

- [機能拡張について](#)
- [はじめに](#)
- [機能拡張サンプル](#)
 - [動画から3秒目の画像を取り出し表示する](#)
 - [いろいろな値をクリップボードにコピーするボタンと動画ページのサムネイルを設置する \(-ver0.47\)](#)
 - [携帯動画変換君に投げ込むボタンを設置する・MP3ファイルを抽出するボタンを設置する \(-ver0.41\)](#)
 - [コメントの推移をグラフ化して表示する](#)
 - [コメントを再生ファイルの時間軸についてグラフ化](#)
 - [コメントを再生ファイルの時間軸についてグラフ化2 \(-ver0.57\)](#)
 - [プレイリストのサムネイルを列挙し、クリックしたら再生する](#)
 - [動画ページへのリンクを改造する\(-ver0.49\)](#)
 - [マイリスト・タグ検索をTubePlayerっぽく表示する\(-ver0.47\)](#)
 - [投稿者コメントのニコニコ動画内リンク\(動画・マイページ\)をクリックするとサムネイルを表示する \(-ver0.42\)](#)
 - [プレイリストを検索しリストをリアルタイム更新、該当する行をクリックすると再生](#)
 - [動画ページを全て表示する](#)
 - [動画ページを全て表示する2 \(-ver0.41\)](#)
 - [プレイリスト一覧を表示し切り替える \(-ver0.41\)](#)
 - [公式のNGリストを取得し表示する \(-ver0.41\)](#)
 - [ダウンロード済みのコメントファイルを結合する \(-ver0.42\)](#)
 - [過去のコメントを取得する \(-ver0.42\)](#)
 - [一つ前のコメントを取得する \(-ver0.44\)](#)
 - [コメントをこっそり投稿する \(-ver0.45\)](#)
 - [NicoPlayerを操作する各種ボタンを取り付ける \(-ver0.46\)](#)
 - [オススメを表示する \(-ver0.46\)](#)
- [過去ログ](#)
- [コメント](#)

機能拡張について

[動画情報ウィンドウ#機能拡張](#)をご覧ください。

はじめに

NicoPlayerは動画情報の表示にIEコンポーネント(InternetExplorerの一部)の機能を使っています。設定は[設定ファイル]で示されている、HTML及びその周辺フォーマット(JavaScript・CSS)で記述されたファイル群で行われています。特にtemplate.htmlを編集することで、通常再生時に表示される動画情報を好みに合わせることが出来ます。またシェルを介することでNicoPlayerを操作したり、他のツールと連携したりといったことも可能です。

大きな可能性を秘めた動画情報ウィンドウですが、あまりに強力なためセキュリティ上の観点からいくつかの機能が標準で禁止されています。以下の機能拡張サンプルではこれらの機能を用いている場合が多いため、危険性を十分に認識した上で次の手順で機能使用の許可設定を行う必要があります。

セキュリティの設定

インターネットエクスプローラ(IE)のメニュー[ツール]から[インターネットオプション]を選択する
 上部のタブから[セキュリティ]を選択し、[イントラネット]・[レベルのカスタマイズ]へ進む
 [ActiveXコントロールとプラグイン]の
 [スクリプトを実行しても安全だとマークされていないActiveXコントロールの初期化とスクリプトの実行]
 を有効にし、
 [ActiveXコントロールに対して自動的にダイアログを表示]
 を無効にしてください。

これでセキュリティ上は全てのサンプルが機能するようになりました。ただし一部のサンプルではフレームワークという、開発を容易にするソフトウェアを使用しています。次の手順で導入してください。

追加JavaScriptライブラリ ダウンロード：[prototype.js](#) [excanvas.js](#)

prototype.jsの導入

[prototype](#)のページを開き、Download(右上)をクリックする
 [Download the latest stable version]をクリックし、ファイルを保存する

ダウンロードしたprototype.js(設定によっては.jsが表示されないことがあります)をvideoinfoフォルダへ移動させる

これでほとんどのサンプルは機能するはずですが、もしうまくいかなかったら、このページの下のほうにあるコメントに書き込んでみましょう。誰かが手助けしてくれるかもしれません。自力で解決できた場合は、ぜひ躰いた点とその解決策を書き込んでください！

機能拡張サンプル

動画から3秒目の画像を取り出し表示する

グラフの背景(サムネイルを拡大表示)があまりに無理やりなので、自前で用意してみました。

```
<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
var pathOf3GPCConverter = "携帯動画変換君のパス";
function showThumbnail()
{
var shellObject = new ActiveXObject( "WScript.Shell" );
shellObject.Run(
pathOf3GPCConverter + " \ \cores \ \ffmpeg.exe -i \ "%__MEDIA_PATH2_%\" -f image2 -pix_fmt png"
+ " -vframes 1 -ss 3 -s 260x200 -an -deinterlace \ "%__MEDIA_PATH2_%.png\" ", 0, true );
wshShell = null;

$( "target1" ).innerHTML = "<img src= \ "%__MEDIA_PATH2_%.png\" ">";
}
</script>
</head>
<body onload="showThumbnail()">
<span id="target1" />
</body>
</html>
```

いろいろな値をクリップボードにコピーするボタンと動画ページのサムネイルを設置する (-ver0.47)

<td>コピー</td>の次の行から、実際に何をコピーするかの定義が始まります。ボタンを追加したい場合は行ごとコピー&ペーストで増やして、シングルクォーテーション(')で挟まれている値をお好みの内容に変更してください。このページの上のほうにある[パラメータ]の項を参考にするとよいです。

```
<!-- NicoPlayer Option="-open_inside" -->
<html>
<head>
<script type="text/javascript">
function copyTextToClipboard( value )
{
clipboardData.setData( "Text", value );
}
</script>
</head>
<body>
<table border="1">
<tr>
<td>コピー</td>
<td><button onclick="copyTextToClipboard( 'http://www.nicovideo.jp/watch/%__MOVIE_NAME__' )">アドレス</button></td>
<td><button onclick="copyTextToClipboard( '%__MEDIA_PATH2_%' )">ファイル</button></td>
<td><button onclick="copyTextToClipboard( '%__MOVIE_NAME__' )">ID</button></td>
</tr>
</table>
<br />
<iframe src="http://www.nicovideo.jp/thumb/%__MOVIE_NAME__" scrolling="no"
width="350" height="180" style="border:solid 1px #000ccc;" frameborder="0">
</body>
</html>
```

携帯動画変換君に投げ込むボタンを設置する・MP3ファイルを抽出するボタンを設置する (-ver0.41)

"携帯動画変換君のパス"という所を、各自の環境に合うよう変更してください(パスの区切り文字はではなく、もしくは/を使ってください)。前提として、動画ファイルをドラッグ&ドロップするだけで変換できるようになっていなければなりません。MP3ファイル抽出を追加しました。"FFMPEGのパス(\\ffmpeg.exeまで含む)"を環境に合わせて変更してください。FFMPEGは[ここ](#)

[ら](#)からどうぞ、追記:保存先フォルダを指定するダイアログを表示するようにしました。

```
<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript">
  var pathOf3GPConverter = "携帯動画変換君のパス";
  var pathOfFfmpeg = "FFMPEGのパス(\\ \ ffmpeg.exeまで含む)";

  function convertWith3GPConverter()
  {
    var shellObject = new ActiveXObject( "WScript.Shell" );
    shellObject.Run( " \" + pathOf3GPConverter + " \ 3GP_Converter.exe \" \ \"%_MEDIA_PATH2_% \" );
    wshShell = null;
  }

  function extractMp3( path )
  {
    var shell = new ActiveXObject( "WScript.Shell" );
    shell.Run( " \" + pathOfFfmpeg + " \" -codec copy -y -i \" + path + " \" \" + path + ".mp3 \" , 0, true );
    shell = null;

    shell = new ActiveXObject( "Shell.Application" );
    var folder = shell.BrowseForFolder( 0, "保存先のフォルダを選択してください", 1, "ssfDESKTOP" );
    if( folder != null ) {
      folder.CopyHere( path + ".mp3" );
    }
  }
</script>
</head>
<body>
<table border="1">
<tr>
<td>変換</td>
<td><button onClick="convertWith3GPConverter()">3GP</button></td>
<td><button onClick="extractMp3( '%_MEDIA_PATH2_%' )">MP3</button></td>
</tr>
</table>
</body>
</html>
```

コメントの推移をグラフ化して表示する

縦軸がコメント数、横軸が投稿日時です。canvasタグの描画にexcanvas.jsを使用しているので[こちら](#)からダウンロードして同じパスにおいてください。
赤線がコメントを表し、青線は100コメントごとの区切りです。

```
<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript" src="excanvas.js"></script>
<script type="text/javascript">
  var canvasWidth = 260;
  var canvasHeight = 200;
  function drawGraph()
  {
    // XMLパーサオブジェクトを生成
    xmlDocObject = new ActiveXObject( "Microsoft.XMLDOM" );
    xmlDocObject.async = false;
    xmlDocObject.load( "%_COMMENT_PATH2_%" );

    // コメント数・投稿日時の尺をキャンバスサイズに合わせる
    var nodes = xmlDocObject.getElementsByTagName( "chat" );
    var firstChatDate = nodes[0].getAttribute( "date" );
    var lastChatDate = nodes[nodes.length - 1].getAttribute( "date" );
    var hscale = ( lastChatDate - firstChatDate ) / canvasWidth;
    var vscale = nodes.length / canvasHeight;

    var context = document.getElementById( "canvas" ).getContext( "2d" );

    // 背景画像を描画
    var imgObj = new Image( canvasWidth, canvasHeight );
    imgObj.src = "http://tn-skr.smilevideo.jp/smile?i=%_MOVIE_ID_%";
    context.drawImage( imgObj, 0, 0 );
```

```

var j = 0;
var table = new Array( 15 );

// コメントの遷移を表示
context.strokeStyle = "rgba( 255, 0, 0, 1 )";
context.moveTo( 0, canvasHeight );
for( var i = 0; i < nodes.length; i++ ) {
    context.lineTo( ( nodes[i].getAttribute( "date" ) - firstChatDate ) / hscale, canvasHeight - ( i / vscale ) );

    if( i % 100 == 99 ) {
        table[j++] = canvasHeight - ( i / vscale );
    }
}
context.stroke();

// コメント数の水準線を表示
context.strokeStyle = "rgba( 0, 0, 255, 1 )";
for( var i = 0; i < j; i++ ) {
    context.moveTo( 0, table[i] );
    context.lineTo( canvasWidth, table[i] );
}
context.stroke();
}
</script>
</head>
<body onload="drawGraph()">
<canvas id="canvas" width="260" height="200" />
</body>
</html>

```

コメントを再生ファイルの時間軸についてグラフ化

横軸が再生時間、縦軸がコメント数(秒単位)です。グラフをクリックすると該当する位置にシーク(再生位置が移動)します。弾幕のありそうなところに当たりをつけて直前に飛ぶ、といった使い方を想定しています。

```

<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript" src="excanvas.js"></script>
<script type="text/javascript">
    var canvasWidth = 260;
    var canvasHeight = 200;

    function drawGraph()
    {
        xmlDocObject = new ActiveXObject( "Microsoft.XMLDOM" );
        xmlDocObject.async = false;
        xmlDocObject.load( "%_COMMENT_PATH2_" );

        // コメントの発言時間を集計(秒単位)
        var vposHash = {}, maxCount = 0, lastVpos = 0;
        var nodes = xmlDocObject.getElementsByTagName( "chat" );
        for( var i = 0; i < nodes.length; i++ ) {
            var nVpos = Math.round( nodes[i].getAttribute( "vpos" ) / 100 );
            if( vposHash[ nVpos ] == undefined ) {
                vposHash[ nVpos ] = 1;
            } else {
                vposHash[ nVpos ]++;
            }
            lastVpos = Math.max( lastVpos, nVpos );
            maxCount = Math.max( maxCount, vposHash[ nVpos ] );
        }

        // スケール決定
        var hscale = lastVpos / canvasWidth;
        var vscale = maxCount / canvasHeight;

        var context = document.getElementById( "canvas" ).getContext( "2d" );

        // 背景画像を描画
        var imageObject = new Image( canvasWidth, canvasHeight );
        imageObject.src = "http://tn-skr.smilevideo.jp/smile?i=%_MOVIE_ID_";
        context.drawImage( imageObject, 0, 0 );

        context.strokeStyle = "rgba( 255, 0, 0, 1 )";

```

```

context.moveTo( 0, canvasHeight );

// グラフを描画
for( var i = 0; i < lastVpos; i++ ) {
  var count = vposHash[i];
  if( count != undefined ) {
    context.lineTo( i / hscale, canvasHeight - ( count / vscale ) );
  }
}
context.stroke();
}

function seek()
{
  var pos_percent = Math.round( ( event.offsetX / canvasWidth ) * 100 );
  var shellObject = new ActiveXObject( "WScript.Shell" );
  shellObject.Run( " \ "%__SOFT_DIR2__% \ NicoPlayer.exe \ " player -seek pos_percent=" + pos_percent );
}
</script>
</head>
<body onload="drawGraph()">
<canvas id="canvas" width="260" height="200" onclick="seek()" />
<span />
</body>
</html>

```

コメントを再生ファイルの時間軸についてグラフ化2 (-ver0.57)

こちらではカラーチャートを使ってみました。無茶な計算をしまくっているの、大変重いです。

```

<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<style>
div.canvas { height:30px; filter:blur(strength=2); position:relative; }
span.item { width:2px; height:30px; position:absolute; bottom:0px; }
span.cursor { height:78px; bottom:0px; width:3px; position:absolute; background-color:white; }
</style>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript" src="nicoplayer.js"></script>
<script type="text/javascript">
  var splitNumber = 100;
  var vposMax = 0, vposList = [];
  window.onload = function()
  {
    var xml = new ActiveXObject( "Microsoft.XMLDOM" );
    xml.async = false;
    xml.load( "%__COMMENT_PATH2__%" );
    var nodes = xml.getElementsByTagName( "chat" );
    for( var i = 0; i < nodes.length; i++ ) {
      var vpos = nodes[i].getAttribute( "vpos" ) - 0;
      vposList.push( vpos );
      if( vposMax < vpos ) { vposMax = vpos; }
    }
    vposList = vposList.sort( function( l, r ) { return l - r; } );
    drawChart();
    setInterval( interval, 1000 );
  }

  var drawChart = function() {
    var r = 0, g = 0, b = 0, l = [];
    for( var i = 0; i < 766; i++ ) {
      l.push( r + g + b );
      if( 510 <= i ) { r += 0x010000; g -= 0x000100; }
      if( 255 <= i && i < 510 ) { g += 0x000100; b -= 0x000001; }
      if( i < 255 ) { b += 0x000001; }
    }

    for( var i = 0; i < 30; i++ ) {
      ls.push( l[i * 20 + 185] );
    }

    var target = document.getElementById( "target" );
    var canvas = document.createElement( "div" );
    canvas.className = "canvas";

```

```

var lastIndex = 0; counter = 0;
for( var i = 0; i < splitNumber; i++ ) {
  counter += vposMax / splitNumber;

  var colorIndex = 0;
  for( var j = lastIndex; vposList[j] < counter; j++ ) {
    colorIndex++;
    lastIndex = j;
  }

  if( 29 < colorIndex ) { colorIndex = 29; }
  var item = document.createElement( "span" );
  item.index = i;
  item.className = "item";
  item.onclick = function() {
    np.execCmd( "player -seek percent=" + this.index );
  }
  item.style.backgroundColor = ls[colorIndex];
  item.style.left = i * 2;
  canvas.appendChild( item );
  lastColorIndex = colorIndex;
}
cursor = document.createElement( "span" );
cursor.className = "cursor";
canvas.appendChild( cursor );
target.appendChild( canvas );

timer = setInterval( interval, 1000 );
}

var cursor;
var ls = [];
var interval = function() {
  cursor.style.left = Math.round( splitNumber * ( ( np.getTime() - 0 ) / ( vposMax * 10 ) ) ) * 2;
}
</script>
</head>
<body>
<div id="debug"></div>
<div id="exchange" style="display:none;"></div>
<div id="target"></div>
</body>
</html>

```

プレイリストのサムネイルを列挙し、クリックしたら再生する

プレイリストにサムネイルが表示されたいいな、と思っていたので力づくでやってみました。NicoPlayerの機能は全然使っていません、動画情報ですらないです。巨大なプレイリストで使うと、問題がありそうです。ActiveXを使っているので携帯動画変換君に投げ込むボタンと同じ条件でないと機能しません。

```

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=shift_jis">
<script language="javascript">
<!--
function listupThumbnails()
{
  var strtt = "", fso = new ActiveXObject( 'Scripting.FileSystemObject' );
  var file = fso.OpenTextFile("NicoPlayerのインストールパス \ default.m3u");
  splited = file.ReadAll().match( /sm[0-9]+/ig );
  for( var i = 0; i < splited.length; i++ )
  {
    strtt += "<img src=\ "http://tn-skr.smilevideo.jp/smile?i=" + splited[i].match( /[0-9]+/ ) + "\ " ";
    strtt += " onClick=\ "play(\ '" + splited[i] + "\ ' ); \ ">";
    strtt += ( i % 5 == 4 ) ? "<br>\n" : "\n";
  }
  document.getElementsByTagName( "span" )[0].innerHTML = strtt;
}

function play( strg )
{
  var fso = new ActiveXObject( 'Scripting.FileSystemObject' );
  var filesCollection = fso.GetFolder( "動画のダウンロードパス" ).Files;
  for( var file = new Enumerator( filesCollection ); !file.atEnd(); file.moveNext() )
  {

```

```

var fileName = file.item().Name;
if( fileName.match( new RegExp( strg + ".*\ .flv" ) ) )
{
var wshShell = new ActiveXObject( 'WScript.Shell' );
wshShell.Run( "NicoPlayerのインストールパス\ \NicoPlayer.exe" "動画のダウンロードパス\ \ ' + fileName + ' \ ' );
return;
}
}
}
}
//-->
</script>
</head>
<body onLoad="listupThumbnails();">
<span/>
</body>
</html>

```

動画ページへのリンクを改造する(-ver0.49)

動画タイトル・再生/コメント/マイリスト登録数をツールチップとして表示しつつ、ダウンロード済みなら青文字・そうでないなら赤文字にします。それぞれクリックすることで再生・ダウンロードが始まります。うまく表示されない場合や、ダウンロード後リンクから再生したい場合はF5で再表示してください。

```

<!-- saved from url=(0017)http://localhost/ -->
<script type="text/javascript">
window.onload = function() {
for( var i = 0; i < document.links.length; i++ ) {
var link = document.links[i];
link.onclick = new Function( "return false;" );
if( link.href.match( /watch\/(sm[0-9]+)/ ) ) {
link.id = RegExp.$1;
var http = new ActiveXObject( "Msxml2.XMLHTTP" );
http.open( "GET", "http://www.nicovideo.jp/thumb/" + RegExp.$1, true );
http.onreadystatechange = function() {
if( http.readyState == 4 && http.status == 200 ) { onpage( http ); }
}
http.send();
}
}
}

onpage = function( response ) {
var text = response.responseText;
var name = text.match( /watch\/(sm\d+)/ ) ? RegExp.$1 : "";
var play = text.match( /再生:<strong>([0-9,]*)/ ) ? RegExp.$1 : "";
var title = text.match( /<img alt="([^\"]+)/ ) ? RegExp.$1 : "";
var mylist = text.match( /マイリスト:<strong>([0-9,]*)/ ) ? RegExp.$1 : "";
var comment = text.match( /コメント:<strong>([0-9,]*)/ ) ? RegExp.$1 : "";

var link = this[name];
link.title = title + "\n再生:" + play + " コメント:" + comment + " マイリスト:" + mylist;

var path = "%_DOWNLOAD_DIR2_%\ \ " + title + "(" + name + ").flv";
var url = "http://www.nicovideo.jp/watch/" + name;
with( exist( path ) ? {href:path, color:"blue"} : {href:url, color:"red"} ) {
link.href = href;
link.style.cssText = "color:" + color + ";";
}
link.onclick = new Function( "exec(this);return false;" );
}

exist = function( path ) {
if( !this.fs ){ this.fs = new ActiveXObject( "Scripting.FileSystemObject" ); }
return this.fs.FileExists( path );
}

exec = function( node ) {
( new ActiveXObject( "WScript.Shell" ) )
.Run( "%_SOFT_DIR2_%\ \NicoPlayer.exe" " " + node.href + " -inactive" );
}
</script>

```

マイリスト・タグ検索をTubePlayerっぽく表示する(-ver0.47)

マイリストやタグ検索のリンクをクリックとウィンドウをポップアップしてTubePlayerっぽく一覧表示します。文字が青色ならすでにダウンロード済みなので、クリックするとすぐに再生できます。そうではなく赤色だとまだダウンロードしておらず、クリックするとダウンロードを開始します。ウィンドウを閉じるときは右上の×をクリックしてください。EscapeUTF8は[こちら](#)からお借りしました(感謝(-人-))。

ランキング・検索については、[スクリプト](#)の[検索・ランキングを一覧表示する]へ移動しました。

```
<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<style>
#template { position:absolute; visibility:hidden; }
#target { position:absolute; top:0pt; left:0pt; width:100% }
#table { font-size:9pt; width:100%; background-color:white }
th { background-color:#000000; color:#ffffff; }
.even { background-color:#ddddd; cursor:pointer }
.odd { background-color:#ffffff; cursor:pointer }
</style>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
var pathOfNicoPlayer = "%_SOFT_DIR_%";
var pathOfDownload = "%_DOWNLOAD_DIR_%";

// マイリストアイテム構造( 情報名:{ tag:タグ名 index:位置 param:パラメタ(省略時innerHTML) regex:正規表現}
// ex)date:{ tag:"strong", index:0 } > 0番目のstrongタグの内容
var mylistStructures = { root:{ tag:"table", index:6 }, block:{ tag:"tr" }, values:{
  date:{ tag:"strong", index:0 }, time:{ tag:"strong", index:1 }, title:{ tag:"a", index:1 }
  , comment:{ tag:"strong", index:3 }, play:{ tag:"strong", index:2 }
  , name:{ tag:"a", index:1, param:"href", regex:"(sm[0-9]+)" } }
}

//タグアイテム構造
var tagStructures = { root:{ tag:"table", index:8 }, block:{ tag:"td" }, values:{
  time:{ tag:"strong", index:0 }, comment:{ tag:"strong", index:2 }, title:{ tag:"a", index:1 }
  , play:{ tag:"strong", index:1 }, mylist:{ tag:"strong", index:3 }
  , name:{ tag:"a", index:1, param:"href", regex:"(sm[0-9]+)" } }
}

var localFlvFiles;

var $_ = function( target, key ) {
  return $( target.getElementsByTagName( key ) );
}

// 検索文字エンコード
EscapeUTF8=function(str){
  return str.replace(/[\^*+.-9A-Z_a-z-]/g,function(s){
    var c=s.charCodeAt(0);
    return (c<16?"%"+c.toString(16):c<128?"%"+c.toString(16)
    :c<2048?"%"+(c>>6|192).toString(16)+"%"+(c&63|128).toString(16)
    :"%"+(c>>12|224).toString(16)+"%"+(c>>6&63|128).toString(16)+"%"+
    +(c&63|128).toString(16)).toUpperCase()
  });
};

// リンク編集
var decorateLinks = function() {
  searchFlvFiles();
  $( document.links ).each( function( link ) {
    link.onclick = new Function( "return false;" );
    if( link.href.match( /mylist \\/( \d+ ) / ) ) {
      link.onclick = new Function( "showDecoratedMylist(this);return false;" );
    } else if( link.href.match( /tag \\/(.+) / ) ) {
      link.onclick = new Function( "showDecoratedTag(this);return false;" );
    }
  } );
}

// マイリストページ取得要求
var showDecoratedMylist = function( link ) {
  var url = link.href;
  new Ajax.Request( url, { method : "get", onComplete : function( request, args ) {
    parseBase( request, args, mylistStructures );
  } } );
}
```

```

}

// タグページ取得要求
var showDecoratedTag = function( link ) {
  var url = "http://www.nicovideo.jp/tag/" + EscapeUTF8( link.innerHTML );
  new Ajax.Request( url, { method : "get", onComplete : function( request, args ) {
    parseBase( request, args, tagStructures );
  } } );
}

// ページ解釈
var parseBase = function( response, args, structures ) {
  var container = document.createElement( "div" );
  container.innerHTML = response.responseText;
  with( structures ) {
    var rootNode = $( container, root.tag )[root.index];
    var items = parse( $( rootNode, block.tag ), values );
  }
  showTable( items );
}

// アイテム抽出
parse = function( itemNodes, structures ) {
  var tempList = {};
  itemNodes.each( function( node ) {
    var templItem = {};
    $H( structures ).keys().each( function( key ) {
      var tag = structures[key].tag;
      var index = structures[key].index;
      var param = structures[key].param;
      var regex = structures[key].regex;
      var temp = $( node, tag )[index][param ? param : "innerHTML"];
      if( regex ) {
        RegExp( regex, "" ).exec( temp );
        temp = RegExp.$1;
      }
      templItem[key] = temp;
    } );
    tempList[templItem.name] = templItem;
  } );
  return tempList;
}

// テーブル表示
var showTable = function( items ) {
  var table = $( "template" ).cloneNode( true );
  table.setAttribute( "id", "table" );
  $H( items ).each( function( pair, index ) {
    var tr = document.createElement( "tr" );
    tr.className = ( index % 2 ) ? "odd" : "even";
    tr.style.cssText = "color:" + ( localFlvFiles[pair.key] ? "blue;" : "red;" );
    tr.setAttribute( "onclick", "action( '" + pair.value.name + "' );" );
    // tr.onclick = new Function( "action( '" + item.name + "' );" );
    var td = document.createElement( "td" );
    var text = document.createTextNode( index + 1 );
    td.appendChild( text );
    tr.appendChild( td );
    var keys = $( new Array( "time", "play", "comment", "mylist", "title", "name" ) );
    keys.each( function( key ) {
      var td = document.createElement( "td" );
      var value = pair.value[key];
      var text = document.createTextNode( value ? value : "-" );
      td.appendChild( text );
      tr.appendChild( td );
    } );
    table.appendChild( tr );
  } );
}

// 表示(再描画のため空文字列""を追加挿入)
$( "target" ).innerHTML = "";
$( "target" ).appendChild( table );
$( "target" ).innerHTML += "";
}

// ダウンロード済みFLVファイル検索
var searchFlvFiles = function() {
  if( !localFlvFiles ) {

```

```

    localFlvFiles = $H();
    var fs = new ActiveXObject( "Scripting.FileSystemObject" );
    var files = new Enumerator( fs.GetFolder( pathOfDownload ).Files );
    while( !files.atEnd() ) {
        if( files.item().Name.match( /(sm\d+).*\.flv/ ) ) {
            localFlvFiles[RegExp.$1] = files.item();
        }
        files.moveNext();
    }
}
}
}

// NicoPlayerコマンド実行
execBuildupCommand = function( argument ) {
    var shellObject = new ActiveXObject( "WScript.Shell" );
    shellObject.Run( " \" " + pathOfNicoPlayer + " \ NicoPlayer.exe \" " + argument + " -inactive", 0, true );
    shellObject = null;
}

// ダウンロード済みなら再生・そうでないならダウンロード
action = function( movieName ) {
    if( localFlvFiles[movieName] ) {
        execBuildupCommand( " \" " + localFlvFiles[movieName].Path + " \" " );
    } else {
        execBuildupCommand( "http://www.nicovideo.jp/watch/" + movieName );
    }
}
</script>
</head>
<body onload="decorateLinks();" >
%__COMMENT__%
<br />
%__TAGS__%
<table id="template">
<thead>
<tr>
<th> </th><th>Time</th><th>P</th><th>C</th><th>M</th><th>Title</th>
<th align="right"><span style="cursor:pointer" onclick="$('target').innerHTML='';" x </span></th>
</tr>
</thead>
<tbody />
</table>
<div id="target"></div>
<div id="debug"></div>
</body>
</html>

```

投稿者コメントのニコニコ動画内リンク(動画・マイページ)をクリックするとサムネイルを表示する (-ver0.42)

ブログパーツと同じ表示です。次にどの動画に飛ぶかな、というときの参考にどうぞ。表示した内容をさらに編集かけて、リンクをクリックで再生とかダウンロードとかコメント更新とかにすれば、さらに使い勝手がよくなるのではないのでしょうか。スタイルシートがないとリンクが分からないので最小限入れました。

```

<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<style type="text/css">
span#link{ color:blue; text-decoration:underline; cursor:pointer }
</style>
<link rel="stylesheet" type="text/css" href="http://www.nicovideo.jp/css/thumb.css">
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
function showThumbnail( path )
{
    var url = "http://www.nicovideo.jp/" + path;
    var ajax = new Ajax.Updater( "target", url, { method : "get" } );
    ajax = null;
}

function editCommentLinks()
{
    var comment = $( "comment" );
    var atags = $( comment.getElementsByTagName( "a" ) );
    atags.each( function( atag ) {
        var href = atag.getAttribute( "href" );

```

```

    if( href.match( /watch\/\((sm[0-9]+)\)\/ ) ) {
      atag.outerHTML = "<span id=\"link\" onclick=\"showThumbnail( 'thumb/' + RegExp.$1 + '\" ) \">>\" + RegExp.$1 + \"</span>\" + \"</span>\"";
    } else if( href.match( /mylist\/\([0-9]+\)\([0-9]+\)\/ ) ) {
      atag.outerHTML = "<span id=\"link\" onclick=\"showThumbnail( 'thumb_mylist/' + RegExp.$1 + '\" ) \">>mylist/' + RegExp.$1 + \"</span>\"";
    } );
  } );
</script>
</head>
<body onload=editCommentLinks()>
<span id=comment>%_COMMENT_</span>
<br/>
<span id=target></span>
</body>
</html>

```

プレイリストを検索しリストをリアルタイム更新、該当する行をクリックすると再生

NicoPlayerで表示するとonclickが機能しないので検索ボタンを押してください、ブラウザで見るとリアルタイムで更新します。動画情報でやることではないですが、プレイリストにこんな機能がつくと嬉しいな！という感じで。

```

<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=shift_jis">
<script language="javascript">
  var pathOfNicoPlayer = "NicoPlayerのインストールパス";
  var pathOfDownload = "動画ファイルのダウンロードパス";

  function getPlaylistWithRequest( keyword )
  {
    if( keyword == "" )
    {
      keyword = document.getElementsByTagName( "input" )[0].value;
    }

    var fso = new ActiveXObject( 'Scripting.FileSystemObject' );
    var playlistFile = fso.OpenTextFile( pathOfNicoPlayer + "\\default.m3u", 1 );
    var filePathList = playlistFile.ReadAll().split( "\n" );
    playlistFile.Close();

    var str = "";
    for( var i = 0; i < filePathList.length; i++ )
    {
      var fileName = fso.GetFileName( filePathList[i] );
      var regex = new RegExp( keyword );
      if( fileName.match( regex ) )
      {
        str += "<span onClick= \"play( this.innerHTML ) \">>\" + fileName + "</span><br>";
      }
    }
    document.getElementsByTagName( "span" )[0].innerHTML = keyword + str;
  }

  function play( fileName )
  {
    var wshShell = new ActiveXObject( 'WScript.Shell' );
    wshShell.Run( pathOfNicoPlayer + "\\NicoPlayer.exe \"\" + pathOfDownload + "\\\" + fileName + "\" );
  }
</script>
</head>
<body onload=getPlaylistWithRequest( '.' );>
<input type="text" onkeyup=getPlaylistWithRequest( this.value )>
<input type="submit" value = "検索" onclick=getPlaylistWithRequest( '' )>
<br/>
<span/>
</body>
</html>

```

動画ページを全て表示する

ページ総行数制限に引っかかったため削除。(2008/01/02 13:33)

動画ページを全て表示する2 (-ver0.41)

上記[動画ページを全て表示する]と違って公式Flashも機能します。適当なポータルサイトを指定すると、便利になるかもしれません。ちなみに<meta http-equiv="refresh" content="0;url=http://~">を使うと新規ウィンドウを開こうとし、標準ブラウザに制御が

移ってしまいます。

```
<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript">
function redirect()
{
var shellObject = new ActiveXObject( "WScript.Shell" );
// 動画ページ直通(動画情報ウィンドウの中で動画が再生される一発芸)
shellObject.Run( " \ "%_SOFT_DIR2_% \ NicoPlayer.exe \ " videoinfo -move href= \ "http://www.nicovideo.jp/watch//%_MO
// ブログパーツ(サムネイル・再生数やコメント数など、こちらのほうが実用的)
// shellObject.Run( " \ "%_SOFT_DIR2_% \ NicoPlayer.exe \ " videoinfo -move href= \ "http://www.nicovideo.jp/thumb/%_
shellObject = null;
}
</script>
</head>
<body onload="redirect()" />
</html>
```

プレイリスト一覧を表示し切り替える (-ver0.41)

プレイリストファイルを気軽に切り替えられます。[スクリプト](#)や上記の拡張を使ってマイページやタグ、検索ワードごと大量のプレイリストを作成している場合特に有効です。CSS要素を省くと異様にしょぼくなりましたが仕様です。

```
<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
function execBuildupCommand( argument )
{
var shellObject = new ActiveXObject( "WScript.Shell" );
// 0:ウィンドウ非表示(-6まであり) true:終了待機フラグ
shellObject.Run( " \ "%_SOFT_DIR2_% \ NicoPlayer.exe \ " " + argument, 0, true );
shellObject = null;
}

function showPlaylists()
{
var str = "<table>";
var fsObject = new ActiveXObject( "Scripting.FileSystemObject" );
var files = fsObject.GetFolder( "%_SOFT_DIR2_%" ).Files;
for( var file = new Enumerator( files ); !file.atEnd(); file.moveNext() ) {
var fileName = file.item().Name;
if( fsObject.GetExtensionName( fileName ) == "m3u" ) {
str += "<tr><td onclick=\ "setPlaylist( ' " + file.item().Path.gsub( /\ \ /, " \ \ \ \" ) + " ' ) \ " >" + fileName +
}
}
$( "target2" ).innerHTML += str + "</table>";
files = null;
}

function setPlaylist( val )
{
execBuildupCommand( "playlist -clear" );
execBuildupCommand( " \ \" + val + " \ " -addlist" );
}
</script>
</head>
<body>
<span id="target1" onclick="showPlaylists()">プレイリスト</span>
<br />
<span id="target2"></span>
</body>
</html>
```

公式のNGリストを取得し表示する (-ver0.41)

それだけです。これといって有効な連携が思い浮かばなかった....

```
<html>
<head>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
```

```

function showIgnorelist()
{
  var xml = new ActiveXObject( "Microsoft.XMLDOM" );
  xml.async = false;
  xml.load( "http://www.nicovideo.jp/api/configurengclient?mode=get" );
  var nodes = $( xml.getElementsByTagName( "source" ) );

  var strt = "";
  nodes.each( function ( node ) {
    strt += node.text + " ";
  } );
  $( "target" ).innerHTML = strt;

  xml = null;
}
</script>
</head>
<body onload="showIgnorelist()">
  <span id="target"></span>
</body>
</html>

```

ダウンロード済みのコメントファイルを結合する (-ver0.42)

1000コメントと500コメントのファイルがあれば、1500コメントがまとまった1つのファイルが出来ます(プレイリストに追加登録)。本家開発とかぶってる感ふんぷんですが、どんな感じになるのか先行体験してみたいかたはどうぞ。

```

<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript" src="nicoplayer.js"></script>
<script type="text/javascript">
  function execBuildupCommand( argument )
  {
    var shellObject = new ActiveXObject( "WScript.Shell" );
    shellObject.Run( " \ "%_SOFT_DIR2_% \ NicoPlayer.exe \ " " + argument, 0, true );
    shellObject = null;
  }

  function uniteComment()
  {
    // コメントファイルの検索
    var filePath = [];
    var fsObject = new ActiveXObject( "Scripting.FileSystemObject" );
    var files = fsObject.GetFolder( "%_DOWNLOAD_DIR2_%" ).Files;
    for( var file = new Enumerator( files ); !file.atEnd(); file.moveNext() ) {
      var item = file.item();
      if( item.Name.match( /\ \ (%_MOVIE_NAME_% \ ) \ [.\+\ ] \ .xml/ ) ) {
        filePath.push( item.Path );
      }
    }

    // 読み込み
    var nodes = [], threadNode, viewCounterNode;
    xml = new ActiveXObject( "Microsoft.XMLDOM" );
    xml.async = false;

    filePath.each( function( filePath ) {
      xml.load( filePath );
      nodes = nodes.concat( $( xml.getElementsByTagName( "chat" ) ) );
    } );
    threadNode = xml.getElementsByTagName( "thread" )[0];
    viewCounterNode = xml.getElementsByTagName( "view_counter" )[0];
    xml = null;

    // 重複除去
    var unique = {};
    nodes = nodes.findAll( function ( node ) {
      var no= node.getAttribute( "no" );
      if( unique[no] ) {
        return false;
      }
      unique[no] = true;
      return true;
    } );

```

```

// ソート
nodes.sortBy( function( node ) {
    return parseInt( node.getAttribute( "no" ) );
} );

// 新規コメントファイル作成
xml = new ActiveXObject( "Microsoft.XMLDOM" );
var packetNode = xml.createElement( "packet" );
xml.appendChild( packetNode );
// $( "target" ).innerHTML = nodes.last().getAttribute( "no" );
threadNode.setAttribute( "last_res", nodes.last().getAttribute( "no" ) );
packetNode.appendChild( threadNode );

packetNode.appendChild( viewCounterNode );

nodes.each( function( node ) {
    packetNode.appendChild( node );
} );
var newCommentPath = "%_DOWNLOAD_DIR2_% \ \%_TITLE_%(%_MOVIE_NAME_%).xml"
xml.save( newCommentPath );
packetNode = null;
xml = null;

// プレイリストに追加
execBuildupCommand( " \ " + newCommentPath + " \ " -addlist" );
}
</script>
</head>
<body>
<button onclick="uniteComment()">コメントを結合する</button>
</body>
</html>

```

過去のコメントを取得する (-ver0.42)

ページ総行数制限に引っかかったため削除, (2008/01/02 13:30)

一つ前のコメントを取得する (-ver0.44)

今再生している、一番最初のコメントの日時を使って取得します。日時指定をする必要がない点を除けば、過去のコメントを取得する、とほとんど同じです(prototype.jsのAJAXオブジェクトを使って書き換えました)。

```

<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
    $_ = function( target, key ) { return $( target.getElementsByTagName( key ) ); }
    $_A = function( target, key ) { return target.getAttribute( key ); }

function execBuildupCommand( argument )
{
    var shellObject = new ActiveXObject( "WScript.Shell" );
    shellObject.Run( " \ "%_SOFT_DIR2_% \ NicoPlayer.exe \ " " + argument, 0, true );
    shellObject = null;
}

var parametersHash = $H();
var when;

show = function() {
    var url, param, vid = "%_MOVIE_NAME_%";

// 動画ページへアクセス
url = "http://www.nicovideo.jp/watch/" + vid;
new Ajax.Request( url, { method : "get", asynchronous : false } );

// ダウンロード元サーバ・スレッド番号・ユーザIDを取得
url = "http://www.nicovideo.jp/api/getflv?v=" + vid;
new Ajax.Request( url, { method : "get", asynchronous : false, onComplete : getParameters } );

// キー取得
url = "http://www.nicovideo.jp/api/getwaybackkey?thread=" + parametersHash["thread_id"];
new Ajax.Request( url, { method : "get", asynchronous : false, onComplete : getParameters } );

// 取得時間取得

```

```

var xml = new ActiveXObject( "Microsoft.XMLDOM" );
xml.load( "%__COMMENT_PATH2__%" );
when = $A( $( xml, "chat" )[0], "date" );
xml = null;

// コメント取得
param = "<thread res_from= \"-1000 \" \"
+ \" when= \" \" + when + \" \" \"
+ \" waybackkey= \" \" + parametersHash[\"waybackkey\"] + \" \" \"
+ \" user_id= \" \" + parametersHash[\"user_id\"] + \" \" \"
+ \" version= \"20061206 \" \"
+ \" thread= \" \" + parametersHash[\"thread_id\"] + \" \" \"
+ \"/>";
url = parametersHash["ms"];
new Ajax.Request( url, { postBody : param, asynchronous : false, onComplete : getComment } );
}

// パラメタをハッシュへ格納
function getParameters( request ) {
var parametersList = $( unescape( request.responseText ).split( "&" ) );
parametersList.each( function( parameter ) {
if( parameter.match( /([^\=]+)=(.+)/ ) ) {
parametersHash[RegExp.$1] = RegExp.$2;
}
} );
}

function getComment( request ) {
var xml = new ActiveXObject( "Microsoft.XMLDOM" );
xml.loadXML( request.responseText );

var date = new Date();
date.setTime( when * 1000 );
var path = "%__DOWNLOAD_DIR2__\" \" \%__TITLE__\" \" \%__MOVIE_NAME__\" [\"
+ date.getFullYear() + \".\" + date.getMonth() + \".\" + date.getDate() + \" \"
+ date.getHours() + \"時\" + date.getMinutes() + \"分\".xml\";

xml.save( path );
xml = null;

// プレイリストに追加
execBuildupCommand( \" \" + path + \" \" -addlist\" );
}
</script>
</head>
<body>
<button onclick=\"show()\">取得</button><br />
<span id=\"target\"></span>
</body>
</html>

```

コメントをこっそり投稿する (-ver0.45)

ページ総サイズ(50KB)制限に引っかかったため削除。(2008/01/10 13:04)

NicoPlayerを操作する各種ボタンを取り付ける (-ver0.46)

[起動オプション](#)のsendmsgを用いてNicoPlayerを操作しています。

今後のバージョンアップでコマンドのIDが変更され動作しなくなる可能性があるので注意して下さい。

```

<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<meta http-equiv=\"MSThemeCompatible\" content=\"yes\">
<script type=\"text/javascript\">
function sendCommand( commandId )
{
var shellObject = new ActiveXObject( \"WScript.Shell\" );
shellObject.Run( \" \" \%__SOFT_DIR2__\" \" NicoPlayer.exe \" sendmsg msg=273 wparam= \" + commandId + \" -inactive\" );
}
</script>
</head>
<body>
<button onclick=\"sendCommand( 8300 )\">再生/一時停止</button>
<button onclick=\"sendCommand( 8301 )\">停止</button>

```

```

<button onclick="sendCommand( 8310 )" ">前へ</button>
<button onclick="sendCommand( 8311 )" ">次へ</button>
<button onclick="sendCommand( 8332 )" ">速度 減</button>
<button onclick="sendCommand( 8331 )" ">速度 増</button>
<button onclick="sendCommand( 8333 )" ">速度 標準</button>
<button onclick="sendCommand( 8307 )" ">ミュート</button>
<br>
<button onclick="sendCommand( 8800 )" ">プレイリスト</button>
<button onclick="sendCommand( 8810 )" ">ダウンロードウィンドウ</button>
<button onclick="sendCommand( 8820 )" ">動画情報ウィンドウ</button>
<button onclick="sendCommand( 8830 )" ">コメントリスト</button>
<button onclick="sendCommand( 8860 )" ">情報ウィンドウ</button>
<br>
<button onclick="sendCommand( 8870 )" ">画面 標準</button>
<button onclick="sendCommand( 8871 )" ">画面 1x</button>
<button onclick="sendCommand( 8872 )" ">画面 2x</button>
<button onclick="sendCommand( 8874 )" ">シンプルモード</button>
<button onclick="sendCommand( 8875 )" ">フルスクリーン</button>
<button onclick="sendCommand( 8880 )" ">最前面表示切り替え</button>
<br>
<button onclick="sendCommand( 8530 )" ">コメントの枠表示</button>
<button onclick="sendCommand( 8879 )" ">色調を反転</button>
<br>
<button onclick="sendCommand( 8500 )" ">コメント 表示/非表示</button>
<button onclick="sendCommand( 8600 )" ">コメント表示量 自動</button>
<button onclick="sendCommand( 8610 )" ">コメント表示量 全て</button>
<button onclick="sendCommand( 8510 )" ">フィルタ 有効/無効</button>
<br>
<button onclick="sendCommand( 8350 )" ">通常再生</button>
<button onclick="sendCommand( 8351 )" ">繰り返し再生</button>
<button onclick="sendCommand( 8352 )" ">連続再生</button>
<button onclick="sendCommand( 8353 )" ">ランダム再生</button>
<br>
<button onclick="sendCommand( 8322 )" ">元URLを開く</button>
<button onclick="sendCommand( 8323 )" ">最新コメントをダウンロードする</button>
<button onclick="sendCommand( 8324 )" ">最新情報をダウンロードする</button>
</body>
</html>

```

オススメを表示する (-ver0.46)

```

<!-- saved from url=(0017)http://localhost/ -->
<html>
<head>
<link rel="stylesheet" type="text/css" href="http://res.nicovideo.jp/css/thumb.css">
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
show = function() {
var xml = new XMLHttpRequest( "Microsoft.XMLDOM" );
xml.async = false;
xml.load( "http://www.nicovideo.jp/api/getrelation?video=%__MOVIE_NAME__%" );
$( xml.getElementsByTagName( "url" ) ).each( function( url ) {
var url = url.text.replace( "watch", "thumb" );
new Ajax.Request( url, { method:"get", onComplete:function( response ) {
$( "target" ).innerHTML += response.responseText;
}
} );
} );
} );
} );
</script>
</head>
<body onload="show();" >
<div id="target"></div>
</body>
</html>

```

過去ログ

[動画情報ウィンドウ 機能拡張サンプルII](#)

コメント

- みんな爆笑してるな~、と思いつつ違和感を感じてブラウザで開いたらフィルタのwでした...というわけで[ニコフィルター(ry)]を追加。解釈が力押しな上にマッチングがだめすぎますが(;'`)既知>英数字1文字以外無反応(当然)・文字次第で多分化ける(マスクせず多重変換してるので)。Ver0.50って起動オプションで2度以上ダウンロード(http: ~ 指定)しようとするが無反応になりませんか? - 名無しさん 2008-02-08 02:34:52
- [ニコフィルター(ry)]を若干修正。<や>をエスケープしてますが、ブラウザで表示する関係上戻ってしまうのでgt/ltに置換してください。元の仕様だと文中に含まれるキーワードも置換しますが、ニコプレフィルタでそれをやるとchatやshitaなど制御文字まで置換して動作不能になるので却下(-x-)。またsssssなどキーワードを繰り返すとその回数に応じて置換しますが、正規表現のみで対応できないのでやはり却下(-x-)無理にすると制御文字に累が及ぶ、というわけで、フィルタの置換機能に対象指定を希望! [LastReplace Word]みたいな感じでどうでしょう。もっともそれでも、フィルターは投稿者が編集できるので、再生中の過去コメントと今のフィルタが合致しないと無意味に。コメントファイルと紐付けて保存する機能が本体につくと一番うれしいですが(´`´) どうでしょ。 - 名無しさん 2008-02-10 18:55:39
- 触発(!?)されて投稿者設定のフィルタに対応してみました。フィルタの設定はいろいろな事情で動画情報ファイルに保存しています。置き換えはとりあえずコメントのみを対象とした[Replace Word]を追加しました。 - 輪廻(rinne) 2008-02-10 23:23:15
- こちらのサンプルを二つ以上同時に使用したい場合気をつける点はありますか? どうもうまくいかないのですが。 - moe 2008-02-12 01:41:12
- 一つ一つは矛盾がないようにしてありますが、複数を単純にコピペでくっつけると構文がおかしかったり同じ名前の機能がかぶったり、同じ名前でも微妙に仕様が異なったりします。どれとどれを同時に使おうとされていますか? - 名無しさん 2008-02-12 05:40:24
- 3 対応お疲れ様です。(。・。)/[ニコフィルター(ry)]は早速二軍落ち。いまさらですが、分割してくださった方ありがとうございます。 - 名無しさん 2008-02-12 05:49:25
- 元々のtemplateにオススメを表示するを追加したらおすすみが表示されなかったのですが再設定で表示されました。 - moe 2008-02-17 14:15:40
- バージョンアップで上書きばかりだから内容が古くて調子が悪かったのかtemplate部分をwiki掲載の最新の内容に変えておススメ表示部分を追加したら表示されました。検証に時間がかかりすみませんでした。 - moe 2008-02-17 14:17:53
- コメントグラフ化を導入していたのですが、「<!-- saved 」の1行を消さないとうまく動きませんでした。あとセキュリティの設定はうちの環境ではなぜか必要なかったようです。環境はVista32bit IE8 - 名無しさん 2010-06-15 13:52:25
- getContextエラーでコメントグラフ化が動かないlorz - 名無しさん 2011-02-13 14:56:02

名前:

書き込む