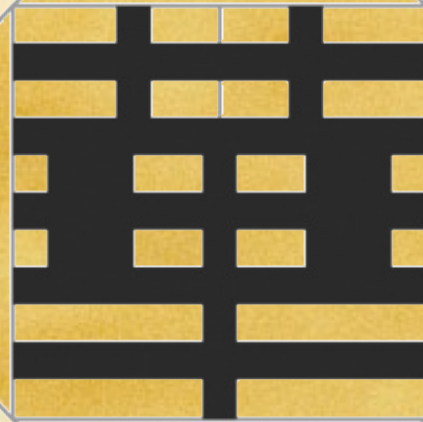


# 梨の季節

MULTI PURPOSE SHIORI SYSTEM



"KAWARI"

2008/11/3

華和梨開発チーム

さとー

Site: <http://kawari.sourceforge.net>  
Blog: <http://d.hatena.ne.jp/satos/>  
Mail: [shobu@users.sourceforge.net](mailto:shobu@users.sourceforge.net)



# アウトライン

☞はじめに

☞初級：いいエディタを使いましょう

☞中級：ログを読みましょう

☞上級：ツールで楽をしましょう

☞上級：ログを活用しましょう

☞デバッグ実演

☞おわりに



# はじめに

❧ 華和梨は難しい？

❧ 実はそうでもないよ！

❧ もっと使って欲しいので、デバッグの  
ノウハウを紹介します

❧ 紹介するノウハウを実演します



初級：  
いいエディタを使いましょう





# いいエディタとは？

- ❧ 華和梨(KIS)の辞書を、notepadで編集していませんか？
- ❧ キーワードや括弧が色付けされると、格段に作業効率が上がります
- ❧ 括弧対応バグは、ほぼこれで原因特定できます
- ❧ ぜひテキストエディタを使いましょう

# 分かりやすさの違い

テキストエディタ

notepad

The screenshot shows the SakuraScript editor with a dark background and syntax highlighting. Red dashed boxes and arrows highlight specific features:

- さくらスクリプト強調表示** (SakuraScript highlighting): Points to the initial comment block.
- ダブルクォート内強調表示** (Double quote highlighting): Points to the initialization of the 'kis' variable.
- コメント強調表示** (Comment highlighting): Points to the function definition for 'setDefaultBase'.
- カーソル位置の対応括弧強調表示** (Bracket highlighting at cursor position): Points to the closing parenthesis of the 'setDefaultBase' function.
- コマンド強調表示** (Command highlighting): Points to the 'setstr' command within the function.

```
95 # 句点、記号など
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111 w4
112 w8
113 w16
114 振動
115 ガツ
116
117
118
119
120
121 # サーフィス定義
122
123 # サーフィスの初期化
124
125 =kis
126 # トーク開始時のためのサーフィス初期化
127 # デフォルト基底関数
128 # 第1引数: 対象キャラのスコープ番号
129 # 第2引数: 対象キャラの基底状態サーフィス番号
130 # 第3引数: 初期設定タグ
131 # 第4引数: デフォルトとして使うキャラクタの表情指定エントリ名
132 function デフォルト基底 $(
133     if ${length $@arg[4]} == 0 }
134     setstr @surface "${arg[2]}";
135     setstr @ename "${arg[3]}";
136     setstr @ename $@arg[4];
137 );
138
139 echo $@arg[3];
140 setstr CharScope $@arg[1];
141 echo ${@ename};
142
143 # 華和製で開始するサーフィス初期化
144 function デフォルト $(
145     デフォルト基底 0 0 %p[3]%-1] %p[2] %s[2000] %1 %s[1000] %0 $@arg[1];
146 );
```

The screenshot shows the Notepad editor with a white background and no syntax highlighting. The code is identical to the SakuraScript editor but lacks all visual aids.

```
character.kis - メモ帳
ファイル(E) 編集(E) 書式(O) 表示(V) ヘルプ(H)

# 改行
n : #n$(setstr CRLFflag 1)
# 次のスコープ変更で改行しない
noCR : $(clear CRLFflag)

-----
# 句点、記号など
、 : 、#w4
。 : 。#w8
? : ?#w8
! : !#w8
♪ : ♪#w8
~ : ~#w8
... : ...#w8
..... : .....#w8...#w8
..... : .....#w8...#w8...#w8
.. : ..#w8
... : ...#w8...#w8
..... : .....#w8...#w8...#w8
. : .#w4
w4 : #w4
w8 : #w8
w16 : #w16
振動 : $(if $[ ! ${sw_noshake} ] #!["raise,OnUserStruck"])#w8#w8
ガツ : ガツ$振動



-----
# サーフィス定義
# サーフィスの初期化

-----
# サーフィスの初期化

=kis
# トーク開始時のためのサーフィス初期化
# デフォルト基底関数
# 第1引数: 対象キャラのスコープ番号
# 第2引数: 対象キャラの基底状態サーフィス番号
# 第3引数: 初期設定タグ
# 第4引数: デフォルトとして使うキャラクタの表情指定エントリ名
function デフォルト基底 $(
    if ${length $@arg[4]} == 0 }
        setstr @surface "${arg[2]}";
        setstr @ename "surface";
    ) else $(
        setstr @ename $@arg[4];
    );
    echo $@arg[3];
    setstr CharScope $@arg[1];
    echo ${@ename};
);

# 華和製で開始するサーフィス初期化
function デフォルト $(
    デフォルト基底 0 0 %p[3]%-1] %p[2] %s[2000] %1 %s[1000] %0 $@arg[1];
);

# 里々で開始するサーフィス初期化
function デフォルト里々 $(
```



## 特に…

☞ 括弧対応エラーは、絶対にエディタに探してもらった方がいいです！

☞ 経験者k女史談

# おすすめエディタ

☞ サクラエディタ + おぼろさんの華和梨キーワード色分け設定をお薦めします。

☞ サクラエディタ

☞ [http://members.at.infoseek.co.jp/sakura\\_editor/index.html](http://members.at.infoseek.co.jp/sakura_editor/index.html)

☞ 華和梨キーワード色分け設定（おぼろさん）

☞ <http://kemonomimisippo.hp.infoseek.co.jp/kawari/hw.html>

☞ Site top : <http://>

[kemonomimisippo.hp.infoseek.co.jp/](http://kemonomimisippo.hp.infoseek.co.jp/)





中級:  
ログを読みましょう



# ログは全てを知っています

- ❧ 閉じ括弧がない、タイポで間違ったコマンドを実行している等のミスは、ログですぐ分かります
- ❧ 実は日本語で表示できます
  - ❧ あまり知られていませんが…

# ログを日本語化する方法

☞Kawarirc.kisに、次のように記述する

```
=kis  
# エラーメッセージを日本語 (Shift_JIS) に設定  
rccharset Shift_JIS;  
=end
```

…これだけです！

# ログを日本語化すると…

## ☞ 日本語化前

```
Debugger: on  
SecurityLevel: middle  
C:\WORK\sandbox\errorlog\errordic.kis 8: error: unexpected EOL in  
quoted string.  
C:\WORK\sandbox\errorlog\errordic.kis 13: error: ') ' not found.  
runaway script?
```

## ☞ 日本語化

```
Debugger: on  
SecurityLevel: middle  
C:\WORK\sandbox\errorlog\errordic.kis 8: error: クォート文字列が改  
行に達しました。  
C:\WORK\sandbox\errorlog\errordic.kis 13: error: ') ' がありません。  
インラインスクリプトブロックが正しく閉じられていません。
```

# ログで分かる情報

- ⌘ ダブルクォート等の閉じ忘れ
- ⌘ 括弧の閉じ忘れ
- ⌘ 存在しないコマンドを呼んだ
- ⌘ 存在しないエントリを呼んだ
- ⌘ Etc...

- ⌘ 華和梨同梱の「エラーメッセージ一覧」に、ログに出てくるエラーメッセージの意味が出ています

⌘ <http://kawari.sourceforge.net/8.2.5/errormes.html>



上級:  
ツールで楽をしましょう



# ログファイルを楽しめる

- ❧ 華和梨のログは、ファイルに書き出す方式です
- ❧ 里々、YAYAの専用ログツールと比べて不便だと思っていないですか？
- ❧ 実は「tail」というツールを使うと、専用ログツールと同様に、リアルタイムでログが見れます。
- ❧ キーワード強調も出来ます（一部）



# tail系ツールの紹介

## ☞ Tail for Windows

☞ <http://www.vector.co.jp/soft/winnt/util/se241>

☞ キーワード強調はないが、軽量シンプル

## ☞ Tail for Win32

☞ <http://tailforwin32.sourceforge.net/>

☞ キーワード強調が出来て高機能

☞ 洋物なので英語のみ







上級:  
ログを活用しましょう



# ログは読むだけではありません

- ゴースト側から、ログに情報を残すことができます (logprintコマンド)
- キーとなるエントリの中身をログに書き出すと、デバッグが進みます (「printfデバッグ」法)

# 例

```
sentence: $(
    logprint "ステータス=" ${status};
    if ${ ${status} } == "最悪" ) (
        logprint "→ステータスは最悪";
        ¥0¥s[4]うへーorz¥e
    ) else (
        logprint "→ステータスは通常";
        ¥0¥s[0]がんばろう。¥e
    )
)
```

・キーとなるエントリをログに出力  
・万が一、次のif文が異常動作した時、ログを見れば条件文の動作は担保できる

# バルーンよりログを使いましょう

- ☞ ゴーストのデバッグ情報を、バルーン内に表示する方をよく見かけます
- ☞ バルーン内表示だと、「\」を含む情報表示でエスケープする手間が必要です
  - ☞ ファイルのパス表示で頻出します
- ☞ Tailで見ると、ログはバルーン表示より見やすいです


# 幸水+ログ=最強デバッグ環境

- ❧ 幸水は、KISをゴーストに送って実行できるため、異常動作部分を手動ステップ実行できます
- ❧ ログにステップ実行の中間過程を残していくと、おかしい箇所をほぼ特定できます



実演タイム！





# おわりに

- ❧ プログラムの上達には、いっぱい失敗するのが王道です
- ❧ ログはデバッグの一番の道具です
- ❧ 華和梨で楽しくデバッグしてください



# 参考文献

🌀なごやか。（おば9様）

🌀<http://kemonomimisippo.hp.infoseek.co.jp/>

🌀華和梨の学習総合サイト

🌀サクラエディタの設定もこちらに