Android Calendar的学习与运用

2013-05-24 10:56:10 0个评论 收藏 我要投稿

文章 推荐

· SGU126-Boxes
· SGU276-Andrew's Troubles
· c++学习笔记(6.类的封装)
· Moderate 整数打印读法 @CareerCup
· 最小生成树kruskal算法
· Moderate 最大连续序列之和 @CareerCup
· C++基础学习笔记----第四课（函数的重
· hdu1151Air Raid

```
    [java]
mport java.text.DateFormat;
import java.text.ParsePosition;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;
```

```
/**
 * Android Calendar的运用
 * @author Administrator
 *
 */
public class CalendarUtil {

    private int weeks = 0;// 用来全局控制 上一周，本周，下一周的周数变化
    private int MaxDate; // 一月最大天数
    private int MaxYear; // 一年最大天数

    public static void main(String[] args) {
        CalendarUtil tt = new CalendarUtil();
        System.out.println("获取当天日期:" + tt.getNowTime("yyyy-MM-dd"));
        System.out.println("获取本周一日期:" + tt.getMondayOFWeek());
        System.out.println("获取本周日的日期:" + tt.getCurrentWeekday());
        System.out.println("获取上周一日期:" + tt.getPreviousWeekday());
        System.out.println("获取上周日日期:" + tt.getPreviousWeekSunday());
        System.out.println("获取下周一日期:" + tt.getNextMonday());
        System.out.println("获取下周日日期:" + tt.getNextSunday());
        System.out.println("获得相应周的周六的日期:" + tt.getNowTime("yyyy-MM-dd"));
        System.out.println("获取本月第一天日期:" + tt.getFirstDayOfMonth());
        System.out.println("获取本月最后一天日期:" + tt.getDefaultDay());
        System.out.println("获取上月第一天日期:" + tt.getPreviousMonthFirst());
```

点击排行

· Android Studio安装配置详细步骤（图

· Android Studio中如何引用图片资源

· RecyclerView完全解析,让你从此爱上它

· [Android Studio 权威教程] 断点调

· Android之开发常用颜色

· Android M新控件之AppBarLayout，Nav

· 最新2017（Android）安卓面试题级答案

· Android Studio打包全攻略---从入门到

```
    System.out.println("获取上月最后一天的日期:" + tt.getPreviousMonthEnd());
    System.out.println("获取下月第一天日期:" + tt.getNextMonthFirst());
    System.out.println("获取下月最后一天日期:" + tt.getNextMonthEnd());
    System.out.println("获取本年的第一天日期:" + tt.getCurrentYearFirst());
    System.out.println("获取本年最后一天日期:" + tt.getCurrentYearEnd());
    System.out.println("获取去年的第一天日期:" + tt.getPreviousYearFirst());
    System.out.println("获取去年的最后一天日期:" + tt.getPreviousYearEnd());
    System.out.println("获取明年第一天日期:" + tt.getNextYearFirst());
    System.out.println("获取明年最后一天日期:" + tt.getNextYearEnd());
    System.out.println("获取本季度第一天:" + tt.getThisSeasonFirstTime(11));
    System.out.println("获取本季度最后一天:" + tt.getThisSeasonFinallyTime(11));
    System.out.println("获取两个日期之间间隔天数2008-12-1~2008-9.29:"
        + CalendarUtil.getTwoDay("2008-12-1", "2008-9-29"));
    System.out.println("获取当前月的第几周：" + tt.getWeekOfMonth());
    System.out.println("获取当前年份：" + tt.getYear());
    System.out.println("获取当前月份：" + tt.getMonth());
    System.out.println("获取今天在本年的第几天：" + tt.getDayOfYear());
    System.out.println("获得今天在本月的第几天(获得当前日)：" + tt.getDayOfMonth());
    System.out.println("获得今天在本周的第几天：" + tt.getDayOfWeek());
    System.out.println("获得半年后的日期："
        + tt.convertDateToString(tt.getTimeYearNext()));
}



public static int getYear() {
```

```
    return Calendar.getInstance().get(Calendar.YEAR);
}



public static int getMonth() {
    return Calendar.getInstance().get(Calendar.MONTH) + 1;
}



public static int getDayOfYear() {
    return Calendar.getInstance().get(Calendar.DAY_OF_YEAR);
}



public static int getDayOfMonth() {
    return Calendar.getInstance().get(Calendar.DAY_OF_MONTH);
}



public static int getDayOfWeek() {
    return Calendar.getInstance().get(Calendar.DAY_OF_WEEK);
}



public static int getWeekOfMonth() {
    return Calendar.getInstance().get(Calendar.DAY_OF_WEEK_IN_MONTH);
}


public static Date getTimeYearNext() {
    Calendar.getInstance().add(Calendar.DAY_OF_YEAR, 183);
    return Calendar.getInstance().getTime();
}


public static String convertDateToString(Date dateTime) {
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
    return df.format(dateTime);
}


public static String getTwoDay(String sj1, String sj2) {
    SimpleDateFormat myFormatter = new SimpleDateFormat("yyyy-MM-dd");
    long day = 0;
    try {
        java.util.Date date = myFormatter.parse(sj1);
        java.util.Date mydate = myFormatter.parse(sj2);
        day = (date.getTime() - mydate.getTime()) / (24 * 60 * 60 * 1000);
    } catch (Exception e) {
        return "";
    }
    return day + "";
}



public static String getWeek(String sdate) {
    // 再转换为时间
    Date date = CalendarUtil.strToDate(sdate);
    Calendar c = Calendar.getInstance();
    c.setTime(date);
    // int hour=c.get(Calendar.DAY_OF_WEEK);
    // hour中存的就是星期几了，其范围 1~7
    // 1=星期日 7=星期六，其他类推
    return new SimpleDateFormat("EEEE").format(c.getTime());
}



public static Date strToDate(String strDate) {
    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    ParsePosition pos = new ParsePosition(0);
    Date strtodate = formatter.parse(strDate, pos);
    return strtodate;
}
```

```
public static long getDays(String date1, String date2) {
    if (date1 == null || date1.equals(""))
        return 0;
    if (date2 == null || date2.equals(""))
        return 0;
    // 转换为标准时间
    SimpleDateFormat myFormatter = new SimpleDateFormat("yyyy-MM-dd");
    java.util.Date date = null;
    java.util.Date mydate = null;
    try {
        date = myFormatter.parse(date1);
        mydate = myFormatter.parse(date2);
    } catch (Exception e) {
    }
    long day = (date.getTime() - mydate.getTime()) / (24 * 60 * 60 * 1000);
    return day;
}



public String getDefaultDay() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    Calendar lastDate = Calendar.getInstance();
    lastDate.set(Calendar.DATE, 1);// 设为当前月的1号
    lastDate.add(Calendar.MONTH, 1);// 加一个月，变为下月的1号
    lastDate.add(Calendar.DATE, -1);// 减去一天，变为当月最后一天

    str = sdf.format(lastDate.getTime());
    return str;
}



public String getPreviousMonthFirst() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    Calendar lastDate = Calendar.getInstance();
    lastDate.set(Calendar.DATE, 1);// 设为当前月的1号
    lastDate.add(Calendar.MONTH, -1);// 减一个月，变为下月的1号
    // lastDate.add(Calendar.DATE,-1);//减去一天，变为当月最后一天

    str = sdf.format(lastDate.getTime());
    return str;
}



public String getFirstDayOfMonth() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    Calendar lastDate = Calendar.getInstance();
    lastDate.set(Calendar.DATE, 1);// 设为当前月的1号
    str = sdf.format(lastDate.getTime());
    return str;
}


public String getCurrentWeekday() {
    weeks = 0;
    int mondayPlus = this.getMondayPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, mondayPlus + 6);
    Date monday = currentDate.getTime();

    DateFormat df = DateFormat.getDateInstance();
    String preMonday = df.format(monday);
    return preMonday;
}


public String getNowTime(String dateformat) {
    Date now = new Date();
    SimpleDateFormat dateFormat = new SimpleDateFormat(dateformat);// 可以方便地修改日期格式
    String hehe = dateFormat.format(now);
    return hehe;
}



private int getMondayPlus() {
    Calendar cd = Calendar.getInstance();
    // 获得今天是一周的第几天，星期日是第一天，星期二是第二天......
    int dayOfWeek = cd.get(Calendar.DAY_OF_WEEK) - 1; // 因为按中国礼拜一作为第一天所以这里减1
    if (dayOfWeek == 1) {
        return 0;
    } else {
        return 1 - dayOfWeek;
    }
}



public String getMondayOFWeek() {
    weeks = 0;
    int mondayPlus = this.getMondayPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, mondayPlus);
    Date monday = currentDate.getTime();

    DateFormat df = DateFormat.getDateInstance();
    String preMonday = df.format(monday);
```

```
        return preMonday;
    }



    public String getSaturday() {
        int mondayPlus = this.getMondayPlus();
        GregorianCalendar currentDate = new GregorianCalendar();
        currentDate.add(GregorianCalendar.DATE, mondayPlus + 7 * weeks + 6);
        Date monday = currentDate.getTime();
        DateFormat df = DateFormat.getDateInstance();
        String preMonday = df.format(monday);
        return preMonday;
    }



    public String getPreviousWeekSunday() {
        weeks = 0;
        weeks--;
        int mondayPlus = this.getMondayPlus();
        GregorianCalendar currentDate = new GregorianCalendar();
        currentDate.add(GregorianCalendar.DATE, mondayPlus + weeks);
        Date monday = currentDate.getTime();
        DateFormat df = DateFormat.getDateInstance();
        String preMonday = df.format(monday);
        return preMonday;
```

```
}



public String getPreviousWeekday() {
    weeks--;
    int mondayPlus = this.getMondayPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, mondayPlus + 7 * weeks);
    Date monday = currentDate.getTime();
    DateFormat df = DateFormat.getDateInstance();
    String preMonday = df.format(monday);
    return preMonday;
}



public String getNextMonday() {
    weeks++;
    int mondayPlus = this.getMondayPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, mondayPlus + 7);
    Date monday = currentDate.getTime();
    DateFormat df = DateFormat.getDateInstance();
    String preMonday = df.format(monday);
    return preMonday;
}
```

```
public String getNextSunday() {

    int mondayPlus = this.getMondayPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, mondayPlus + 7 + 6);
    Date monday = currentDate.getTime();
    DateFormat df = DateFormat.getDateInstance();
    String preMonday = df.format(monday);
    return preMonday;
}

private int getMonthPlus() {
    Calendar cd = Calendar.getInstance();
    int monthOfNumber = cd.get(Calendar.DAY_OF_MONTH);
    cd.set(Calendar.DATE, 1);// 把日期设置为当月第一天
    cd.roll(Calendar.DATE, -1);// 日期回滚一天，也就是最后一天
    MaxDate = cd.get(Calendar.DATE);
    if (monthOfNumber == 1) {
        return -MaxDate;
    } else {
        return 1 - monthOfNumber;
    }
}
```

```
public String getPreviousMonthEnd() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    Calendar lastDate = Calendar.getInstance();
    lastDate.add(Calendar.MONTH, -1);// 减一个月
    lastDate.set(Calendar.DATE, 1);// 把日期设置为当月第一天
    lastDate.roll(Calendar.DATE, -1);// 日期回滚一天，也就是本月最后一天
    str = sdf.format(lastDate.getTime());
    return str;
}



public String getNextMonthFirst() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    Calendar lastDate = Calendar.getInstance();
    lastDate.add(Calendar.MONTH, 1);// 减一个月
    lastDate.set(Calendar.DATE, 1);// 把日期设置为当月第一天
    str = sdf.format(lastDate.getTime());
    return str;
}
```

```
public String getNextMonthEnd() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    Calendar lastDate = Calendar.getInstance();
    lastDate.add(Calendar.MONTH, 1);// 加一个月
    lastDate.set(Calendar.DATE, 1);// 把日期设置为当月第一天
    lastDate.roll(Calendar.DATE, -1);// 日期回滚一天，也就是本月最后一天
    str = sdf.format(lastDate.getTime());
    return str;
}



public String getNextYearEnd() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    Calendar lastDate = Calendar.getInstance();
    lastDate.add(Calendar.YEAR, 1);// 加一个年
    lastDate.set(Calendar.DAY_OF_YEAR, 1);
    lastDate.roll(Calendar.DAY_OF_YEAR, -1);
    str = sdf.format(lastDate.getTime());
    return str;
}



public String getNextYearFirst() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    Calendar lastDate = Calendar.getInstance();
    lastDate.add(Calendar.YEAR, 1);// 加一个年
    lastDate.set(Calendar.DAY_OF_YEAR, 1);
    str = sdf.format(lastDate.getTime());
    return str;

}



private int getMaxYear() {
    Calendar cd = Calendar.getInstance();
    cd.set(Calendar.DAY_OF_YEAR, 1);// 把日期设为当年第一天
    cd.roll(Calendar.DAY_OF_YEAR, -1);// 把日期回滚一天。
    int MaxYear = cd.get(Calendar.DAY_OF_YEAR);
    return MaxYear;
}

private int getYearPlus() {
```

```
    Calendar cd = Calendar.getInstance();
    int yearOfNumber = cd.get(Calendar.DAY_OF_YEAR);// 获得当天是一年中的第几天
    cd.set(Calendar.DAY_OF_YEAR, 1);// 把日期设为当年第一天
    cd.roll(Calendar.DAY_OF_YEAR, -1);// 把日期回滚一天。
    int MaxYear = cd.get(Calendar.DAY_OF_YEAR);
    if (yearOfNumber == 1) {
        return -MaxYear;
    } else {
        return 1 - yearOfNumber;
    }
}



public String getCurrentYearFirst() {
    int yearPlus = this.getYearPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, yearPlus);
    Date yearDay = currentDate.getTime();
    DateFormat df = DateFormat.getDateInstance();
    String preYearDay = df.format(yearDay);
    return preYearDay;
}

// 获得本年最后一天的日期 *
public String getCurrentYearEnd() {
    Date date = new Date();
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy");// 可以方便地修改日期格式
    String years = dateFormat.format(date);
    return years + "-12-31";
}

// 获得上年第一天的日期 *
public String getPreviousYearFirst() {
    Date date = new Date();
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy");// 可以方便地修改日期格式
    String years = dateFormat.format(date);
    int years_value = Integer.parseInt(years);
    years_value--;
    return years_value + "-1-1";
}

// 获得上年最后一天的日期
public String getPreviousYearEnd() {
    weeks--;
    int yearPlus = this.getYearPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, yearPlus + MaxYear * weeks
            + (MaxYear - 1));
    Date yearDay = currentDate.getTime();
    DateFormat df = DateFormat.getDateInstance();
```

```
        String preYearDay = df.format(yearDay);
        return preYearDay;
    }



    public String getThisSeasonFirstTime(int month) {
        int array[][] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 }, { 10, 11, 12 } };
        int season = 1;
        if (month >= 1 && month <= 3) {
            season = 1;
        }
        if (month >= 4 && month <= 6) {
            season = 2;
        }
        if (month >= 7 && month <= 9) {
            season = 3;
        }
        if (month >= 10 && month <= 12) {
            season = 4;
        }
        int start_month = array[season - 1][0];
        int end_month = array[season - 1][2];

        Date date = new Date();
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy");// 可以方便地修改日期格式
        String years = dateFormat.format(date);
        int years_value = Integer.parseInt(years);

        int start_days = 1;// years+"-"+String.valueOf(start_month)+"-1";//getLastDayOfMonth(years_value,start_month);
        int end_days = getLastDayOfMonth(years_value, end_month);
        String seasonDate = years_value + "-" + start_month + "-" + start_days;
        return seasonDate;

    }



    public String getThisSeasonFinallyTime(int month) {
        int array[][] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 }, { 10, 11, 12 } };
        int season = 1;
        if (month >= 1 && month <= 3) {
            season = 1;
        }
        if (month >= 4 && month <= 6) {
            season = 2;
        }
        if (month >= 7 && month <= 9) {
            season = 3;
        }
        if (month >= 10 && month <= 12) {
            season = 4;
        }
        int start_month = array[season - 1][0];
        int end_month = array[season - 1][2];

        Date date = new Date();
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy");// 可以方便地修改日期格式
        String years = dateFormat.format(date);
        int years_value = Integer.parseInt(years);

        int start_days = 1;// years+"-"+String.valueOf(start_month)+"-1";//getLastDayOfMonth(years_value,start_month);
        int end_days = getLastDayOfMonth(years_value, end_month);
        String seasonDate = years_value + "-" + end_month + "-" + end_days;
        return seasonDate;

    }



    private int getLastDayOfMonth(int year, int month) {
        if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8
                || month == 10 || month == 12) {
            return 31;
        }
        if (month == 4 || month == 6 || month == 9 || month == 11) {
            return 30;
        }
        if (month == 2) {
            if (isLeapYear(year)) {
                return 29;
            } else {
                return 28;
            }
        }
        return 0;
    }



    public boolean isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
    }



    public boolean isLeapYear2(int year) {
        return new GregorianCalendar().isLeapYear(year);
    }
}
```

```
        import java.text.DateFormat;
import java.text.ParsePosition;
import java.text.SimpleDateFormat;
import java.util.Calendar;

import java.util.Date;

import java.util.GregorianCalendar;

        /**

 * Android Calendar的运用

 * @author Administrator

 *

 */

public class CalendarUtil {

            private int weeks = 0;// 用来全局控制 上一周，本周，下一周的周数变化

    private int MaxDate; // 一月最大天数

    private int MaxYear; // 一年最大天数


    public static void main(String[] args) {

        CalendarUtil tt = new CalendarUtil();

        System.out.println("获取当天日期:" + tt.getNowTime("yyyy-MM-dd"));

        System.out.println("获取本周一日期:" + tt.getMondayOFWeek());

        System.out.println("获取本周日的日期:" + tt.getCurrentWeekday());

        System.out.println("获取上周一日期:" + tt.getPreviousWeekday());

        System.out.println("获取上周日日期:" + tt.getPreviousWeekSunday());

        System.out.println("获取下周一日期:" + tt.getNextMonday());

        System.out.println("获取下周日日期:" + tt.getNextSunday());

        System.out.println("获得相应周的周六的日期:" + tt.getNowTime("yyyy-MM-dd"));

        System.out.println("获取本月第一天日期:" + tt.getFirstDayOfMonth());

        System.out.println("获取本月最后一天日期:" + tt.getDefaultDay());
        System.out.println("获取上月第一天日期:" + tt.getPreviousMonthFirst());
        System.out.println("获取上月最后一天的日期:" + tt.getPreviousMonthEnd());
        System.out.println("获取下月第一天日期:" + tt.getNextMonthFirst());
        System.out.println("获取下月最后一天日期:" + tt.getNextMonthEnd());
        System.out.println("获取本年的第一天日期:" + tt.getCurrentYearFirst());
        System.out.println("获取本年最后一天日期:" + tt.getCurrentYearEnd());
        System.out.println("获取去年的第一天日期:" + tt.getPreviousYearFirst());
        System.out.println("获取去年的最后一天日期:" + tt.getPreviousYearEnd());
        System.out.println("获取明年第一天日期:" + tt.getNextYearFirst());
        System.out.println("获取明年最后一天日期:" + tt.getNextYearEnd());
        System.out.println("获取本季度第一天:" + tt.getThisSeasonFirstTime(11));
        System.out.println("获取本季度最后一天:" + tt.getThisSeasonFinallyTime(11));
        System.out.println("获取两个日期之间间隔天数2008-12-1~2008-9.29:"
                + CalendarUtil.getTwoDay("2008-12-1", "2008-9-29"));
        System.out.println("获取当前月的第几周：" + tt.getWeekOfMonth());
        System.out.println("获取当前年份：" + tt.getYear());
        System.out.println("获取当前月份：" + tt.getMonth());
        System.out.println("获取今天在本年的第几天：" + tt.getDayOfYear());
        System.out.println("获得今天在本月的第几天(获得当前日)：" + tt.getDayOfMonth());
        System.out.println("获得今天在本周的第几天：" + tt.getDayOfWeek());
        System.out.println("获得半年后的日期："
                + tt.convertDateToString(tt.getTimeYearNext()));
    }


    public static int getYear() {
```

```
    return Calendar.getInstance().get(Calendar.YEAR);
}


public static int getMonth() {
    return Calendar.getInstance().get(Calendar.MONTH) + 1;
}


public static int getDayOfYear() {
    return Calendar.getInstance().get(Calendar.DAY_OF_YEAR);
}


public static int getDayOfMonth() {
    return Calendar.getInstance().get(Calendar.DAY_OF_MONTH);
}


public static int getDayOfWeek() {
    return Calendar.getInstance().get(Calendar.DAY_OF_WEEK);
}


public static int getWeekOfMonth() {
    return Calendar.getInstance().get(Calendar.DAY_OF_WEEK_IN_MONTH);
}
```

```
public static Date getTimeYearNext() {
    Calendar.getInstance().add(Calendar.DAY_OF_YEAR, 183);
    return Calendar.getInstance().getTime();
}

public static String convertDateToString(Date dateTime) {
    SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");
    return df.format(dateTime);
}

public static String getTwoDay(String sj1, String sj2) {
    SimpleDateFormat myFormatter = new SimpleDateFormat("yyyy-MM-dd");
    long day = 0;
    try {
        java.util.Date date = myFormatter.parse(sj1);
        java.util.Date mydate = myFormatter.parse(sj2);
        day = (date.getTime() - mydate.getTime()) / (24 * 60 * 60 * 1000);
    } catch (Exception e) {
        return "";
    }
    return day + "";
}
```

```
public static String getWeek(String sdate) {
    // 再转换为时间
    Date date = CalendarUtil.strToDate(sdate);
    Calendar c = Calendar.getInstance();
    c.setTime(date);
    // int hour=c.get(Calendar.DAY_OF_WEEK);
    // hour中存的就是星期几了，其范围 1~7
    // 1=星期日 7=星期六，其他类推
    return new SimpleDateFormat("EEEE").format(c.getTime());
}


public static Date strToDate(String strDate) {
    SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd");
    ParsePosition pos = new ParsePosition(0);
    Date strtodate = formatter.parse(strDate, pos);
    return strtodate;
}


public static long getDays(String date1, String date2) {
    if (date1 == null || date1.equals(""))
        return 0;
    if (date2 == null || date2.equals(""))
        return 0;
    // 转换为标准时间
    SimpleDateFormat myFormatter = new SimpleDateFormat("yyyy-MM-dd");
    java.util.Date date = null;
    java.util.Date mydate = null;
    try {
        date = myFormatter.parse(date1);
        mydate = myFormatter.parse(date2);
    } catch (Exception e) {
    }
    long day = (date.getTime() - mydate.getTime()) / (24 * 60 * 60 * 1000);
    return day;
}


public String getDefaultDay() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
            Calendar lastDate = Calendar.getInstance();
    lastDate.set(Calendar.DATE, 1);// 设为当前月的1号
    lastDate.add(Calendar.MONTH, 1);// 加一个月，变为下月的1号
    lastDate.add(Calendar.DATE, -1);// 减去一天，变为当月最后一天
            str = sdf.format(lastDate.getTime());
    return str;
}


public String getPreviousMonthFirst() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

        Calendar lastDate = Calendar.getInstance();
    lastDate.set(Calendar.DATE, 1);// 设为当前月的1号
    lastDate.add(Calendar.MONTH, -1);// 减一个月，变为下月的1号
    // lastDate.add(Calendar.DATE,-1);//减去一天，变为当月最后一天

        str = sdf.format(lastDate.getTime());
    return str;
}


public String getFirstDayOfMonth() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

        Calendar lastDate = Calendar.getInstance();
    lastDate.set(Calendar.DATE, 1);// 设为当前月的1号
    str = sdf.format(lastDate.getTime());
    return str;
}


public String getCurrentWeekday() {
    weeks = 0;
    int mondayPlus = this.getMondayPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, mondayPlus + 6);
    Date monday = currentDate.getTime();
            DateFormat df = DateFormat.getDateInstance();
    String preMonday = df.format(monday);
    return preMonday;
}


public String getNowTime(String dateformat) {
    Date now = new Date();
    SimpleDateFormat dateFormat = new SimpleDateFormat(dateformat);// 可以方便地修改日期格式
    String hehe = dateFormat.format(now);
    return hehe;
}


private int getMondayPlus() {
    Calendar cd = Calendar.getInstance();
    // 获得今天是一周的第几天，星期日是第一天，星期二是第二天......
    int dayOfWeek = cd.get(Calendar.DAY_OF_WEEK) - 1; // 因为按中国礼拜一作为第一天所以这里减1
    if (dayOfWeek == 1) {
        return 0;
    } else {
        return 1 - dayOfWeek;
    }
}
```

```
public String getMondayOFWeek() {
    weeks = 0;
    int mondayPlus = this.getMondayPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, mondayPlus);
    Date monday = currentDate.getTime();
            DateFormat df = DateFormat.getDateInstance();
    String preMonday = df.format(monday);
    return preMonday;
}


public String getSaturday() {
    int mondayPlus = this.getMondayPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, mondayPlus + 7 * weeks + 6);
    Date monday = currentDate.getTime();
    DateFormat df = DateFormat.getDateInstance();
    String preMonday = df.format(monday);
    return preMonday;
}


public String getPreviousWeekSunday() {
    weeks = 0;
```

```
        weeks--;
        int mondayPlus = this.getMondayPlus();
        GregorianCalendar currentDate = new GregorianCalendar();
        currentDate.add(GregorianCalendar.DATE, mondayPlus + weeks);
        Date monday = currentDate.getTime();
        DateFormat df = DateFormat.getDateInstance();
        String preMonday = df.format(monday);
        return preMonday;
    }

    public String getPreviousWeekday() {
        weeks--;
        int mondayPlus = this.getMondayPlus();
        GregorianCalendar currentDate = new GregorianCalendar();
        currentDate.add(GregorianCalendar.DATE, mondayPlus + 7 * weeks);
        Date monday = currentDate.getTime();
        DateFormat df = DateFormat.getDateInstance();
        String preMonday = df.format(monday);
        return preMonday;
    }

    public String getNextMonday() {
        weeks++;
        int mondayPlus = this.getMondayPlus();
        GregorianCalendar currentDate = new GregorianCalendar();
        currentDate.add(GregorianCalendar.DATE, mondayPlus + 7);
        Date monday = currentDate.getTime();
        DateFormat df = DateFormat.getDateInstance();
        String preMonday = df.format(monday);
        return preMonday;
    }


    public String getNextSunday() {
                int mondayPlus = this.getMondayPlus();
        GregorianCalendar currentDate = new GregorianCalendar();
        currentDate.add(GregorianCalendar.DATE, mondayPlus + 7 + 6);
        Date monday = currentDate.getTime();
        DateFormat df = DateFormat.getDateInstance();
        String preMonday = df.format(monday);
        return preMonday;
    }
            private int getMonthPlus() {
        Calendar cd = Calendar.getInstance();
        int monthOfNumber = cd.get(Calendar.DAY_OF_MONTH);
        cd.set(Calendar.DATE, 1);// 把日期设置为当月第一天
        cd.roll(Calendar.DATE, -1);// 日期回滚一天，也就是最后一天
        MaxDate = cd.get(Calendar.DATE);
        if (monthOfNumber == 1) {
            return -MaxDate;
        } else {
            return 1 - monthOfNumber;
        }
    }


    public String getPreviousMonthEnd() {
        String str = "";
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

                Calendar lastDate = Calendar.getInstance();
        lastDate.add(Calendar.MONTH, -1);// 减一个月
        lastDate.set(Calendar.DATE, 1);// 把日期设置为当月第一天
        lastDate.roll(Calendar.DATE, -1);// 日期回滚一天，也就是本月最后一天
        str = sdf.format(lastDate.getTime());
        return str;
    }


    public String getNextMonthFirst() {
        String str = "";
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

                Calendar lastDate = Calendar.getInstance();
        lastDate.add(Calendar.MONTH, 1);// 减一个月
        lastDate.set(Calendar.DATE, 1);// 把日期设置为当月第一天
        str = sdf.format(lastDate.getTime());
        return str;
    }


    public String getNextMonthEnd() {
        String str = "";
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
                Calendar lastDate = Calendar.getInstance();
        lastDate.add(Calendar.MONTH, 1);// 加一个月
        lastDate.set(Calendar.DATE, 1);// 把日期设置为当月第一天
        lastDate.roll(Calendar.DATE, -1);// 日期回滚一天，也就是本月最后一天
        str = sdf.format(lastDate.getTime());
        return str;
    }


    public String getNextYearEnd() {
        String str = "";
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
                Calendar lastDate = Calendar.getInstance();
        lastDate.add(Calendar.YEAR, 1);// 加一个年
        lastDate.set(Calendar.DAY_OF_YEAR, 1);
        lastDate.roll(Calendar.DAY_OF_YEAR, -1);
        str = sdf.format(lastDate.getTime());
        return str;
    }
```

```
public String getNextYearFirst() {
    String str = "";
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
            Calendar lastDate = Calendar.getInstance();
    lastDate.add(Calendar.YEAR, 1);// 加一个年
    lastDate.set(Calendar.DAY_OF_YEAR, 1);
    str = sdf.format(lastDate.getTime());
    return str;
        }


private int getMaxYear() {
    Calendar cd = Calendar.getInstance();
    cd.set(Calendar.DAY_OF_YEAR, 1);// 把日期设为当年第一天
    cd.roll(Calendar.DAY_OF_YEAR, -1);// 把日期回滚一天。
    int MaxYear = cd.get(Calendar.DAY_OF_YEAR);
    return MaxYear;
}
        private int getYearPlus() {
    Calendar cd = Calendar.getInstance();
    int yearOfNumber = cd.get(Calendar.DAY_OF_YEAR);// 获得当天是一年中的第几天
    cd.set(Calendar.DAY_OF_YEAR, 1);// 把日期设为当年第一天
    cd.roll(Calendar.DAY_OF_YEAR, -1);// 把日期回滚一天。
    int MaxYear = cd.get(Calendar.DAY_OF_YEAR);
    if (yearOfNumber == 1) {
        return -MaxYear;
    } else {
        return 1 - yearOfNumber;
    }
}


public String getCurrentYearFirst() {
    int yearPlus = this.getYearPlus();
    GregorianCalendar currentDate = new GregorianCalendar();
    currentDate.add(GregorianCalendar.DATE, yearPlus);
    Date yearDay = currentDate.getTime();
    DateFormat df = DateFormat.getDateInstance();
    String preYearDay = df.format(yearDay);
    return preYearDay;
}
        // 获得本年最后一天的日期 *
public String getCurrentYearEnd() {
    Date date = new Date();
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy");// 可以方便地修改日期格式
    String years = dateFormat.format(date);
    return years + "-12-31";
}
```

```
    // 获得上年第一天的日期 *
public String getPreviousYearFirst() {
  Date date = new Date();
  SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy");// 可以方便地修改日期格式
  String years = dateFormat.format(date);
  int years_value = Integer.parseInt(years);
  years_value--;
  return years_value + "-1-1";
}
    // 获得上年最后一天的日期
public String getPreviousYearEnd() {
  weeks--;
  int yearPlus = this.getYearPlus();
  GregorianCalendar currentDate = new GregorianCalendar();
  currentDate.add(GregorianCalendar.DATE, yearPlus + MaxYear * weeks
      + (MaxYear - 1));
  Date yearDay = currentDate.getTime();
  DateFormat df = DateFormat.getDateInstance();
  String preYearDay = df.format(yearDay);
  return preYearDay;
}


public String getThisSeasonFirstTime(int month) {
  int array[][] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 }, { 10, 11, 12 } };
  int season = 1;
```

```
    if (month >= 1 && month <= 3) {
      season = 1;
    }
    if (month >= 4 && month <= 6) {
      season = 2;
    }
    if (month >= 7 && month <= 9) {
      season = 3;
    }
    if (month >= 10 && month <= 12) {
      season = 4;
    }
    int start_month = array[season - 1][0];
    int end_month = array[season - 1][2];
        Date date = new Date();
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy");// 可以方便地修改日期格式
    String years = dateFormat.format(date);
    int years_value = Integer.parseInt(years);
        int start_days = 1;// years+"-
"+String.valueOf(start_month)+"-1";//getLastDayOfMonth(years_value,start_month);
    int end_days = getLastDayOfMonth(years_value, end_month);
    String seasonDate = years_value + "-" + start_month + "-" + start_days;
    return seasonDate;
      }
public String getThisSeasonFinallyTime(int month) {
    int array[][] = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 }, { 10, 11, 12 } };
    int season = 1;
    if (month >= 1 && month <= 3) {
      season = 1;
    }
    if (month >= 4 && month <= 6) {
      season = 2;
    }
    if (month >= 7 && month <= 9) {
      season = 3;
    }
    if (month >= 10 && month <= 12) {
      season = 4;
    }
    int start_month = array[season - 1][0];
    int end_month = array[season - 1][2];

        Date date = new Date();
    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy");// 可以方便地修改日期格式
    String years = dateFormat.format(date);
    int years_value = Integer.parseInt(years);

        int start_days = 1;// years+"-
"+String.valueOf(start_month)+"-1";//getLastDayOfMonth(years_value,start_month);
    int end_days = getLastDayOfMonth(years_value, end_month);
    String seasonDate = years_value + "-" + end_month + "-" + end_days;
    return seasonDate;
        }


private int getLastDayOfMonth(int year, int month) {
    if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8
            || month == 10 || month == 12) {
        return 31;
    }
    if (month == 4 || month == 6 || month == 9 || month == 11) {
        return 30;
    }
    if (month == 2) {
        if (isLeapYear(year)) {
            return 29;
        } else {
            return 28;
        }
    }
    return 0;
}


public boolean isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
    }


    public boolean isLeapYear2(int year) {
        return new GregorianCalendar().isLeapYear(year);
    }
}
```

点击复制链接 与好友分享! 回本站首页

相关TAG标签 Android Calendar的学习与运用

上一篇：android bitmap compress（图片压缩）

下一篇：Android游戏开发之一

相关文章

android libaray的创建和使用

android中调用webservice

android中json转换成List

Android ListView Adapter

android AudioTrack不能播放awr

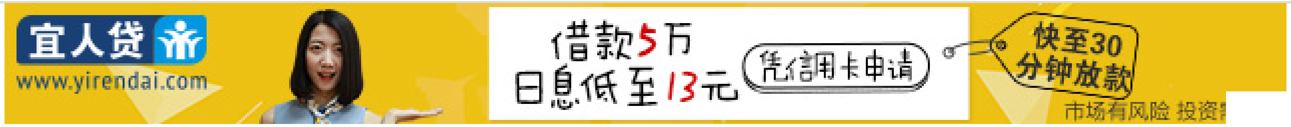
Android异步操作AsyncTask

android-仿QQtab

android database sqlite的使用

Android之ImageSwitcher,Gallery用法 Android面试系列之一

热门专题推荐 python div+css css教程 html5 html教程 jquery Android SDK php mysql oracle



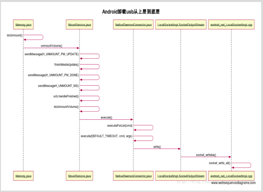
图文推荐

拉绳位移传感器

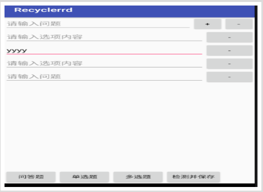
android学习

网页游戏

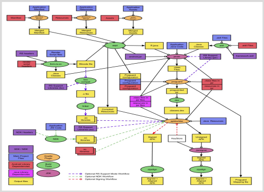
大闸蟹礼盒



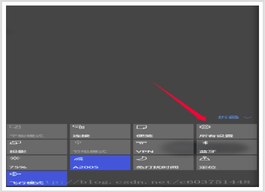
Android手动卸载usb从



Android Kotlin开发



Gradle 完整指南（A



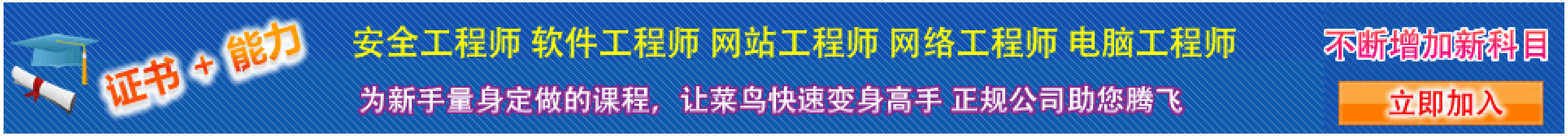
Win10环境下AndroidS

登录

来说两句吧...

还没有评论，快来抢沙发吧！

红黑联盟正在使用畅言


证书 + 能力
安全工程师 软件工程师 网站工程师 网络工程师 电脑工程师
为新手量身定做的课程，让菜鸟快速变身高手 正规公司助您腾飞
不断增加新科目
立即加入