

```

//スプレッドシートを指定
var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
//シートを指定
const sheet = spreadsheet.getSheets()[0];
//最終行を取得
let lastRow = sheet.getLastRow();
const address=Session.getActiveUser().getUserLoginId();

function newres() {

    //過去ログ入りしたスレを消去
    deleterows();
    //スレの元の状態を取得
    const boards = sheet.getRange(2, 1, lastRow - 1, 4).getValues();
    //空の配列を用意
    var elements = [];
    var newress = [];
    var oldress = [];
    //新しく記入する内容
    boards.forEach(function (board) {
        var box = get(board[3]);
        elements.push(box);
        if (board[2] < box[2] && board[2]) {
            //送信する内容
            newress.push(box);
        }
        else if(timeCheck(board[1]))
        {
            oldress.push(box);
        }
    })
    //メール送信
    if (newress.length) {
        var template = text(newress);
        mail(template);
    }
    if(oldress.length){
        var oldtemplate = falltext(oldress);
        mail(oldtemplate);
    }
    //記述
    sheet.getRange(2, 1, lastRow - 1, 4).setValues(elements);
}
// ファイルが開いたときに動作する処理
function onOpen() {

```

```

// カスタムメニューを追加する処理
spreadsheet.addMenu('スクリプト実行', [
  { name: 'レス確認', functionName: 'newres' }
]);
}
function get(url) {
  let html = UrlFetchApp.fetch(url).getContentText();
  let elements = html.match(/<h1 id="threadTitle">.*?<|>スレッドは.*?頃に落ちま
す|>このスレッドは過去ログ倉庫に格納されています|setting={.*?}/g);
  //空の配列を用意
  var box = [];
  //スレタイを入れる
  box.push(elements[0].replace(/<.*?>|</g, ' '));
  //スレが落ちる時間を入れる
  box.push(elements[1].replace('>', ' '));
  //新しいレス番号を入れる
  var setting = elements[2].split(',');
  let num = setting[1].replace(/\\D/g, '');
  box.push(Number(num));
  //urlも足す
  box.push(url);
  return box;
}
//メールの内容を整える
function text(newress) {

  var textTemplate = "スレの確認メールです。/n ";
  var bodyTemplate = 'スレの確認メールです。<br>';
  newress.forEach(function (newres) {
    textTemplate += newres[0] + ' /n ' + newres[3] + ' /n ';
    bodyTemplate += "<br><a href=" + newres[3] + ">" + newres[0] + "<a><br>";

  })
  textTemplate += "上記のスレで最新レスがつけました";
  bodyTemplate += "<br>上記のスレにて最新レスがつけました";

  return ([textTemplate, bodyTemplate])
}
function falltext(olddress){
  var textTemplate = "スレの確認メールです。/n ";
  var bodyTemplate = 'スレの確認メールです。<br>';
  oldress.forEach(function (oldres) {
    textTemplate += oldres[0] + ' /n ' + oldres[3] + ' /n ';
    bodyTemplate += "<br><a href=" + oldres[3] + ">" + oldres[0] + "<a><br>";
  })
}

```

```

    })
    textTemplate += "上記のスレがもうすぐ過去ログに行きます";
    bodyTemplate += "<br>上記のスレがもうすぐ過去ログに行きます";

    return ([textTemplate, bodyTemplate])
}

function deleterows() {
    for (var i = 2; i < lastRow; i++) {
        //スレが落ちる時刻
        var time = sheet.getRange(i, 2).getValue();
        //urlの場所
        var cell = sheet.getRange(i, 4);
        //過去ログまたはurlの書かれていない行を削除
        if (time == "このスレッドは過去ログ倉庫に格納されています" || cell.isBlank()) {
            var start_row = i;
            var num_row = 1;
            sheet.deleteRows(start_row, num_row);
            i = i - 1;
            lastRow-=1;
        }
    }
}

//メールを送信する
function mail(template) {
    //メールの件名を記述する
    let mailTitle = "スレの状況報告";
    var textTemplate = template[0];
    var bodyHtml = template[1];

    //オプションでHTMLメール本文を設定する
    let options = {
        from: address, //送り元
        "htmlBody": bodyHtml,
    };
    //MailAppで宛先、件名、本文、添付ファイルを引数にしてメールを送付
    MailApp.sendEmail(address, mailTitle, textTemplate, options);
}

```