

■ スクリプトによる機能拡張

- [検索・マイリスト・ランキングをTubePlayerっぽく一覧表示しつつ画像の一覧表示もする](#)
- [ローカルファイルを検索し表示](#)
- [マイページに登録されているうち、ダウンロード済みアイテムのプレイリストを作成し開く](#)
- [プレイリストウィンドウにプレイリストファイルをドラッグ&ドロップする\(Cランタイム使用\)](#)

■ コメント

スクリプトによる機能拡張

検索・マイリスト・ランキングをTubePlayerっぽく一覧表示しつつ画像の一覧表示もする

ソースファイルをメモ帳などに貼り付け、拡張子(.hta)でNicoPlayerのインストールフォルダに保存し実行してください(同じ場所にprototype.jsも置いてください)。もし他の場所に起きたい場合はpathOfDownloadとpathOfNicoPlayerをそれぞれの環境に合わせて編集してください(セパレータはスラッシュ/もしくはダブルバックスラッシュ\\です)。タイトルや再生数などのパラメータを表形式及び画像で一覧表示します。フィードバックを強化しました。青字はダウンロード済みでクリックすると再生、赤字はまだでクリックすると該当行を強調してダウンロードを開始します。この強調は再表示で解除されます。TitleやPなどヘッダをクリックすると該当項目でソートします(基本降順以後トグル)。

```
<hta:application maximizebutton="no" selection="yes"
    navigable="no" scroll="no" singleinstance="yes"/>
<html>
<head>
<meta http-equiv="MSThemeCompatible" content="yes">
<style>
#display { border:3px double red; color:red; text-align:center; }
#target { width:100%; height:90%; overflow-y:scroll; }
body { background-color:#f7f7f7; }
div { border:solid 1px #ccc; }
table { font-size:9pt; width:100%; }
th { background-color:black; color:white; cursor:pointer; }
img { width:98%; margin:3px; cursor:pointer; }
.even { background-color:#f7f7f7; cursor:pointer; }
.odd { background-color:#ffffff; cursor:pointer; }
.omo { background-color:#e0ffe0; }
.download { color:red; } .play { color:blue; }
.downloaded { color:blue; background-color:#b7ffb7; }
.downloading{ color:red; background-color:#ffb7b7; }
.playing { color:blue; background-color:#b7b7ff; }
.activet { background-color:#ccc; }
</style>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
var pathOfDownload = ""; // ダウンロードパス(任意)
var pathOfNicoPlayer = ""; // NicoPlayerインストールパス(任意)
var width = 800, height = 800;
var lastSortKey = "", download = {};
var fileTemplate = "#{title}({name}).flv";

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

window.onload = function() {
resizeTo( width, height );

var display = document.getElementById( "display" );
try { Prototype } catch( e ) {
display.innerHTML = "prototype.jsが見つかりません。<a href='http://www.prototypejs.org/'>"
+ "こちら</a>よりダウンロードして同じフォルダにおいてください。";
return;
}
if( !pathOfNicoPlayer ) {
unescape( window.location ).match( /file: \\\/\\\/(.*?)\/ / );
pathOfNicoPlayer = RegExp.$1;
}
if( !ini.load( pathOfNicoPlayer + "/nicoplayer.ini" ) ) {
display.innerHTML = "NicoPlayerが見つかりません。同じフォルダにおいてください。";
return;
}
if( !pathOfDownload ) {
pathOfDownload = ini.download.SavePath;
}
display.innerHTML = "起動中です。しばらくお待ちください。";
```

```

movies.load( pathOfDownload );

Element.hide( "display" );

resizeTo( width, height );
setInterval( "interval();", 5000 );
}

// 周期的(5000ms)にダウンロードリストを走査
interval = function() {
for( name in download ) {
if( movies.exist( name ) ) {
movies[name].className = "downloaded";
var target = $( name );
if( target ) {
with( Element ) {
target.className = target.className.replace( /downloading/g, "" );
target.className += " downloaded";
}
}
delete download[name];
}
}
}

movies = {
init : function() {
init( this, "(init|load|exist)$" );
},
load : function( path ) {
this.init();
var fs = new ActiveXObject( "Scripting.FileSystemObject" );
var files = new Enumerator( fs.GetFolder( path ).Files );
while( !files.atEnd() ) {
if( files.item().Name.match( /((sm|ax)\d+).*\.flv$/ ) ) {
this[RegExp.$1] = { path:files.item().Path, exist:true, className:"play" };
}
files.moveNext();
}
},
exist : function( name ) {
var fs = new ActiveXObject( "Scripting.FileSystemObject" );
if( fs.FileExists( this[name].path ) ) {
this[name].exist = true;
return true;
}
}
}

init = function( node, mask ) {
for( var key in node ) {
if( !RegExp( mask, "" ).test( key ) ) {
delete node[key];
}
}
}

// nicoplayer.iniが対象(一般的なiniファイルにも有効)
ini = {
// 読み込み済み項目があれば削除(操作プロパティinit/loadを除く)
init : function() {
init( this, "(init|load)$" );
},
// iniファイルのパスを引数に取りini以下にセクションと値の組を展開
load : function( path ) {
this.init();
try {
var file = ( new ActiveXObject( "Scripting.FileSystemObject" )
).OpenTextFile( path );
} catch ( e ) {
return;
}
}

var buffer, section;
while( !file.AtEndOfLine ) {
var line = file.ReadLine();
// セクション取得

```

```

if( line.match( /^\[([.+]\$)/ ) ) {
  var newSection = RegExp.$1;
  if( section && !section.match( /^(init|load)\$/ ) ) {
    this[section] = buffer;
  }
  buffer = {};
  section = newSection;
}
// 設定値取得
if( buffer && line.match( /^([^=]+)=(.*)\$/ ) ) {
  buffer[RegExp.$1] = RegExp.$2;
}
}
file.Close();
return this;
}
}

list = {
  list : [],
  index : -1,
  load : function( newlist ) {
    if( 0 <= this.index ) {
      this.list = this.list.slice( 0, this.index + 1 );
    }
    this.list.push( newlist );
    this.index++;
  },
  add : function( addlist ) {
    ( !this.list[this.index] )
    ? ( this.load( addlist ) )
    : ( this.list[this.index] = this.list[this.index].concat( addlist ) );
  },
  reverse : function() {
    if( !this.list[this.index] ) { return; }
    this.list[this.index] = this.list[this.index].reverse();
  },
  back : function() {
    if( 0 < this.index ) { this.index--; }
  },
  forward : function() {
    if( this.index < this.list.length - 1 ) { this.index++; }
  },
  active : function() {
    return this.list[this.index];
  },
  sort : function( key, order ) {
    if( !this.list[this.index] ) { return; }
    if( key == lastSortKey ) {
      this.reverse();
    } else {
      this.list[this.index] = this.list[this.index].sortBy( function( items ) {
        var value = items[key] + "";
        if( key == "title" ) {
          return value;
        } else {
          value = value.gsub( "[^0-9]", "" ) - 0;
        }
        return ( order == "desc" ) ? value : -value;
      } );
      lastSortKey = key;
    }
  }
}

EscapeUTF8=function(str){
  return str.replace(/[^\*+.-9A-Z_a-z-]/g,function(s){
    var c=s.charCodeAt(0);
    return (c<16?"%0"+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()
  })
};

search = function() {
  list.load( [] );
}

```

```

lastSortKey = "";
this.index = 0;
this.timer = setInterval( "search.exec();", 1000 );
this.request = function() {
  if( 3 <= ++this.index ){ clearInterval( this.timer ); search.instance = null; }
  var key = EscapeUTF8( $( "keyword" ) );
  var option = $( "sort" ).value + "&page=" + this.index;
  var url = "http://www.nicovideo.jp/search/" + key + "?" + option;
  new Ajax.Request( url, { method:"get", onSuccess:function( response ) {
    var buffer = document.createElement( "div" );
    buffer.innerHTML = response.responseText;
    var pageGuides = { root:{ tag:"table", index:7 }, blocks:{ tag:"td" } };
    var blocksGuides = [ ["time", "strong", 0], ["play", "strong", 1], ["src", "img", 1, "src"]
      , ["comment", "strong", 2], ["mylist", "strong", 3]
      , ["title", "a", 1], ["name", "a", 1, "href", "((sm|ax) \ [0-9]+)"]
    ];
    var blocks = parsePage( buffer, pageGuides );
    var items = parseBlocks( blocks, blocksGuides );
    list.add( items );
    refreshTable();
  } } );
}
this.request();
}
search.instance;
search.exec = function() {
  if( !this.instance ) {
    this.instance = new this();
  } else {
    this.instance.request();
  }
}

ranking = function() {
  lastSortKey = "";
  var option = [$( "rsort" ), $( "rspan" ), $( "rgenre" )];
  var url = "http://www.nicovideo.jp/ranking/" + option.join( "/" );
  new Ajax.Request( url, { method:"get", onSuccess:function( response ) {
    var buffer = document.createElement( "div" );
    buffer.innerHTML = response.responseText;
    var pageGuides = { root:{ tag:"table", index:7 }, blocks:{ tag:"tr" } };
    var blocksGuides = [ ["time", "strong", 0], ["date", "strong", 1], ["play", "strong", 2]
      , ["comment", "strong", 3], ["target", "p", 1, "([0-9,]+) " ], ["src", "img", 1, "src"]
      , ["title", "a", 1], ["name", "a", 1, "href", "((sm|ax) \ [0-9]+)"]
    ];
    var blocks = parsePage( buffer, pageGuides );
    blocks = blocks.findAll( function( block, index ) { return ( index % 2 ) == 0; } );
    var items = parseBlocks( blocks, blocksGuides );
    list.load( items );
    refreshTable();
  } } );
}

mylist = function() {
  lastSortKey = "";
  var url = "http://www.nicovideo.jp/mylist/" + $( "mylistkeyword" );
  new Ajax.Request( url, { method:"get", onSuccess:function( response ) {
    var buffer = document.createElement( "div" );
    buffer.innerHTML = response.responseText;
    var pageGuides = { root:{ tag:"table", index:6 }, blocks:{ tag:"tr" } };
    var blocksGuides = [ ["date", "strong", 0], ["time", "strong", 1]
      , ["play", "strong", 2], ["comment", "strong", 3], ["src", "img", 1, "src"]
      , ["title", "a", 1], ["name", "a", 1, "href", "((sm|ax) \ [0-9]+)"]
    ];
    var blocks = parsePage( buffer, pageGuides );
    var items = parseBlocks( blocks, blocksGuides );
    list.load( items );
    refreshTable();
  } } );
}

parsePage = function( buffer, guides ) {
  with( guides ) {
    return $( $( buffer, root.tag )[root.index], blocks.tag )
  }
}

```

```

parseBlocks = function( blocks, guides ) {
  var items = [];
  blocks.each( function( block ) {
    var buffer = {};
    guides.each( function( g ) {
      with( { key:g[0], tag:g[1], index:g[2], param:g[3], regex:g[4] } ) {
        var value = $_( block, tag )[index][param ? param : "innerHTML"];
        if( regex ) {
          new RegExp( regex, "" ).exec( value );
          value = RegExp.$1;
        }
        buffer[key] = value;
      }
    } );
    items.push( buffer );
    if( !movies[buffer.name] ) {
      var fileName = ( new Template( fileTemplate ) ).evaluate( buffer );
      movies[buffer.name] = { path : pathOfDownload + "\ \ " + fileName, className : "download" };
    }
  } );
  return items;
}

```

```

lastMode = "string";
var refreshTable = function( mode ) {
  switch( mode ) {
    case "string" :
      $( "rstring" ).className = "activet";
      $( "rimage" ).className = "";
      refreshStringTable();
      lastMode = mode;
      break;
    case "image" :
      $( "rstring" ).className = "";
      $( "rimage" ).className = "activet";
      refreshImageTable();
      lastMode = mode;
      break;
    default :
      refreshTable( lastMode );
      break;
  }
}

```

```

var header = [ [ " ", "index", "asc" ], [ "Title", "title", "desc" ], [ "P", "play", "asc" ]
, [ "C", "comment", "asc" ], [ "M", "mylist", "asc" ]
, [ "Time", "time", "asc" ], [ "ID", "name", "desc" ] ];

```

```

refreshStringTable = function() {
  var ths = "";
  header.each( function( item, index ) {
    ths += ( new Template( "<th onclick='list.sort(\ \"#{key}\ \", \ \"#{order}\ \");refreshTable();'"
+ " onmouseover='this.style.cssText=\ \"background-color:white; color:black \ \""
+ " onmouseout='this.style.cssText=\ \" \ \>#{label}</th>" )
).evaluate( { label:item[0], key:item[1], order:item[2] } ) );
  } );
  var thead = "<thead><tr>" + ths + "</tr></thead>";

  var trs = "";
  list.active().each( function( items, index ) {
    if( !items.index ){ items.index = index + 1; }
    var tds = "";
    header.each( function( value, index ) {
      tds += "<td>" + ( items[value[1]] ? items[value[1]] : "-" ) + "</td>";
    } );
    var trsvalue = { c1 : ( index % 2 ) ? "odd" : "even"
//, c2 : movies[items.name].exist ? " play" : " download"
, className : movies[items.name].className
, omover : "this.className += \" omo \ ";"
, omout : "this.className = this.className.replace(/ omo/g, \ \ );"
, oc : "action( this );", id:items.name
};
    trs += ( new Template( "<tr class='#{c1} #{c2} #{className}' onclick='#{oc}'"
+ " onmouseover='#{omover}' onmouseout='#{omout}' id='#{id}'>" )
).evaluate( trsvalue ) + tds + "</tr>";
  } );
  var tbody = "<tbody>" + trs + "</tbody>";

```

```

$( "target" ).innerHTML = "<table>" + thead + tbody + "</table>";
}

refreshImageTable = function() {
var trs = "<tr>";
var tds = "";
list.active().each( function( items, index ) {
if( !( index % 5 ) ) {
trs += tds + "</tr>";
tds = "<tr>";
}
items.state = movies[items.name].exist ? "playing" : "downloading";
tds += ( new Template( "<td id='#{name}' align='center' class='#{state}' onclick='action( this )'>"
+ "<img src='#{src}' title='#{title}' \n再生:#{play} コメント:#{comment}' /></td>" )
).evaluate( items );
if( index == list.active().length - 1 ) {
trs += tds + "</tr>";
}
} );
var tbody = "<tbody>" + trs + "</tbody>";
$( "target" ).innerHTML = "<table>" + tbody + "</table>";
}

execBuildupCommand = function( argument )
{
var shell = new ActiveXObject( "WScript.Shell" );
shell.Run( "\" + pathOfNicoPlayer + "\\NicoPlayer.exe\" " + argument + " -inactive", 0, true );
shell = null;
}

playingBefore = undefined;
action = function( node ) {
var movieName = node.id;
if( movies[movieName].exist ) {
if( playingBefore ) {
movies[playingBefore].className = "play";
var bnode = $( playingBefore );
if( bnode ) {
bnode.className = bnode.className.replace( / playing/g, "" );
bnode.className += " play";
}
}
movies[movieName].className = "playing";
node.className = node.className.replace( / downloaded/g, "" );
node.className = node.className.replace( / play/g, "" );
node.className += " playing";
playingBefore = movieName;
execBuildupCommand( "\" + movies[movieName].path + \" \" );
} else {
movies[movieName].className = "downloading";
node.className.replace( / download/g );
node.className += " downloading";
download[movieName] = false;
execBuildupCommand( "http://www.nicovideo.jp/watch/" + movieName );
}
}
}
</script>
</head>
<body>
<div id="display">JavaScriptを有効にしてください。</div>
<table id="controller">
<tr><form onsubmit="search.exec();return false;"><td>
<input type="text" id="keyword" value="初音ミク">
<select id="sort">
<option value="">投稿日時が新しい</option>
<option value="order=a">投稿日時が古い</option>
<option value="sort=v" selected>再生が多い</option>
<option value="sort=v&order=a">再生が少ない</option>
<option value="sort=n">コメントが新しい</option>
<option value="sort=n&order=a">コメントが古い</option>
<option value="sort=r">コメントが多い</option>
<option value="sort=r&order=a">コメントが少ない</option>
</select>
<input type="submit" value="検索" />
</td></form><form onsubmit="mylist();return false;"><td>
<input type="text" id="mylistkeyword" value="2949389/2632878" />

```

```



```

ローカルファイルを検索し表示

最上部のテキストボックスにキーワードを入力してTabキーを押すと、ローカルファイルを検索しマッチするファイルをリストアップします。好きなファイルをクリックして再生してください、onchangeではなくonkeyupを使うとインクリメンタル検索になるのですが、このロジックだと重すぎて実用的ではないのでやめました。

```

<hta:application maximizebutton="no" selection="yes"
  navigable="no" scroll="yes" singleinstance="yes"/>
<html>
<head>
<meta http-equiv="MSThemeCompatible" content="yes">
<style>
  ul { list-style-type:none; margin:1px; }
  li { cursor:pointer; width:100%; border:3px double; margin:3px; font-size:12px }
</style>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript">
  var pathOfDownload = "ダウンロードパス"; // ex) D:\ \ Download
  var pathOfNicoPlayer = "NicoPlayerインストールパス";

  var localFlvFiles;

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

  onLoad = function( link ) {
    resizeTo( 500, 800 );
    showList();
  }

  showList = function() {
    searchFlvFiles();

    // 検索
    var keyword = $( "keyword" );
    if( keyword ) {
      localFlvFiles = localFlvFiles.findAll( function( file ) {
        return RegExp( keyword, "i" ).test( file.Name );
      } );
    }
  }

```

```

// リストクリア
var ul = $( "target" );
$( ul.childNodes ).each( function( child ) {
    ul.removeChild( child );
} );

// リスト追加
localFlvFiles.each( function( file, index ) {
    var li = document.createElement( "li" );
    li.setAttribute( "onclick", new Function( "play( \' + index + \' );" ) );
    file.Name.match( /(.*)\.flv$/ );
    var text = document.createTextNode( RegExp.$1 );
    li.appendChild( text );
    ul.appendChild( li );
} );

searchFlvFiles = function() {
    if( !localFlvFiles ) {
        localFlvFiles = $( );
    } else {
        localFlvFiles.clear();
    }

    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.push( files.item() );
        }
        files.moveNext();
    }
}

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

play = function( index ) {
    execBuildupCommand( "\" + localFlvFiles[index].Path + "\" );
}
</script>
</head>
<body onload="onLoad();">
<input type="text" id="keyword" onchange="showList();" />
<ul id="target" />
</body>
</html>

```

マイページに登録されているうち、ダウンロード済みアイテムのプレイリストを作成し開く

ソースファイルをメモ帳などに貼り付け、拡張子(.js)で保存してください。
pathOfNicoPlayerとpathOfDownloadをそれぞれの環境にあったパスに変更してください。
使う時はクリップボードにマイページのアドレス(http://~/0000000/000000:省略不可)を
コピーしてダブルクリックしてください。
またNicoPlayerのプレイリストウィンドウが表示されていれば、自動的に開きます。

```

// 環境設定
var pathOfNicoPlayer = "NicoPlayerのインストールパス(パスセパレータ \\)";
var pathOfDownload = "動画ファイルのダウンロードパス(同上)";

// クリップボードからマイページのアドレスを取得
var ieObject = new ActiveXObject( "InternetExplorer.Application" );
ieObject.Navigate( "about:blank" );
while ( ieObject.Busy ) {
    WScript.Sleep( 100 );
}
var addressOfMypage = ieObject.Document.parentWindow.clipboardData.getData( "text" );
ieObject.Quit();

// マイページアドレスが取得できなかった場合入力ウィンドウを表示(Excelのインストールが必要)
if( !addressOfMypage.match( /.*\$/([0-9]+)\$/([0-9]+)/ ) ) {

```

```

var excelObject = WScript.CreateObject( "Excel.Application" );
if( excelObject != null )
{
    addressOfMypage = excelObject.InputBox( "マイページのアドレスを入力してください" );
    excelObject.Quit();
}
}

// Msxml2(IE6標準)によりマイページのGETリクエスト送出
var httpObject = WScript.CreateObject( "Msxml2.XMLHTTP" );
httpObject.onreadystatechange = function()
{
    if( httpObject.readyState == 4 )
    {
        getRequestPage( httpObject );
    }
}
// open( , , false )は同期指定(さもないと取得前にプログラムが終了する)
httpObject.open( "GET", addressOfMypage, false );
httpObject.send( "" );

function getRequestPage( httpObject )
{
    // マイページのテキストを検索しID(sm[0-9]+)をリストアップ
    var requestPageText = httpObject.responseText;
    var mypageIdList = requestPageText.match( /sm[0-9]+/g );
    if( mypageIdList == null ) {
        return; // IDが見つからず
    }

    // ダウンロードフォルダを検索し.flvファイルをIDに基づいてリストアップ
    var filesHash = {};
    var fsoObject = WScript.CreateObject( "Scripting.FileSystemObject" );
    var filesCollection = fsoObject.GetFolder( pathOfDownload ).Files;
    for( var file = new Enumerator( filesCollection ); !file.atEnd(); file.moveNext() ) {
        var fileName = file.item().Name;
        if( fsoObject.GetExtensionName( fileName ) == ".flv" ) {
            filesHash[fileName.match( /sm[0-9]+/ )] = file.item().Path;
        }
    }

    // マイページのIDとローカルファイルを紐付けプレイリスト形式にする
    var newPlaylistText = "", newDownloadListText = "";
    for( var i = 0; i < mypageIdList.length; i += 2 ) {
        if( filesHash[mypageIdList[i]] != undefined ) {
            newPlaylistText += filesHash[mypageIdList[i]] + "\n";
        }
    }

    // プレイリストファイル(.m3u)に落とす
    addressOfMypage.match( /.*\:\/\/([0-9]+)\:\/\/([0-9]+)/ );
    var newPlylistFileName = RegExp.$1 + "-" + RegExp.$2 + ".m3u";
    // OpenTextFile( , 2, true )の2は書出指定(読込1・追記8)、trueは新規作成あり
    var pathOfNewPlaylistFile = pathOfNicoPlayer + "\\ " + newPlylistFileName;
    var newPlaylistFile = fsoObject.OpenTextFile( pathOfNewPlaylistFile, 2, true );
    newPlaylistFile.Write( newPlaylistText );
    newPlaylistFile.Close();

    // NicoPlayerが.m3uファイルのD&Dに対応すると、以下の全ては次の1行になります
    // shellObject.Run( "\" + pathOfNicoPlayer + "\\NicoPlayer.exe\" \" \" + pathOfNewPlaylistFile + "\" );

    // プレイリストファイルパスをクリップボードへコピー
    var ieObject = new ActiveXObject( "InternetExplorer.Application" );
    ieObject.Navigate( "about:blank" );
    while ( ieObject.Busy ) {
        WScript.Sleep( 100 );
    }
    ieObject.Document.parentWindow.clipboardData.setData( "text", pathOfNewPlaylistFile );
    ieObject.Quit();

    // NicoPlayerのプレイリストウィンドウに登録する
    var shellObject = WScript.CreateObject( "WScript.Shell" );
    shellObject.AppActivate( "NicoPlayer - プレイリスト" );
    WScript.Sleep( 100 );
    shellObject.SendKeys( "^O" );
    WScript.Sleep( 100 );
    shellObject.SendKeys( "^V" );
}

```

```
WScript.Sleep( 100 );
shellObject.SendKeys( "%O" );
}
```

プレイリストウィンドウにプレイリストファイルをドラッグ&ドロップする(Cランタイム使用)

使い勝手向上のためぜひともD&Dを実装したかったが、スクリプトのみでは実装できなかった。
WindowsAPIをVBAでラップすれば可能だが、あまりに煩雑なので断念した。

[マイページに登録されているうち、ダウンロード済みアイテムのプレイリストを作成し開く]の
V/ NicoPlayerが.m3uファイルのD&Dに対応すると、以下の全ては次の1行になります
以下を次のように置き換える。

```
var shellObject = new ActiveXObject( "WScript.Shell" );
shellObject.Run( " \ランタイム名.exe \ " \ "C:\Windows\NicoPlayer - プレイリスト\" \ " + pathOfNewPlaylistFile +
shellObject = null;
```

ランタイム(正確には違うが)のコードは以下の通り。
開発環境は[Microsoft公式](#)を参考にしてください。

```
#define STRICT
#include <windows.h>
#include <cstring>

// DnD操作におけるOS領域のメモリ構造定義
typedef struct drop_files{
    DWORD pFiles;
    POINT pt;
    bool fNC;
    bool fWide;
} DropFiles;

int main( int argc, char *argv[] )
{
    // パラメータが3個([0]は実行ファイルのパス)未満なら終了
    if( argc < 4 ) {
        return 0;
    }

    // 指定窓が見つからなければ終了
    // ([1]:クラス名・[2]タイトル、空文字列も検索条件、無効にするにはNULL)
    HWND hWndTarget = FindWindowExA( NULL, NULL, argv[1], argv[2] );
    if( !hWndTarget ) {
        return 0;
    }

    // D&D操作のための情報を作成
    DropFiles df = { sizeof( DropFiles ), { 0, 0 }, false, false };
    char *file = argv[3];

    // OS領域のメモリリソースを確保し情報を転送
    HGLOBAL hGlobalMemory = GlobalAlloc( GHND, sizeof( DropFiles ) + strlen( file ) + 1 );
    void *p = GlobalLock( hGlobalMemory );
    memcpy( p, ( void * )&df, sizeof( DropFiles ) );
    memcpy( ( char * )p + sizeof( DropFiles ), ( void * )file, strlen( file ) + 1 );
    GlobalUnlock( hGlobalMemory );

    // 指定窓に通知
    PostMessageA( hWndTarget, WM_DROPFILES, ( WPARAM )hGlobalMemory, 0 );

    // OS領域のメモリリソースを開放
    GlobalFree( hGlobalMemory );

    return 0;
}
```

コメント

- スクリプト実行時の制約を軽減(マウス・キーボードを触ると失敗しやすい等)したが、敷居が大幅に高くなってしまった。やはり本家に特定拡張子(.m3u / .lst等)のD&D対応をお願いしたいところ。 - 名無しさん 2007-11-24 17:58:53
- [ローカルファイルを検索し表示]を追加。こちらはサイズ制限が先の話なので調子に乗ってます (^ - ^)。膨張したライブラリの任意再生にどうぞ。 - 名無しさん 2008-01-15 02:54:44

[検索を表示する]を追加。パーサをシンプルにしようとしたら余計にカオスに…。 - 名無しさん 2008-01-19 16:33:52

- [検索・ランキング・マイページをTubePlayerっぽく一覧表示する]でNicoPlayer以外で開くにはactionの中を書き換えるといいです。ブラウザで開くならwindow.open("~ watch/" + movieName, "", ""); HTAからSleipnirを開くなら(new ActiveXObject("Sleipnir.API")).NewWindow("~ watch/"+movieName,true);でいけます。ちなみに拡張子を.htaではなく.htm[]で保存してもブラウザでほぼ動きます(ただし警告でまくり、この場合Sleipnirでもwindow.open()で開ける)。 - 名無しさん 2008-01-26 17:29:21
- [TubePlayerっぽく一覧表示]のフィードバック他を変更しました。テキストボックスでEnterがきくようになり、ソートの不具合が改善されました。 - 名無しさん 2008-01-27 11:01:14
- [TubePlayerっぽく一覧表示]にランキングの画像表示を統合しました(マイリスト・検索にも対応)。一覧と画像一覧の作りこみの差が手に取るように分かりますがネ(。 - 名無しさん 2008-01-27 19:20:31
- [TubePlayerっぽく一覧表示]のpathOf ~ を自動取得するようにしてみました(NicoPlayerと同じフォルダにあることが前提)。任意に設定すればスルーするので、好きな場所におきたい方はそちらで、ini以下にnicoplayer.iniを階層維持のまま展開しているので何かに使えるかも、ini.main.RecentOpen0とかini.playlist.ListPathとかとか。 - 名無しさん 2008-01-30 23:08:16
- [TubePlayerっぽく一覧表示]にエラー処理(prototype.jsがない\NicoPlayerが見つからない\JavaScriptが無効)を追加。iniを拡張して全域から参照可能にしました。ini.load(ファイルパス)で読み込み、ini.initで初期化。構造上init/loadセクションはあっても読み飛ばします。必要ならプロパティ名と例外判定/^(init|load)\$/2箇所をいじってください。 - 名無しさん 2008-01-31 21:06:54
- [TubePlayerっぽく一覧表示]のリスト管理を整理。ついでに戻ると進むを追加。検索・マイページ・ランキングを開くと進む、ソートは最後の並びが残ります。戻った状態で新たに開くと、それより先の記録はリセットされます。ブラウザの戻る・進むと同じ挙動です。listに格納する情報を追加すれば、履歴一覧とかも楽にいけます。 - 名無しさん 2008-02-02 02:24:59
- [TubePlayerっぽく一覧表示]のバグバグを修正(検索時ページが更新されない・同じ動画を複数表示するとインデックスが狂う・他)。ダウンロードパスの全検索を起動時のみにし、以後はファイル名きめうちの二段構えにしてファイル数が多い場合体感で分かるほど軽く(ダウンロードが終了すると数秒以内に赤背景>緑背景になりますが、タイトルとファイル名が違う場合(*が含まれる場合など)は再起動するまで認識しません)。フィードバックをページ横断式に変更(同じ動画ならページを切り替えても強調が維持されます)。結果としてデザインとロジックが融合した素敵コードに...行数も500間近...そろそろ分離しないと。 - 名無しさん 2008-02-02 09:13:34

名前:

書き込む