

```
#!/usr/local/bin/perl
```

```
# _____  
# | Cosmo Navigator : navi.cgi - 2011/09/08  
# | Copyright (c) KentWeb  
# | http://www.kent-web.com/  
# | _____
```

```
# モジュール宣言
```

```
use strict;  
use CGI::Carp qw(fatalsToBrowser);  
use lib "./lib";  
use Jcode;
```

```
# 設定ファイル認識
```

```
require "./init.cgi";  
my %cf = &init;
```

```
# データ受理
```

```
my %in = &parse_form;
```

```
# 条件分岐
```

```
if ($in{site} > 0) { &jump_site; }  
if ($in{mode} eq 'find') { &find_data; }  
if ($in{mode} eq 'cate') { &cate_page; }  
if ($in{mode} eq 'new') { &new_list; }  
if ($in{mode} eq 'rank') { &rank_list; }  
if ($in{mode} eq 'note') { &note_page; }  
if ($in{mode} eq 'form') { &form_page; }  
if ($in{mode} eq 'edit' || $in{mode} eq 'delete') { &form_mente; }  
&menu_top;
```

```
#-----
```

```
# メニュー画面
```

```
#-----
```

```
sub menu_top {  
    open(IN,"$cf{numfile}") or &error("open err: $cf{numfile}");  
    my $num = <IN>;  
    close(IN);  
  
    my ($all) = (split(/,/,$num))[1];
```

```

# テンプレート読込
open(IN,"$cf{tmpldir}/top.html") or &error("open err: top.html");
my $tmpl = join("", <IN>);
close(IN);

# 管理者登録
if ($cf{useType} == 0) {
    $tmpl =~ s/<!-- form_begin -->.+<!-- form_end -->//s;
}

# 文字置き換え
$tmpl =~ s!/cgi_title!/ $cf{cgi_title}/g;
$tmpl =~ s!/([a-z]+_cgi)!/ $cf{$1}/g;
$tmpl =~ s!/homepage!/ $cf{homepage}/g;
$tmpl =~ s!/all!/$all/g;
$tmpl =~ s!/top_msg!/ $cf{top_msg}/g;

# テンプレート分割
my ($head,$loop,$foot);
if ($tmpl =~ /(.)<!-- loop_begin -->(.)<!-- loop_end -->(.)s) {
    ($head,$loop,$foot) = ($1,$2,$3);
} else {
    &error("テンプレートが不正です");
}

# ヘッダ表示
print "Content-type: text/html¥n¥n";
print $head;

# カテゴリ展開
my ($i,$tmp);
foreach (0 .. $#{$cf{categ}}) {
    $i++;
    my @cat = split(/,/ , $cf{categ}->[$_]);
    my $sub;
    foreach my $s (1 .. $#cat) {
        $sub .= qq|<a href="$cf{navi_cgi}?mode=cate&cat=$_&sub=$s">$cat[$s]</a>, |;
    }
    $sub =~ s/[,$s]+$//;

    if ($i % 2 == 1) { $tmp = $loop; }
}

```

```

$tmp = ~ s|!cate-$i| |<a href="$cf{navi_cgi}?mode=cate&cat=$_">$cat[0]</a>|;
$tmp = ~ s|!sub-$i| |$sub|;

if ($i % 2 == 0) {
    print $tmp;
    $i = 0;
}
}
# 端数あり
if ($i % 2) {
    $tmp = ~ s|!cate-[12]!|&nbsp;/g;
    $tmp = ~ s|!sub-[12]!|&nbsp;/g;
    print $tmp;
}

print $foot;
exit;
}

#-----
# サイト移動
#-----
sub jump_site {
    # 汚染チェック
    $in{site} = ~ s/¥D//g;
    return if (!$in{url} || !$in{site});

    # 正当性チェック
    if ($in{url} !~ /^https?:¥/¥/[¥w-!~*()¥/?¥@&=+¥$,%#]+$/) {
        &error("不正なリクエストです");
    }

    # データ存在チェック
    if (!-e "$cf{cntdir}/$in{site}.dat") {
        open(DAT,"> $cf{cntdir}/$in{site}.dat");
        close(DAT);
        chmod(0666, "$cf{cntdir}/$in{site}.dat");
    }

    # IP 取得
    my $addr = $ENV{REMOTE_ADDR};

```

```

# カウントデータ
open(DAT,"+< $cf{cntdir}/$in{site}.dat");
eval "flock(DAT, 2)";
my $log = <DAT>;

# データ分解
my ($cnt,$ip) = split(/:/, $log);

# カウントアップ
if ($ip ne $addr) {
    $cnt++;
    seek(DAT, 0, 0);
    print DAT "$cnt:$addr";
    truncate(DAT, tell(DAT));
}
close(DAT);

# 移動
if ($ENV{PERLXS} eq "PerlIS") {
    print "HTTP/1.0 302 Temporary Redirection¥r¥n";
    print "Content-type: text/html¥n";
}
print "Location: $in{url}¥n¥n";
exit;
}

#-----
# 新着ページ
#-----
sub new_list {
    open(IN,"$cf{numfile}") or &error("open err: $cf{numfile}");
    my @num = <IN>;
    close(IN);

    chomp($num[1]);
    my @new = split(/./, $num[1]);
    my %new;
    foreach (@new) {
        $new{$_}++;
    }

    my @log;

```

```

open(IN,"$cf{logfile}") or &error("open err: $cf{logfile}");
while (<IN>) {
    my ($no) = (split(/<>))[0];

    if (defined($new{$no})) { push(@log,$_); }
}
close(IN);

# カテゴリ情報
my %cat = &categ;

# テンプレート読込
open(IN,"$cf{tmpldir}/new.html") or &error("open err: new.html");
my $tmpl = join("", <IN>);
close(IN);

# 文字置き換え
$tmpl =~ s!/cgi_title!/$cf{cgi_title}/g;
$tmpl =~ s!/([a-z]+_cgi)!/$cf{$1}/g;
$tmpl =~ s!/homepage!/$cf{homepage}/g;

# テンプレート分割
my ($head,$loop,$foot);
if ($tmpl =~ /(.(+)<!-- loop_begin -->(.(+)<!-- loop_end -->(.(+)/s) {
    ($head,$loop,$foot) = ($1,$2,$3);
} else {
    &error("テンプレートが不正です");
}

# 時間取得
my $now = time;

# ヘッダ表示
print "Content-type: text/html¥n¥n";
print $head;

my $i = 0;
foreach (@log) {
    $i++;
    my ($no,$cat,$scat,$sub,$url,$nam,$eml,$pw,$msg,$tim,$reco) = split(/<>);
    $url = &url_enc($url);

```

```

        my $tmp = $loop;
        $tmp      =~ s|!site-name!|<a href="$cf{navi_cgi}?site=$no&url=$url"
target="$cf{target}">$sub</a>|g;
        $tmp =~ s|!cate!|<a href="$cf{navi_cgi}?mode=cate&cat=$cat">$cat{$cat}{0}</a>|g;
        $tmp      =~ s|!sub!|<a
href="$cf{navi_cgi}?mode=cate&cat=$cat&sub=$scat">$cat{$cat}{$scat}</a>|g;
        $tmp =~ s|!comment!|$msg|g;
        $tmp =~ s|!ico:(¥w+)!||g;
        $tmp =~ s|!num!|$no|g;
        $tmp =~ s|<!-- ico:new -->||g if ($now - $tim <
$cf{new_mark}*24*60*60);
        $tmp =~ s|<!-- ico:recom -->||g if ($reco == 1);
        print $tmp;
    }

    # フッタ
    &footer($foot);
}

#-----
# 人気ランキング
#-----
sub rank_list {
    # ログ展開
    my (@sort,@data);
    open(IN,"$cf{logfile}") or &error("open err: $cf{logfile}");
    while (<IN>) {
        my ($no,$cat,$scat,$sub,$url,$nam,$eml,$pw,$msg,$tim,$reco) = split(/<>/);

        # カウントデータ読み込み
        open(DB,"$cf{cntdir}/$no.dat") || next;
        my $log = <DB>;
        close(DB);

        my ($count) = (split(/:/, $log))[0];

        # アクセス 0 はカウントしない
        next if (!$count);

        push(@data,"$count¥0$no¥0$cat¥0$scat¥0$sub¥0$url¥0$msg¥0$tim¥0$reco");
        push(@sort,$count);
    }
}

```

```

close(IN);

# 降順ソート
@data = @data[sort {$sort[$b] <=> $sort[$a]} 0 .. $#sort];

# カテゴリ情報
my %cat = &categ;

# テンプレート読込
open(IN,"${tmpldir}/rank.html") or &error("open err: rank.html");
my $tmpl = join("", <IN>);
close(IN);

# 文字置き換え
$tmpl =~ s!/cgi_title!/${cf{cgi_title}}/g;
$tmpl =~ s!/([a-z]+_cgi)!/${cf{$1}}/g;
$tmpl =~ s!/homepage!/${cf{homepage}}/g;

# テンプレート分割
my ($head,$loop,$foot);
if ($tmpl =~ /(.)<!-- loop_begin -->(.)<!-- loop_end -->(.)s) {
    ($head,$loop,$foot) = ($1,$2,$3);
} else {
    &error("テンプレートが不正です");
}

# 現在時間
my $now = time;

# ヘッダ表示
print "Content-type: text/html¥n¥n";
print $head;

my ($i,$bef_c,$bef_r);
foreach (@data) {
    $i++;
    my ($cnt,$no,$cat,$scat,$sub,$url,$msg,$tim,$reco) = split(/¥0/);
    $url = &url_enc($url);

    # ランキング
    my $rank = $cnt == $bef_c ? $bef_r : $i;

```

```

        my $tmp = $loop;
        $tmp      =~ s|!site-name!|<a href="$cf{navi_cgi}?site=$no&url=$url"
target="$cf{target}">$sub</a>|g;
        $tmp =~ s|!cate!|<a href="$cf{navi_cgi}?mode=cate&cat=$cat">$cat{$cat}{0}</a>|g;
        $tmp      =~ s|!sub!|<a
href="$cf{navi_cgi}?mode=cate&cat=$cat&sub=$scat">$cat{$cat}{$scat}</a>|g;
        $tmp =~ s|!comment!|$msg|g;
        $tmp =~ s|!ico:(¥w+)!||g;
        $tmp =~ s|!rank!|$rank|g;
        $tmp =~ s|!count!|$cnt|g;
        $tmp =~ s|!num!|$no|g;
        $tmp =~ s|<!-- ico:new -->||g if ($now - $tim <
$cf{new_mark}*24*60*60);
        $tmp =~ s|<!-- ico:recom -->||g if ($reco == 1);
        print $tmp;

        $bef_c = $cnt;
        $bef_r = $rank;
    }

# カウントをリセット
&reset_data if ($cf{reset});

# フッタ
&footer($foot);
}

#-----
#  注意事項
#-----
sub note_page {
    # テンプレート読込
    open(IN,"$cf{tmpldir}/note.html") or &error("open err: note.html");
    my $tmpl = join("", <IN>);
    close(IN);

    # 文字置き換え
    $tmpl =~ s!/([a-z]+_cgi)!$cf{$1}/g;
    $tmpl =~ s!/new_mark!/$cf{new_mark}/g;
    $tmpl =~ s|!ico:(¥w+)!||g;

    # ヘッダ表示

```



```

print "Content-type: text/html¥n¥n";
print $tmpl;
exit;
}

#-----
# 登録フォーム
#-----
sub form_page {
    # カテゴリ
    my $op_cate;
    foreach (0 .. ${$cf{categ}}) {
        my @cat = split(/,/ , $cf{categ}->[$_]);

        foreach my $c (1 .. $#cat) {
            $op_cate .= qq|<option value="$_:$c">$cat[0] &gt; $cat[$c]¥n|;
        }
    }
}

# テンプレート読込
open(IN,"$cf{tmpldir}/form.html") or &error("open err: form.html");
my $tmpl = join("", <IN>);
close(IN);

# 画像認証作成
my ($str_plain,$str_crypt);
if ($cf{use_captcha} > 0) {
    require $cf{captcha_pl};
    ($str_plain, $str_crypt) = cap::make( $cf{captcha_key}, $cf{cap_len} );
    $tmpl =~ s!/str_crypt!/$str_crypt/g;
} else {
    $tmpl =~ s/<!-- captcha_begin -->.+<!-- captcha_end -->!/s;
}

# 文字置き換え
$tmpl =~ s!/([a-z]+_cgi)!/$cf{$1}/g;
$tmpl =~ s/<!-- op_cate -->/ $op_cate/g;
$tmpl =~ s!/mode!/regist/g;
$tmpl =~ s!/fm-url!/http:¥/¥/g;
$tmpl =~ s!/fm-¥w+!/g;

# ヘッダ表示

```

```

print "Content-type: text/html¥n¥n";
print $tmpl;
exit;
}

#-----
#  検索実行ページ
#-----

sub find_data {
    # キーワードなしの場合は無効
    return if ($in{word} eq "");

    # ページ数
    my $pg = $in{pg} || 0;

    # 表示件数
    $in{list} =~ s/¥D//g;
    if ($in{list} <= 0) { $in{list} = 10; }

    # 検索実行
    my ($i,@find) = &search($pg);

    # プルダウン
    my ($op_cond,$op_list) = &pulldown;

    # ページ数案内メッセージ
    my $pgbtn = &page_btn($i,$pg,$in{list});

    # テンプレート読込
    open(IN,"$cf{tmpldir}/find.html") or &error("open err: find.html");
    my $tmpl = join(" ", <IN>);
    close(IN);

    # 文字置き換え
    $tmpl =~ s/!cgi_title!/!$cf{cgi_title}/g;
    $tmpl =~ s/!([a-z]+_cgi)!/!$cf{$1}/g;
    $tmpl =~ s/!word!/!$in{word}/g;
    $tmpl =~ s/!keyword!/!$in{word}/g;
    $tmpl =~ s/!<!-- op_cond -->/!$op_cond/g;
    $tmpl =~ s/!<!-- op_list -->/!$op_list/g;
    $tmpl =~ s/!hit!/!$i/g;
    $tmpl =~ s/!page-btn!/!$pgbtn/g;

```

```

$tpl = ~ s!/mode!/$in{mode}/g;

# テンプレート分割
my ($head,$loop,$foot);
if ($tpl =~ /(.)<!-- loop_begin -->(.)<!-- loop_end -->(.)s) {
    ($head,$loop,$foot) = ($1,$2,$3);
} else {
    &error("テンプレートが不正です");
}

# カテゴリ情報
my %cat = &categ;

# 時間取得
my $now = time;

# ヘッダ表示
print "Content-type: text/html¥n¥n";
print $head;

foreach (@find) {
    my ($no,$cat,$scat,$sub,$url,$nam,$eml,$pw,$msg,$tim,$reco) = split(/<>/);
    $url = &url_enc($url);

    my $tmp = $loop;
    $tmp =~ s|!ico:(¥w+)||g;
    $tmp = ~ s|!site-name!|<a href="$cf{navi_cgi}?site=$no&url=$url"
target="$cf{target}">$sub</a>|g;
    $tmp = ~ s|!cate!|<a href="$cf{navi_cgi}?mode=cate&cat=$cat">$cat{$scat}{0}</a>|g;
    $tmp = ~ s|!sub!|<a
href="$cf{navi_cgi}?mode=cate&cat=$cat&sub=$scat">$cat{$scat}{$scat}</a>|g;
    $tmp = ~ s|!comment!|$msg|g;
    $tmp = ~ s|!num!|$no|g;
    $tmp = ~ s|<!-- ico:new -->||g if ($now - $tim <
$cf{new_mark}*24*60*60);
    $tmp = ~ s|<!-- ico:recom -->||g if ($reco == 1);
    print $tmp;
}

# フッタ
&footer($foot);
}

```

```

#-----
# カテゴリページ
#-----
sub cate_page {
    # ページ数
    my $pg = $in{pg} || 0;

    # カテゴリ名
    my @cat = split(/,/ , $cf{categ}->[$in{cat}]);
    my $keyword = $cat[$in{cat}];
    $keyword .= " &gt; $cat[$in{sub}]" if ($in{sub} ne "");

    # カテゴリ選択ボタン
    my $btn_cate;
    $in{area} ||= 'in';
    my %area = (all => '全検索', in => 'カテゴリ内から検索');
    foreach ('all','in') {
        if ($in{area} eq $_) {
            $btn_cate .= qq|<input type="radio" name="area" value="$_" checked>$area{$_}¥n|;
        } else {
            $btn_cate .= qq|<input type="radio" name="area" value="$_">$area{$_}¥n|;
        }
    }
}

# 検索なし
my ($i,@find);
if ($in{word} eq "") {
    $i = 0;
    open(IN,"$cf{logfile}") or &error("open err: $cf{logfile}");
    while (<IN>) {
        my ($no,$cat,$scat,$sub,$url,$nam,$eml,$pw,$msg,$tim,$rec) = split(/<>/);

        # 大カテ
        next if ($in{cat} != $cat);

        # 小カテあり
        next if ($in{sub} ne " && $in{sub} != $scat);

        $i++;
        next if ($i < $pg + 1);
        next if ($i > $pg + $cf{pg_max});
    }
}

```

```

        push(@find,$_);
    }
    close(IN);

# 検索あり
} else {
    ($i,@find) = &search($pg);
}

# 繰越ボタン作成
my $pg_btn = &page_btn($i,$pg,$cf{pg_max});

# プルダウン
my ($op_cond,$op_list) = &pulldown;

# テンプレート読込
open(IN,"$cf{tmpldir}/find.html") or &error("open err: find.html");
my $tmpl = join("", <IN>);
close(IN);

# 文字置き換え
$tmpl =~ s!/cgi_title!/${cf{cgi_title}}/g;
$tmpl =~ s!/([a-z]+_cgi)!/${cf{$1}}/g;
$tmpl =~ s!/word!/${in{word}}/g;
$tmpl =~ s/<!-- op_cond -->/${op_cond}/g;
$tmpl =~ s/<!-- op_list -->/${op_list}/g;
$tmpl =~ s!/hit!/${i}/g;
$tmpl =~ s!/page-btn!/${pg_btn}/g;
$tmpl =~ s!/keyword!/「${keyword}」/g;
$tmpl =~ s/<!-- btn_cate -->/${btn_cate}/g;
$tmpl =~ s!/area-cat!/${in{cat}}/g;
$tmpl =~ s!/area-sub!/${in{sub}}/g;
$tmpl =~ s!/mode!/${in{mode}}/g;

# テンプレート分割
my ($head,$loop,$foot);
if ($tmpl =~ /(.(+)<!-- loop_begin -->(.)<!-- loop_end -->(.)s) {
    ($head,$loop,$foot) = ($1,$2,$3);
} else {
    &error("テンプレートが不正です");
}

```

```

# カテゴリ情報
my %cat = &categ;

# 時間取得
my $now = time;

# ヘッダ表示
print "Content-type: text/html¥n¥n";
print $head;

foreach (@find) {
    my ($no,$cat,$scat,$sub,$url,$nam,$eml,$pw,$msg,$tim,$reco) = split(/<>/);
    $url = &url_enc($url);

    my $tmp = $loop;
    $tmp =~ s|!ico:(¥w+)!||g;
    $tmp      =~      s|!site-name!|<a      href="$cf{navi_cgi}?site=$no&url=$url"
target="$cf{target}">$sub</a>|g;
    $tmp =~ s|!cate!|<a href="$cf{navi_cgi}?mode=cate&cat=$cat">$cat{$cat}{0}</a>|g;
    $tmp      =~      s|!sub!|<a
href="$cf{navi_cgi}?mode=cate&cat=$cat&sub=$scat">$cat{$cat}{$scat}</a>|g;
    $tmp =~ s|!comment!|$msg|g;
    $tmp =~ s|!num!|$no|g;
    $tmp =~ s|<!-- ico:new -->||g if ($now - $tim <
$cf{new_mark}*24*60*60);
    $tmp =~ s|<!-- ico:recom -->||g if ($reco == 1);
    print $tmp;
}

# フッタ
&footer($foot);
}

#-----
# 繰越ボタン作成
#-----
sub page_btn {
    my ($i,$pg,$pgmax) = @_ ;
    $pgmax || = 10;

    # 表示範囲

```

```

my $from = $pg + 1;
my $to = $pg + $pgmax > $i ? $i : $pg + $pgmax;
my $pg_btn = "$from - $to [ $i 件中 ]";

# ページ繰越ボタン作成
if ($pg - $pgmax >= 0 || $pg + $pgmax < $i) {

    # 引数
    my $param = "mode=${in{mode}}";
    if (${in{word}} ne "") {
        my $word = &url_enc(${in{word}});
        $param .= "&word=$word&cond=${in{cond}}&list=$pgmax";
    }
    $param .= "&cat=${in{cat}}";
    $param .= "&sub=${in{sub}}" if (${in{sub}} ne "");

    # ページボタン
    $pg_btn .= "&nbsp;&nbsp;  Page: ";
    my ($x, $y) = (1, 0);
    while ($i > 0) {
        if ($pg == $y) {
            $pg_btn .= qq(| <b style="color:red">$x</b> );
        } else {
            $pg_btn .= qq(| <a href="$cf{navi_cgi}?$param&pg=$y">$x</a> );
        }
        $x++;
        $y += $pgmax;
        $i -= $pgmax;
    }
    $pg_btn .= "|";
}
return $pg_btn;
}

#-----
# 検索実行
#-----
sub search {
    my ($pg) = @_ ;
    ${in{list}} ||= 10;
    if (${in{cond}} eq "") { ${in{cond}} = 1; }

```

```

# コード変換
Jcode::convert(¥$in{word},'sjis');

# 入力内容調整
$in{word} =~ s/ //g;
my @wd = split(/¥s+/, $in{word});

# 検索
my $i = 0;
my @find;
open(IN,"$cf{logfile}") or &error("open err: $cf{logfile}");
while (<IN>) {
    my ($no,$cat,$scat,$sub,$url,$nam,$eml,$pw,$msg,$tim,$reco) = split(/<>/);

# 大カテ
next if ($in{cat} ne " && $in{cat} != $cat);

# 小カテあり
next if ($in{sub} ne " && $in{sub} != $scat);

my $flg;
foreach my $wd (@wd) {
    if (index("$sub $url $msg", $wd) >= 0) {
        $flg = 1;
        if ($in{cond} == 0) { last; }
    } else {
        if ($in{cond} == 1) { $flg = 0; last; }
    }
}
if ($flg) {
    $i++;
    next if ($i < $pg + 1);
    next if ($i > $pg + $in{list});

    push(@find,$_);
}
}
close(IN);

# 検索結果
return ($i,@find);

```

```

}
```



```

#-----
#   メンテ用フォーム
#-----
sub form_mente {
    my ($flg,$men_ttl,$men_url);
    open(IN,"$cf{logfile}") or &error("open err: $cf{logfile}");
    while (<IN>) {
        my ($no,$cat,$scat,$sub,$url,$nam,$eml,$pw,$msg,$tim,$reco) = split(/<>/);

        if ($in{no} == $no) {
            $men_ttl = $sub;
            $men_url = $url;
            $flg++;
            last;
        }
    }
    close(IN);

    if (!$flg) { &error("該当記事は見当たりません"); }

    # テンプレート読込
    open(IN,"$cf{tmpldir}/mente.html") or &error("open err: mente.html");
    my $tmpl = join("", <IN>);
    close(IN);

    my %mode = (edit => '修正', dele => '削除');
    my $op_mode;
    foreach ('edit','dele') {
        if ($in{mode} eq $_) {
            $op_mode .= qq |<option value="$_" selected>$mode{$_}¥n | ;
        } else {
            $op_mode .= qq |<option value="$_">$mode{$_}¥n | ;
        }
    }
}

# 文字置き換え
$tmpl =~ s/<!-- op_mode -->/ $op_mode /g;
$tmpl =~ s!/site-title!/$men_ttl/g;
$tmpl =~ s!/url!/$men_url/g;
$tmpl =~ s!/([a-z]+_cgi)!/$cf{$1}/g;
$tmpl =~ s!/num!/$in{no}/g;

```

```

print "Content-type: text/html¥n¥n";
print $tmpl;
exit;
}

#-----
# プルダウン作成
#-----
sub pulldown {
    my @cond = qw |OR AND|;
    my ($op_cond,$op_list);
    foreach (1,0) {
        if ($in{cond} eq $_) {
            $op_cond .= qq |<option value="$_" selected>$cond[$_]|;
        } else {
            $op_cond .= qq |<option value="$_">$cond[$_]|;
        }
    }
    foreach (10,20,30,40,50) {
        if ($in{list} eq $_) {
            $op_list .= qq |<option value="$_" selected>$_件¥n|;
        } else {
            $op_list .= qq |<option value="$_">$_件¥n|;
        }
    }
    return ($op_cond,$op_list);
}

```

```

#-----
# フッター
#-----
sub footer {
    my $foot = shift;

    # 著作権表記 (削除・改変禁止)
    my $copy = <<EOM;

```

<p style="margin-top:2em;text-align:center;font-family:Verdana,Helvetica,Arial;font-size:10px;">

- <a href="http://www9.ocn.ne.jp/~asukadjj/" target="\_top">ASUKA High-tech Laboratory</a> -

</p>

EOM

```
if ($foot =~ /(.(+)(<¥/body[^\>]*>.*)/si) {
    print "$1$copy$2¥n";
} else {
    print "$foot$copy¥n";
    print "</body></html>¥n";
}
exit;
}
```

#-----

# カテゴリ情報

#-----

```
sub categ {
    my %cat;
    foreach (0 .. ${#cf{categ}}) {
        my @cat = split(/,/ , $cf{categ}->[$_]);

        # 大カテ
        $cat{$_}{0} = $cat[0];

        # 小カテ
        foreach my $c (1 .. $#cat) {
            $cat{$_}{$c} = $cat[$c];
        }
    }
    return %cat;
}
```

#-----

# URL エンコード

#-----

```
sub url_enc {
    local($_) = @_;

    s/(¥W)"/%' . unpack('H2', $1)/eg;
}
```

```

s/¥s+/g;
$_;
}

#-----
# カウントリセット
#-----

sub reset_data {
    # 時間情報
    my ($mday,$mon,$wday) = (localtime(time))[3,4,6];

    # リセットデータ読み込み
    open(DAT,"+< $cf{resfile}") || &error("open err: $cf{resfile}");
    eval "flock(DAT, 2)";
    my $data = <DAT>;

    my ($m,$d,$w) = split(/:/, $data);

    # 週更新
    my $flg;
    if ($cf{reset} == 1) {
        if ($mday != $d && $wday == 0) { $flg = 1; }

    # 10日更新
    } elsif ($cf{reset} == 2) {
        if ($mday % 10 == 0 && $mday != $d) { $flg = 1; }

    # 月更新
    } elsif ($cf{reset} == 3) {
        if ($mon != $m) { $flg = 1; }
    }

    # 更新
    if ($flg) {

        # リセットデータ更新
        seek(DAT, 0, 0);
        print DAT "$mon:$mday:$wday";
        truncate(DAT, tell(DAT));

        ## --- 全ファイルリセット
        opendir(DIR,"$cf{cntdir}") || &error("open err: $cf{cntdir}");

```

```
my @dir = readdir(DIR);
closedir(DIR);

foreach (@dir) {
    if (/^✪d+✪.dat$/) {
        unlink("$cf{cntdir}/$_");
    }
}
}
close(DAT);
}
```