 7.5.3 Android 4.4后WebVie...
- 7.5.4 WebView文件下载
- 7.5.5 WebView缓存问题
- 7.5.6 WebView处理网页返...
- 7.6.1 Socket学习网络基础准备
- 7.6.2 基于TCP协议的Socket...
- 7.6.3 基于TCP协议的Socket...
- 7.6.4 基于UDP协议的Socke...
- 8.1.1 Android中的13种Draw...
- 8.1.2 Android中的13种Draw...
- 8.1.3 Android中的13种Draw...
- 8.2.1 Bitmap(位图)全解析 P...
- 8.2.2 Bitmap引起的OOM问题
- 8.3.1 三个绘图工具类详解
- 8.3.2 绘图类实战示例

```

/**
 * Created by Jay on 2015/9/2 0002.
 */
public class SharedHelper {

    private Context mContext;

    public SharedHelper() {
    }

    public SharedHelper(Context mContext) {
        this.mContext = mContext;
    }

    //定义一个保存数据的方法
    public void save(String username, String passwd) {
        SharedPreferences sp = mContext.getSharedPreferences("mysp", Context.MODE_PRIVATE);
        SharedPreferences.Editor editor = sp.edit();
        editor.putString("username", username);
        editor.putString("passwd", passwd);
        editor.commit();
        Toast.makeText(mContext, "信息已写入SharedPreferences中",
            Toast.LENGTH_SHORT).show();
    }

    //定义一个读取SP文件的方法
    public Map<String, String> read() {
        Map<String, String> data = new HashMap<String, String>();
        SharedPreferences sp = mContext.getSharedPreferences("mysp", Context.MODE_PRIVATE);
        data.put("username", sp.getString("username", ""));
        data.put("passwd", sp.getString("passwd", ""));
        return data;
    }
}

```

最后是MainActivity.java实现相关逻辑：

```

public class MainActivity extends AppCompatActivity {

    private EditText editname;
    private EditText editpasswd;
    private Button btnlogin;
    private String strname;
    private String strpasswd;
    private SharedHelper sh;
    private Context mContext;

    @Override

```

8.3.3 Paint API之——Mask...

8.3.4 Paint API之——Xferm...

8.3.5 Paint API之——Xferm...

8.3.6 Paint API之——Xferm...

8.3.7 Paint API之——Xferm...

8.3.8 Paint API之——Xferm...

8.3.9 Paint API之——Color...

8.3.10 Paint API之——Colo...

8.3.11 Paint API之——Colo...

8.3.12 Paint API之——Path...

8.3.13 Paint API之——Sha...

8.3.14 Paint几个枚举/常量值...

8.3.15 Paint API之——Type...

8.3.16 Canvas API详解(Part 1)

8.3.17 Canvas API详解(Part...

8.3.18 Canvas API详解(Part...

8.4.1 Android动画合集之帧...

8.4.2 Android动画合集之补...

8.4.3 Android动画合集之属...

8.4.4 Android动画合集之属...

9.1 使用SoundPool播放音...

9.2 MediaPlayer播放音频与...

9.3 使用Camera拍照

9.4 使用MediaRecord录音

10.1 TelephonyManager(电...

10.2 SmsManager(短信管理...

10.3 AudioManager(音频管...

10.4 Vibrator(振动器)

10.5 AlarmManager(闹钟服务)

10.6 PowerManager(电源服...

10.7 WindowManager(窗口...

10.8 LayoutInflater(布局服务)

10.9 WallpaperManager(壁...

10.10 传感器专题(1)——相...

10.11 传感器专题(2)——方...

10.12 传感器专题(3)——加...

10.12 传感器专题(4)——其...

10.14 Android GPS初涉

11.0 《2015最新Android基...

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    mContext = getApplicationContext();
    sh = new SharedHelper(mContext);
    bindViews();
}

private void bindViews() {
    editname = (EditText) findViewById(R.id.editname);
    editpasswd = (EditText) findViewById(R.id.editpasswd);
    btnlogin = (Button) findViewById(R.id.btnlogin);
    btnlogin.setOnClickListener(new View.OnClickListener()
    {
        @Override
        public void onClick(View v) {
            strname = editname.getText().toString();
            strpasswd = editpasswd.getText().toString();
            sh.save(strname, strpasswd);
        }
    });
}

@Override
protected void onStart() {
    super.onStart();
    Map<String, String> data = sh.read();
    editname.setText(data.get("username"));
    editpasswd.setText(data.get("passwd"));
}
}
```

2.读取其他应用的SharedPreferences

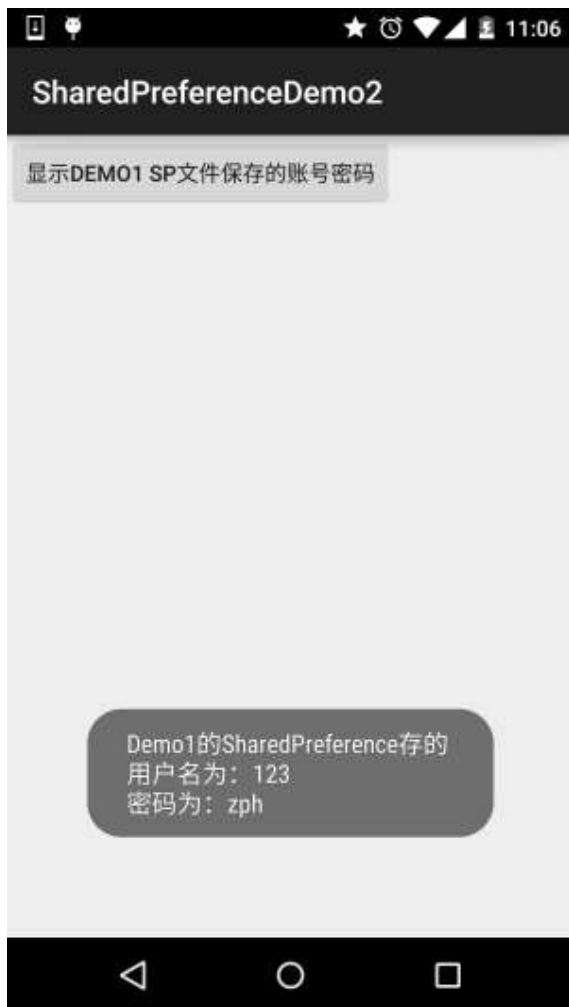
核心：获得其他app的Context,而这个Context代表访问该app的全局信息的接口,而决定应用的唯一标识 是应用的包名,所以我们可以通过应用包名获得对应app的Context 另外有一点要注意的是：其他应用的SP文件是否能被读写的前提就是SP文件是否指定了可读或者 可写的权限，我们上面创建的是MODE_PRIVATE的就不可以了~所以说你像读别人的SP里的数据，很难，另外，一些关键的信息，比如密码保存到SP里，一般都是会做加密的，所以只能自己写自己玩~ 等下会讲下常用的MD5加密方法！

实现流程图：



代码示例：

运行效果图：



代码实现：

我们读取SP的操作放在MainActivity.java中完成，点击按钮后读取SP，并通过Toast显示出来：

```
public class MainActivity extends AppCompatActivity {  
  
    private Context othercontext;  
    private SharedPreferences sp;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {
```



```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
Button btnshow = (Button) findViewById(R.id.btnshow);
btnshow.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //获得第一个应用的包名,从而获得对应的Context,需要对异常进行捕获

        try {
            othercontext = createPackageContext("com.ja
y.sharedpreferencedemo", Context.CONTEXT_IGNORE_SECURITY);
        } catch (PackageManager.NameNotFoundException e
) {
            e.printStackTrace();
        }
        //根据Context取得对应的SharedPreferences
        sp = othercontext.getSharedPreferences("mysp",
Context.MODE_WORLD_READABLE);
        String name = sp.getString("username", "");
        String passwd = sp.getString("passwd", "");
        Toast.makeText(getApplicationContext(), "Demo1
的SharedPreferences存的\n用户名为: " + name + "\n密码为: " + passwd
, Toast.LENGTH_SHORT).show();
    }
});
}
}

```

3.使用MD5对SharedPreferences的重要数据进行加密

嘿嘿，上面我们这样直接把账号密码保存到sp里，如果没root的手机，别的应用倒无法访问手机，如果root了，然后数据给其他应用获取到，然后造成了一些后果，这...就不怪我们了，哈哈，谁叫你root了~，这锅我们不背，的确是这样！但是作为一名有责任心的APP开发人员，我们总不能这样是吧，我们可以使用一些加密算法对用户密码进行加密，另外我们一般加密的都是用户密码！下面我们简画个简单的图帮助大家理解下加密的处理的流程：

1.简单的加密处理流程

流程图如下：

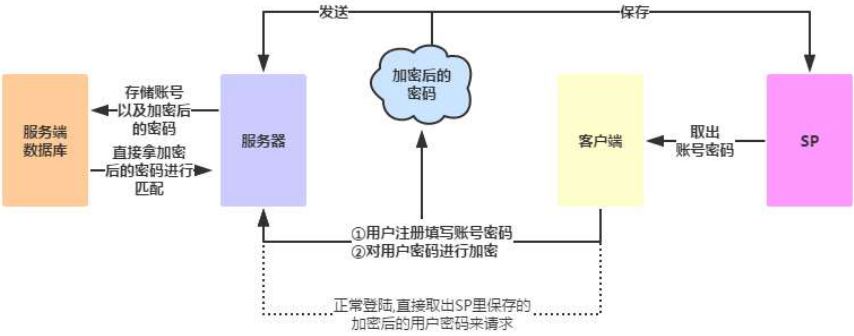


图2.简单的加密处理流程图

流程图解析：

- Step 1.**用户注册账号密码，账号密码校验后(账号是否重复，密码位数 > 6位等)，即账号密码有效，注册成功后，我们提交给服务器的账号，以及本地加密过的密码！
- Step 2.**服务器端将用户提交的账号，加密过的密码保存到服务端的数据库中，也就是服务端并不会保存我们的明文密码(原始)密码！
- Step 3.**说回客户端，如果注册成功或者登陆成功，你想保存账号密码到SP中，保存的密码也需要走一趟加密流程！即明文密码——>加密，再保存！如果不保存，每次请求的时候，明文密码也要走一趟家里流程，然后拿着加密后的密码来请求服务器！
- Step 4.**服务器验证账号以及加密密码，成功，分配客户端一个session标识，后续客户端可以拿着这个session来访问服务端提供的相关服务！

嘿嘿，理解了吧，加密的方法有很多种，小猪也不是这方面的高玩，以前使用过的加密方法是MD5 加密，本节也给大家简单介绍一下这个MD5加密，以及演示下用法~

2.MD5简单介绍：

1) MD5是什么鬼？：

答：Message Digest Algorithm MD5（中文名为消息摘要算法第五版）为计算机安全领域广泛使用的一种散列函数，用以提供消息的完整性保护——摘自《百度百科》简单点说就是一种加密算法，可以将一个字符串，或者文件，压缩包，执行MD5加密后，就可以生产一个固定长度为128bit的串！这个串基本唯一！另外我们都知道:一个十六进制 需要用4个bit来表示，那么对应的MD5的字符串长度就为：128 / 4 = 32位了！另外可能 你看到一些md5是16位的，只是将32位MD5码去掉了前八位以及后八位！不信么，我们来试试 百度一下：md5在线解密，第一个：<http://www.cmd5.com/>

密文: 曹神带我日狗

类型: md5

解密

帮助

查询结果:
md5(曹神带我日狗,32) = 32a16e933ea4bc45f34790b015012ec0
md5(曹神带我日狗,16) = 3ea4bc45f34790b0

2) MD5能破解吗？

答：MD5不可逆，就是说没有对应的算法，无法从生成的md5值逆向得到原始数据！当然暴力破解除外，简单的MD5加密后可以查MD5库~

3) MD5值唯一吗？

答：不唯一，一个原始数据只对应一个MD5值，但是一个MD5值可能对应多个原始数据！

3.MD5加密实现例子：

其实网上有很多写好的MD5的例子，百度或者谷歌一搜一大堆，这里提供下小猪用的MD5加密工具类！

Md5Util.java：

```
/**
 * Created by Jay on 2015/9/2 0002.
 */
public class MD5 {

    public static String getMD5(String content) {
        try {
            MessageDigest digest = MessageDigest.getInstance("MD5");
            digest.update(content.getBytes());
            return getHashString(digest);
        } catch (NoSuchAlgorithmException e) {
            e.printStackTrace();
        }
        return null;
    }

    private static String getHashString(MessageDigest digest) {
        StringBuilder builder = new StringBuilder();
        for (byte b : digest.digest()) {
            builder.append(Integer.toHexString((b >> 4) & 0xf));
        }
        builder.append(Integer.toHexString(b & 0xf));
        return builder.toString();
    }
}
```

MainActivity.java直接调用getMD5这个静态方法：

```
Log.e("HeHe", MD5.getMD5("呵呵"));
```

我们可以看到Logcat上打印出：

E/HeHe : 86d51ce7753a36079041fc15f7248035

这就是加密过后的呵呵了，我们可以把这串密文拷贝到上面这个md5的在线解密网站：




嘿嘿，果然，只是这样加密一次，就直接破解了，有点不安全的样子，那就加密100次咯，就是将加密后的字符串再加密，重复100次，我们在原先的基础上加个加密一百次的方法：

```
public static String getMD5x100 (String content) {  
    String s1 = content;  
    for(int i = 0;i < 100;i++){  
        s1 = getMD5(s1);  
    }  
    return s1;  
}
```

然后调用下，发现打印这个Log：

E/HeHe : 50db93f6f9a34e73ae5d1e6953601941

复制界面网站上：



好的，装B成功~

4.SharedPreference工具类：

每次都要自行实例化SP相关的类，肯定很麻烦，这里贴个SP的工具类，大家可以贴到自己的项目中，工具类来源于鸿洋大神的blog~

SPUtils.java

```
package com.jay.sharedpreferencedemo3;

import android.content.Context;
import android.content.SharedPreferences;

import java.util.Map;

/**
 * Created by Jay on 2015/9/2 0002.
 */
public class SPUtils {
    /**
     * 保存在手机里的SP文件名
     */
    public static final String FILE_NAME = "my_sp";

    /**
     * 保存数据
     */
    public static void put(Context context, String key, Object
obj) {
        SharedPreferences sp = context.getSharedPreferences(FILE
E_NAME, context.MODE_PRIVATE);
        SharedPreferences.Editor editor = sp.edit();
        if (obj instanceof Boolean) {
            editor.putBoolean(key, (Boolean) obj);
        } else if (obj instanceof Float) {
            editor.putFloat(key, (Float) obj);
        } else if (obj instanceof Integer) {
            editor.putInt(key, (Integer) obj);
        } else if (obj instanceof Long) {
            editor.putLong(key, (Long) obj);
        } else {
            editor.putString(key, (String) obj);
        }
        editor.commit();
    }

    /**
     * 获取指定数据
     */
    public static Object get(Context context, String key, Objec
t defaultObj) {
        SharedPreferences sp = context.getSharedPreferences(FILE
E_NAME, context.MODE_PRIVATE);
        if (defaultObj instanceof Boolean) {
            return sp.getBoolean(key, (Boolean) defaultObj);
        } else if (defaultObj instanceof Float) {
            return sp.getFloat(key, (Float) defaultObj);
        } else if (defaultObj instanceof Integer) {
            return sp.getInt(key, (Integer) defaultObj);
        } else if (defaultObj instanceof Long) {
```

```
        return sp.getLong(key, (Long) defaultObj);
    } else if (defaultObj instanceof String) {
        return sp.getString(key, (String) defaultObj);
    }
    return null;
}

/**
 * 删除指定数据
 */
public static void remove(Context context, String key) {
    SharedPreferences sp = context.getSharedPreferences(FILE_NAME, context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sp.edit();
    editor.remove(key);
    editor.commit();
}

/**
 * 返回所有键值对
 */
public static Map<String, ?> getAll(Context context) {
    SharedPreferences sp = context.getSharedPreferences(FILE_NAME, context.MODE_PRIVATE);
    Map<String, ?> map = sp.getAll();
    return map;
}

/**
 * 删除所有数据
 */
public static void clear(Context context) {
    SharedPreferences sp = context.getSharedPreferences(FILE_NAME, context.MODE_PRIVATE);
    SharedPreferences.Editor editor = sp.edit();
    editor.clear();
    editor.commit();
}

/**
 * 检查key对应的数据是否存在
 */
public static boolean contains(Context context, String key)
{
    SharedPreferences sp = context.getSharedPreferences(FILE_NAME, context.MODE_PRIVATE);
    return sp.contains(key);
}
}
```

5.代码下载：

- SharedPreferenceDemo.zip：[下载 SharedPreferenceDemo.zip](#)
- SharedPreferenceDemo2.zip：[下载 SharedPreferenceDemo2.zip](#)
- SharedPreferenceDemo3.zip：[下载 SharedPreferenceDemo3.zip](#)

本节小结：

好的，关于Android存储数据的第二种方式：SharedPreferences保存用户偏好参数的内容就这么多，应该可以满足你日常开发使用SP的需求，如果有什么遗漏，欢迎提出，谢谢~

← 6.1 数据存储与访问之——文件存储读写

6.3.1 数据存储与访问之——初见SQLite数据库 →

服务器，超值预售击穿5折

服务器数据库，预售抵双倍，击穿5折 腾讯云服务器，预售抵双倍，击穿5折



在线实例

- [HTML 实例](#)
- [CSS 实例](#)
- [JavaScript 实例](#)
- [Ajax 实例](#)
- [jQuery 实例](#)
- [XML 实例](#)
- [Java 实例](#)

字符集&工具

- [HTML 字符集设置](#)
- [HTML ASCII 字符集](#)
- [HTML ISO-8859-1](#)
- [HTML 实体符号](#)
- [HTML 拾色器](#)
- [JSON 格式化工具](#)

最新更新

- [Vue.js 入门教程](#)
- [CSS 下拉菜单](#)
- [WordPress Nginx...](#)
- [TIOBE 2015 年12...](#)
- [Swift 正式开源](#)
- [PHP 7 正式发布](#)
- [Shell 编程快速入门](#)

站点信息

- [意见反馈](#)
- [免责声明](#)
- [关于我们](#)
- [文章归档](#)

关注微信



Copyright © 2013-2015 菜鸟教程 [runoob.com](#) All Rights Reserved. 备案号：闽ICP备15012807号-1