

JChecker3.0 操作説明書



1998年5月15日 JChecker1.0 初版
1999年11月10日 JChecker2.0.1版
1999年12月01日 JChecker2.0.2版
2000年01月15日 JChecker2.1.1版
2002年02月15日 JChecker3.0版

もくじ

はじめに

JChecker3.0 の特長

動作環境

著作権、使用、配布について

ユーザーサポート

第1章 とりあえず使ってみよう

1.1 インストール

1.2 JChecker3.0の起動

1.3 HTMLの文法チェック

1.4 JChecker3.0のリビジョン番号確認

第2章 HTML文法チェックの根拠

2.1 DTDと辞書

2.2 インターネット上の代表的なHTML文法チェッカ

第3章 初期設定

3.1 エディタの設定

3.2 カラーリングの設定

3.3 整形の設定

3.4 文法チェックの設定

3.5 リンクチェックの設定

3.6 メッセージの設定

3.7 ブラウザの設定

3.8 自動修正の設定

第4章 ファイル操作

4.1 新規にHTMLを作成する

4.2 ローカル書類を開く

4.3 リモート書類を開く

4.4 最近開いたファイル

4.5 HTMLテンプレート

4.6 書類を閉じる

4.7 書類を保存する

4.8 書類の復帰

4.9 エラーメッセージの保存

4.10 書類の印刷

第5章 FTP転送プログラムとの連動

5.1 設定方法

5.2 使用方法

第6章 編集操作

6.1 編集ウィンドウの各部の名称

6.2 キー操作

- 6.3 取消し(Undo)とやり直し(Redo)
- 6.4 コピー、カット、ペースト
- 6.5 ドラッグ&ドロップ
- 6.6 エLEMENTの選択
- 6.7 タグの削除
- 6.8 日付・時刻のスタンプ

第7章 検索と置換

- 7.1文字列の検索
- 7.2 正規表現について
- 7.3 文字列の置換
- 7.4 マルチファイル検索・置換
- 7.5 辞書の検索
- 7.6 マーク機能
- 7.7 マークリスト
- 7.8 行番号ジャンプ

第8章 チェック

- 8.1 文法チェック
- 8.2 リンクチェック
- 8.3 すべてチェック
- 8.4 自動修正
- 8.5 HTMLの整形
- 8.6 HTMLの最小化
- 8.7 再カラーリング

第9章 他のエディタと連動

- 9.1 ワードサービスでチェック
- 9.2 サービスメニューでチェック

第10章 辞書

- 10.1 辞書ウィンドウ
- 10.2 辞書の生成 (DtdToDic)
- 10.3 辞書ファイルの構造
- 10.4 辞書のカスタマイズ

第11章 ブラウザメニュー

- 11.1 ブラウザの登録
- 11.2 ブラウザでプレビューする

第12章 ウィンドウメニュー

- 12.1 マクロウィンドウ
- 12.2 カラーパレット
- 12.3 すべてを手前に移動
- 12.4 切り替え
- 12.5 タイリング
- 12.6 ズーム

第13章 マクロ

13.1 マクロメニュー

13.2 マクロウィンドウ

13.3 コンテキストメニュー

付録1、 付属マクロについて

付録2、 メッセージについて

付録3、 アップルスクリプト

x

はじめに

すでにインターネットは毎日の生活に欠かせないものになり、ホームページの作成はだれもが行うあたりまえのものになりました。ホームページ作成ツールも、一般向けのものからプロ向けのものまで、様々なWYSIWYG型のツールが発売されています。

ホームページの作成に慣れてHTMLを理解してくると、多くのユーザーはWYSIWYG型のツールに加えて、直接テキストエディタでもHTMLのソースコードをで編集するようになります。それは、WYSIWYG型ツールでは、レイアウト偏重のためHTMLの論理構造を正しく記述できない、動作が重たい、ちょっとした修正などには使いにくい、ツールによっては意味不明のタグが勝手にはいってまう、などの理由からです。

しかし、ソースコードを直接編集するときは、エレメントや属性のスペル、階層関係、終了タグ、出現順序などに十分注意する必要があります。英文を書くときにスペルチェックが有用なのと同じように、HTMLを直接編集するときは、HTML規則に従っているか、リンク先が存在するかなどをチェックしてくれるHTMLチェックは不可欠です。

JCheckerは、Jeditなどのテキストエディタと連携するHTMLチェックとして1998年5月に公開され、JChecker2.0（1999年11月）、JChecker2.1（2000年1月）と改訂を重ね、ご好評をいただけてきました。

そして、このJChecker3.0では、HTMLチェックの枠を超え、Jeditなみの編集、検索、置換、マクロ機能を備えたHTMLソースコードエディタに生まれ変わりました。HTMLチェックとしても、XHTMLやCompact HTMLなど16種類の辞書をサポート、DTDからの辞書の自動生成機能、HTMLエラーの自動修正機能、リンク先のページもまとめてチェックする一括チェック機能など大幅な機能強化が行われています。さらにエディタとしても、軽快で強力なJeditの編集機能を受け継ぐとともに、HTML4.0のタグを網羅したタグ編集マクロ、タグヘルプ機能、外部エディタプロトコル、サービメニューのサポートなど豊富な機能を備えています。

HTML作成の必携ツールとしてJChecker3.0を皆様のお役に立てていただければ幸いです。

最後になりましたが、JChecker1.0以来Netscape NavigatorおよびInternet ExplorerのDTDを快く提供して下さった石野恵一郎氏、HTMLヘルプページへのリンクを快諾していただいたWeb Design GroupのLiam Quinn、加藤泰孝両氏に心からお礼申し上げます。

2002年2月

松本慧（Jedit/JChecker作者）

JChecker3.0 の特長

- ・ Classic OSとMac OS Xの両OSに対応。Mac OS Xではカーボンアプリケーションとしてネイティブに動作します。
- ・ 英語、日本語はもちろんUnicode、ロシア語、中国語など28種類の文字コードセットに対応したHTMLソースコードエディタです。
- ・ 強力なエディタ機能を内蔵しています。Jeditなみの正規表現による検索/置換、アップルスク립ト、マクロ機能、ドラッグ&ドロップ、行番号表示、HTML カラーリング、インターネット設定、無限Undo、コンテキストメニュー、インライン日本語変換などをサポートしています。
- ・ HTML4.0の仕様を網羅するタグ編集マクロ、ヘルプ機能をサポートしています。うる覚えの要素や属性もおまかせください。また、マクロはアップルスク립トで作成されているので、ユーザーで修正、追加が容易に行えます。
- ・ HTML書類について、要素や属性のスペル、階層関係、対応する終了タグ、出現順序、属性値のタイプ など60数種類にわたるチェックを行います。HTML2.0、HTML3.2、HTML4.0、XHTML1.0、Compact HTML1.0、XHTML Basic 1.0など16種類のDTDに基づくチェックが可能です。
- ・ リンクチェックでは、HTML書類のなかのリンク先URLについて存在の有無をチェックします。ローカルリンク、リモートリンクのどちらもチェックできます。また、リンク先のページもまとめてチェックする一括チェック機能をサポートしています。
- ・ 検出したHTMLエラー(20種類)に対して自動修正機能をサポート。単純なエラーはJCheckerが自動的に修正します。
- ・ HTML書類の整形をサポート。タグ階層に応じたインデントやデータ文からの改行コード削除などのオプションを指定できます。また整形のカスタマイズも可能です。
- ・ HTML書類の最小化をサポート。HTML書類から不要なスペース、タブ、改行を取り除きHTML書類の最小化します。
- ・ DTDからJCheckerの辞書を自動生成するプログラム「DtdToDic」をサポート。DTDさえあればユーザーが辞書を生成しJCheckerの辞書メニューに追加できます。またDTDを修正してユーザー独自の辞書を作ることも可能です。
- ・ Netscape Navigator, Internet ExplorerなどのブラウザをJChecker から起動しHTML書類をプレビューできます。ブラウザはブラウザメニューに何種類でも登録できます。

・イメージファイルやHTMLファイルを編集画面にドロップするだけで、イメージタグやアンカータグが自動生成されます。

・外部エディタプロトコルをサポート。FTPプログラム（Interarchy, NetFinder, Transmit, Fetchなど）のメニューから直接、リモートサーバー上のファイルをJCheckerで編集できます。すなわち、WEBサーバー上のhtmlファイルなどをJCheckerで直接編集チェックできます。

・OS Xのサービスマニューに対応しているので、サービスマニュー対応のエディタ（Jedit4.0.9(3)など）で作成中のHTML書類をチェックできます。

・ワードサービスプロトコルに対応し、JeditやBBEditなどのワードサービス対応エディタで作成中のHTML書類をJCheckerで連動チェックできます。

動作環境

- ・JChecker3.0には、Classic OS版とOS X版（カーボン）の2種類があります。
- ・Classic OS版は、漢字Talk8.6以降のMacOSのもとで使用できます。
- ・OS X版は、OS X 10.1以降のOS Xのもとで使用できます。

著作権、使用、配布について

・JChecker3.0の著作権は、松本慧（satoshi@artman21.co.jp）にあります。著作権者の許可なくJChecker3.0の一部または全部を改変することはできません。

・JChecker3.0はシェアウェアです。試用期間(20日)後も継続して利用される場合は、別紙の「シェアウェア登録」を参照の上、ユーザー登録をしてください。

・JChecker3.0の使用によるどのような損害に対しても著作者は一切の責任を負いません。またシェアウェア登録後のJChecker3.0の使用権は第三者に譲渡、貸与することはできません。

・非営利にかぎりJChecker3.0はBBSやインターネットなどで自由に複製、配布ができます。ただし、配布の際にJChecker3.0のファイル構成/内容等に修正を加えることはできません。CD-ROMでの配布や商品にバンドルする場合は事前に著作者（satoshi@artman21.co.jp）の書

面による許可が必要です。

・ JChecker3.0で利用している「WASTE Text Engine」の著作権はMarco Piovanelli氏にあります。(WASTE Text Engine (c) 1993-2002 by Marco Piovanelli, All Rights Reserved.)

・ JChecker3.0の辞書の作成にあたっては、石野恵一郎氏作成のNetscape Navigator及びInternet Explorer用DTDを利用しています。

ユーザーサポート

JChecker3.0 をご利用になりましてご不明な点、不具合、ご要望、改善すべき点などがありましたら、ぜひ開発元までご連絡ください。

連絡先: アートマン21 ユーザーサポート係
internet support@artman21.co.jp
TEL/FAX 0466-24-1028
〒251-0055 神奈川県藤沢市南藤沢16-11

またJChecker3.0の最新版は以下のホームページで公開されています。

<http://www.artman21.net/>

第1章 とりあえず使ってみよう

1.1 インストール

JChecker3.0は、圧縮ファイルの形で提供され、インターネット、BBSまたはCD-ROMなどから入手できます。Classic OS(Mac OS 8.6-9.2)で利用するときは、Classic版JChecker3.0を、Mac OS X で利用するときはOS X版JChecker3.0を、希望に応じてダウンロードしてください。またそれぞれに英語版と日本語版があります。



JChecker301J.sit

Classic (日本語) 版



JChecker301JX.dmg

OS X (日本語) 版

ファイル名のなかの数字「301」はリビジョン番号なので、その時によってリビジョン番号はこの説明書と違うかも知れません。

Classic版のインストール

JChecker301J.sitをダブルクリックすると解凍されて「JChecker3.0 Folder」が現れますので、そのフォルダを起動ディスクのApplications フォルダのなかにマウスでドラッグして移動してください。

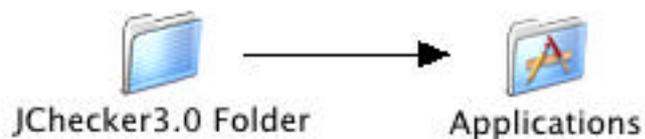


OS X版のインストール

JChecker301JX.dmgをマウスでダブルクリックすると以下のようなディスク JChecker301JX がデスクトップに現れます。



ディスク JChecker301JX をダブルクリックすると、そのなかに「JChecker3.0 Folder」がありますので、そのフォルダを起動ディスクのApplications フォルダのなかにマウスでドラッグして移動してください。



JChecker3.0 Folderを開くと下図のようにファイルやフォルダが格納されています。



以上で、インストールは完了です。なお、JChecker3.0は、システムフォルダにはなにもインストールする必要はありません。

1.2 JChecker3.0の起動

JChecker3.0フォルダのプログラム「JChecker3」をダブルクリックするとJChecker3.0が起動します。



未登録の場合、起動すると以下のような登録ダイアログが現れます。



登録お願いします

 JChecker 3.0 copyright(c)1998-2002
MATSUMOTO Satoshi
All right Reserved.

JCheckerはシェアウェアです。試用期間（20日）の経過後もご
利用される場合は、ユーザー登録をお願いします。登録料の支払
いはArtman21のホームページからオンラインで行えます。
<http://www.artman21.net/>
詳しくは、同梱のファイル「シェアウェア登録」を参照くださ
い。登録が完了すればこのダイアログは表示されなくなります。

ユーザー名:

登録コード:

とりあえず試用する 登録

登録の申し込みは、当社のホームページから直接オンラインで登録を行えます。
URL表示(<http://www.artman21.net/>)をクリックするとブラウザが起動し当社のホーム
ページへリンクします。また、オンライン以外にファックスや郵便でも登録できますので、詳
しくは同梱のファイル「シェアウェア登録」を参照してください。

登録申し込みを行い、当社からユーザー名と登録コードを電子メールで受け取ったら、そのユー
ザー名と登録コードをキー入力し「登録」ボタンを押してください。登録が完了すると以後この
登録ダイアログは表示されなくなります。

登録コードとユーザー名を入力するときは、こちらからお知らせしたとおりに正確に入力して
ください。余分なスペースをつけたり、半角・全角の違いがあると登録できません。

登録せずに試用するときは「とりあえず試用する」ボタンを押して下さい。

また、登録しないまま試用期間の20日間が過ぎると、チェックのたびに試用期間が過ぎた旨の
ダイアログがでるようになります。

使用しない場合は「JChecker3.0フォルダ」をゴミ箱に捨ててください。

1.3 HTMLの文法チェック

JChecker3.0の起動時、またはファイルメニューの「開く...」を選択すると以下のように、
チェックする HTML ソースファイルを聞いてきます。

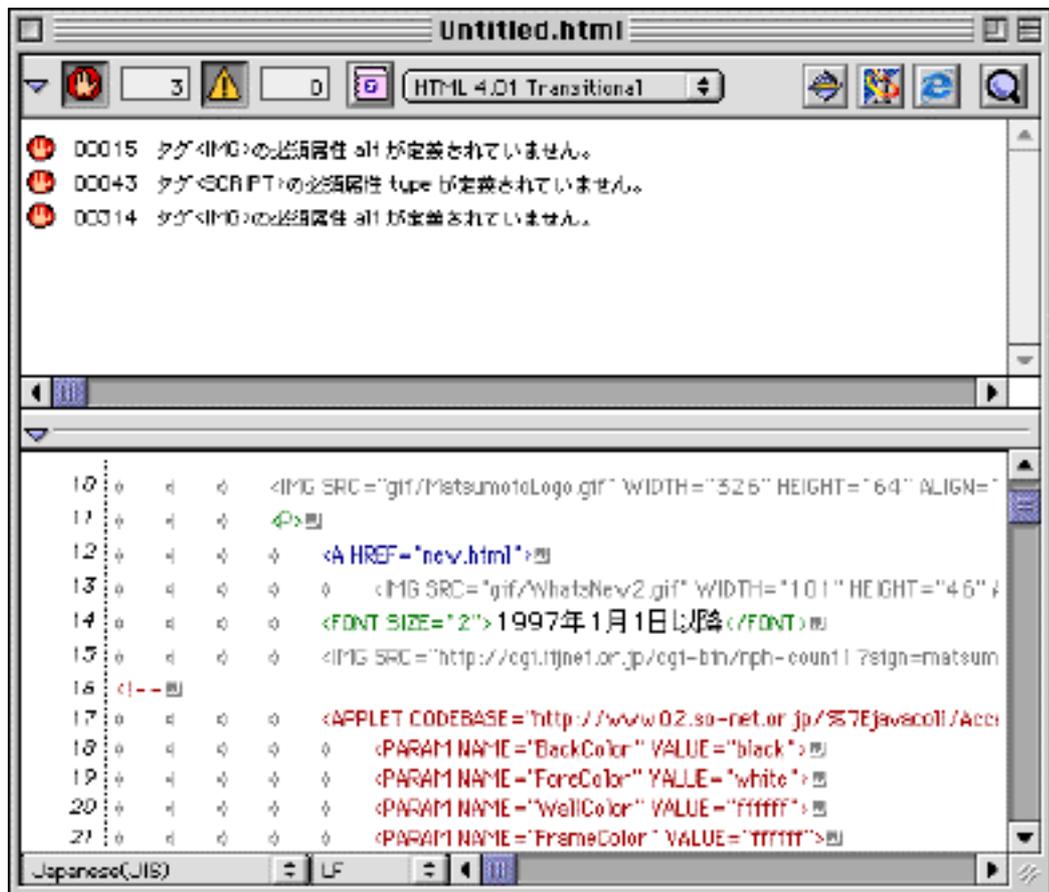


JChecker3.0では、インターネット上の様々な文字コードセット（英語、日本語、中国語、Unicodeなどの28種類）をサポートしています。また改行コードについては、CR(Mac), LF (Unix), CR+LF(DOS)の3種類の改行コードのファイルを扱うことができます。

日本語のHTMLの場合は、「開く」ダイアログ下部の「文字コードセット」を「Japanese (Auto detect)」に、「改行コード」を「Auto detect」にしておくことで読み込んだファイルの漢字コードと改行コードを自動判定します。

希望のHTMLソースファイルを選択し「開く」ボタンを押すと、以下のようにそのソースファイルの内容が、ウィンドウの下半分(エディタウィンドウ)に表示されます。

ウィンドウの上半分(メッセージウィンドウ)には、HTMLの文法チェックの結果が表示されません。



メッセージウィンドウには、エラーと警告の行番号とその内容が表示されます。エラーメッセージをシングルクリックすると、そのエラーに対応するソースリストの部分が下部のエディタウィンドウで選択表示されます。ダブルクリックすると、そのエラーの解説が別ウィンドウに表示されます。

はじめて文法チェックをしたとき、多くのユーザーはそのエラーと警告の多さに驚かれるかもしれません。「一体全体、JChecker3.0はどのような根拠をもってこんなにたくさんエラーや警告を出すのか」と疑問に思われるかたは、是非「第2章、HTML 文法チェックの根拠」をお読みください。

もともとHTMLにエラーがたくさんあっても、プログラムのように止まってしまったりするわけではなく、単に「思ったような見え方にならないかもしれない」だけですので、エラーにあまり神経質になる必要はありません。

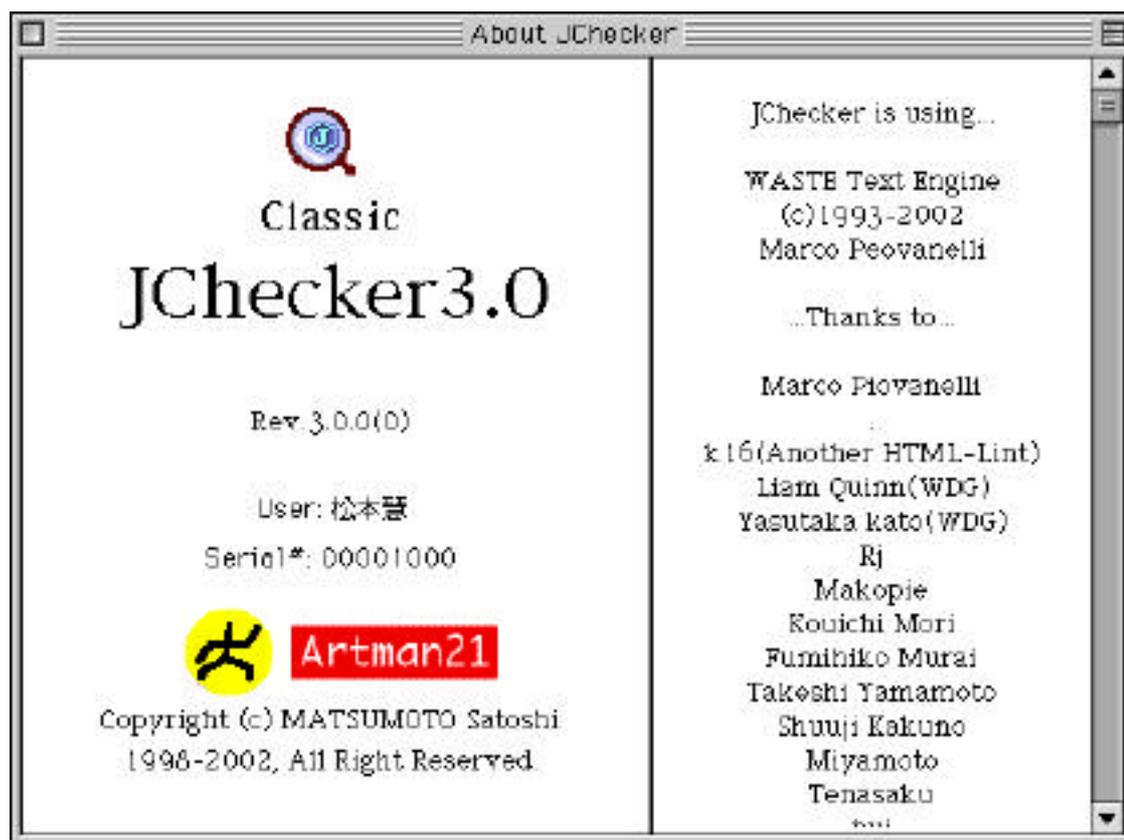
各エラーや警告については「第5章、エラーメッセージについて」を参照してください。

見つかったエラーや警告を修正するには、HTMLについての基礎的な知識がある程度必要です。残念ながらこのマニュアルはHTMLの解説書ではないので、HTMLそのものについては市販の解説書を併せてお読みください。

1.4 JChecker3.0のリビジョン番号確認

自分が使っているJChecker3.0がどの版かを調べるには、アップルメニューの「JCheckerについて...」を選択すると以下のようなウィンドウが現れ、JChecker3.0のリビジョン番号を確認できます。

この例では、JChecker3.0がPowerMac版でリビジョン番号が「3.0.0(0)」であることがわかります。



また、下部にはユーザー名とシリアル番号が表示されます。未登録のときは、この部分に「** * Unregisterd ***」と表示されます。なお、表示されるシリアル番号は、登録時に入力した登録コードとは別の番号です。

第2章 HTML文法チェックの根拠

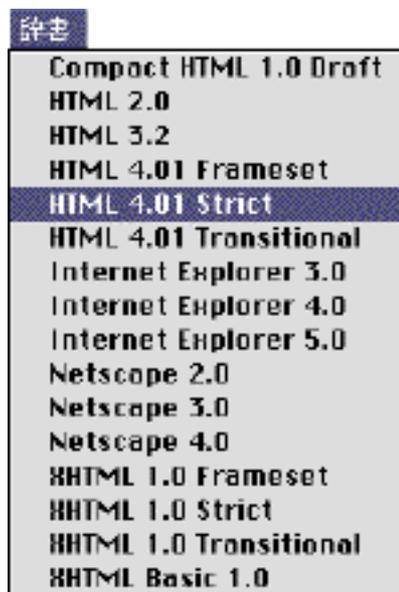
JChecker3.0で文法チェックすると予想以上にエラーや警告が多くでて、驚かれるかもしれません。この章ではJChecker3.0のエラーや警告がどのような根拠に基づいているのかについて説明しています。

2.1 DTDと辞書

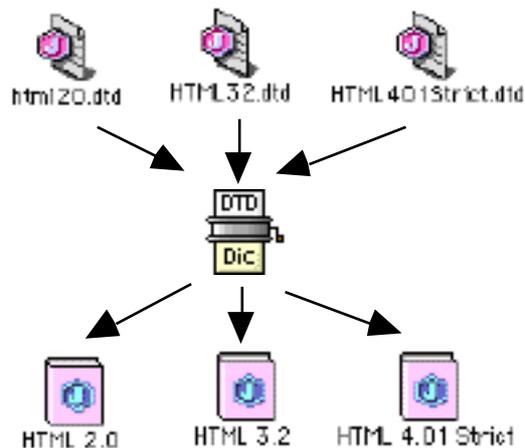
HTMLでは、文書のなかにどんなタグがあるのか、それぞれのタグではどんな属性をサポートしているのか、またタグ間の階層関係や出現順序はどうなっているのかなど、DTD(Document Type Definition)というファイルにすべて規定されています。

JChecker ではこのDTDから辞書を生成し、それをもとにHTML書類をチェックしエラーメッセージや警告の根拠としています。

DTDはHTMLのバージョンごとに存在し、またブラウザによっても異なってきます。JCheckerには、HTML2.0/3.2/4.0、Netscape Navigator2.0/3.0/4.0、Internet Explorer 3.0/4.0/5.0、XHTML1.0、Compact HTML1.0などの辞書をあらかじめ用意しています。JCheckerの辞書メニューで現在JCheckerがサポートしている辞書を一覧できます。



これらの辞書は、JChecker付属のプログラム「DtdToDic」を使ってDTDから自動生成されたものです。



したがって、DTDさえ入手できれば、辞書をユーザーが自由に追加できます。また、DTDをユーザーがカスタマイズして「DtdToDic」で辞書を生成すれば、ユーザー独自の辞書を作成することも可能です。

なお、Netscape Navigator とInternet Explorer については公式DTDが存在しないため、WWW上のHTMLチェッカとして有名な石野恵一郎 氏作成のDTDを許可を得て辞書を作成しました。

Adobe PageMillなどのGUIのHTMLエディタで作成したHTML書類を、JCheckerでチェックすると、多くのエラーや警告がでて驚かれるかもしれません。警告やエラーがでるということは、少なくともDTDに準拠していない部分があるということですが、HTMLはもともとプログラムとは異なり、エラーがあってもハングしたりすることはないので、あまり神経質になる必要はないのかも知れません。しかし、警告やエラーの部分は、それぞれのブラウザが独自に解釈したり無視することになり、結果としてブラウザのバージョンや種類によって見え方が違って来る可能性があります。

2.2 インターネット上の代表的なHTML文法チェッカ

JChecker3.0はパソコン上でチェックしますが、HTML文法チェッカにはインターネット上でチェックしてくれるサービスもあります。JChecker3.0と同じようにDTDに基づいたチェックが行なわれます。もし、JChecker3.0でエラーメッセージで納得できないときは、これらのサービスでもチェックしてみてください。(JChecker3.0よりは権威があります。)

<http://validator.w3.org/>

HTMLの仕様を公開しているHTMLの本家、W3C(World Wide Web Consortium)が行っているサービスです。チェックしたいホームページのURLを入力すると、そのページの文法チェックを行います。日本語を含むHTMLにも対応していますが、エラーメッセージは英語で表示さ

れます。

<http://openlab.ring.gr.jp/k16/htmlint/htmlint.html>

k16(石野恵一郎)氏が作成・運営している日本語対応のHTML文法チェッカです。大変緻密な文法チェックを行います。HTMLの文法チェックに関する有用な情報も入手できます。JChecker3.0の辞書作成でもNetscape Navigator、Internet ExplorerのDTDでお世話になりました。

第3章 初期設定

JCheckerでは、エディタ、カラーリング、整形、文法などのさまざまな条件をメニュー「ファイル」「初期設定...」から設定できます。これらの設定値は、システムの初期設定フォルダのファイル「JChecker3.0 Prefs」に保存されます。



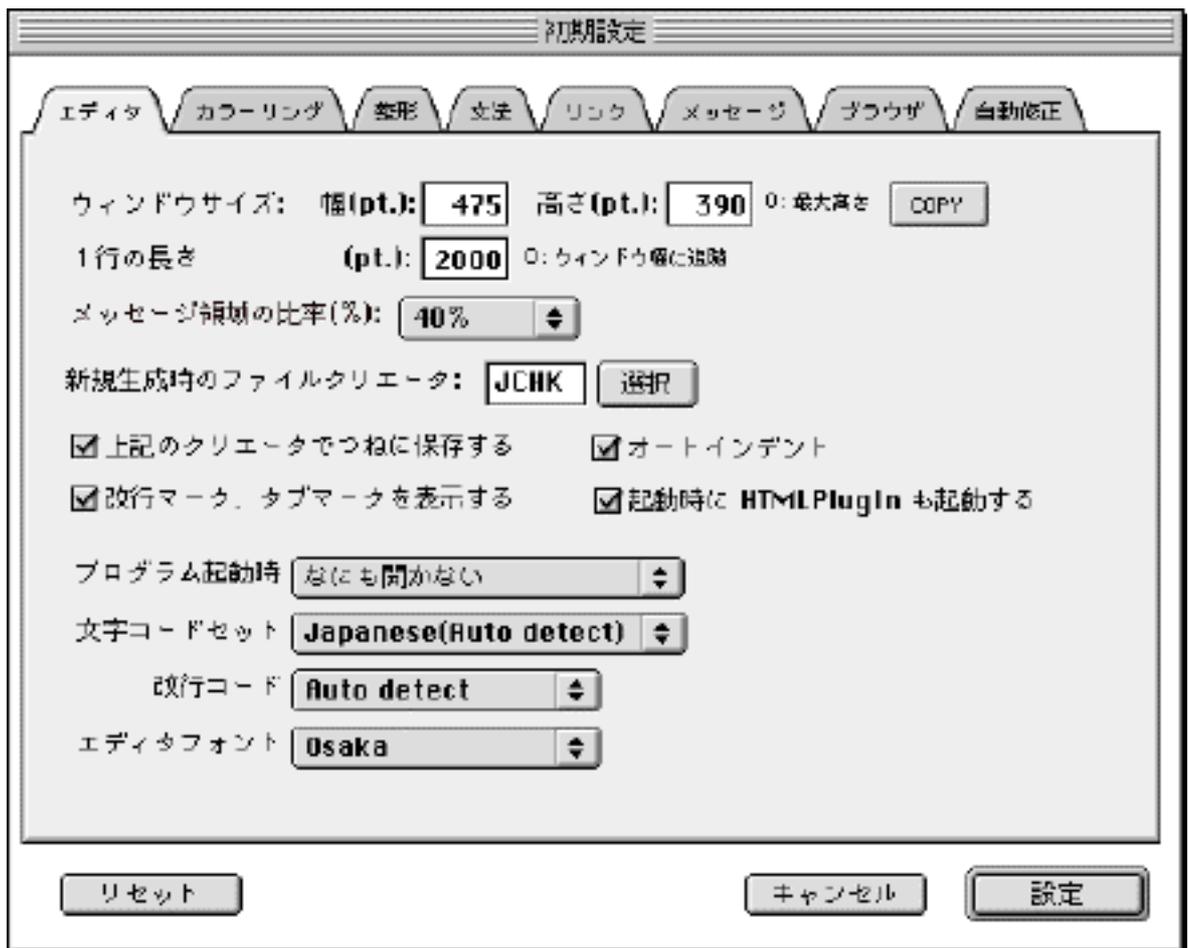
JChecker 3.0 Prefs

Mac OS Xでは、初期設定ファイルはホームディレクトリ下の以下の位置に格納されています。

"~/Library/Preferences/JChecker3.0 Prefs"

3.1 エディタの設定

メニュー「ファイル」「初期設定...」を開き、「エディタ」タブをクリックすると、以下のようなエディタ設定パネルが表示されます。



ウィンドウサイズでは、チェックウィンドウを開いたときの幅と高さを指定します。スクリーン一杯の高さにしたいときは、高さに「0」を指定してください。「COPY」ボタンを押すと

現在の編集ウィンドウの幅と高さがそれぞれのフィールドにセットされます。

1行の長さでは、エディタでの1行の横幅を指定します。「0」を指定すると1行の横幅がウィンドウの横幅に追従ようになります。

メッセージ領域の比率では、編集ウィンドウを開いた直後にメッセージ領域が占める比率を指定します。

新規生成時のファイルクリエータでは、新規HTML書類を保存するときどのようなアイコンにするかを指定できます。たとえばファイルクリエータをJCheckerのクリエータ「JCHK」にするとJCheckerの書類アイコンで保存できます。希望のファイルクリエータがわからないときは「選択」ボタンを押すと以下のようなファイル選択ダイアログが現れるので、希望のアイコンのファイルを選択するとそのクリエータがセットされます。



「上記のクリエータで常に保存する」

オンにしておくと、ファイルの保存時にファイルクリエータを書き換えます。ファイルクリエータを「JCHK」、「保存時のクリエータを書き換える」にすれば、チェックしたHTML書類は以下のアイコンで表示され、チェックしたかどうか一目でわかります。



JCheckerでせっかくエラーを直しても、GUIのHTMLエディタ(PageMillやClaris Homepage)に再読込させるとまたもとに戻ってしまいますが、上記のようにJCheckerのアイコンに書き換えてしまえば、誤って再読込させることを防げます。

「改行、タブマークを表示する」

オンにしておく、改行コードとタブコードが表示されるようになります。タブマークを表示すると、整形後の階層構造がよりわかりやすくなります。

「オートインデント」

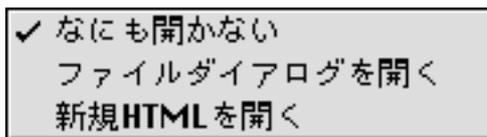
改行コードを挿入したとき、前行にあわせてオートインデントするかどうかを指定します。

「起動時にHTMLPlugInを起動する」

HTMLPlugInを、JCheckerの起動時に自動的に起動するかどうかを指定します。HTMLPlugInを先に起動させておくと、マクロメニューを選択したときマクロの立ち上がりははやくなります。



プログラム起動時メニューでは、JChecker起動時に「新規HTMLを開く」か「ファイル選択ダイアログを開く」か「なにもしない」かを指定できます。



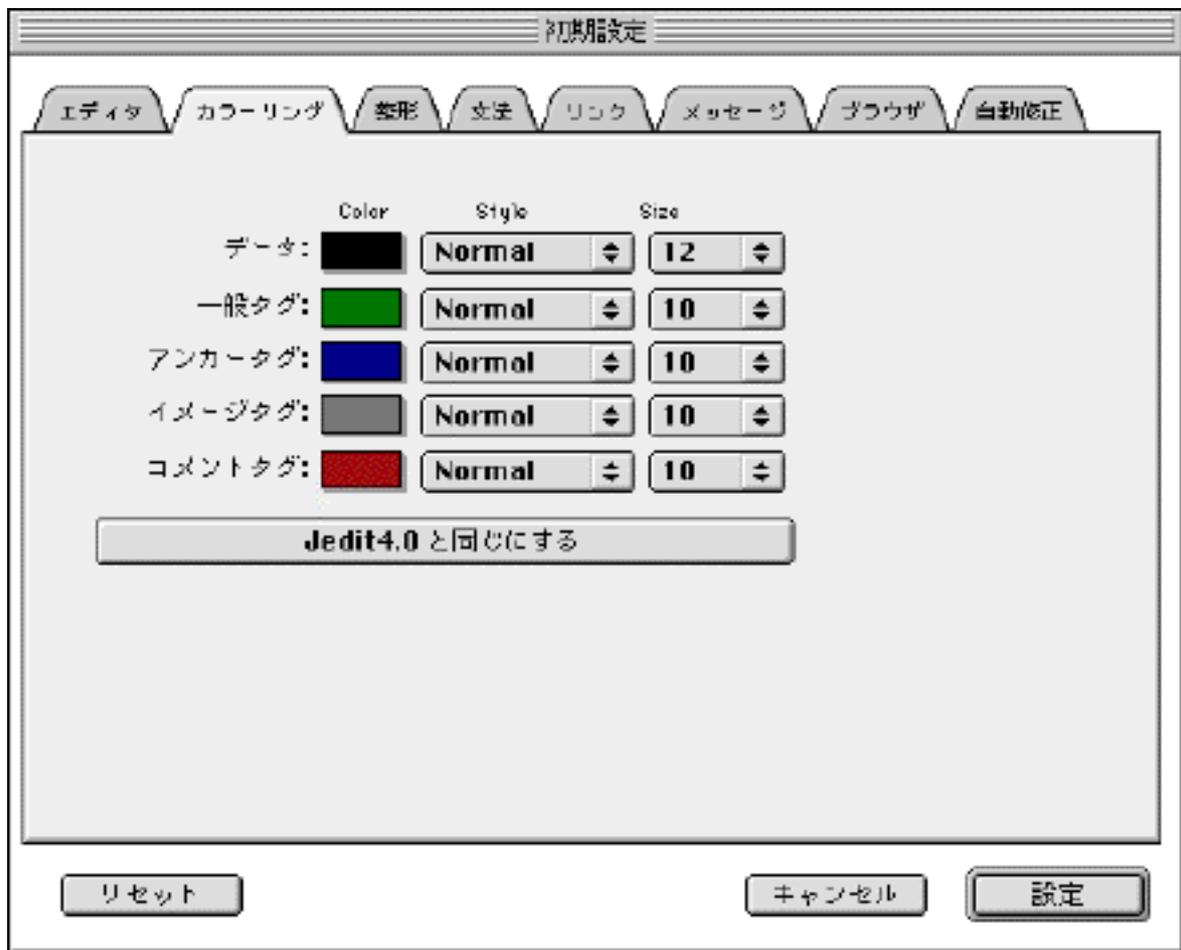
文字コードセットは、読み込むファイルの文字コードセットを指定します。日本語の場合は「Japanese Auto detect」を指定しておけば、読み込み時に Shift-JISか、JISか、EUCかを自動判定します。

改行コードは、読み込むファイルの改行コードの種別を指定します。「Auto detect」にしておけば、CR(Mac)か、LF(Unix)か、CR+LF(Dos)かを自動判定します。

エディタフォントはエディタで使用するフォントです。文字コードセットにマッチしたフォントを指定してください。たとえば、文字コードセットが日本語なのに欧文フォントを指定してしまうと漢字が正しく表示されなくなります。

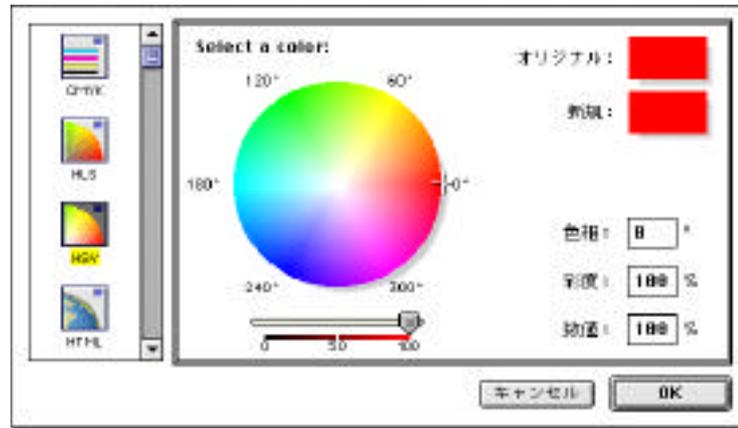
3.2 カラーリングの設定

メニュー「ファイル」「初期設定...」を開き、「エディタ」タブをクリックすると、以下のような設定パネルが表示されます。



HTML書類の構造が見やすくなるように「データ」「一般タグ」、「アンカータグ」<A>、「イメージタグ」、「コメントタグ」<!-- xxxx -->のそれぞれに、希望のスタイル、文字サイズ、文字色を指定できます。

文字色を指定するには、色を表示している部分をクリックすると以下のようなカラーダイアログが現れます。



エディタJedit4.0がすでにインストールしてあれば、「Jedit4.0と同じ」ボタンを押すと、Jedit4.0の初期設定ファイルからカラーリング設定をコピーします。

3.3 整形の設定

メニュー「ファイル」「初期設定...」を開き、「整形」タブをクリックすると、以下のような設定パネルが表示されます。

初期設定

エディタ カラーリング 整形 文法 リンク メッセージ ブラウザ 自動修正

階層に応じたインデントを付加する
インデント(タブ)幅(pt.): 24
エレメント名 小文字化する
属性名 小文字化する

<タグ> ラベルは1行にまとめる
 データ文からタブ/改行を削除し1行にする
 属性値は自動的に''マークで囲む

インデントも改行もしないタグ (文章修飾系)

A, ABBR, ACRONYM, APPLET, B, BASEFONT, BDO, BIG, BUTTON, CITE, CODE, DEFN, EM, FONT, I, FRAME, IMG, INPUT, KBD, LABEL, MAP, OBJECT, Q, S, SAMP, SCRIPT, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TEXTAREA, T, U, VAR

開始タグはインデントするが、終了タグは改行/インデントしない

CAPTION, DD, LI, P, TD, TH, TITLE

インデントはしないが開始タグ直後に改行する

ER, WBR

開始タグ前でインデントし、開始タグ直後に改行する

HR

リセット キャンセル 設定

整形の設定では、メニュー「チェック」「HTML整形」を実行したときのいろいろなオプションを指定できます。

「階層に応じたインデントを付加する」

HTMLの文章は階層構造になっています。<HTML>タグの下には <HEAD> <BODY>タグがあり、さらに<BODY>タグの下には<HR> <TABLE>タグなどが連なっています。この階層構造をインデントで表現するかどうかを指定します。オンにすると以下のように整形が行われます。階層構造の深さに応じてタグの前にタブコードが付加されて以下のようにインデント表示されます。

```
<HTML>  
  <HEAD>  
    <TITLE>タイトル</TITLE>  
  </HEAD>  
<BODY BGCOLOR="#FFFFFF">
```

```
.....  
</BODY>  
</HTML>
```

「<タグ>ラベルは1行にまとめる」

'<マークで始まり'>'マークで終わる 1 つのタグが、複数行にまたがっていたときに1行にまとめるかどうかを指定します。またオンの時は、アンカーに余分なスペースが入らないようにタグは行末に付加されるようになります。

「データ文からタブ/改行を削除し1行にする」

オンにすると、データ文にあるタブ・改行・連続したスペースを1つのスペースに置き換え、データ文を1行にまとめます。ただし、<PRE> <SERVER> <SCRIPT> <STYLE> <LISTING> <XMP> <PLAINTEXT>のエレメントに含まれるデータ文 (CDATA)については処理を行いません。

メニュー「エレメント名」

整形のときに、エレメント名を大文字にするか、小文字にするか、それともなにもしないかをポップアップメニューで指定します。

メニュー「属性名」

整形のときに、属性名を大文字にするか、小文字にするか、それともなにもしないかをポップアップメニューで指定します。

「属性値は自動的に" "マークで囲む」

オンにすると、整形時に" "マークで囲まれていない属性値があったら自動的に" "マークで囲みます。

「インデントも改行もしないタグ (文字修飾系)」

インデントも改行もしないエレメントを登録します。工場出荷時には以下のインラインエレメントが登録されています。

A, ABBR, ACRONYM, APPLETT, B, BASEFONT, BDO, BIG, BUTTON, CITE, CODE, DFN, EM, FONT, I, IFRAME, IMG, INPUT, KBD, LABEL, MAP, OBJECT, Q, S, SAMP, SCRIPT, SELECT, SMALL, SPAN, STRIKE, STRONG, SUB, SUP, TEXTAREA, TT, U, VAR

「開始タグはインデントするが終了タグは改行・インデントしない」

工場出荷時には以下のエレメントが登録されています。

CAPTION, DD, LI, P, TD, TH, TITLE

「インデントはしないが、開始タグ直後で改行する」

工場出荷時には以下のエレメントが登録されています。

BR, WBR

「開始タグ直前でインデントし、開始タグ直後で改行する」

工場出荷時には以下のエレメントが登録されています。

HR

上記のいずれにも指定されていないエレメントについては開始タグと終了タグで行を変えてインデントします。

3.4 文法チェックの設定

メニュー「ファイル」「初期設定...」を開き、「文法」タブをクリックすると、以下のような設定パネルが表示されます。



メニュー「デフォルト辞書」

HTMLの文法チェックをするときに使用する辞書を指定します。通常は「HTML 4.01 Transitional」を指定しておくが無難です。

「DOCTYPEからチェックする辞書を決める」

オンにしておくと、文法チェックに先立ちHTML書類の先頭にDOCTYPE宣言を解釈して、使用する辞書をきりかえるようにします。

「ウィンドウを開いたときに自動チェックする」

オンにしておくと、編集ウィンドウを開いたときに自動的にHTMLの文法チェックをします。

「エラーのほかに警告メッセージも表示する」

オンにしておくと、HTMLの文法チェックにしたときに、エラーメッセージと警告メッセージの両方を表示するようになります。

「エラーメッセージの文字サイズを小さくする」

オンにしておくと、編集ウィンドウのメッセージ領域に表示するエラーメッセージの文字サイズを9ポイントで表示します。オフの時は12ポイントで表示します。

「ソースを修正したとにエラー位置も更新する」

チェック後にソースを修正すると、エラーメッセージに対応するソース上のエラー位置がずれてしまいます。このチェックボックスをオンにしておくと、ソース修正時に自動的にエラー位置の更新も行うようになります。動作速度の遅い機種ではキー入力の応答が遅くなってしまうのでオフにしたほうがいいかもしれません。

「チェックの終了をサウンドで知らせる」

オンにしておくと、HTMLの文法チェックが終わったときサウンドで知らせます。エラーがあったときはアヒルの鳴き声のような「グエッ」という音で、警告だけでエラーがないときは心地よい「ピロン」という音で知らせます。また、エラーも警告もなかったときはファンファーレで知らせます。

 ボタンを押すとそれぞれのサウンドを試せます。また「変更」ボタンを押すと以下のようなダイアログがあらわれ、システムにある別のサウンドに変更することができます。



「メッセージファイルのクリエータ」

「ファイル」「メッセージの保存」メニューでエラーメッセージを保存するときのファイルクリエータを指定します。デフォルトは「ttxt」（SimpleText）になっています。右の「選択」ボタンを押すと、ファイル選択ダイアログが開き、希望のファイルを指定すると、そのファイルのクリエータがセットされます。

3.5 リンクチェックの設定

メニュー「ファイル」「初期設定...」を開き、「リンク」タブをクリックすると、以下のような設定パネルが表示され、リンクチェックでのいろいろな条件を設定できます。



「タイムアウト」

インターネット上のそれぞれのリンクを調べるときの制限時間を指定します。回線速度によっても違ってきますが通常は20秒程度を指定しておいてください。

「ローカルリンクをチェックする」

ローカルリンクとは、リンク先が自分のハードディスク内にあるファイルのとき、または、同一HTMLファイル内のアンカーのときです。ローカルリンクはインターネットへのアクセスを伴わないので高速にチェックできますので、通常はオンにしておいてください。

「リモートリンクをチェックする」

リモートリンクとは、リンク先がインターネット上のファイルのときです。Macintoshがインターネットにつながっていないときや、とりあえずローカルリンクだけをチェックしたいときなどにはオフにしてください。

「未サポートのプロトコルは警告する」

JCheckerでチェックできるリンクプロトコルは、「http://.....」「https://.....」「file://.....」の3種類です。このチェックをオンにしていると、チェックできないプロトコル（ftp:, mailto: など）があったときに警告メッセージを表示します。

高速チェック（HEAD ACCESS併用）

オンにしておく、リンク先のページの存在を調べるときに、そのすべてを読み込む（GETアクセス）のではなく、HEAD情報だけを読み込みます。すべてを読まないのリンクチェックが高速化します。

3.6 メッセージの設定

メニュー「ファイル」「初期設定...」を開き、「メッセージ」タブをクリックすると、以下のような設定パネルが表示されます。HTMLの文法チェックして検出した結果を、どのようなモードで出力するかを指定できます。



チェックの結果出力される70種類近くのメッセージのそれぞれについて、エラーとして表示するか、警告として表示するか、あるいは無視するかを指定できます。

ボタン「出荷時の状態にもどす」を押すと、メッセージのモードが出荷時の状態にもどります。出荷時には、HTMLの規則(DTD)に反するものはエラー、違反していないが好ましくないものは警告に設定してあります。

メッセージのなかの「%1」「%2」の文字列は、メッセージが実際に出力されるときに、ふさわしい文字列が代入されます。

3.7 ブラウザの設定

メニュー「ファイル」「初期設定...」を開き、「ブラウザ」タブをクリックすると、以下のような設定パネルが表示されます。編集ウィンドウのHTMLをプレビューするときのWebブラウザや、イメージを参照するときの画像ブラウザを設定します。



Webブラウザの設定

新たにWebブラウザを登録・追加するときは、ボタン「新規」をクリックする以下のように追加するWebブラウザを聞いてきます。希望のブラウザプログラムを指定して「開く」ボタンを押すと追加されます。



またそのブラウザのプレビューのショートカットキーも指定できます。

登録されているブラウザを変更するときは、リストのなかでそのブラウザを選択し、ボタン「変更」をクリックします。

登録されているブラウザを削除するときは、リストのなかでそのブラウザを選択し、ボタン「削除」をクリックします。

画像ブラウザの設定

ボタン「設定」をクリックする以下のように設定する 画像ブラウザを聞いてきますので、希望のイメージ編集プログラムを指定してください。



3.8 自動修正の設定

HTMLの文法チェックで検出したエラーのうち、約20種類のエラーについては、JCheckerで自動修正が可能です。メニュー「ファイル」「初期設定...」を開き、「自動修正」タブをクリックすると、以下のように自動修正できる項目がリスト表示されます。

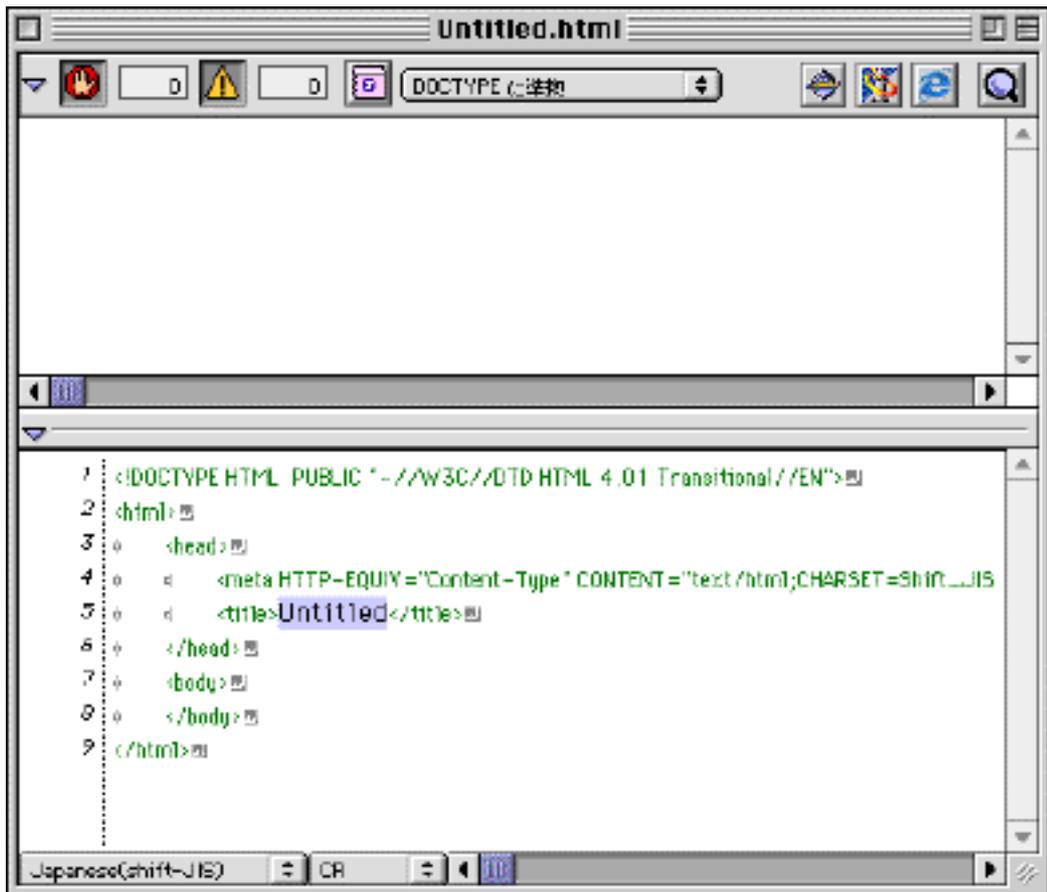


各項目のチェックボックスをオフにすると、その項目については自動修正から除外されます。

第4章 ファイル操作

4.1 新規にHTMLを作成する

HTML書類を新規に作成するときは、「ファイル」「新規HTML」メニューを選択します。すると、以下のような必要最小限のタグが埋め込まれたHTML書類が開きます。



ファイルメニュー「初期設定」で新規HTMLのDocument Type や文字セットを、あらかじめ指定しておくことができます。

4.2 ローカル書類を開く

(1)開きたいJChecherの書類をダブルクリックする。

図のようなJChecherの書類アイコンの書類であれば、ダブルクリックするとJChecherが起動しその書類が開きます。



なお、同じハードディスクに旧版JChecher2.1があると、誤って古いJChecher2.1が起動してしまう場合があります。JChecher2.1が起動しないように、旧版JChecher2.1はゴミ箱へ捨ててください。

(2)開きたい書類を、JChecherのアイコンにドロップする。

この方法では、JChecherの書類アイコンでないHTMLファイルでも開くことができます。

(3)ファイルメニュー「開く...」「選択ダイアログ...」から開くファイルを選択する。

ファイルメニュー「開く...」「選択ダイアログ...」を選択すると既存のHTMLファイルを開くことができます。



4.3 リモート書類を開く

インターネットのWebサーバー上のHTML書類を開くには、「ファイル」「アドレスを開く...」を選択します。選択すると図のようなダイアログが現れ、URL(インターネット上のアドレス)を聞いてきます。

*



URLはいちいちキー入力しなくても、たとえばブラウザのURL表示部分をこのダイアログのURLフィールドへドラッグコピーすれば、ブラウザで表示しているHTMLのチェックを行うことができます。

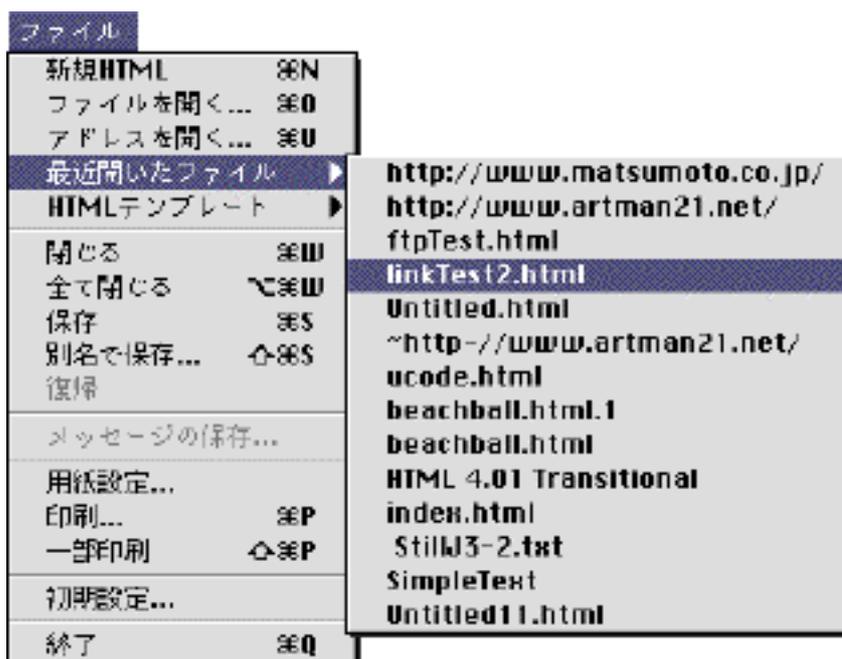
なお、「アドレスを開く...」では、httpプロトコルでファイルを読みだすため、編集したファイルをリモートサーバーへ書き戻すことはできません。

後述の「FTPプログラムとの連動」で説明する方法を使うとリモートファイルの読み込みも書き込みも行えます。

4.4 最近開いたファイル

ファイルメニュー「最近開いた書類」を選択すると図のような階層メニューに、最近開いたファイルが表示されます。そのなかにかきたいファイルがあれば、それを選択するとメニューからワンタッチで開きます。階層メニューに登録されるファイルの最大数は15です。

また、optionキーを押しながら階層メニューの項目を選択すると、その項目を階層メニューから削除できます。



4.5 HTMLテンプレート

JChecker3.0フォルダの「Templates」フォルダに自分がよく使うHTML書類のひな形を格納しておく、と、「ファイル」「HTMLテンプレート」階層メニューに一覧表示され、それらをワンタッチで開くことができます。ホームページのスケルトンHTMLなどを、ここに格納しておく、と便利です。



4.6 書類を閉じる

「ファイル」「閉じる」メニュー

開いている HTML 編集ウィンドウを閉じます。HTMLの書類を修正しているときは書類を保存するかどうかが聞いてきます。

「ファイル」「すべて閉じる」メニュー

開いているすべてのHTML編集ウィンドウを閉じます。

4.7 書類を保存する

「ファイル」「保存」メニューを選択すると、編集中のテキストをファイルに保存します。

編集している書類が「新規書類」のとき、またはファイルメニュー「別名で保存...」を選択したときは、以下のような保存ダイアログが表示されるので、保存先のファイル名を指定し保存してください。



保存ダイアログでは、保存先の文字コードセットと改行コードのタイプを指定できます。

「ファイル」「別名で保存...」メニューを選択すると、開いているHTML編集ウィンドウの内容を、希望のファイル名、希望の文字セット、希望の改行コードでディスクに保存します。

4.8 書類の復帰

「ファイル」「復帰」メニューを選択すると、編集ウィンドウで編集した内容を無効にして、書類を読み込んだ直後の状態、保存しているときは最後に保存したときの状態にもどします。

4.9 エラーメッセージの保存

「ファイル」「メッセージの保存」メニューを選択するとHTMLのチェック結果（メッセージウィンドウの内容）を、希望のファイル名でディスクに保存します。保存時のファイルクリエータ（アイコン）は、「初期設定」「文法」で指定できます。デフォルトはSimpleTextのアイコンで保存されます。

4.10 書類の印刷

「ファイル」「用紙設定...」メニューを選択すると、用紙設定ダイアログが表示され印刷時の用紙を設定できます。

「ファイル」「印刷...」メニューを選択すると、印刷ダイアログが表示され、編集ウィンドウを印刷できます。

「ファイル」「一部印刷...」メニューを選択すると、印刷ダイアログなしに一部だけ編集ウィンドウを印刷します。

第5章 FTP転送プログラムとの連動

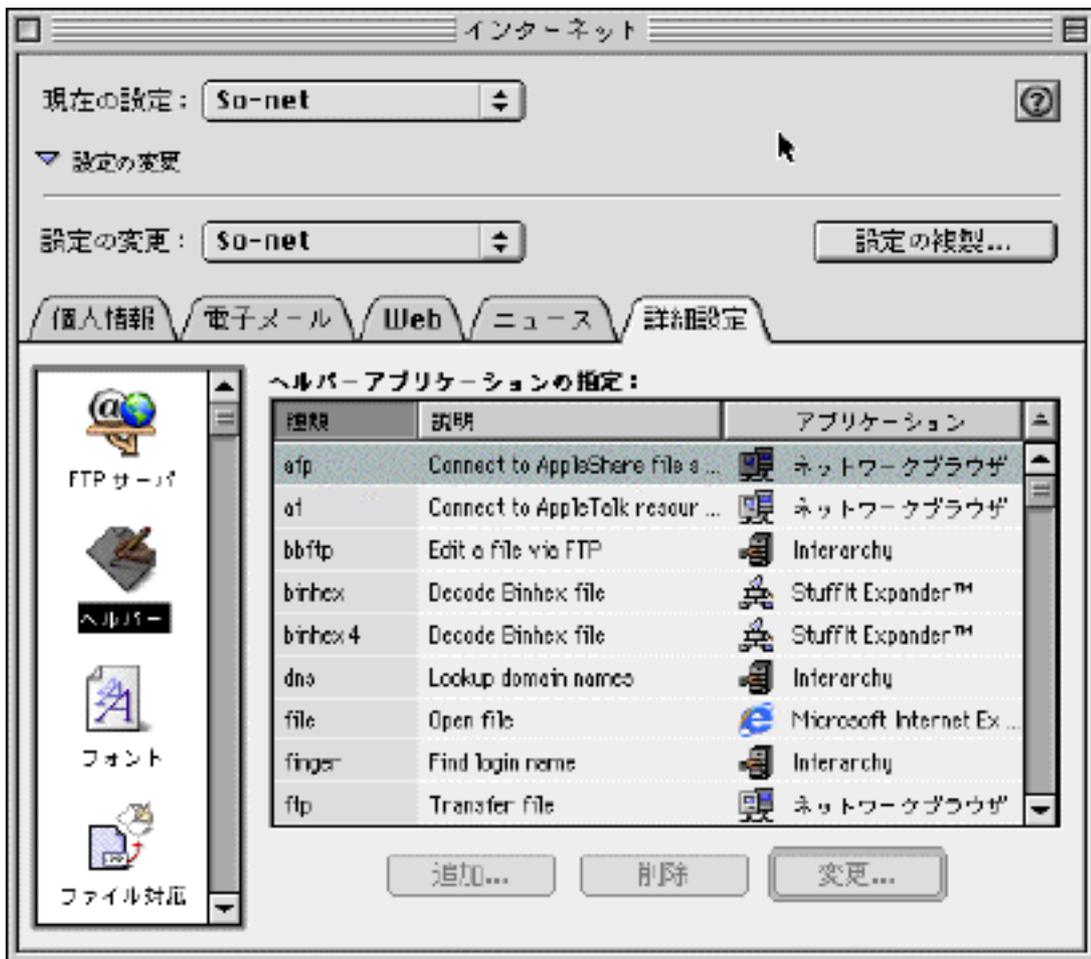
JCheckerは、**外部エディタプロトコル** をサポートしているので、FTPプログラム（Interarchy, NetFinder, Transmit, Fetchなど）のメニューから直接、リモートサーバー上のファイルをJCheckerで編集できます。すなわち、WEBサーバー上のhtmlファイルなどをJCheckerで直接編集できます。（ただし、外部エディタプロトコルは、Classic OS上でのみ有効で、残念ながらOS X上では動作しません。）

5.1 設定方法

InterarchyなどのFTP転送プログラムのメニューには「Edit with BBEdit」があります。これはFTPサーバー上のファイルをBBEditで直接編集するための機能ですが、以下のようにするとそのメニューを「Edit with JChecker」に変更できます。

1) アップルメニュー>コントロールパネル>インターネットを選択し、さらにそのなかの「詳細設定タブ」を選択します。左側のヘルパーアイコンをクリックすると以下のようにヘルパーアプリケーションがリスト表示されます。

もし詳細設定タブが表示されていないときは、編集メニュー「利用者モード」を選択して、利用者モードを「情報を管理」に変更してください。



2) リストに種類「ftp-editor」がすでに登録されているかどうかチェックしてください。登録されていないときは「追加...」ボタンを、登録されているときは「変更」ボタンを押し、以下のように種類に「ftp-editor」、説明に「editor」、ヘルパーアプリケーションに「JChecker3」を指定してください。「OK」ボタンを押せば設定完了です。



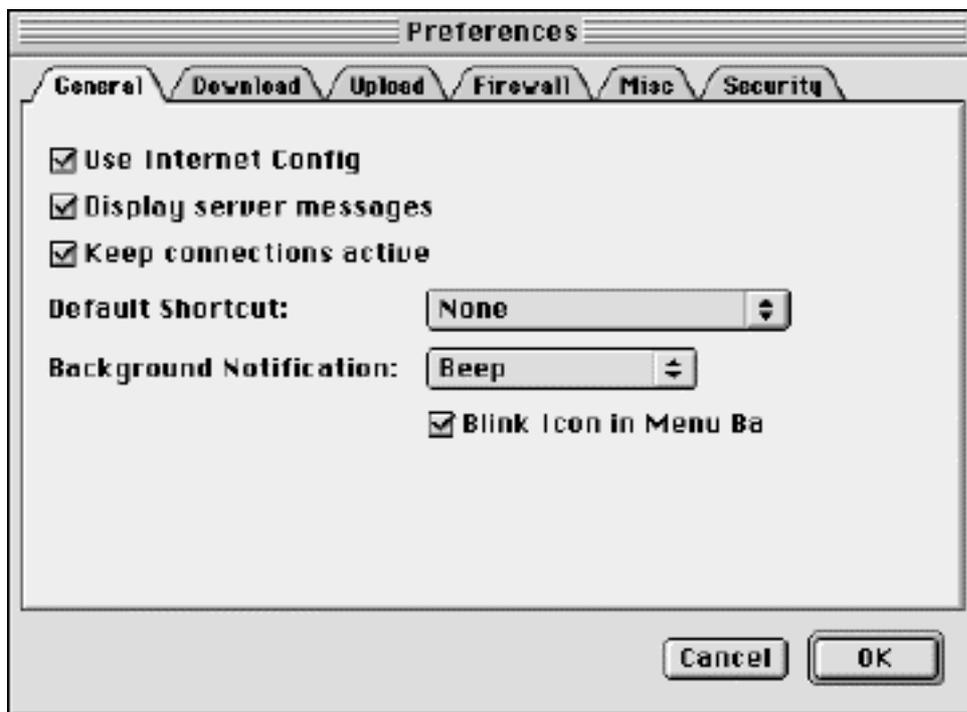
3) たとえばInterarchy4.1以降では、メニューが以下のように変わってInterarchyから

JCheckerを直接使えるようになります。

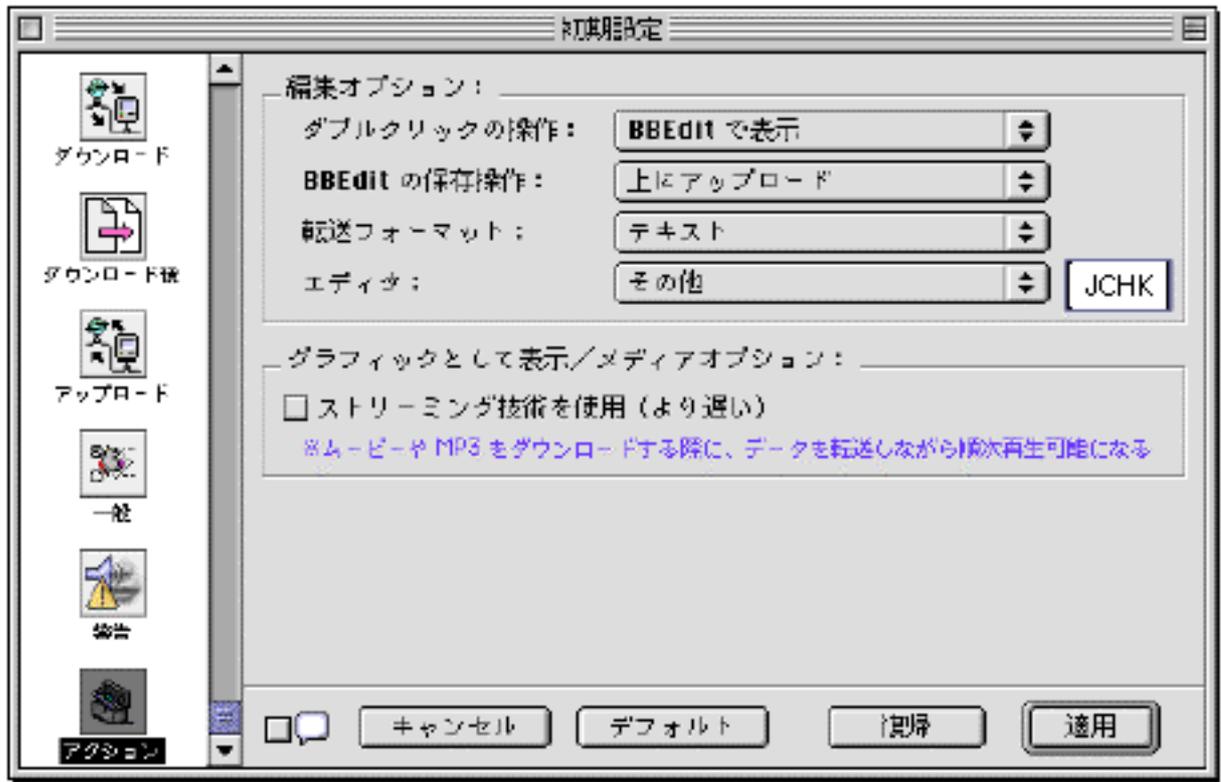
Listing

Download	
View as Text	⌘L
Edit with JChecker	⌘J
Upload	⌘U
Rename/Move	⇧⌘R
Create Directory	⌘N
Delete	⌘⌫
Open Parent	⌘↑
Export Listing	
Set Permissions	⇧⇧⌘P
Set Upload Permissions	⇧⇧⌘U
Send Raw FTP Command	

Fetch4.0では、メニュー「Customize > Preferences > General」で以下のようにUse Internet Config をオンにしてください。オフのときは2)でセットの内容がFetchに反映されません。



NetFinderの場合は、メニュー「編集」>「NetFinder初期設定」を選択し、アイコンリストの最後にある「アクション」アイコンをクリックし、以下のようにエディタに「その他」を指定し、そのクリエータに「JCHK」を指定してください。NetFinderの場合はメニューは「BBeditで編集」のまま変わりません。



5.2 使用方法

Interarchy4.1を例に説明します。

Interarchy4.1でFTPサーバーのディレクトリをリスト表示させ、編集したいファイルを選択します。その状態でListingメニューの「Edit with JChecker」を選択すると、JCheckerでそのファイルが開き編集できます。

編集が終わったら、JCheckerの「保存」メニューを選択すると、その内容が自動的にFTPサーバー上のファイルに保存されます。

JCheckerの「最近開いた書類」メニューには、上記で編集したファイルも自動的に登録されます。「最近開いた書類」メニューからそのファイルを選択すると、自動的にInterarchy4.1が起動され、そのファイルをFTPサーバーから読み込んできてJCheckerで編集できます。

ノート

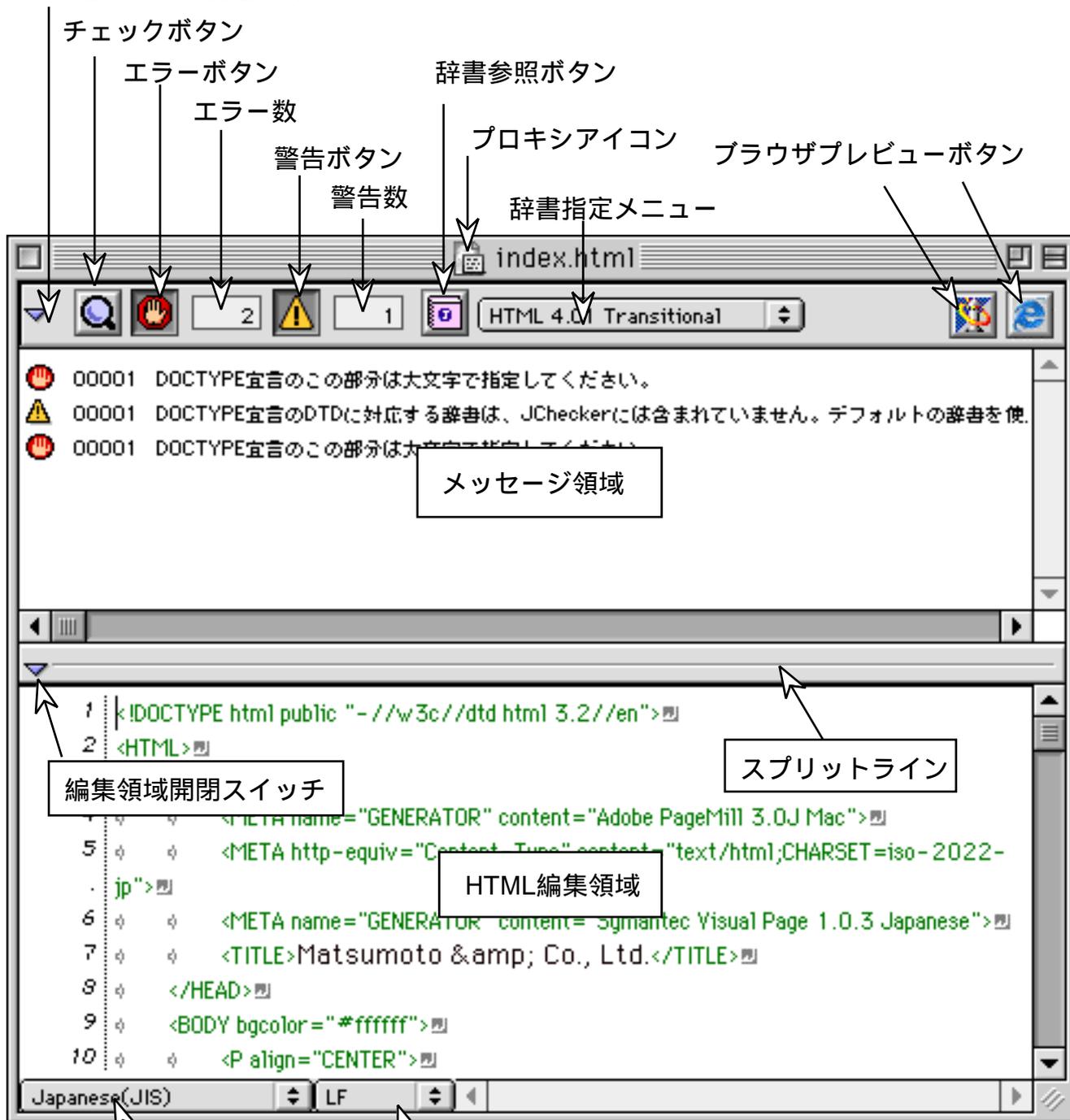
外部エディタプロトコルはClassic OS上に限られ、残念ながらMac OS X版のJCheckerでは動作しません。

第6章 編集操作

6.1 編集ウィンドウの各部の名称

編集ウィンドウは、上部のメッセージ領域と下部のHTML編集領域の2つに分かれています。メッセージ領域と編集領域の比率は中央のスプリットラインをマウスで上下にドラッグして変更されます。

メッセージ領域開閉スイッチ



プロキシアイコン

ウィンドウタイトル左の小さなアイコンをプロキシアイコンといいます。
このアイコンは、その書類のファイルアイコンと同じようにFinder上へドラッグして、ファイルを移動したりコピーしたりすることができます。
文書が変更されてまだ保存されていないときは、プロキシアイコンがグレイ表示になります。
グレイ表示のときは、アイコンをドラッグすることはできません。

ディレクトリメニュー

タイトル部分をcmdキーを押しながらマウスでクリックすると以下のように編集ファイルのディレクトリメニューがポップアップします。

この例では、ファイル「第4章編集操作」は、ドライブ「FireWire HD」のなかのフォルダ「CW6」のなかのフォルダ「Jedit4.0開発」のなかのフォルダ「マニュアル」にあることがわかります。メニューの各項目を選択すると対応するフォルダがウィンドウとしてFinder上に開きます。



ウィンドウ上左端の三角のメッセージ領域開閉スイッチをクリックするとメッセージ領域の開閉を行えます。

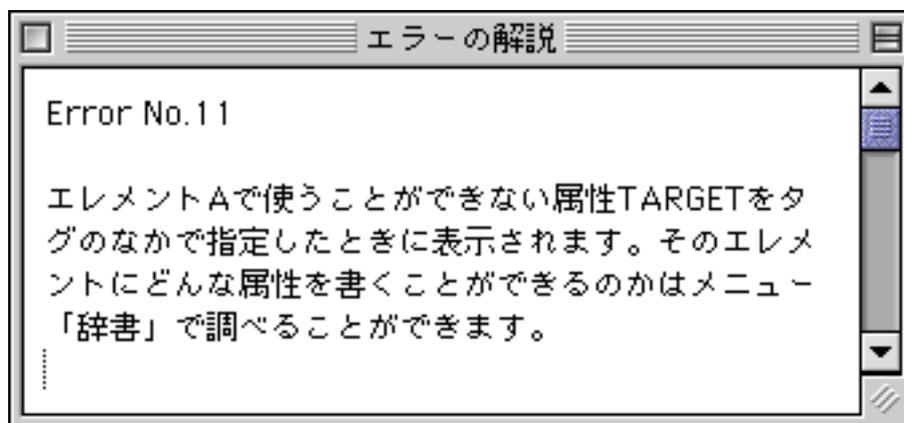
中央左の三角の編集領域開閉スイッチをマウスでクリックすると編集領域の開閉を行えます。

チェックボタン

クリックすると、編集領域のHTML文書の文法チェックを行います。
オプションキーを押しながらクリックすると文法チェック後自動修正を行います。
シフトキーを押しながらクリックするとリンクチェックを行います。

エラーや警告を検出すると、それらをメッセージ領域に表示します。また、表示されたメッセージをマウスでクリックすると、対応するHTML文書の該当部分がエディタウィンドウに選択表示されます。

メッセージをダブルクリックすると、そのメッセージのさらに詳しい解説が表示されます。



なお、文法チェック/リンクチェックは、メニュー「チェック」「文法チェック」/「リンクチェック」からもおこなえます。



エラーボタン

オンとオフの状態をもち、オンのときは検出したエラーメッセージをメッセージ領域に表示します。オフにすると、エラーメッセージを隠すことができます。



警告ボタン

オンとオフの状態をもち、オンのときは検出した警告メッセージをメッセージ領域に表示します。オフにすると、警告メッセージを隠すことができます。



ブラウザプレビューボタン

初期設定「ブラウザ」で登録されているブラウザがボタン表示されます。ボタンを押すと、対応するブラウザで、編集領域のHTMLをプレビューできます。初期設定「ブラウザ」では登録できるブラウザ数に制限はありませんが、ブラウザプレビューボタンとしては先頭の4つのブラウザがボタン表示されます。ボタンで表示されないブラウザは、メニューブラウザから選択できます。

辞書指定メニュー

文法チェックするときの辞書を指定します。「DOCTYPEに準拠」を指定しておく、チェックのときにHTMLの先頭のDOCTYPE宣言に対応する辞書を自動的に選択します。



辞書ボタン

カーソル位置のタグについて、以下のような辞書ウィンドウを開き、そのエレメントを調べることができます。



文字コードセット指定メニュー

ファイルを読み込んだときの文字コードセットを示し、次にファイルへ保存するとき、どの文字セットで保存するかをこのメニューで変更できます。

改行コード指定メニュー

ファイルを読み込んだときの改行コードを示し、次にファイルへ保存するとき、どの改行コードで保存するかをこのメニューで変更できます。

スプリットライン

このラインのどこかをマウスで上下方向にドラッグすると、メッセージ領域と編集ウィンドウの比率を変更できます。

6.2 キー操作

カーソル移動キー

方向キーで以下のようにカーソルの移動を行えます。

カーソルを1文字左へ キー

カーソルを1文字右へ	キー
カーソルを1行上へ	キー
カーソルを1行下へ	キー
カーソルを1ワード左へ	option+ キー
カーソルを1ワード右へ	option+ キー
カーソルを段落の先頭へ	command+ キー
カーソルを段落の末尾へ	command+ キー
カーソルをページ最上行へ	option+ キー または PageUpキー
カーソルをページ最下行へ	option+ キー または PageDownキー
カーソルを文書の先頭へ	command+ キー または Homeキー
カーソルを文書の末尾へ	command+ キー または Endキー

その他のキー操作

- ・ controlキーを押しながらマウスをクリックするとコンテキストメニューが開きます。
- ・ cmdキーを押しながら相対パスのURL文字列どこかをクリックすると、リンク先がテキストファイルのときは新しい編集ウィンドウを開きそのファイルを表示します。リンク先がイメージファイル（ファイルタイプがJPEG、GIFf、PNGf）のときは、そのファイルを画像ブラウザで開きます。
- ・ ファンクションキーF1=Undo、F2=Cut、F3=Copy、F4=Paste。
- ・ cmdキーを押しながらreturnキーを押すと、カーソル位置へジャンプします。
- ・ shiftキーを押しながらdeleteキーを押すと、キャレット右隣の文字が削除されます。
- ・ shiftキーを押しながら「次を検索」(cmd+G)で文頭への検索になります。
- ・ shiftキーを押しながら「選択文字列で検索」(cmd+H)で文頭への検索になります。
- ・ optionキーを押しながらdeleteキーを押すと、キャレット右隣から段落末までの文字が削除されます。
- ・ optionキーを押しながらファイルメニューの「最近開いた書類」の項目を選択するとその項目をリストから削除できます。
- ・ optionキーを押しながらブックマークリストのファイル項目をダブルクリックすると、ジャンプ先ファイルを指定し直すことができます。

6.3 取消し(Undo)とやり直し(Redo)

編集ウィンドウでのテキストを変更するようなすべての操作について「取消し」と「やり直し」が可能です。

編集メニュー「取消し」(Undo)を選択すると、直前の操作を取消し、元の状態にもどすことができます。

また、編集メニュー「やり直し」(Redo)を選択すると、直前に取り消した操作を再度行えます。

JCheckerでは無限Undoをサポートしているので、ファイルを読み込んだときまで、「取り直し」で順次さかのぼることができます。またそこから「やり直し」で最新の状態まで順次復元できます。

6.4 コピー、カット、ペースト

一般的なMacintoshのアプリケーションと同様、編集中のテキストについて、コピー、カット、ペーストが行えます。

領域の選択について

コピーまたはカットしたい領域は、通常はマウスでドラッグすれば指定できますが、以下の方法もサポートしています。

- ・ダブルクリックでワード選択、トリプルクリックでパラグラフ選択します。
- ・shiftキーを押しながら、クリックすると選択領域の追加となります。
- ・shiftキーを押しながら、カーソル起動キーを使用すると移動した領域が選択されます。
- ・編集ウィンドウ左の行番号表示部分をマウスでドラッグすると行選択を行えます。

6.5 ドラッグ&ドロップ

ドラッグ&ドロップ機能を使うと、コピー&ペーストを使用することなくウィンドウ間あるいは他のアプリケーションとの間でデータの移動/複写を簡単に行うことができます。

たとえば、テキストの一部を別のウィンドウへコピーするには以下のようにします。

まず編集ウィンドウのコピーしたい部分をマウスで選択します。マウスポインタを選択領域のなかにもっていくと、ポインタがの形になります。その状態でマウスボタンを押すとポインタはに変わりますので、ボタンを押したままマウスを移動し、マウスポインタがコピー先のウィンドウのなかにはいったら、ボタンをはなしてください。以上で、選択領域が別のウィンドウへコピーされます。

同一のウィンドウ内で、ドラッグ&ドロップを行なうと選択された部分の「移動」になりま

す。このとき、optionキーを押していれば「移動」ではなく「複写」になります。

デスクトップへドロップを行うと、以下のようなクリッピングファイルが作成されます。



クリッピングファイルを編集ウィンドウにドロップすると、クリッピングファイルの内容がペーストされます。

テキストファイルを編集ウィンドウへドロップすると、アンカータグが自動生成され、そのファイルの URL が相対パスで挿入されます。



ドロップ

```
<A href="[そのファイルの相対URL]" ></A>
```

イメージファイル (JPEG, GIF, PNG) を編集ウィンドウへドロップすると、IMGタグが自動生成され、そのファイルの URL が相対パスで挿入されます。またイメージの大きさに応じた属性値がwidthとheightにセットされます。



ドロップ

```
<IMG src="[そのファイルの相対URL]" width="999" height="999" border=0 alt="[ファイル名]">
```

また、テキストファイルやイメージファイルをシフトキーを押しながら編集ウィンドウへドロップすると、そのファイルの相対パスだけが挿入されます。

6.6 エレメントの選択

カーソルをタグのどこかにセットして、メニュー「編集」「エレメントを選択」を実行すると、そのエレメント全体 (開始タグから終了タグまで) が選択されます。終了タグが省略されている場合でも、そのエレメントの範囲を正しく選択します。

6.7 タグの削除

タグを削除したい領域をあらかじめ選択して、メニュー「編集」「タグの削除」を実行すると、その領域の全てのタグ ('<'と'>'マークで囲まれた部分) が削除されデータ文だけになります。HTMLから通常のテキストに戻したいときなどに使用すると便利です。

6.8 日付・時刻のスタンプ

編集メニュー「編集」「日付スタンプ」または「時刻スタンプ」を選択すると、カーソル位置へ、「日付」または「時刻」がペーストされます。
このときshiftキーを押していると、「長い日付」または「秒付きの時刻」がペーストされま
す。

短い日付 99.3.17
長い日付 1999年 3月 17日 水曜日

短い時刻 2:43 PM
長い時刻 2:43:36 PM

第7章 検索と置換

JChecker3.0は、正規表現、マルチファイル検索、マークリストなど、エディタ Jedit4.0とほぼ同等の強力な検索・置換機能を備えています。

7.1文字列の検索

編集中の文書から文字列を検索するには、まずメニュー「検索」「検索...」を選択します。



検索したい文字列を検索フィールドにキー入力して、「検索」ボタンを押すと現在のカーソル位置から検索が始まり、該当の文字列が見つかったらその部分が選択状態になって表示されます。

さらに「検索」ボタンを押せば、次の該当文字列が表示されます。該当文字列がないときは、ピープ音とともに検索が終了します。検索ボタンのかわりにメニューの「検索」「次を検索」でも検索できます。

「一括検索」ボタンを押すと、文書の最後まで検索し該当箇所を以下のマークリストにまとめ

て表示します。



マークリストの希望の項目をダブルクリックすれば、その箇所が編集ウィンドウに表示されま
す。

検索フィールド右横の上の  ボタンを押すと、最近検索した文字列（最大10項目）がメ
ニューとしてポップアップします。希望の項目を選択すれば検索フィールドにセットされま
す。

検索フィールド右横の下の  ボタンを押すと、よくつかう正規表現のパターンメニューがあ
らわれます。



メニューの先頭の「パターンに登録」を選択すると、現在検索フィールドに表示されている文字列をこのメニューへ登録できます。

また、一度登録したメニューは、optionキーを押しながら、そのメニュー項目を選択するとメニューから削除されます。

検索文字列に制御コードを指定したいときは、以下のようにエスケープコード「¥」を使って表現します。

¥t = TAB, ¥r = CR, ¥n = LF, ¥s =SPACE, ¥b = BS

エスケープコード「¥」を使った表現についてさらに詳しくは後述の正規表現を参照してください。正規表現と同じエスケープコード「¥」表現が、通常の検索でも使用できます。

検索メニュー「選択域の取込み」を行うと、文書を選択領域の文字列が検索条件ダイアログの検索文字列としてとりこまれます。

検索メニュー「選択文字列で検索」を行うと、文書を選択領域の文字列を検索文字列として検索が行われます。このとき検索文字列は検索条件ダイアログには取り込まれません。

チェックボックス「先頭へ検索」をオンにすると、現在のカーソル位置から文書の先頭へ向けて検索を行います。

チェックボックス「書類を巡回」をオンにすると、文章の最後までいっても見つからなかったときには、文書の先頭にもどって検索を続けます。

チェックボックス「大文字小文字の区別」をオフにすると大文字、小文字によらず指定の文字列を検索します。(オフのときは多少検索速度が遅くなります。)

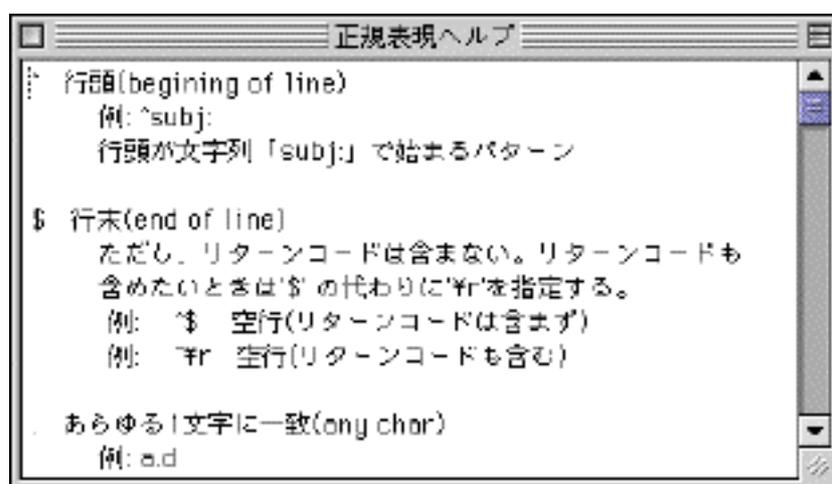
チェックボックス「全角半角の区別」をオフにすると全角、半角によらず指定の文字列を検索します。(オフのときは多少検索速度が遅くなります。)

チェックボックス「ワード単位」がオンのときは、見つかった文字列が語(ワード)として独立している場合のときだけ該当するようになります。(ただし、検索対象が日本語の文章の場合はこのチェックボックスはあまり意味がありません。)

チェックボックス「正規表現」がオンのときは、検索文字列に正規表現を指定できます。

正規表現を使うと、単一文字列以外に指定のパターンにマッチする複数の文字列を一括して検索できます。

 ボタンを押すと以下のような正規表現についてのヘルプウィンドウがひらきます。



7.2 正規表現について

JCheckerの正規表現は2バイトコードに対応していて、以下の記号を使って検索したい文字列のパターンを指定します。

^ 行頭(beginning of line)
例: ^subj:
行頭が文字列「subj:」で始まるパターン

\$ 行末(end of line)
ただし、リターンコードは含まない。リターンコードも含めたいときは'\$'の代わりに'\$r'を指

定する。

例: `^\$` 空行(リターンコードは含まず)

例: `^\r` 空行(リターンコードも含む)

あらゆる1文字に一致(any char)

例: `a.d`

aで始まり、つぎになにか1文字つづき、dで終わる文字列。「and」「add」「dandy」「and」などがマッチする。

* パターンに0文字以上連続でマッチします。

例: `an*d`

aで始まり0回以上のnの繰り返しを含みdで終わる文字列。「and」「advance」「add」などがマッチする。

*? 最短一致でパターンに0文字以上連続でマッチします。

例えば、「aaa bbb bbb bbb cccc」という文字列のなかで、

「`.*bbb`」を検索すると「aaa bbb bbb bbb」がマッチするが、

最短一致で「`.*?bbb`」と指定すると「aaa bbb」がマッチするようになる。

+ パターンに1文字以上連続でマッチします。

例: `an+d`

aで始まり1回以上のnの繰り返しを含みdで終わる文字列。

「and」「annd」はマッチするが、

「advance」「add」などはマッチしない。

+? 最短一致でパターンに1文字以上連続でマッチします。

? パターンに0文字か1文字でマッチします。

例: `an?d`

aで始まり1文字のnまたはすぐにdで終わる文字列。

「and」「add」はマッチするが「annd」はマッチしない。

?? 最短一致でパターンに0文字か1文字でマッチします。

¥ エスケープ。

¥`^`, ¥`$`, ¥`*`, ¥`+`, ¥`?`, ¥¥のように「`^`」「`$`」など正規表現として意味をもつ特殊記号を単なる文字として指定したいときに使用します。

また、以下のようなメタ文字を使うと制御コードや特定の文字種をまとめて表現できます。

¥`t` = TAB,

¥`r` = CR,

¥`n` = LF,

¥`s` = SPACE,

¥`b` = BS,

¥`d` = [0-9] (数字)

¥w = [A-Za-z_] (アルファベットとアンダースコア)

さらに文字を16進コード指定するときは、以下のように¥hをつけて指定します。

1バイト文字は ¥hxx で、
2バイト文字は ¥hxxxx で

例: ¥h8140 全角スペース(shift JIS漢字コード 8140)

1バイト文字「¥hxx」を指定したときは、2バイト文字はその検索の対象にはなりません。
すなわち1バイト文字「¥h81」を検索しても全角スペース(0x8140)の前半バイトがヒットすることはありませんので注意してください。

[] 文字クラス。

文字にはもちろん全角文字も指定できます。
文字クラスのなかでは範囲をあらわす「-」や否定をあらわす「^」も指定できます。

例: [abc]
文字a,b,cのどれかに一致する文字

例: [A-Z]
文字AからZまでに一致する文字(大文字アルファベット)

例: [^a-z]
文字aからz(半角小文字)以外の文字にマッチ。

例: [a-xABC]
文字aからxまでとA,B,Cのどれかにマッチする1文字。

例: [ア-ケー]+
全角カタカナが1文字以上連続した文字列。

例: [¥t¥s¥h8140]+\$
行末にある連続したタブ、スペース(半角/全角)。

{ } グループ化

検索文字列をグループ化すると、置換文字列に各グループの文字列を代入することができます。なお、{ と } のかわりに、¥(と ¥) でも指定できます。

たとえば検索文字列を、日本語日付として

{[0-9]+}年{[0-9]+}月{[0-9]+}日

のように3つのグループを指定します。

検索した日付を英語表記 (mm.dd.yy)に置き換えるには置換文字列を以下のように指定します。

¥2.¥3.¥1

「¥2」は検索文字列の2番目のグループすなわち月データを表し、「¥3」は日データ、「¥1」は年データを表します。

上記の正規表現で検索と置換を行うと、

1997年11月25日 ----> 11.25.1997

のようになります。

！ 複数文字列の同時検索

以下のように「！」マークを使うと、複数の正規表現を同時に検索できます。

<検索文字列1> | <検索文字列2> ! <検索文字列3>

上記の例では、<検索文字列1>、<検索文字列2>、<検索文字列3>のどれかに該当する文字列をいっぺんに検索します。

なお、！ は、文字クラス[] やグループ{ } の内部に含めることはできません。

文字列にグループ指定があるときは、以下のように検索文字列ごとにグループ番号が割り当てられます。

たとえば、半角の日付 (99.99.99) と全角の日付 (9 9 . 9 9 . 9 9) を同時に検索し、「99年99月99日」に置き換える場合を考えると、

検索文字列は、

$\{[0-9]+\}\{[0-9]+\}\{[0-9]+\}\{[0-9]+\}\{[0-9]+\}\{[0-9]+\}$
¥1 ¥2 ¥3 ¥1 ¥2 ¥3

置換文字列は、

¥1年 ¥2月 ¥3日

にします。

7.3 文字列の置換

検索した文字列を置き換えるには、検索条件ダイアログで置換したい文字列を指定し、メニュー「検索」「置換」または「置換&検索」を実行します。

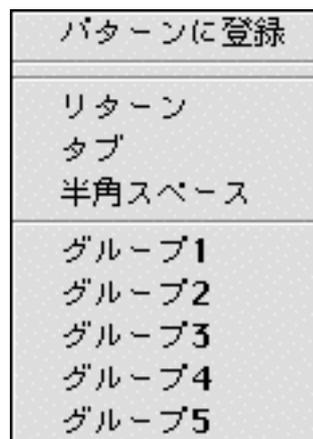
「置換&検索」の場合は、置き換えてから次の候補の検索を行います。検索ダイアログの「置換」ボタン、「置換&検索」ボタンからも行えます。

編集メニューの「すべてを置換」またはダイアログの「一括置換」ボタンを実行すると、選択領域のなかの該当文字列をすべて置き換えます。

領域が選択されていないときは、現在のカーソル位置から文末までを対象として該当文字列をすべて置き換えます。

置換フィールド右横の上の  ボタンを押すと、最近置換した文字列（最大10項目）がメニューとしてポップアップします。希望の項目を選択すれば置換フィールドにセットされます。

置換フィールド右横の下の  ボタンを押すと、よくつかう置換文字列のパターンメニューがあらわれます。



メニューの先頭の「パターンに登録」を選択すると、現在置換フィールドに表示されている文字列をこのメニューへ登録できます。

また、一度登録したメニューは、optionキーを押しながら、そのメニュー項目を選択するとメニューから削除されます。

7.4 マルチファイル検索・置換

JChecker3.0では、ディスク上の複数のファイルを対象に文字列の検索を行うことができます。

検索条件ダイアログの下半分でマルチファイル検索での対象ファイルを指定します。



通常の検索モードからマルチファイル検索モードへ切り替えるには、 ボタンを押しま

す。ボタンが凹んだ  の表示になりマルチファイル検索モードになります。

また、ボタン左の  マークをクリックするとファイル指定部分の表示をオンオフできます。

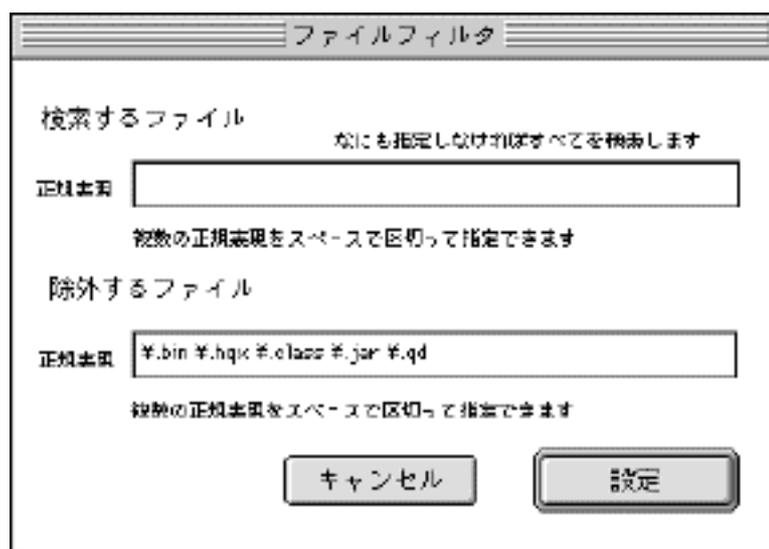
検索対象となるファイルを指定するには「追加」ボタンを押すと以下のような選択ダイアログが現れます。ダイアログには候補としてテキストファイルと「初期設定」「オープン」「読み込めるファイルタイプ」で指定しているファイルが一覧表示されます。



ファイルを複数同時に選択できますし、またフォルダも選択できます。フォルダを選択するとそのなかのすべてのファイル（サブフォルダも含む）が検索リストに追加されます。

あるいは、検索したいファイルやフォルダを直接Finder上からドラッグし検索リストへドロップしても検索ファイルを追加できます。

フォルダなどでまとめて検索ファイルを登録するとき、特定のファイルだけを登録したい、あるいは特定のファイルを除外したいときがあります。そのようなときに、ファイルフィルタを使用します。「ファイルフィルタ...」ボタンを押すと、以下のように検索するファイル、検索から除外するファイル名の特徴を正規表現で指定できます。この例では、拡張子.bin, .hqx, .class, .jar, .qdをもつファイルを検索対象から除外するように指定してます。

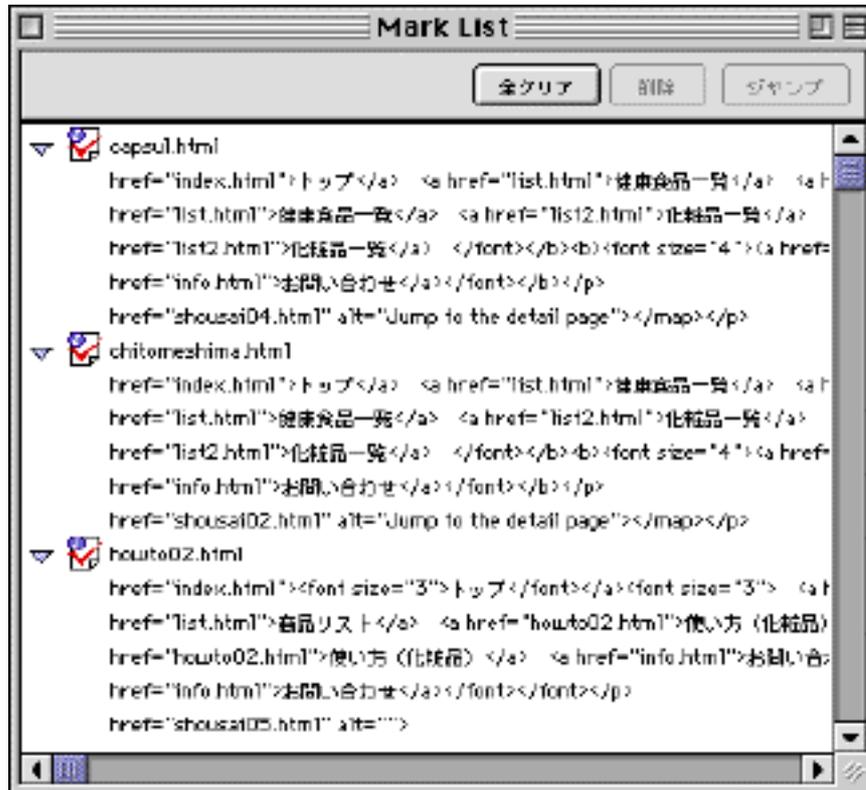


「検索」ボタンまたは「一括検索」ボタンを押すとマルチファイル検索が始まります。



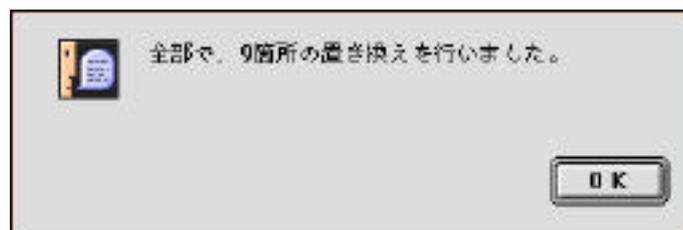
「検索」ボタンを押したときは、該当文字列が見つかるたびに編集ウィンドウが開き該当箇所が選択表示されます。さらに検索するには「検索」ボタンまたは検索メニュー「次を検索」を選択します。検索と同時に置換も行うときは「置換&検索」ボタンまたは検索メニュー「置換し次を検索」を選択します。

「一括検索」ボタンを押したときは、検索リストのすべてのファイルについて検索し該当箇所が、ファイルごとに図のようなマークリストに表示されます。



マークリストの希望の項目をダブルクリックすれば、その箇所が編集ウィンドウに表示されま
す。ファイル項目をダブルクリックするとそのファイルが開きます。

「一括置換」ボタンを押すと、検索リストのすべてのファイルについて一括置換を行います。
置換が発生したときは、そのファイルを編集ウィンドウに開き、最後に以下のような確認ダイ
アログが表示されます。



置換結果を実際のファイルへ反映させるには、それらの編集ウィンドウを閉じるときに保存を
選択してください。（直接ファイルの内容を置換するのではないことに注意してください。保
存しない限りファイルへは反映されません。）

初期設定「保存」で「自動保存」をオンにしておくと、一括置換で沢山開いた編集ウィンドウ
をファイルへ書き戻すときに便利です。（「自動保存」がオンのときは、ウィンドウを閉じ
るときに、いちいち保存するかどうかを聞かずに自動的に保存されます。）

検索ファイルリストの操作

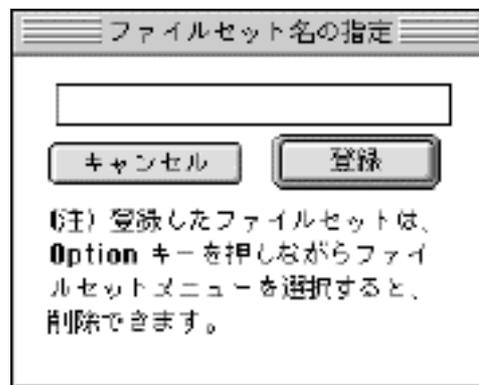
すでに登録した検索ファイルを検索リストから削除するには「削除」ボタンを、すべての検索ファイルをクリアするには「クリア」ボタンを押してください。



検索リストのファイル名部分をダブルクリックするとそのファイルが編集ウィンドウに開きます。

ファイル名左の「」マークはこれから検索するファイルであることを示しています。「」マークの部分をクリックすると「」マークが消え検索対象から除外されしながら「」マークの部分をクリックすると、そのファイルよりも前にある全てのファイルの「」マークを消すことができます。

対象リストを今後もよく使う場合はファイルセットとして登録しておくとう便利です。「登録」ボタンを押すと以下のダイアログが現れます。



ファイルセット名を入力し「登録」ボタンを押すとファイルセットとして登録されます。

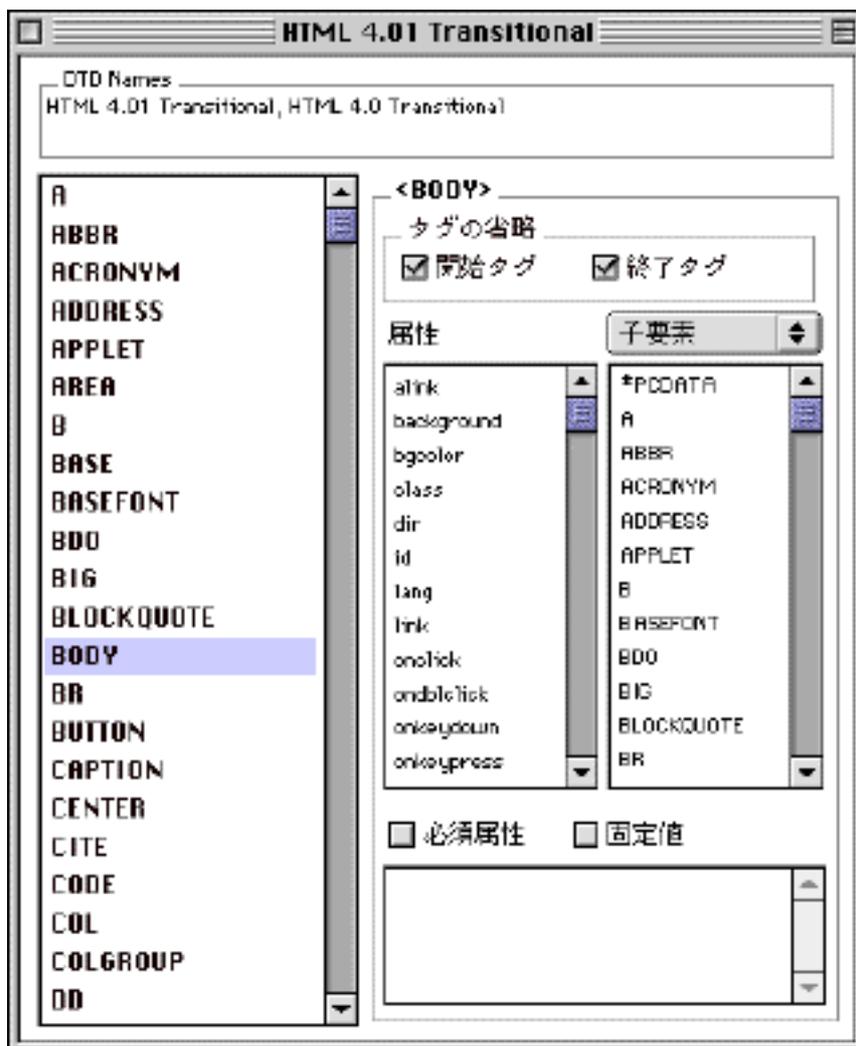
登録したファイルセットを呼び出すには、「ファイルセット」ボタンを押すとファイルセットがメニューとしてポップアップしますので希望のセットを選択してください。

登録したファイルセットをメニューから削除するにはOptionキーを押しながら「ファイルセット」ボタンを押しメニューから削除したいファイルセットを選択してください。

7.5 辞書の検索

あるエレメントについて辞書で調べたいときは、カーソルをそのタグのどこかにセットして、メニュー「検索」「辞書を検索」を実行すると、以下のような辞書ウィンドウが開きます。辞書ウィンドウについて詳しくは第9章を参照してください。

また、メニューではなく、ツールバーの  辞書ボタンを押しても、辞書の検索を行えます。

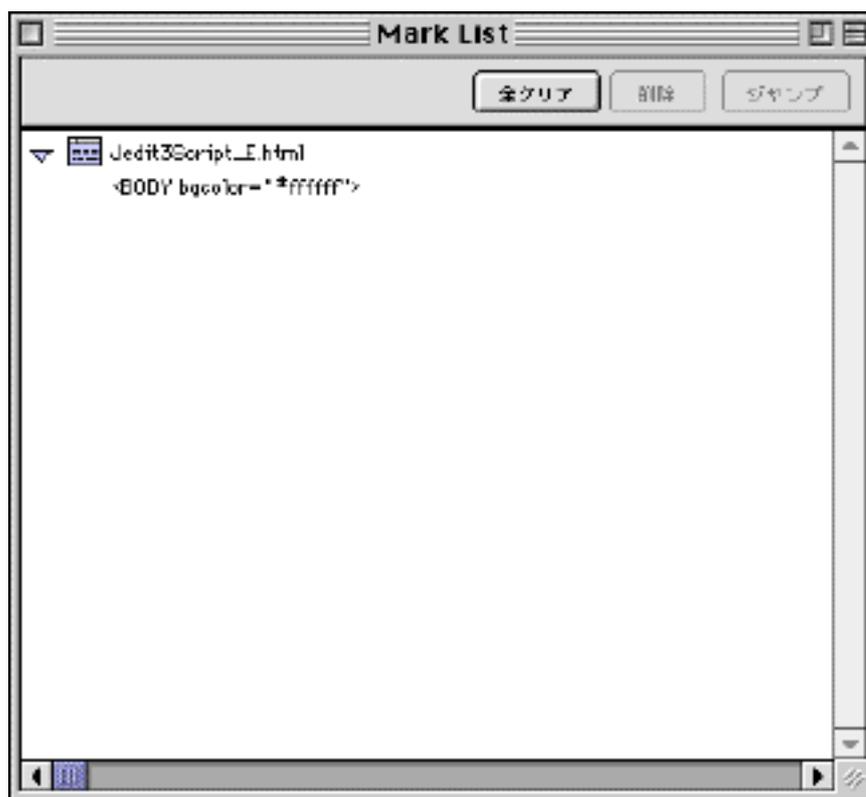


7.6 マーク機能

テキストファイルの任意の場所に印（マーク）をつけ、あとでワンタッチでジャンプできるようにする機能がマーク機能です。

マークしたい部分を選択し、マウスでドラッグし、マークリストのなかへドロップすればマークとして登録できます。

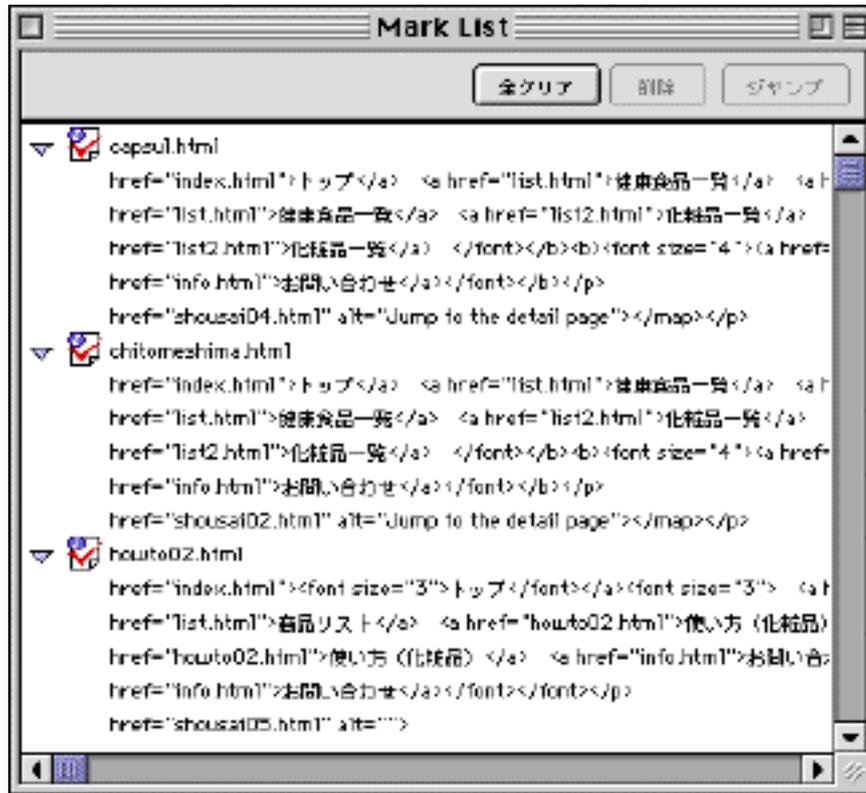
または、検索メニュー「マーク」を選択しても選択部分をマークリストに登録できます。



マークした箇所を開くには、マークリストの希望のマークをダブルクリックしてください。

7.7 マークリスト

検索メニュー「マークリスト」を選択すると以下のようなマークリストが開きます。



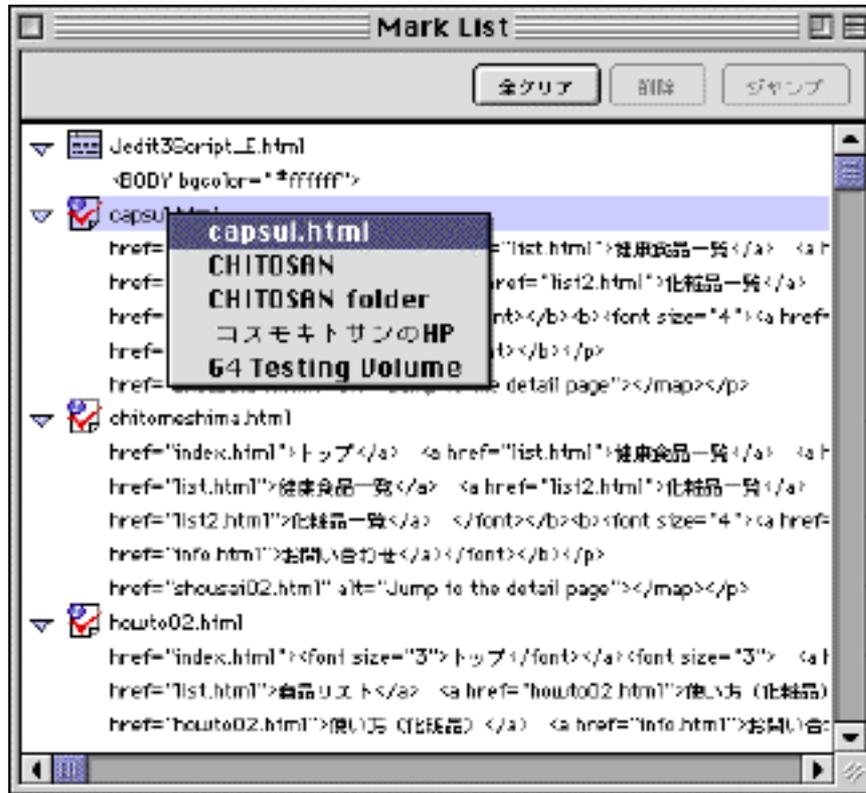
このマークリストには、「一括検索」で見つかった行や「マーク」した行がファイルごとに表示されます。

ファイルアイコン左の ▾ マークをクリックするとそのファイルの表示をオンオフできます。optionキーを押しながら ▾ マークをクリックすると全てのファイルの表示をオンオフできます。

ファイルパスの表示

マークリストにはファイル名しか表示されないなので、そのファイルがどこにあるかがわかりません。ファイルのパス（位置）を調べたいときは、そのマークリストのなかのファイルアイコンを cmdキーを押しながらクリックすると下図のように、そのファイルのディレクトリメニューがポップアップします。

メニューから、ディレクトリを選択すれば、そのウィンドウがFinder上に開きます。



ジャンプ

ファイル項目（ファイルアイコンのある行）をダブルクリックすると、そのファイルが対応するアプリケーションで開きます。行項目（アイコンのない行）をダブルクリックすると、Jedit4.0でそのファイルを開きその行が編集ウィンドウに選択表示されます。ダブルクリックするかわりに「ジャンプ」ボタンを押しても同じです。

ファイルが編集ウィンドウに開いているときは、ファイル項目のアイコンがウィンドウアイコン  になります。

項目の削除

「削除」ボタンを押すと選択されている行がリストから削除されます。ファイル項目が選択されているときは、そのファイルに属する全ての行が削除されます。「全クリア」ボタンを押すと全ての項目がクリアされます。

マークリストは自動的にクリアされることはなく、検索結果やマークを行うと新しい項目がマークリストに追加されます。

7.8 行番号ジャンプ

検索メニュー「指定の行番号へ...」を選択すると以下のようなダイアログが現れます。



ジャンプ先の行番号と、行番号かパラグラフ番号かを指定し「ジャンプ」ボタンを押してください。

第8章 チェック

8.1 文法チェック

メニュー「チェック」「文法チェック」を選択するか、 チェックボタンをクリック、編集ウィンドウのHTML文書の文法チェックを行います。エラーや警告を検出すると、それらをメッセージ領域に表示します。また、表示されたメッセージをマウスでクリックすると、対応するHTML文書の該当部分が編集領域に選択表示されます。ダブルクリックすると、そのエラーまたは警告の解説ウィンドウが表示されます。

文法チェックは「辞書」をもとに行われます。どの「辞書」を使うかは、ツールバーの辞書メニューで指定できます。また、辞書メニューで「DOCTYPEに準拠」を指定しておくと、HTMLのDOCTYPE宣言に対応する辞書が自動的に選択されます。

通常、辞書メニューでは「DOCTYPEに準拠」か「HTML 4.01 Transitional」を指定しておくが無難です。

チェックを行ったとき、場合によっては、エラーや警告が100件以上もでて驚くかもしれません。

まず、単純なエラーや警告については、メニュー「チェック」「自動修正」を実行してまとめて修正してください。JCheckerでは約20種類のエラーについて自動修正をサポートしていません。

対応する終了タグがなかったり、含むことができないサブエレメントを指定しているなどのHTMLの階層構造を壊すようなエラーは、その行以降の階層構造をすべて壊してしまいうので、1つのエラーがたくさんのエラーを誘発します。終了タグがあるのに「終了タグが見つからない」というエラーがでたりするのは、そのエレメントの中に階層構造を壊すエラーを含んでいるからです。もともになるエラーを修正すれば、まとめてたくさんのエラーを減らすことができます。

8.2 リンクチェック

メニュー「チェック」「リンクチェック」を選択するか、shiftキーを押しながら チェックボタンをクリックすると、編集ウィンドウのHTML文書内のリンクについて、そのリンク先が実際に存在するかどうかをチェックします。エラーや警告を検出すると、それらをメッセージウィンドウに表示します。初期設定「リンク」でローカルリンクとリモートリンクのそれぞれについてチェックするかどうかを指定できます。

リンクチェックは、辞書で属性値のタイプに「CDATAURI」が指定されている属性値について行います。具体的には

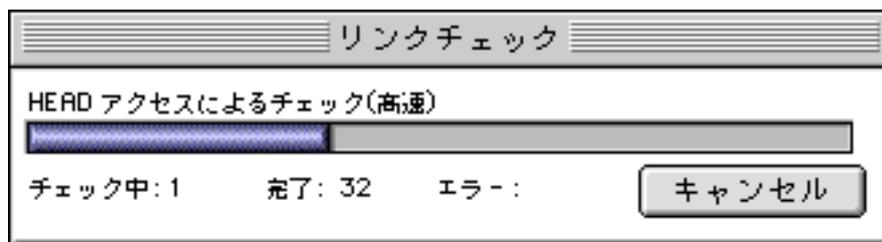
"A", "BODY", "LAYER", "BGSOUND", "EMBED", "MARQUEE", "OBJECT",
"ILAYER", "FORM", "DEL", "INS", "INPUT", "IMG", "APPLET", "AREA", "LINK",
"BASE"

の要素にあらわれる属性

"background", "background-image", "cite", "data", "classid",
"longdesc", "pluginspage", "profile", "action", "src", "usemap", "codebase",
"href"

のなどの値(URI)です。

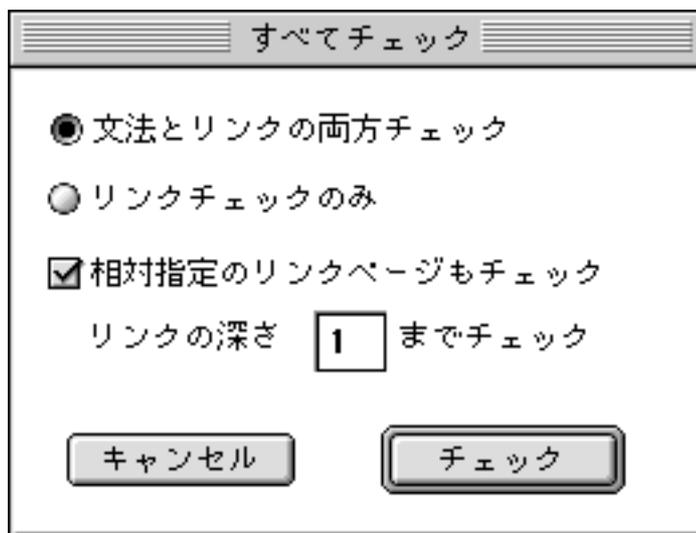
メニューを実行すると以下のようなプログレスバーが表示され進行状況がわかります。



8.3 すべてチェック

メニュー「チェック」「すべてチェック...」では、リンクチェックをしながら、さらにそのリンク先の HTML書類のチェックも行います。すなわち、トップページで「すべてチェック」を実行すれば、そこから相対リンクしている下流のページをまとめてチェックできます。

メニュー「チェック」「すべてチェック...」を選択すると以下のようなダイアログが現れます。



ラジオボタン「文法とリンクチェック」を指定すると、リンクチェックと文法チェックの両方を行います。

「相対指定のリンクページもチェック」をオンにして、リンクの深さを1以上にすると、リンク先のページもチェックします。深さ1のときは、最初のページからリンクしているページ（子ページ）までをチェックします。深さ2のときは、子ページからリンクしている孫ページまでチェックします。

リンクの深さを大きくすると、チェックするページ数が幾何級数的に増大してしまうので、必要以上には大きくしないでください。

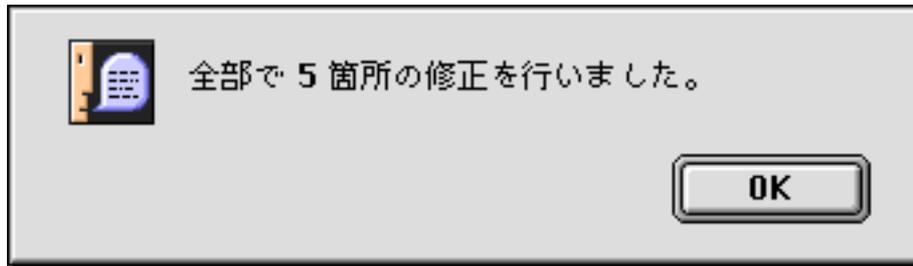
ボタン「チェック」を押すと、チェックがはじまり、以下のような経過ダイアログで調べたページ数が表示されます。



チェックを中止したいときは、同時に表示されているプログレスバーのキャンセルボタンを押してください。

8.4 自動修正

メニュー「チェック」「自動修正」を実行すると、単純な20種類のエラーについてJCheckerが自動修正します。自動修正が終わると以下のような修正した数が表示されます。



自動修正する項目は以下の通りです。

- タグにalt属性を自動付加する。
- <AREA>タグにalt属性を自動付加する。
- <SCRIPT>タグの属性languageをtypeに置きかえる。
- <STYLE>タグに属性type="text/css"を自動付加。
属性値を"マークで囲む。
- URL中の などの「安全でない文字」を16進表記に置きかえる。
- NULLキャラクターを削除する。
- データ中の&マークを&に置きかえる。
- XML宣言を追加する。
- DOCTYPE宣言を追加する。
- カラー属性値の16進表記で#がないものに#を挿入する。
- 属性値で、終わりの"マークが抜けているものを修正する。
- <TITLE>エレメントを追加する。
- キャラクタセット指定x-sjis, x-euc-jpをそれぞれShift_JIS, EUC-JPに置きかえる。
- 属性値が固定(#FIXED)のとき、正しい固定値に修正する。
- XMLで属性値がないものに属性名と同じ属性値をセットする。
- 属性名と属性値が同じでなければならない属性の誤りを修正する。
- <HEAD>とそのなかに<TITLE>エレメントを追加する。
- HTMLエディタが編集のために独自に挿入した属性(NATURALSIZEFLAG, coolなど)を取り除く。
- DOCTYPE宣言のなかの大文字小文字の間違いを修復する。

メニュー「チェック」「自動修正の設定...」を選択すると、初期設定「自動修正」で、上記の各項目について自動修正するかどうかを個別に指定できます。



8.5 HTMLの整形

メニュー「チェック」「HTML整形」を実行すると、エレメントの階層構造などを一目で分かりやすくなるようにHTML文書をインデントをつけて整形します。

チェックしたエラーメッセージで原因がよくわからないようなときは、この「HTML整形」で、文書を階層構造で表記してみるとエラーの原因をみつけやすくなります。

また、整形するときに階層構造を解釈する上でエラーがあると「正しく整形できないことがある」旨の警告が表示されます。

整形には、初期設定の「整形」でいろいろなオプションをつけることができます。詳しくは、「3.3 整形の設定」を参照してください。

8.6 HTMLの最小化

HTMLでは、連続したホワイトスペース（半角スペース、タブコード、改行コード）は1つの

半角スペースに置き換わります。したがって、ソースリスト上にあるたくさんのスペース、タブコード、改行コードはHTMLの表示では意味を持ちません。そこで、そのような無駄なホワイトスペースをソースリストから取り除いてしまうのが最小化です。

メニュー「チェック」「最小化」を実行すると、以下のように最小化したときの効果が表示されます。最小化するときは「はい」ボタンを押します。



最小化するとおおむね10%程度の縮小が可能です。そのぶん、インターネット上の転送時間が短くなり、またサーバー上のスペースも節約できます。

ただし、最小化を行うとソースリストの視認性が著しく低下しますので、ソースを変更するときは「HTML整形」メニューで整形してから編集してください。

なお、PRE、SCRIPT、XMP、LISTING、PLAINTEXT、STYLE、SERVERなどのデータタイプがCDATAの要素については最小化は行いません。

8.7 再カラーリング

JCheckerのエディタでは、タグの種類などに応じて表示色やスタイルを変えて表示（カラーリング）できるようになっていますが、なにかの具合でカラーリングが乱れてしまったようなとき、このメニュー「チェック」「再カラーリング」を選択してください。再カラーリングを行います。

第9章 他のエディタと連動

JCheckerでは、ワードサービスとサービスメニューをサポートし、これらの機能を使って他のエディタで編集中のHTML文書の文法チェックも行うことができます。

9.1 ワードサービスでチェック

Jeditなどのワードサービス対応のエディタで編集中のHTML書類を、JCheckerで文法チェックできます。Jeditの場合は、Classic OSでもOS Xでも、ワードサービス経由でJCheckerを呼び出すことができますが、その他アプリケーションでは、ワードサービスはClassic OSのものでないと動作しません。

JChecker3.0と連動するワードサービス対応アプリケーションとしては、Jeditの他にBBEdit、NisusWriter、WordPerfect、Eudora、ClarisWorks など様々なものがあります。ワードサービスに対応したアプリケーション、テキスト処理プログラムについては、<http://www.wordservices.org/> から情報を入手できます。

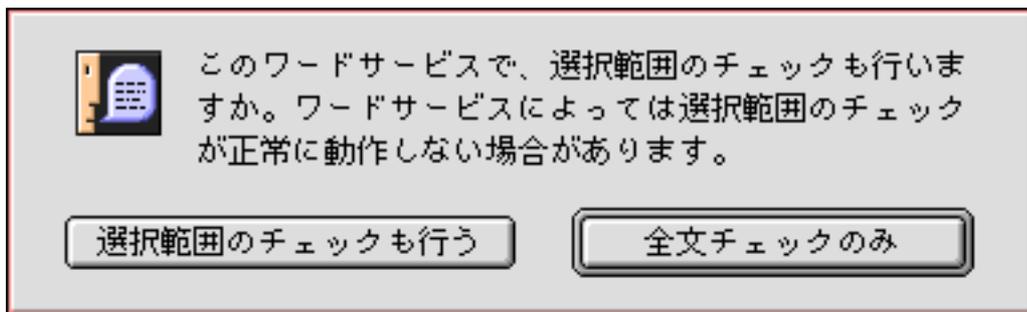
Jeditとの連動を例に説明します。

JCheckerをJedit4.0のメニューに登録する

Jedit4.0の「ツール」メニューの「ワードサービス」「チェッカの追加」を選択します。



すると上の図のように、JChecker3.0の場所を聞いてきますので、プログラムファイルJChecker3を指定してください。



すると上の図のように、選択範囲のチェックを行うかどうかを聞いてきます。JCheckerの場合は「全文チェックのみ」を選択してください。

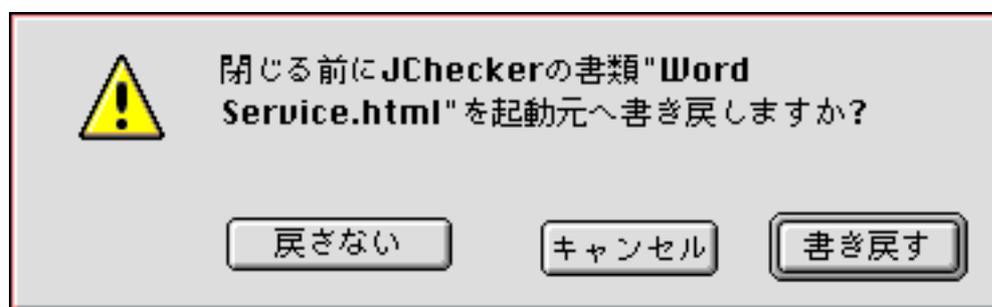
以上で、ワードサービスのサブメニューにJCheckerのメニュー「JChecker3.0」が追加されます。



JeditからJCheckerを起動する

Jeditのメニュー「ツール」「ワードサービス」の「Check HTML」を選択すると、JChecker3.0が自動的に起動し、Jeditで編集集中のHTML書類の文法チェックが行えます。

そして、JChecker3.0でエラーや警告を取り除きチェックウィンドウを閉じるときに、



と聞いてきます。「書き戻す」ボタンを押すとJChecker3.0内で修正した内容が、Jeditの内の編集ウィンドウへ書き戻されます。「戻さない」のショートカットは cmd+D キー です。

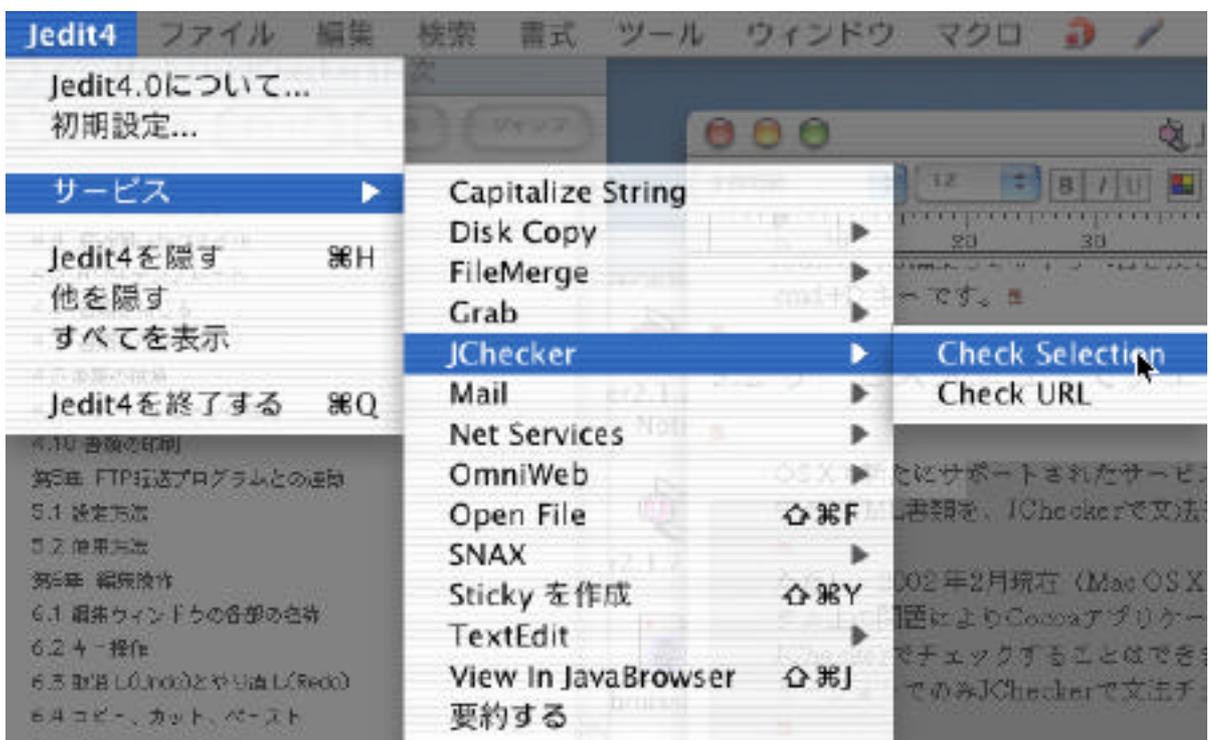
9.2 サービスメニューでチェック

OS Xで新たにサポートされたサービスメニューを使うと、Jeditなどのエディタで編集中のHTML書類を、JCheckerで文法チェックできます。

ただし、2002年2月現在（Mac OS X 10.1.2のもとでは）、サービスメニューのシステム上の問題によりCocoaアプリケーション（TextEditなど）のHTML書類をJCheckerでチェックすることはできません。サービスメニュー対応のCarbonアプリケーションについてJCheckerで文法チェックできます。

JeditからJCheckerを起動する

まず、Jeditで編集中のHTML書類をメニュー「編集」「全て選択」で全選択します。そして、メニュー「Jedit4」「サービス」を開くと以下のようにJCheckerを呼び出せます。



メニュー「Check Selection」を選択すると、JCheckerが開きJeditの選択領域の文法チェックを行います。また、「Check URL」を選択すると、Jeditで選択されているURLをJCheckerで開きチェックを行います。

なお、サービスメニューでは、JCheckerで編集した結果を起動元に書き戻すことはサポートしていません。

第10章 辞書

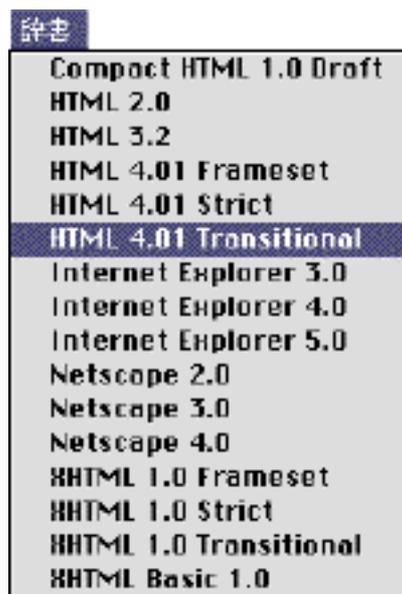
HTMLには、HTML2.0, HTML3.2, HTML4.01, XHTML1.0など多くのバージョンがあり、バージョンによってエレメントや属性の仕様が異なります。JCheckerでは、それらのバージョンごとにその規則をおさめた辞書を用意し、バージョンに応じた文法チェックができるようになっています。また辞書の追加やカスタマイズも行えます。

辞書は、JChecker3.0フォルダ内の「Dictionaries」フォルダのなかに格納されています。

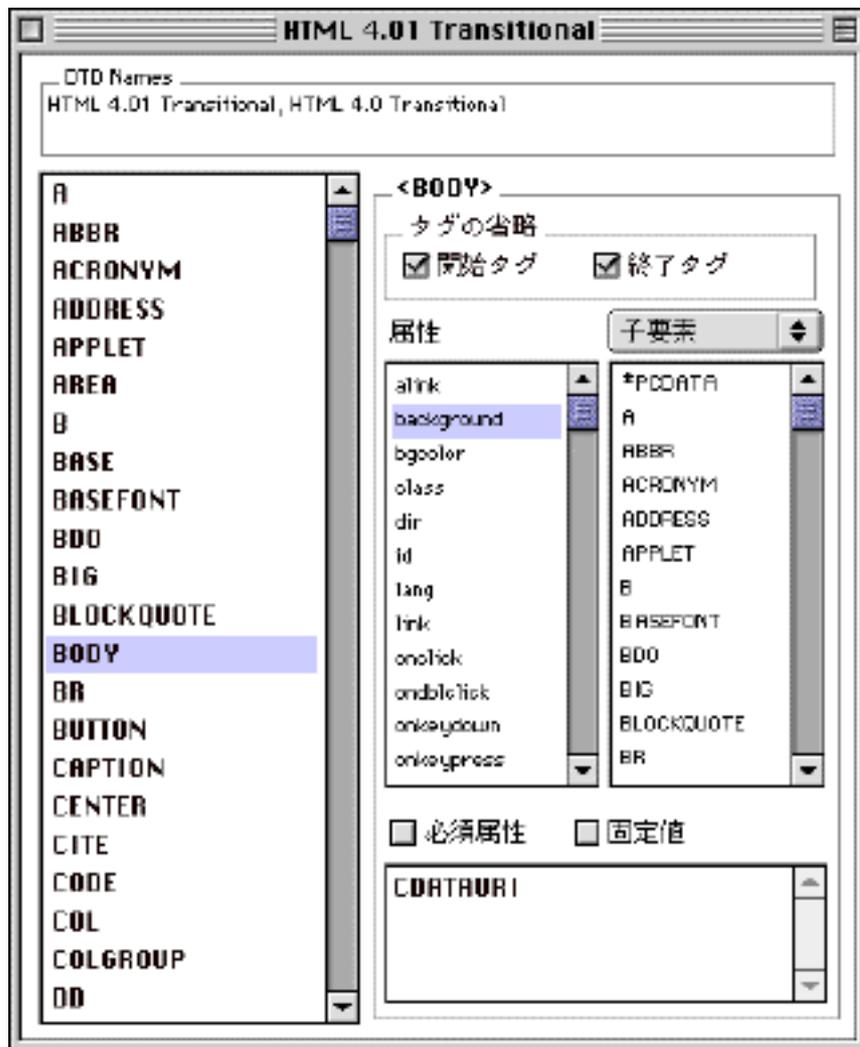


10.1 辞書ウィンドウ

メニュー「辞書」を開くと、以下のように「Dictionaries」フォルダに格納されている辞書がアルファベット順に表示されます。



希望の辞書を選択しメニューを実行すると、その内容が辞書ウィンドウに表示されます。



ウィンドウ上部の「DTD Names」には、DOCTYPE宣言でのこの辞書のDTD名が表示されます。上記の例では、HTMLのDOCTYPE宣言で「HTML 4.01 Transitional」か「HTML 4.0 Transitional」が指定されたときに、この辞書でチェックします。

ウィンドウ左には、使用できるエレメントがアルファベット順にリストされています。リストから希望のエレメントを選択すると、そのエレメントに関する情報が右側に表示されます。

「タグの省略」では、このエレメントの開始タグ、終了タグが省略可能かどうかチェックボックスで表示されます。（なお、辞書ウィンドウは参照専用なので、クリックしても設定をかえることはできません。）

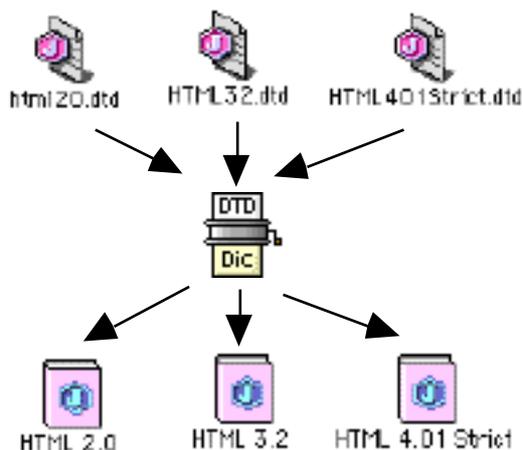
「属性」には、このエレメントで指定できる属性がリストされています。リストのどこかをクリックすると、その属性が必須属性かどうか、値が固定値かどうか、そしてその属性値のタイプが下に表示されます。

ウィンドウ右のメニューが「子要素」になっているときは、このエレメントで含むことができるサブエレメントが下にリストされます。リストのなかのエレメントをクリックすると、そのエレメントについての表示に切り替わります。

ウィンドウ右のメニューが「親要素」になっているときは、このエレメントを含むことができる親エレメントが下にリストされます。

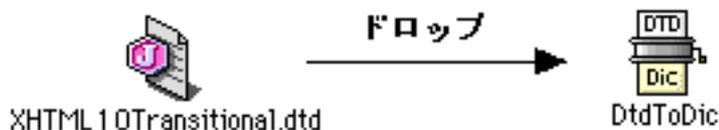
10.2 辞書の生成 (DtdToDic)

JCheckerの辞書は、公開されているDTDから付属プログラム「DtdToDic」で自動生成することができます。



JChecker3.0のフォルダDTDには、辞書のもとになったDTDファイルが格納されています。またファイル「DTDの入手先」には、それらのDTDが公開されてるURLが記されてます。

辞書を生成するには、DTDファイルをDtdToDicにプログラムアイコンにドロップしてください。



すると以下のように、DOCTYPE宣言で指定されるDTD名を聞いてきます。DTD名は複数指定可能です。



「Continue」ボタンを押すと辞書の生成が始まり、DTDファイルと同じフォルダのなかに以下のような辞書ファイルが生成されます。



生成された辞書を「Dictionaries」フォルダに移動し、JCheckerを再起動すれば、辞書メニューにその辞書名があらわれます。

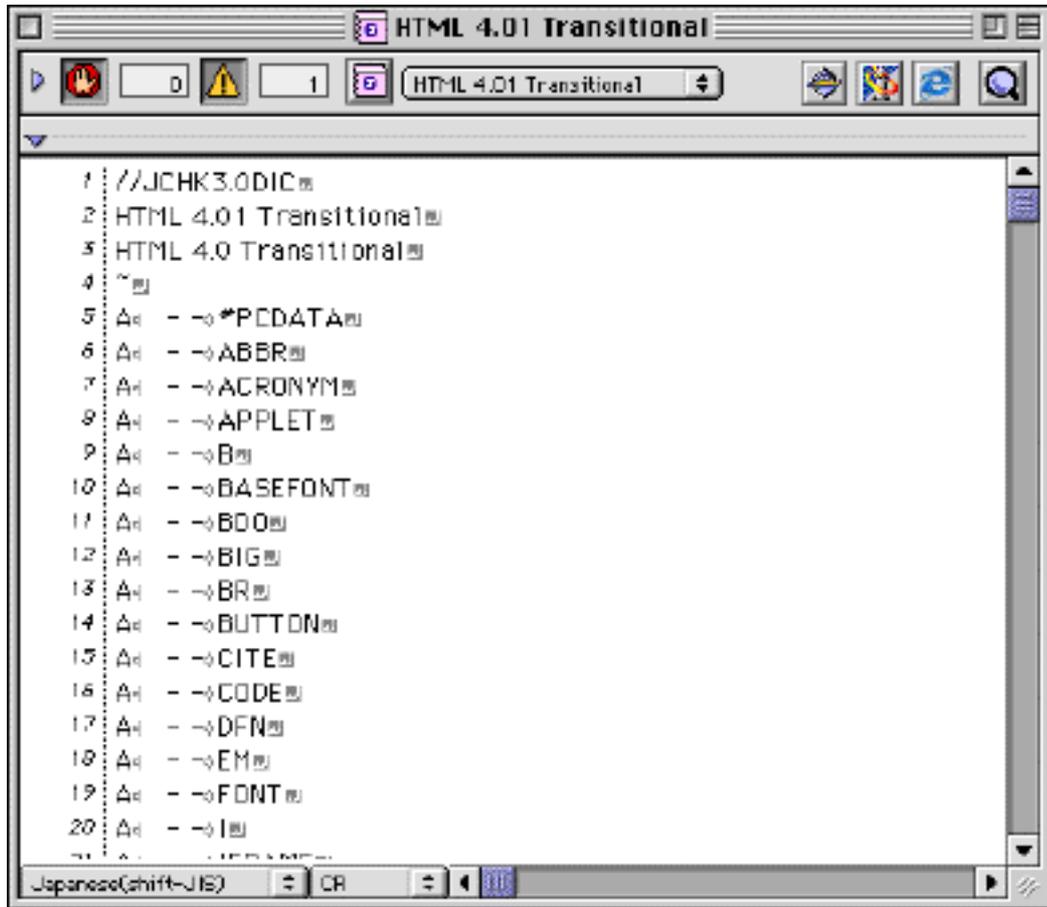
DtdToDic 留意点

- ・ DTDそのものに誤りがあるようなときは、正しく辞書を生成できない場合があります。
- ・ DtdToDicはカーボンアプリケーションです。

10.3 辞書ファイルの構造

辞書ファイルは、テキストファイルなのでエディタやJCheckerでその内容を直接見ることができ、また構造も簡単なので、ユーザーが直接書き換えることも可能です。

辞書ファイルをダブルクリックすると、JCheckerの編集ウィンドウにその内容が表示されます。



辞書は、(1) DTD名を宣言している部分、(2) エLEMENTについて記述している部分、(3) 属性について記述している部分の3つから構成され、それぞれの部分は、「~」だけの行で区切られています。

- 第1行目「//JCHK3.0DIC」は、JChecker3.0の辞書であることを宣言。
- (1) 第2行目から「~」の行まで、この辞書のDTD名。(複数指定可)
- (2) 「~」の行から次の「~」の行まで、ELEMENTに関する記述
- (3) 次の「~」の行から最後まで、属性に関する記述

ELEMENTに関する記述では、以下のようなフォーマットでAELEMENTからアルファベット順に記述されます。1つのELEMENTは、そのELEMENTが含むことができる子ELEMENTを1行として複数の行から構成されます。

ELEMENT名、タブコード、タグ省略情報、タブコード、子ELEMENT名

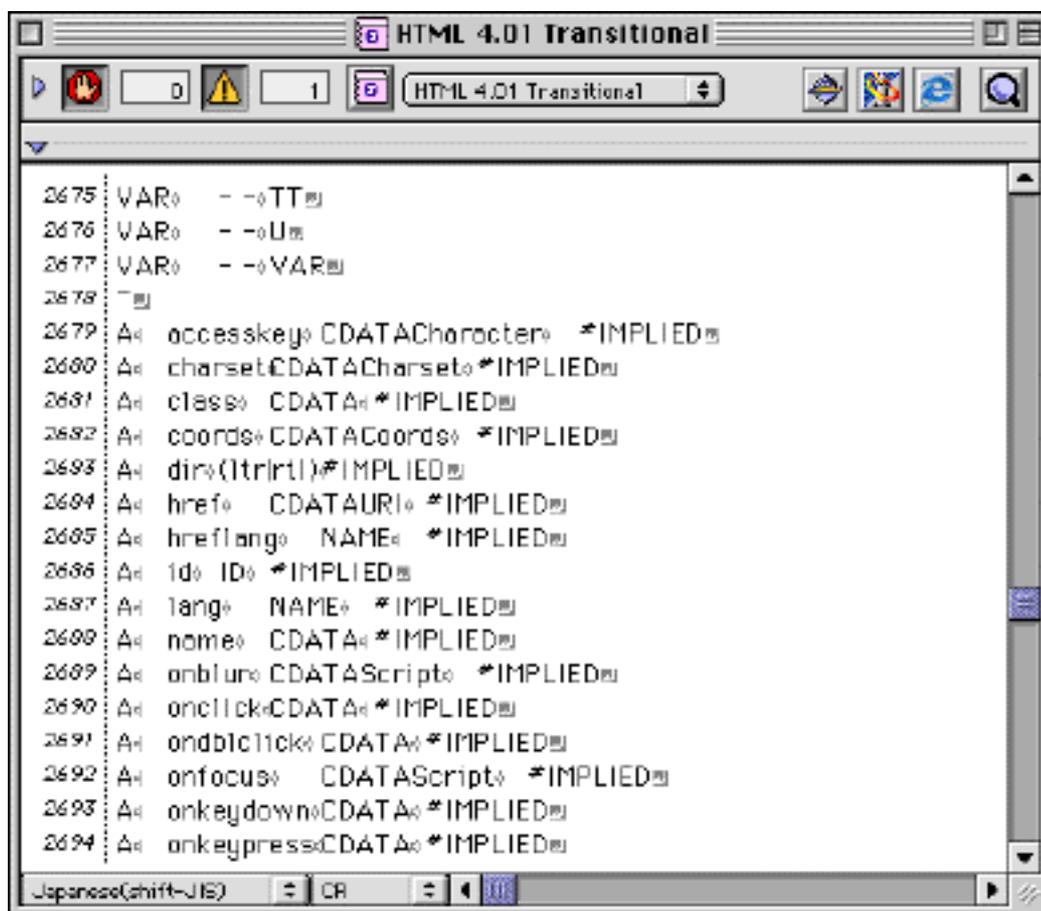
タグ省略情報とは、3バイトの文字でタグが省略可能かどうかを示します。

- 開始タグも終了タグも省略不可
- O 終了タグだけ省略可
- OO 開始タグも終了タグも省略可

なお、文字Oは省略を意味する「omit」の頭文字です。

タグ省略情報は同じエレメント名のときはすべて同じでなければなりません。

以下の図では、2677行目までがエレメントに関する記述で、2678行の「~」を境に、2679行目から属性に関する記述がはじまっています。



属性に関する記述では、以下のようなフォーマットでAエレメントからアルファベット順に記述されます。1つのエレメントは、そのエレメントで指定できる属性を1行として複数の行から構成されます。

エレメント名、タブコード、属性名、タブコード、タイプ、タブコード、状態

タイプには、CDATA、NAME、ID、NUMBERなど属性値のタイプを示す文字列、あるいは、とりうる値が指定されます。

状態には、属性値のデフォルト値、あるいは以下の#付きキーワードが指定されます。

- #IMPLIED 親エレメントから属性値が継承され、デフォルトの値がユーザーエージェントからあたえられる
- #REQUIRED そのエレメントに必ず必要な属性。必須属性。
- #FIXED 属性値が指定のもので固定されている。

例えば

CAPTION align (top|bottom|left|right) #IMPLIED

ではエレメントCAPTIONの属性alignには、top、bottom、left、rightのどれかを指定でき、省略されたときは親エレメントまたはユーザーエージェントのデフォルト値が採用されます。

IMG alt CDATA Text #REQUIRED

ではエレメントIMGの属性altの属性値はテキストで、必ず指定しなければならない属性であること示しています。

IFRAME scrolling (yes|no|auto) auto

ではエレメントIFRAMEの属性scrollingには、yes、no、autoのどれかを指定でき、属性が省略されたときは、autoが指定されたものとして動作します。

10.4 辞書のカスタマイズ

辞書をカスタマイズするには以下の2つ方法があります。

- 1) 辞書ファイルをエディタで直接修正する
- 2) DTDファイルをエディタで修正し、DtdToDicプログラムで辞書を生成し直す。

ここでは、HTML4.01 Transitionalの辞書のbodyエレメントに、属性leftmarginとtopmarginを追加する場合を例に説明します。

辞書ファイルを直接修正する場合

辞書のbodyエレメントの属性に関する部分を修正します。

HTML4.01 Transitionalの辞書のbodyエレメントの属性に関する部分は、2859行目から始まり以下のようになっています。

```
BODY   alink   CDATA   #IMPLIED
BODY   background CDATAURI #IMPLIED
BODY   bgcolor CDATA   #IMPLIED
.....
.....
BODY   text   CDATA   #IMPLIED
BODY   title  CDATA   #IMPLIED
BODY   vlink  CDATA   #IMPLIED
```

leftmarginとtopmarginの属性を追加すると以下のようになります。

```
BODY    alink    CDATA    #IMPLIED
BODY    background CDATAURI #IMPLIED
BODY    bgcolor  CDATA    #IMPLIED
.....
BODY    lang NAME    #IMPLIED
BODY    leftmargin CDATA #IMPLIED
BODY    link CDATA  #IMPLIED
.....
BODY    title    CDATA    #IMPLIED
BODY    topmargin CDATA #IMPLIED
BODY    vlink    CDATA    #IMPLIED
```

上記2行を追加して辞書を保存しなおせば修正が完了です。

DTDから修正する場合

DTDから修正するには、DTDの読み方を知らないとは修正できませんが、詳しくは以下のような市販の解説書を参照してください。

岡部友計著
「図解最新テクノロジーXHTML」株式会社ナツメ社2001年

ビレッジセンターHTML&SGML研究チーム著
「正しいHTML4.0リファレンス&作法」ビレッジセンター出版局 1998年

HTML4.01 TransitionalのDTDファイル「HTML401Transitional.dtd」をエディタで開くと、その313行目に以下のようなbodyの属性定義部分があります。

```
<!ATTLIST BODY
  %attrs;                                -- %coreattrs, %i18n, %events --
  onload          %Script;    #IMPLIED -- the document has been loaded --
  onunload        %Script;    #IMPLIED -- the document has been removed --
  background      %URI;       #IMPLIED -- texture tile for document
                                     background --
  %bodycolors;    -- bgcolor, text, link, vlink, alink --
>
```

leftmarginとtopmarginの属性を追加すると以下のようになります。

```
<!ATTLIST BODY
  %attrs;                                -- %coreattrs, %i18n, %events --
  onload          %Script;    #IMPLIED -- the document has been loaded --
  onunload        %Script;    #IMPLIED -- the document has been removed --
```

```
background      %URI;      #IMPLIED -- texture tile for document
                background --
%bodycolors;    -- bgcolor, text, link, vlink, alink --
leftmargin      CDATA      #IMPLIED
topmargin       CDATA      #IMPLIED
>
```

上記2行を追加し保存して、そのDTDファイルをDtdToDicプログラムで辞書を生成し、Dictionariesフォルダの古い辞書と入れ替えてください。

第11章 ブラウザメニュー

ブラウザメニューは、現在編集集中のHTMLが、実際のブラウザでどのような見え方をするか確認するためのメニューです。

11.1 ブラウザの登録

出荷時には、ブラウザはメニューに登録されていないので、はじめにユーザーがブラウザを登録する必要があります。ブラウザメニューには、いくつでもブラウザを登録することができます。

メニュー「ブラウザ」「ブラウザの設定…」を選択すると、以下のように初期設定「ブラウザ」パネルが開きます。実際の登録方法は、第3章 初期設定「3.7 ブラウザの設定」を参照してください。



11.2 ブラウザでプレビューする

メニュー「ブラウザ」を開くと、以下のように登録されたブラウザが表示されます。

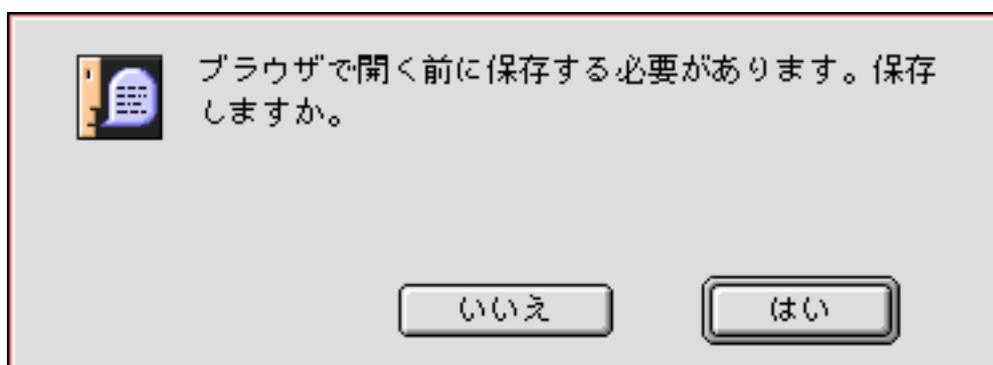


この例では、ブラウザに Internet Explorer、Netscape Communicator、iCab の3種類を登録してあります。希望のブラウザを選択すれば、そのブラウザが起動し、編集集中のHTMLをプレビューできます。

また、メニューからでなく、ツールバーのブラウザプレビューボタンをマウスでクリックしてもブラウザを起動できます。



ブラウザへはファイルにしてデータを渡しますので、まだ一度も保存していないHTMLをプレビューしようとする時、



と聞いてきますので、ボタン「はい」を押してファイルとして保存してください。

プレビューでは、オリジナルファイルが書き変わってしまわないように、オリジナルファイル

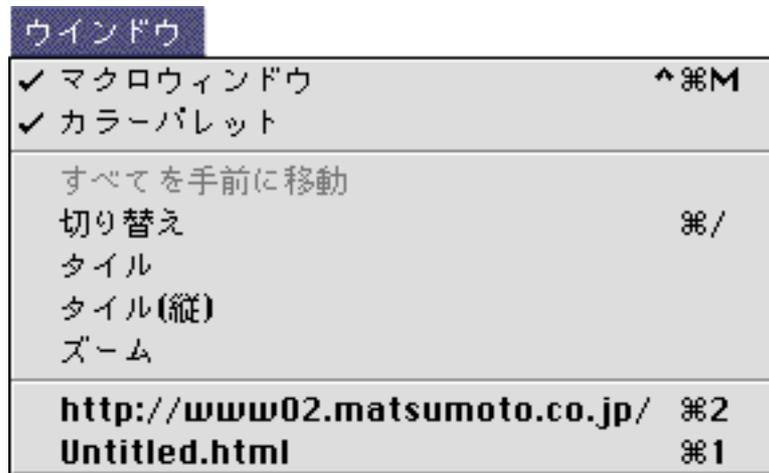
のファイル名に～マークをつけた一時ファイルを使用します。



たとえば、「index.html」というオリジナルファイルのときは、「index.html」という名前の一時ファイルが生成されます。この一時ファイルは、編集ウインドウを閉じたときに自動的に削除されます。

第12章 ウィンドウメニュー

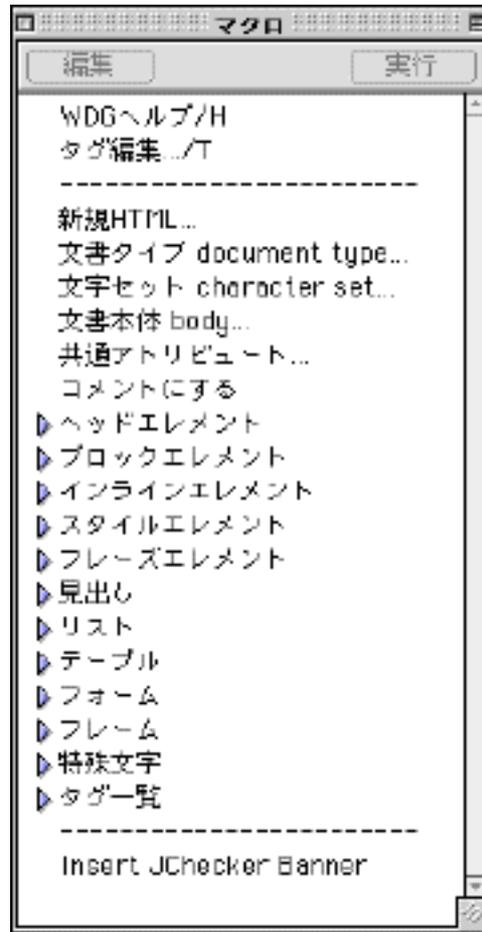
JCheckerで開いているウィンドウを操作するためのメニューです。



ウィンドウメニューの後半（仕切り線以降）には、現在JCheckerで開いているウィンドウがメニューに表示され、選択すると対応するウィンドウが最前面に表示されます。

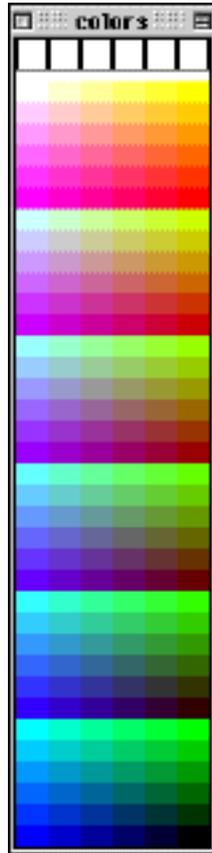
12.1 マクロウィンドウ

メニュー「ウィンドウ」「マクロウィンドウ」を選択すると、以下のようにマクロウィンドウが表示されます。マクロウィンドウの操作について、第13章マクロ「13.2 マクロウィンドウ」を参照してください。



12.2 カラーパレット

メニュー「ウィンドウ」「カラーパレット」を選択すると、以下のようにカラーパレットウィンドウが表示されます。



カラーパレットウィンドウにWeb Safeな216色が表示されます。希望の色をマウスでクリックすると、その色の16進表現が編集ウィンドウのカーソル位置へ挿入されます。

カラーパレット上部には、クリックした色の履歴が6色まで表示されます。

12.3 すべてを手前に移動

JCheckerで表示しているすべてのウィンドウを手前に表示します。

このメニューはOS Xでだけ選択できます。Classic OSのときは、JCheckerをアクティブにすれば、すべてウィンドウは自動的に手前に移動します。

12.4 切り替え

メニュー「切り替え」を選択すると、現在JCheckerで開いているウィンドウで一番奥に隠れているウィンドウが最前面に表示されます。

12.5 タイリング

「タイル」を選択すると、JCheckerで開いている全ての編集ウィンドウが上下にわかれてスクリーン全体に表示されます。

「タイル（縦）」を選択すると、JCheckerで開いている全ての編集ウィンドウが左右にわかれてスクリーン全体に表示されます。

12.6 ズーム

最前面の編集ウィンドウをズームします。編集ウィンドウのズームボタンを押したのと同じです。

第13章 マクロ

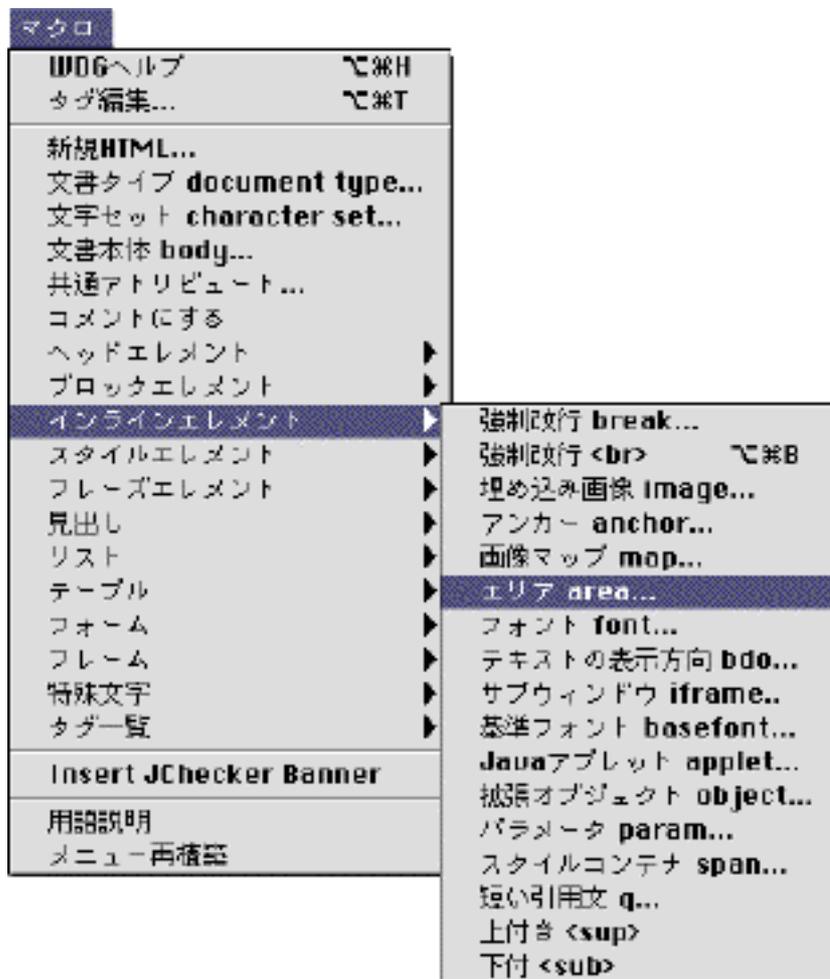
JCheckerはアップルスクリプトに対応しているので、タグ編集などよく使う操作やめんどろな操作をアップルスクリプトで記述し、1つの命令すなわち「マクロ」としてメニューに登録しておくことができます。

JCheckerのマクロ/アップルスクリプト機能は、当社のエディタJedit4.0とほぼ同等の仕様と能力を備えています。

なお、アップルスクリプトそのものについては付録「JCheckerのスクリプトマニュアル」を参照してください。

13.1 マクロメニュー

ユーザーが作成したアップルスクリプトをマクロとして登録すると、マクロメニューまたはマクロウィンドウからワンタッチでスクリプトを実行できます。また、マクロは階層構造にできるので、たくさんのスクリプトを分類・登録できます。



JCheckerのマクロメニューには、すでにHTML4.01の仕様を網羅したタグ編集マクロ一式が登録されていますので、とりあえずはユーザーが作らなくても間に合ってしまうかもしれませんが、オリジナルのマクロをつくる時は以下の手順で行います。（付属マクロについては「付録1、付属マクロ」を参照してください。）

まずスクリプト編集プログラムで希望の動作をアップルスクリプトで記述しコンパイル済みスクリプトとして保存します。



そしてそのファイルを、JChecker3.0フォルダの「Macro Menu Items」フォルダのなかにコピーします。

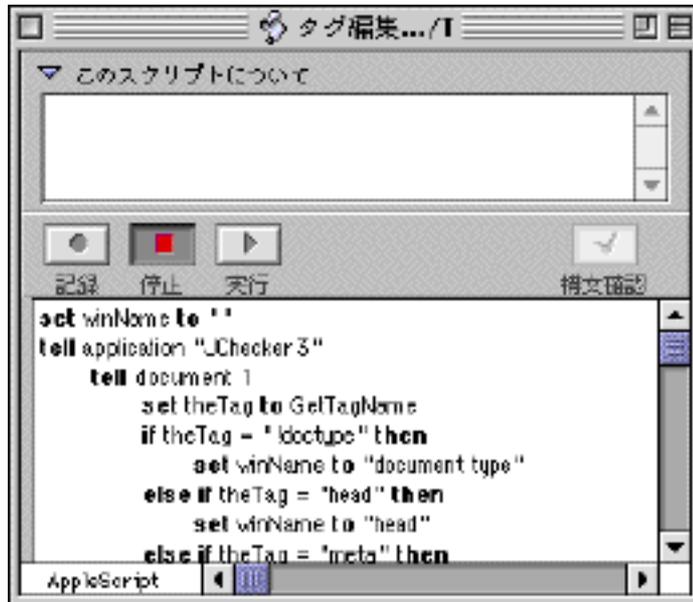


これで、JChecker3.0を起動するとそのスクリプトがマクロメニューとマクロウィンドウに現れます。

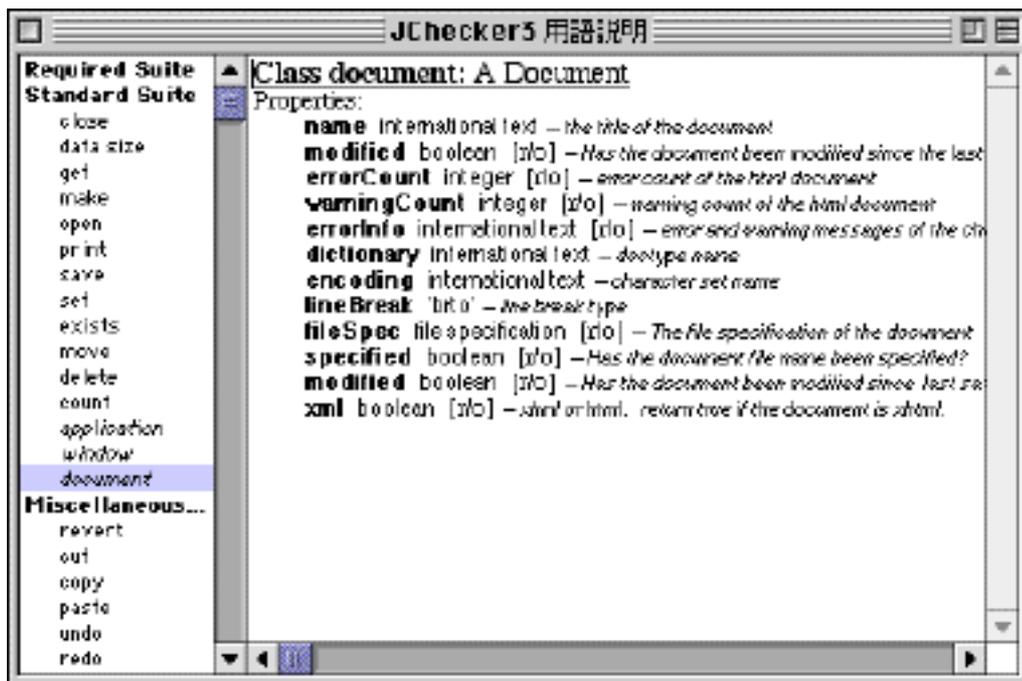
マクロメニューを階層化するには、Macro Menu Itemsフォルダのなかに、さらに別のフォルダを作成しそこにスクリプトファイルを格納すると、そのスクリプトが階層メニューとしてあらわれます。（ただし、階層の深さは2階層までです。）



メニューに登録したスクリプトのプログラムを参照・修正するには、コントロールキーを押しながらそのスクリプトのメニューを選択すると、自動的にスクリプト編集プログラムが起動しスクリプトを編集できます。



マクロメニューの「用語説明」を選択すると、JChecker3.0でサポートしているアップルスクリプトの用語説明ウィンドウが開きます。スクリプトを作成するときに活用できます。



マクロメニューの再構築

「Macro Menu Items」フォルダのなかのマクロファイルを削除したり新しいマクロファイルを追加したときは、そのままではマクロメニューに反映されません。JChecker3.0を再起動するか、マクロメニュー「メニュー再構築」を選択するとマクロメニューが再構築されます。

マクロのショートカットキー

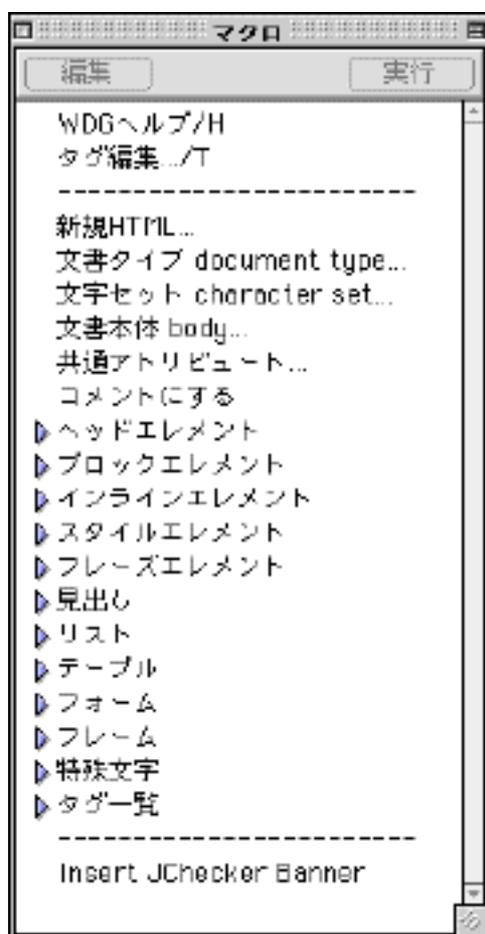
マクロにショートカットキーを割り当てるには、マクロファイル名の末尾に「/」とショートカット文字コード(半角大文字のアルファベット)を付加してください。たとえばファイル名を「タグ編集.../T」とすると、ショートカットoption+cmd+Tでそのマクロを実行できます。

マクロメニューの仕切り線

マクロメニューに分類のための仕切り線を挿入したいときは、ファイル名が「-」で始まるダミーのスク립トファイル(中身は空でもOK)をマクロメニューに登録すると、マクロメニューに仕切り線となって表示されます。この仕切り線の位置を変更するには、後述のマクロウィンドウから行ってください。

13.2 マクロウィンドウ

メニュー「ウィンドウ」「マクロウィンドウ」選択するとマクロウィンドウが開きます。マクロウィンドウはフローティングウィンドウになっているので、常に最前面にあります。



マクロウィンドウからも、マクロの「実行」「編集」を行えます。

階層メニューの項目には三角アイコンが付いていて、マウスでクリックするとその階層が開いたり閉じたりします。

optionキーを押しながら三角アイコンをクリックすると、下位の階層をまとめて開いたり閉じたりできます。

マクロの表示順序の変更

マクロウィンドウで、表示順序をかえたい項目をまず選択状態にして、それを希望の位置ヘドラッグすると表示順序を変更できます。この変更はマクロメニューにも直ちに反映されます。

スクリプトファイルのドロップ

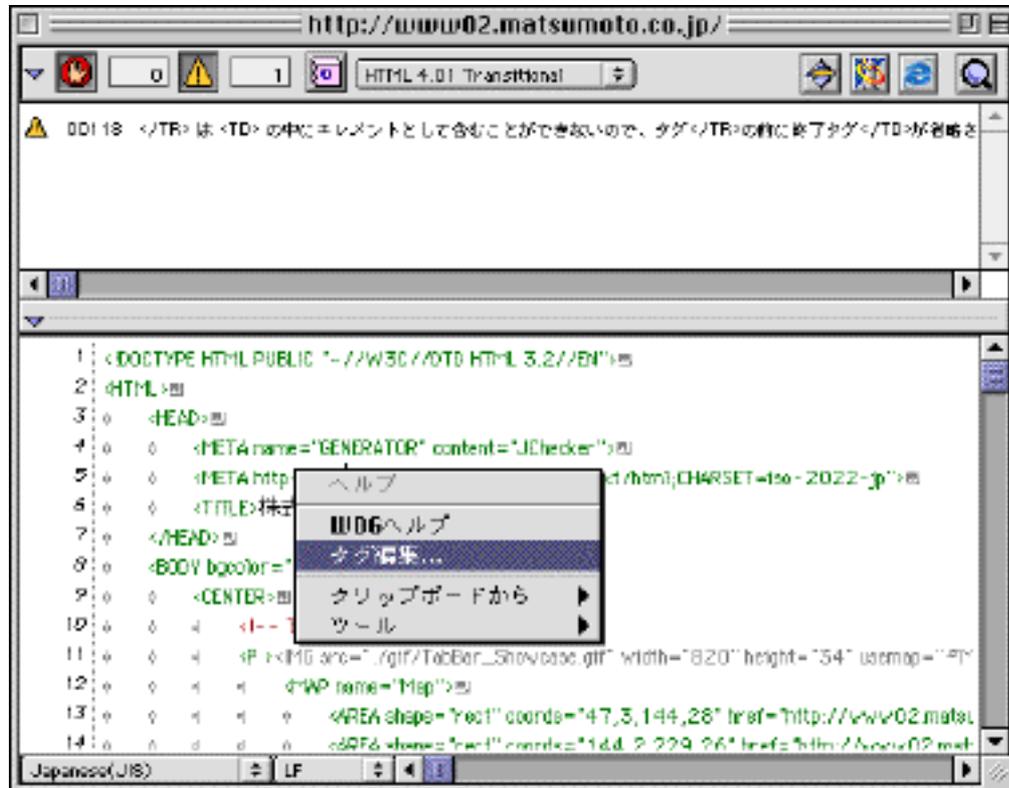
コンパイル済みのスクリプトファイルを、マクロウィンドウヘドロップすると、そのファイルがMacro Menu Itemsフォルダ内に移動しマクロメニューに登録されます。また、マクロメニューの項目をドラッグし、Finder上の「ごみ箱」ヘ捨てると対応するスクリプトファイルが「ごみ箱」ヘ移動しメニュー項目が削除されます。

MenuOrder.dat

Macro Menu Itemsフォルダとそのサブフォルダ内には、MenuOrder.datというファイルが自動的に作成されます。MenuOrder.dat はマクロメニューの表示順序をコントロールするためのもので、ここにマクロメニューの表示順序が保存されています。

13.3 コンテキストメニュー

マウスカーソルが編集ウィンドウのなかにあるときに、controlキーを押しながらマウスをクリックすると、JCheckerのコンテキストメニューがポップアップします。



コンテキストメニューの項目はマクロメニューと同じようにユーザーが自由に追加・作成でき

ます。

実行したい機能をスクリプト編集プログラムで作成し、スクリプトをコンパイル済みスクリプトとして保存します。そしてそのファイルを、JChecker3.0と同じフォルダにある「Context Menu Items」フォルダのなかにコピーします。



これで、JChecker3.0を再起動するかマクロメニュー「メニュー再構築」を選択するとそのスクリプトがコンテキストメニューに現れます。

ただし、コンテキストメニューはマクロメニューと異なりメニューの階層化、区切り線の登録、項目の順序のコントロールは行えません。

付録1、付属マクロについて

JChecker3.0の「Macro Menu Items」フォルダのなかには、HTMLの編集に役立つ付属マクロ一式があらかじめ格納されています。



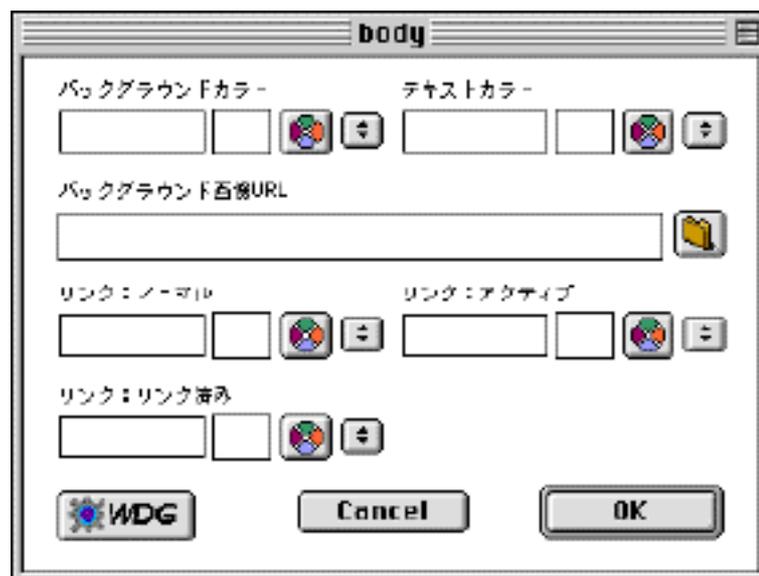
付属マクロは、HTML4.01の仕様をほぼ網羅し以下の強力な機能を備えています。

- (1)編集集中のHTMLの既存タグの解析と修正
- (2)新規タグの挿入
- (3)エレメントや属性についてのヘルプ
- (5)その他

付属マクロには50種類ほどのマクロが登録されていますが、マクロの名前の末尾に「...」がついているものと、そうでないものの2種類があります。

「...」がついていないものは、選択するとすぐ実行し終了するマクロです。

名前の末尾に「...」がついているものは、選択すると、たとえば以下のようなダイアログが表示されます。



これらのダイアログはすべて、マクロの内部から起動された付属プログラムHTMLPlugInが表示しています。

HTMLPlugInについて

HTMLPlugInは、開発ツール「FaceSpan3.0」(Digital Technology International社)のもとでアップルスクリプトで作成されたアプリケーションです。HTMLPlugInのファイルは、マクロファイルと一緒に「Macro Menu Items」フォルダのなかに格納されています。



HTMLPlugIn

FaceSpanシステムを入手すれば、ユーザーがHTMLPlugInのダイアログのレイアウトを変更したり、アップルスクリプトを参照・修正することもできます。(FaceSpanについて詳しくは、<http://www.faceSpan.com/> を参照してください。)

HTMLPlugInを使用するマクロを、JCheckerからはじめて起動したとき、HTMLPlugInの場所を聞いてくる場合があります。そのときは、「Macro Menu Items」フォルダのなプログラムHTMLPlugInを指定してください。

また初期設定「エディタ」で「起動時にHTMLPlugInを起動する」をチェックしておく、JChecker起動時にHTMLPlugInも自動的に起動するので、マクロの起動が速くなります。

なお、2002年2月現在、FaceSpanは残念ながらOS Xのネイティブモードに対応していません。OS X上でHTMLPlugInを起動すると、OS Xのクラシックモードの動作になります。FaceSpanの開発元では、現在OS Xに完全対応したFaceSpan4.0の開発がすすめられています。FaceSpan4.0がリリースされ次第、当社でもHTMLPlugInのOS Xネイティブ化を行う予定です。

WDGヘルプ

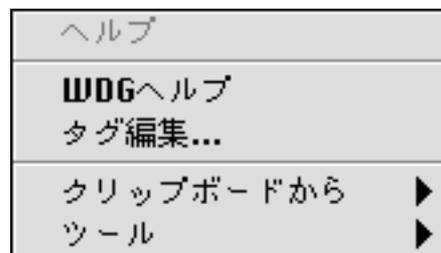
カーソルを調べたいエレメントのタグのなかにセットして、この「WDGヘルプ」を選択すると、そのエレメントについてのヘルプ情報が、ブラウザに表示されます。このヘルプ情報はWDG (Web Design Group)によって作成されたものです。

たとえば、カーソルをmetaタグのなかにセットして、マクロ「WDGヘルプ」を実行すると、以下のようにmetaエレメントについての解説が、ブラウザに表示されます。



WDGヘルプはインターネット経由でヘルプ情報を表示しますので、お使いのパソコンがインターネットに接続できる状態になっている必要があります。

また、WDGヘルプは、コンテキストメニューからも実行できます。

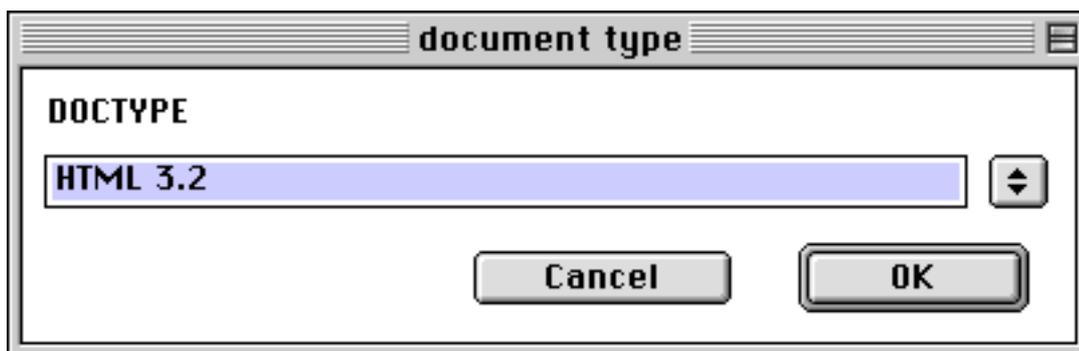


マクロ「タグ編集...」

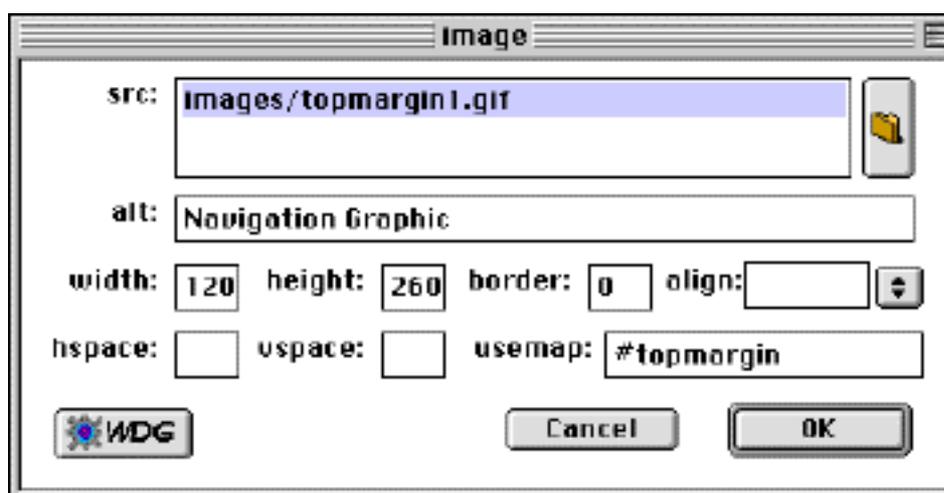
マクロ「タグ編集...」は、大変強力で、いちばん使いでのあるマクロです。

まず、カーソルを、編集したいタグのなかにセットして、マクロ「タグ編集...」を実行してみてください。タグの要素を解析して、その要素に応じたタグ編集ダイアログが表示されます。

たとえば、カーソルがDOCTYPE宣言にあるときにマクロ「タグ編集...」を実行すると、以下のようなDOCTYPE宣言編集ダイアログがあらわれます。



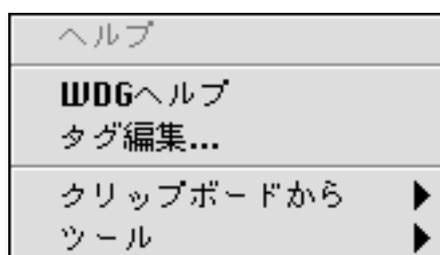
カーソルがイメージタグになかにあるときは、以下のようなIMGタグ編集ダイアログがあらわれます。



マクロ「タグ編集...」では、以下のエレメントをサポートしています。

doctype、head、meta、link、title、base、style、script、body、a、img、br、font、div、p、h1~h6、heading、hr、blockquote、ins、del、table、caption、col、colgroup、thead、tbody、tfoot、tr、td、th、form、legend、button、input、label、select、optgroup、option、textarea、applet、object、param、basefont、bdo、iframe、map、area、q、span、frame、frameset

また、WDGマクロ「タグ編集...」は、コンテキストメニューからも実行できます。

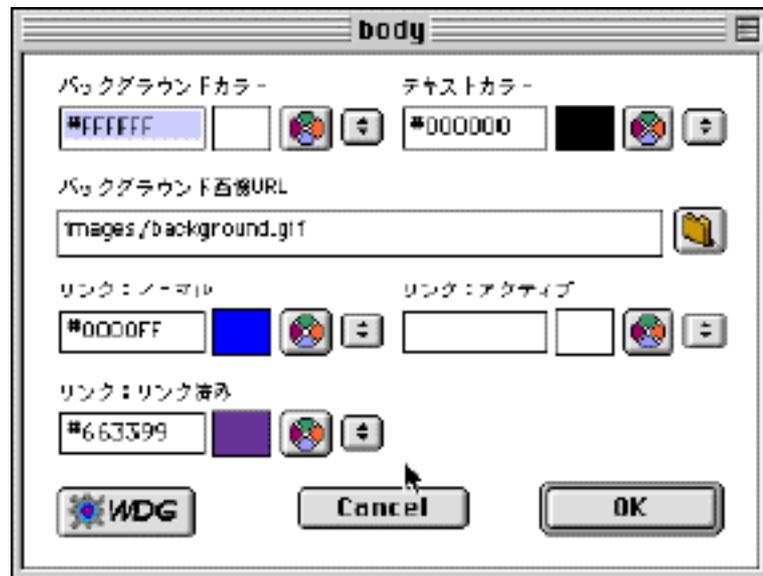


編集ダイアログの使い方

タグ編集ダイアログは50種類ほどありますが、使い方はほぼ共通していますので、BODYタ

グを例に説明します。

カーソルを編集ウィンドウのなかのBODYタグのなかにセットして、マクロ「文書本体 body...」を選択すると、以下のようにBODYタグ編集ダイアログがあらわれます。



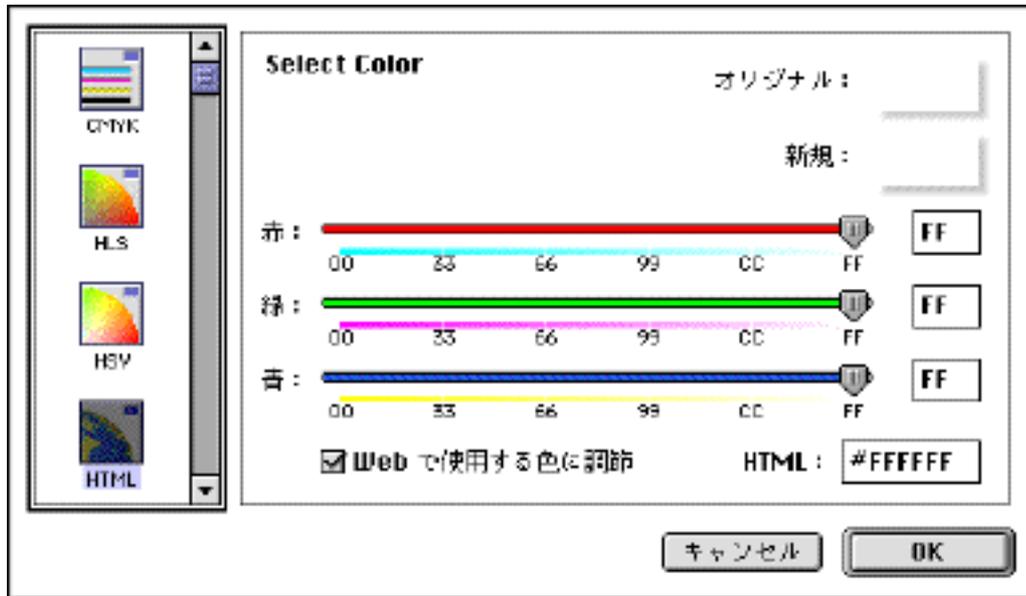
マクロは自動的に現在のタグのなかで指定されている属性を解析して、上記のように現在セットされてる属性値が表示されます。属性値が空白のものは、属性そのものがタグのなかで指定されていないことを表します。

属性値をダイアログから希望の値に変更し「OK」ボタンを押すと、タグが書き変わります。

属性値に空白を指定すると、対応する属性名と属性値の両方がタグから削除されます。



は、カラーパレットボタンです。クリックすると以下のようなカラーパレットが表示されます。



は、ファイルまたはフォルダのURLを指定するボタンです。クリックすると以下のようにファイルまたはフォルダの場所を聞いてきます。



希望のファイルを選択して「開く」ボタンを押すと、編集中のHTMLファイルから選択したファイルへの相対パス(URL)が属性値にセットされます。編集中のHTMLにファイルが割り当てられていないときは「file://」ではじまる絶対パスがセットされます。



は、WDGヘルプボタンです。クリックすると、そのダイアログに応じたヘルプがブラウザに表示されます。BODYタグ編集ダイアログのなかでこのボタンをクリックすれば、以下のようにBODY元素についての情報が表示されます。



一般にタグ編集ダイアログは、カーソルが対応するタグのなかにあるときに選択するとタグ編集モードになりますが、カーソルが対応するタグの外にあるときはタグの新規挿入モードになります。

なお、例外としてマクロ「文書タイプ document type...」「文字セット character set...」「文書本体 body...」では、カーソルが対応するタグのなかになくても、タグを自動的に検索し、対応するタグがあるときはタグ編集モードで動作します。

付録2、メッセージについて

HTMLをチェックをしたときに、JCheckerから出力されるメッセージについて解説しています。

メッセージには、エラーと警告の2種類があります。検出した問題点をエラーとするか、警告とするか、あるいは全く無視してしまうかは、初期設定「メッセージ」でユーザーが自由に指定できます。初期値としては、HTMLを解釈するうえで明らかに支障をきたすような誤り（DTDに違反している誤り）はエラー、DTDに反してはいないがあまり勧められないような表現については警告がセットされています。

No.1

「<」マークとエレメント名の間に空白が含まれています。タグでないなら「<」マークは、< と表記してください。

タグの始まりを表す「<」マークとエレメント名の間にスペース、タブまたはリターンコードが含まれているときに表示されます。Internet Explorer では空白があってもタグと解釈しますが、Netscape Navigator dでは「<」をタグの始まりではなく単なるデータと解釈します。「<」が単なるデータあるなら< と表記してください。

No.2

xxxx のようなエレメント名は存在しません。

タグのなかに存在しないエレメント名を指定したときに表示されます。エレメント名のスペルに誤りがないか確認してください。使用できるすべてのエレメント名はタグ辞書に登録されています。もし、独自のエレメントを使用する場合は辞書のカスタマイズを行って下さい。

No.3

タグのなかに「<」マークがあります。または「<」マークをデータとして表記するときは、< と表記してください。

「<」マークから始まるタグのなかで別の「<」マークが「>」の前に現れた時に表示されます。「<」を単なるデータとして指定したいときは「< 」と表記してください。

No.4

終了タグ</xxxx>に対応する開始タグ<xxxx>が同じ階層位置にないか、対応する<xxxx>が見つかりません。

終了タグ</xxxx>に対応する開始タグ<xxxx>がないときや、タグの階層構造が壊れてしまっているために、対応する開始タグが同じ階層位置にないときに表示されます。

一見対応する開始タグがあるのにこのエラーが表示され戸惑うときがあるかも知れません。そのような場合は開始タグ<xxxx>と終了タグ</xxxx>間に階層構造を壊してしまうようなタグがあるはずです。

たとえば、以下のような例です。

```
<HTML><HEAD>
<B>
<TITLE>タイトル</TITLE>
</B>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
本文
</BODY>
</HTML>
```

この文書を文法チェックすると、5行目と9行目には「対応する開始タグ<HEAD>と<HTML>が見つかりません。」というエラーが表示されます。1行目には<HTML>と<HEAD>がちゃんとあるのにどうしてエラーになるのでしょうか。

実は、本当のエラーの原因は2行目のにあるのです。タグは<HTML>のなかにも<HEAD>のなかにも書けないタグなので、が現れた時点で<HEAD>と<HTML>が終了し、終了タグ</HEAD>と</HTML>があるものと見なされて以下のように解釈されてしまいます。見なし終了タグは警告メッセージで警告されます。

```
<HTML>
  <HEAD>
    </HEAD> <----見なし終了タグ
  </HTML> <----見なし終了タグ
  <B>
    <TITLE>タイトル</TITLE>
  </B>
  </HEAD> <----対応する開始タグがない! ?
  <BODY BGCOLOR="#FFFFFF"> 本文</BODY>
</HTML> <----対応する開始タグがない! ?
```

見なし終了タグ</HEAD>と</HTML>が3行目と4行目にあるため、8行目と10行目の終了タグ</HEAD>と</HTML>がエラーになるわけです。

以上のように、一見開始タグがあるのにこのエラーメッセージが出て原因がわからないときは、

メニュー「HTML整形」で階層構造がわかるように整形し、見なし終了タグの警告を調べてみてください。

 No.5

yyyy は xxxx の中にエレメントとして含むことができないので、タグ<yyyy>の前に終了タグ</xxxx>が省略されているものと見なします。

エレメント(xxxx)には終了タグを省略できるものがありますが、そのエレメント(xxxx)のなかに含むことができない別のエレメント(yyyy)の開始タグが現れたときに、そこでエレメント(xxxx)が終了し終了タグ</xxxx>が省略されているものと見なされこの警告が表示されます。終了タグが省略できるかどうか、またどんなエレメントを含むことができるかは、メニュー「辞書」で調べられます。

終了タグを省略しているつもりがないのにこの警告がでたときは、後行に書いてある終了タグがエラーになるので注意してください。

 No.6

この「>」マークに対応する「<」マークがありません。 または「>」マークをデータとして表記するときは、> と表記してください。

データ文のなかに「>」マークが現れたときに表示されます。「>」を単なる文字としてデータ文のなかに記述したいときは「> 」と表記してください。

 No.7

タグ<xxxx>に対応する終了タグ</xxxx>がありません。

エレメントxxxx が終了タグの必要なエレメントであるのに、対応する終了タグが見つからないときに表示されます。そのエレメントが終了タグの必要なエレメントかどうかは、メニュー「辞書」で確かめることができます。

 No. 8

xxxx は、yyyy のなかにエレメントとして含めることはできません。

yyyyの要素として、エレメント xxxxを含めることができないとき表示されます。yyyyのなかにどのようなエレメントを書くことができるかは、メニュー「辞書」で調べることができます。

 No. 9

タグ<xxxx>と</xxxx>の間が空です。

終了タグと開始タグの間になにも書かれていないとき警告としてこのメッセージが表示されま

す。テーブルなどで空の項目があるときは意図してそうする場合もありますが、そうでないときは無駄なタグなので開始タグと終了タグを削除してください。

 No.10

この属性値は、「" "」マークでくくる必要があります。

属性値が「" "」マークでくくられていないとき表示されます。ただし、72文字以内で英数字(ハイフン'-'、ピリオド'.'も可)の場合は「" "」マークでくくる必要はないのでエラーにはなりません。

 No.11

属性zzzzは、スペルに誤りがあるか、タグ<xxxx>のなかで使うことはできません。

エレメントxxxxで使うことができない属性zzzzをタグのなかで指定したときに表示されます。そのエレメントにどんな属性を書くことができるのかはメニュー「辞書」で調べることができます。

 No.12

タグ<xxxx>の必須属性 zzzz が定義されていません。

エレメントxxxxのタグのなかで必ず指定しなければいけない属性zzzzが指定されていないときに表示されます。たとえばタグのなかでは必ず「src」属性を指定する必要があります。必須属性はメニュー「辞書」で調べることができます。

 No.13

文字リファレンスの表記に誤りがあります。または、& マークを表記するときは & と表記してください。

データや属性値の指定のなかで '&' マークを使うと、それは文字リファレンスの先頭を意味し別の文字に置き換えられます。たとえば「<」は文字「<」に置き換えられます。&を単なる文字として指定したいときは「&」と表記してください。

 No.14

属性値の終わりに「"」マークが付いていない。

属性値のはじめに「"」マークがついているのに、終わりに「"」マークが付いていないときに表示されます。

 No.15

タグ<HTML>を、文書の先頭に書くことをお勧めします。

HTMLの文書は<!DOCTYPE>宣言とタグ<HTML>ではじまり</HTML>で終わるべきなの

で、<HTML>で始まっていないときにこの警告が表示されます。ただし、<!DOCTYPE>宣言あるいは<HTML>がなくても間違いではありません。

 No.16

xxxx のなかで、エレメント<yyyy>が正しい順番で定義されていません。

<HTML>の子エレメント<HEAD> <BODY> <FRAMESET> <PLAINTEXT>や<TABLE>の子エレメント<CAPTION> <COL> <COLGROUP> <THEAD> <TFOOT> <TBODY> <TR>は、正しい順番で指定する必要があります。その順番が正しくないときにこのエラーが表示されません。

 No.17

HEAD 内には<TITLE>文を定義することをお勧めします。

HEADエレメントのなかに<TITLE>が書かれていないときこの警告が表示されます。<TITLE>タグはブラウザのブックマークなどで使われますので、省略せずに正しく書いてください。

 No.18

データを xxxx の要素として直接書くことはできません。

エレメントxxxxの要素にデータを書くことが許されていないのに、データを指定するとこのエラーが表示されます。例えば、<TR>の内容には<TD><TH>を指定することはできますが、データを直接書くことはできません。エレメントとしてデータを指定することができるかどうかは「辞書」で調べることができます。

 No.19

エレメント<xxxx>の属性 zzzz の値には、1バイトの文字しか指定できません。

1バイトの文字(%Character)しか属性値に指定できない属性に2バイト以上の属性値を指定したときにエラーとして表示されます。

 No.20

属性 zzzzz には、属性値を指定する必要があります。

属性 zzzzz に属性値がなにも指定されていないかたときに表示されます。

 No.21

属性 zzzza の値には、(aaa|bbb|ccc....)のうちどれかを指定してください。

属性値に、あらかじめ決められた文字列(aaa, bbb, cccc)のうちどれかしか指定できない属性(例:alignなど)があります。そのような属性の値に決められた文字列以外の値を指定したとき

にエラーとして表示されます。どのような文字列を指定できるかは、メニュー「辞書」で調べられます。



No.22

属性 zzzz のURL値 "uuuu" に、スペースまたは¥(バックスラッシュ)が含まれていません。

値としてURL値を指定する属性の属性値のなかにスペースまたは¥(バックスラッシュ)が含まれていたときに表示されます。



No.23

属性 zzzz のURL値 "uuuu" に、制御コードまたは非ASCII文字が含まれています。

値としてURL値を指定する属性の属性値のなかに制御コードまたは非ASCII文字が含まれていたときに表示されます。



No.24

属性 zzzz のURL値に含まれる 'x' は '%yyyy' と表記したほうが安全です。

URL文字列のなかの記号 ' ' などは%付き16進数で表記したほうが安全です。



No.25

属性 zzzzz の値 'vvvv' とこのファイルのエンコーディングが一致していません。

文書をファイルから読み込んだときのエンコーディングと、属性charsetで指定しているエンコーディングが一致しなかったときこの警告が表示されます。



No.26

キャラクタセット値 'vvvv' は 'www' と表記したほうが安全です。

属性charset の指定で、x-sjis または x-euc-jp が指定されているときは、Shift_JIS または EUC-JP にするように警告が表示されます。x-sjis、x-euc-jp は公式に定められた名称ではありません。



No.27

半角カタカナは正しく表示できない場合があるので使用しないほうがいいです。

半角カタカナは、インターネットのやりとりでは使ってはいけないことになっています。使わないほうが無難です。



No.28

タグ<HTML>を <!DOCTYPE>宣言の直後に書くことをおすすめします。

<!DOCTYPE>宣言があるのにその次にタグ<HTML>がかかれていないときこの警告が表示されます。ただし、<HTML>がなくても間違いではありません。

 No.29

属性 zzzz の値'cccc'はHTML仕様で決められている基本16色以外の色です。

色を指定するときに、基本16色 ("aqua", "black", "blue", "fuchsia", "gray", "green", "lime", "maroon", "navy", "olive", "purple", "red", "silver", "teal", "white", "yellow") 以外の色文字列を指定したときに表示されます。Netscape Navigator と Internet Explorer では140種類のカラー文字列を指定できるので間違いではありません。

 No.30

機種依存文字は、機種やフォントによって正しく表示できない場合があるので使用しないほうがいいです。

使用しているフォントや機種によって定義が異なるため、機種やフォントによっては正しく表示できない文字を機種依存文字といいます。shift-JISコードで0x8450から0x889Eまでと、0xEB40から0xEFFCまでの文字があったとき、機種依存文字として警告します。エディタフォントが日本語フォントでないときは、このチェックを行いません。

 No.31

アンカー "vvvvv" がみつかりません。

タグで参照しているリンク先が見つかりません。リンク先のが正しく定義されているか確認してください。。

 No.32

マップ "vvvvv" がみつかりません。

属性 USEMAP="#vvvvv" で参照しているマップが見つかりません。マップ<MAP NAME="vvvvv" >が正しく定義されているか確認してください。

 No.33

保存先または<BASE>が指定されていないため相対パス"vvvvvv"の解析ができません。

相対パスから絶対パスを求めるためには、その基準となるHTMLファイルの保存先が先に割り当てられている必要があります。HTMLファイルを保存してファイルを割り当てるか、<BASE>タグで基準となるURLを指定してください。

 No.34

ローカル書類 "vvvvvv" がみつかりませんでした。

属性"zzzzz"で指定されているファイル"vvvvvv"を、ハードディスク内に見つけることができませんでした。

 No.35

書類"vvvvv"があるホストへの接続に失敗しました。

属性"zzzzz"で指定されているファイル"vvvvvv"のあるホストへのHTTPアクセスに失敗しました。インターネットが正しく設定されているかどうかチェックしてください。

 No.36

このURL "vvvvv" はJCheckerではチェックできません。

JCheckerでチェックできるリンクは、ローカルファイルまたは http, https でアクセス可能なリンクです。残念ながら mailto, ftp などのプロトコルのチェックはできません。

 No.37

リモート書類 "vvvv"が見つかりませんでした。(応答コード)

インターネット上に書類"vvvv"が存在しない。400番台の応答コードはクライアント側でのエラーを示し、500番台はサーバー側でエラーが発生したことを示している。

 No.38

この書類のURLは恒久的に"xxxxx"から"yyyyyy"へ移動しています。

ページのURLが"xxxxx"から"yyyyyy"にすでに変わってしまっています。HTML内のURLの指定を"yyyyyy"へ書き換える必要があります。

 No.39

この書類のURLは一時的に"xxxxx"から"yyyyyy"へ移動しています。

URLが一時的に、"xxxxx"から"yyyyyy"へ変わっていることを示しています。

 No.40

ヌルコードを検出しました。除去してください。

ヌルコード(0x00)は通常のテキストに現れるはずのないコードです。ヌルコードがあるとブラウザの文字セット判定などが正しく動作しなくなる場合もありますので、除去することをおすすめします。

 No.41

DOCTYPE宣言の内容に誤りがあり、DTDを特定することができません。

DOCTYPEの書式がまちがっていると、JCheckerは辞書を特定することができません。例えば、HTML3.2の場合は以下のように宣言します。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
```

No.42

DOCTYPE宣言のDTDに対応する辞書は、JCheckerには含まれていません。デフォルトの辞書を使用します。

JCheckerの辞書メニューには、このDOCTYPEで指定されているDTDに対応する辞書がありません。DTD名が正しいかどうかチェックしてください。正しいときはそのDTDを入手し、DtdToDicで辞書に変換し、辞書メニューに追加してください。

No.43

DOCTYPE宣言がないが無効です。

通常、HTMLの先頭のDOCTYPE宣言によって、文書型(DTD)を指定します。文書型にはいろいろなバージョンがあるので、DOCTYPEがないとどのDTDに基づいているのか判断できません。DOCTYPEは宣言するようにしてください。

No.44

このエレメントの属性 zzzz の値は"vvvv"に固定されています。

このエレメントの属性 zzzz の値には"vvvv"以外を指定することはできません。

No.45

XMLでは、属性名は省略できません。zzzz="zzzz" のように指定してください。

HTMLでは属性値を単独に指定するだけでその属性が「真」であることを指定できる属性（最小化形式）がありますが、XMLでは最小化形式がなく属性名は省略できません。

No.46

この属性 zzzz の属性値には"zzzz"を指定するか、HTMLのときは属性名を省略し単にzzzz だけを指定してください。

この属性 zzzz は、真か偽かだけを指定するためのものなので、真の時は zzzz="zzzz" のように指定してください。HTMLのときは属性名を省略し単に zzzz だけの指定でもかまいません。

No.47

エレメント <xxxx>の属性 zzzz の値には数字以外は指定できません。

エレメント <xxxx>の属性 zzzz の属性値はNUMBER(0から9までの数字を組み合わせたもの)でなければなりません。

 No.48

エレメント <xxxx>の属性 zzzz の値は英字ではじまる名前文字列でなければなりません。

エレメント <xxxx>の属性 zzzz の属性値はNAME(英字ではじまり、英字、数字、ハイフン、アンダースコア、コロン、ピリオドの組み合わせ)でなければなりません。

 No.49

エレメント <xxxx>の属性 zzzz の値はNMTOKEN(名前文字列)でなければなりません。

エレメント <xxxx>の属性 zzzzz の属性値はMNTOKEN(英字、数字、ハイフン、アンダースコア、コロン、ピリオドの組み合わせ)でなければなりません。

 No.50

エレメント <xxxx>の属性 zzzz のID値はすでに他で定義されています。

エレメント <xxxx>の属性 zzzzz の属性値はIDタイプなので、他で使われていないユニークな値でなければなりません。

 No.51

属性 zzzz のキャラクタセット値は、IANAに登録されていないセット名です。

キャラクタセット値には、IANA(Internet Assigned Numbers Authority)に登録されている名前を指定します。登録されているセット名は以下で一覧できます。

<http://www.iana.org/assignments/character-sets>

 No.52

属性 zzzz の値は、整数値(*)または%付きの数値でなければなりません。

この属性の値は、MultiLength型なので、数値%、数値、数値*、* のいずれかでなければなりません。

 No.53

属性 zzzz の値は、整数値(*)または%付きの数値をカンマで区切って複数指定したものでなければなりません。

属性 zzzz の値は、整数値(*)または%付きのMultiLength型の値をカンマで区切って複数指定したものでなければなりません。

 No.54

エレメント<xxxx>には、終了タグを書くことはできません。

エレメント<xxxx>には、終了タグを書くことはできません。

 No.55

属性 xxxx の値は、整数値または%付きの数値でなければなりません。

この属性の値は、Length型なので、数値%、数値 のいずれかでなければなりません。

 No.56

属性 zzzz の値は、整数値または%付きの数値をカンマで区切って複数指定したものでなければなりません。

属性 zzzz の値は、整数値または%付きのLength型の値をカンマで区切って複数指定したものでなければなりません。

 No.57

属性 zzzz の値 "vvvv" はカラー値として解釈できません。

たとえば、カラー値に「赤」を指定したいときは、あらかじめ決められ文字列"red"を指定するか、あるいはRGB値として"#FF0000"のように指定します。

 No.58

エレメント<xxxx>は空エレメントではないので、終了タグは省略できません。

XTMLでは、空エレメントのタグは<xxxx />として終了タグを省略できますが、空エレメントでない場合は省略できません。

 No.59

XHTMLでは先頭にXML宣言 <?xml verion="1.0" ?> を記述することを奨めます。

エンコーディングがUTF-8かUTF-16のときは、XML宣言 <?xml verion="1.0" ?> は省略できますが、書略しないことをお奨めます。

 No.60

書類のエンコーディングがUnicodeでないので、XML宣言 <?xml version="1.0" encoding="文字セット" ?> が必要です。

書類のエンコーディングがUTF-8/UTF-16以外のときは、XML宣言 <?xml verion="1.0" encoding="文字セット" ?> で、文字セットを指定しなければなりません。

 No.61

CDATAセクションの終了タグ `]]>` がありません。

XHTMLのCDATAセクションは、`<![CDATA[` ではじまり `]]>` で終わらなければなりません。

 No.62

時間切れのためリモート書類 "xxxx" へ接続できませんでした。

初期設定で指定しているタイムアウト秒数以内に、ホストからの応答が得られない。リンク先が存在していても回線やホストが混んでいて、このエラーがでる場合があります。URLに間違いがなければ、再チェックしてみてください。

 No.63

URLアクセスマネージャでエラーが発生しチェックできませんでした。

URLアクセスマネージャのバグのため、リンクチェックできませんでした。URL文字列が非常に長いときなど、Mac OS Xで発生する場合があります。

 No.64

エレメント`<HTML>`の先頭には、`<HEAD>`文を定義することをお奨めします。

`<HEAD>`文とそのなかに`<TITLE>`文を書くことをお奨めします。`<TITLE>`タグはブラウザのブックマークなどに使われます。

 No.65

この属性 `zzzz` はHTMLエディタ独自のもので、DTDには定義されていません。

PageMill, GoliveなどのHTMLエディタが、編集の便宜のために独自に挿入する属性です。DTDには存在しません。

 No.66

DOCTYPE宣言のこの部分は大文字で指定してください。

DOCTYPE宣言文のなかでは、大文字小文字の区別があります。

 No.67

属性 `zzzz` のURL"`vvvv`" の先頭のスキームに誤りがあります。

URLのスキームには、`http: https: ftp: gopher: nntp: telnet: wais: mailto: news: file:` のどれかを指定してください。

付録3、アップルスクリプト

JCheckerはアップルスクリプトをサポート*し、スクリプト編集プログラムなどの外部プログラムからJCheckerを操作することができます。また、作成したアップルスクリプトをJCheckerのマクロメニューに登録すれば、自由にJCheckerの機能をカスタマイズできます。さらにマクロメニューから他のアップルスクリプト対応ソフトを操作することも可能です。

スクリプトからは、起動・終了、編集ウィンドウのオープン・クローズ・移動、書類の保存などのほかに、バイト単位、文字単位、ワード単位、行単位、段落単位でHTMLの様々な編集操作を行えます。JCheckerのアップルスクリプトは、Jeditとほぼ同等の機能に備え、さらにHTMLのタグや属性を操作するためのコマンドも備えています。

なお、この説明書ではJCheckerで使用できるアップルスクリプトについてのみ述べられていますので、アップルスクリプト全般については、以下のホームページあるいは市販のアップルスクリプトの入門書を参照してください。

AppleScript 入門用 AppleScript ガイド
<http://www.apple.co.jp/applescript/begin/>

また、以下のホームページからアップルスクリプト関連の役に立つ情報を入手できます。

<http://www.asep.apple.co.jp/Techdoc/AppleScript/>

*JCheckerのほとんどのテキスト操作はアップルスクリプトから行えるようになっていますが、アップルスクリプトからではできない操作も残念ながら一部あります。またJCheckerのテキスト操作はレコーダブルではありません。

Jeditとの互換性について

JChecker3.0のアップルスクリプトは、エディタ Jedit4.0から移植されたものなので、基本的にJedit4.0と互換性があります。

ただし、以下のコマンドはHTML専用なのでJCheckerにしかない独自コマンドです。

check、linkCheck、correct、checkAll、minimize、beautify、recolor、GetTagName、GetAtrbValue、SelectTag、SelectAtrb、GetColor、GetRelativePath、SetAtrb、HTMLColotToRGB、GetDtdName

また、以下のオブジェクトとコマンドはHTMLの編集には必要がないのでサポートしていません。

サポートしていないオブジェクト

style run、 picture、 sound、 aliasFile、 movie

サポートしていないコマンド

play、 speak、 remove break、 detab、 entab、 convert kanji space to ascii、 toYomi、 kanjiToKana、 toKatakana、 toHiragana、 toZenkaku、 toHankaku、 convert kana to zenkaku、 convert ascii to hankaku、 remove trailing space、 remove null、 jsort、 uniq、 reverse、 toUnicode、 fromUnicode、 OpenDict、 OpenChecker、 clear undo、

オブジェクト(クラス)

オブジェクトには、 application、 window、 document、 byte、 character、 word、 line、 paragraph、 selection があります。

application オブジェクト

適用可能な命令:

run、 launch、 quit、 get、 set、 exists、 preDialog、 postDialog、 suspend、 OpenURL、 select、 updateMacroMenu

プロパティ:

sound boolean -- チェックが終わったとき結果をサウンドで知らせるかどうか。
autoCheck boolean -- 書類を開いたとき自動的に文法チェックを行うかどうか。
dictionary international text -- デフォルトの辞書。
encoding international text -- デフォルトのエンコーディング。
lineBreak auto/mac/mac/unix/dos -- デフォルトの改行タイプ。
allDTDs a list of international text [r/o] -- 登録されている辞書の全リスト
allCharsets a list of international text [r/o] -- サポートしている全文字セット

batchMenu international text [r/o] -- ワードサービスメニューに表示す留文字列。
appLoc alias [r/o] -- JChecker3.0のファイルエイリアス(ワードサービス用)
batchIcon 'SICN' [r/o] -- JChecker3.0のアイコン(ワードサービス用)

window オブジェクト

適用可能な命令:

open、 close、 count、 delete、 exists、 get、 make、 move、 select、 set

プロパティ:

bounds bounding rectangle -- ウィンドウの領域
closeable boolean [r/o] -- クローズボックスをもっているかどうか?
titled boolean [r/o] -- タイトルをもっているかどうか?
index integer -- そのウィンドウの番号
floating boolean [r/o] -- フローティングウィンドウかどうか?

modal boolean [r/o] -- ウィンドウがモーダルかどうか?
resizable boolean [r/o] -- ウィンドウが伸縮可能かどうか?
zoomable boolean [r/o] -- ウィンドウがズーム可能かどうか?
zoomed boolean -- ウィンドウがズームされているかどうか?
name string -- ウィンドウのタイトル
visible boolean [r/o] -- ウィンドウがビジブルかどうか?
position point [r/o] -- ウィンドウの左上座標

document オブジェクト

適用可能な命令:

make, open, close, print, save, revert, close, data size, get, set, insert, find, replace, mark, exist, cut, copy, paste, select, undo, redo, preDialog, postDialog, suspend, check, linkCheck, correct, checkAll, minimize, beautify, recolor, GetTagName, GetAtrbValue, SelectTag, SelectAtrb, GetColor, GetRelativePath, SetAtrb, HTMLColorToRGB, GetDtdName

プロパティ:

name string -- 書類のタイトル (変更可)
modified boolean [r/o] -- 最後に保存してから、変更があったかどうか?
specified boolean [r/o] -- ファイルが割り当てられているかどうか?
fileSpec file specification [r/o] -- ファイルスペック
errorCount integer [r/o] -- 検出したエラーの数
warningCount integer [r/o] -- 検出した警告の数
errorInfo international text [r/o] -- エラーと警告メッセージの内容
dictionary international text -- 辞書名
encoding international text -- 文字セット名
lineBreak auto/mac/mac/unix/dos -- 改行タイプ
xml boolean [r/o] -- htmlのときはfalseを、xhtmlのときはtrueをかえす

byte オブジェクト

byte(バイト)は、1バイト単位のオブジェクトです。英数文字のときはbyteとcharacterは同じですが、全角文字のときは2 bytesで1 characterになります。

適用可能な命令:

print, data size, get, set, insert, find, replace, mark, count, cut, copy, select, change case

プロパティ:

length integer [r/o] -- オブジェクトのバイトサイズ
byteoffset integer [r/o] -- オブジェクトのdocument内のオフセット(1から数える)
line number integer [r/o] -- オブジェクトを含む行の行番号
paragraph number integer [r/o] -- オブジェクトを含む段落の段落番号

character オブジェクト

character(文字)は、1文字を表すオブジェクトです。英数文字も、全角文字もそれぞれ1 characterとしてカウントします。

適用可能な命令:

print, data size, get, set, insert, find, replace, mark, count, cut, copy, select, change case

プロパティ:

length integer [r/o] -- オブジェクトのバイトサイズ
byteoffset integer [r/o] -- オブジェクトのdocument内のオフセット(1から数える)
line number integer [r/o] -- オブジェクトを含む行の行番号
paragraph number integer [r/o] -- オブジェクトを含む段落の段落番号

word オブジェクト

wordは、空白(半角スペース、リターンコード、タブコード)や記号で区切られた英数文字列を1 wordとしてカウントします。和文の場合は、便宜的に同じ字種(漢字、ひらがなあるいはカタカナ)のものをそれぞれまとめて1 wordとしてカウントしますが、語本来の意味は持ちません。

適用可能な命令:

print, data size, get, set, insert, find, replace, mark, count, cut, copy, select, change case

プロパティ:

length integer [r/o] -- オブジェクトのバイトサイズ
byteoffset integer [r/o] -- オブジェクトのdocument内のオフセット(1から数える)
line number integer [r/o] -- オブジェクトを含む行の行番号
paragraph number integer [r/o] -- オブジェクトを含む段落の段落番号

line オブジェクト

line は、JCheckerの編集ウィンドウ上での物理的な1行を、1 lineとしてカウントします。lineオブジェクトの内容は、編集ウィンドウで文書の横幅を変更すると変化します。

適用可能な命令:

print, data size, get, set, insert, find, replace, mark, count, cut, copy, select, change case

プロパティ:

length integer [r/o] -- オブジェクトのバイトサイズ
byteoffset integer [r/o] -- オブジェクトのdocument内のオフセット(1から数える)
line number integer [r/o] -- オブジェクトを含む行の行番号
paragraph number integer [r/o] -- オブジェクトを含む段落の段落番号

paragraph オブジェクト

paragraph は、リターンコードで区切られた文字列を、1 paragraphとしてカウントします。
paragraphは、lineと異なり、文書の横幅を変えても内容は変わりません。

適用可能な命令:

print, data size, get, set, insert, find, replace, mark, count, cut, copy, select, change
case

プロパティ:

length integer [r/o] -- オブジェクトのバイトサイズ
byteoffset integer [r/o] -- オブジェクトのdocument内のオフセット(1から数える)
line number integer [r/o] -- オブジェクトを含む行の行番号
paragraph number integer [r/o] -- オブジェクトを含む段落の段落番号

selection オブジェクト

selection は、document内の選択されている部分をあらわすオブジェクトです。サイズが0のときはカーソルを意味します。

適用可能な命令:

print, data size, get, set, insert, find, replace, mark, count, cut, copy, select, change
case

プロパティ:

length integer [r/o] -- オブジェクトのバイトサイズ
byteoffset integer [r/o] -- オブジェクトのdocument内のオフセット(1から数える)
line number integer [r/o] -- オブジェクトを含む行の行番号
paragraph number integer [r/o] -- オブジェクトを含む段落の段落番号

コマンド(イベント)

run 命令

run variable OR Reference

アプリケーションやスクリプトオブジェクトをファインダからダブルクリックしたのと同じように起動します。

```
run application "JChecker3"
```

run命令は、次のlaunch命令と異なり、OpenApplicationのイベントを送るので、すでにアプリケーションが起動しているときでも新規ウィンドウがそのたびに開かれます。

launch 命令

```
launch variable OR Reference
```

アプリケーションやスクリプトオブジェクトをファインダからダブルクリックしたのと同じように起動します。

```
launch application "JChecker3"
```

run命令と異なり、すでにアプリケーションが起動しているときは新規ウィンドウを開くことはありません。

make 命令

```
make new type class --新規に作成するクラス名  
Result: reference -- 新エレメントのオブジェクトスペシファイア
```

新しいdocumentまたはwindowを開きます。documentとwindow以外のオブジェクトに対しては使用できません。

```
make new document
```

```
make new window
```

open 命令

```
open alias -- list of objects to open
```

open命令では、アプリケーションや書類を開くことができます。例えば、Macintosh HDの文書フォルダにあるHTMLファイル「index.html」をJCheckerで開くには以下のようにします。

```
tell application "JChecker3"
```

```
open file "Macintosh HD:mySite:index.html"
end tell
```

print 命令

```
print alias -- list of objects to print
```

print命令では、ファイルや書類の印刷をおこないます。JCheckerのメニュー「ファイル」「一部の印刷」を選択したときと同じ機能です。print命令の対象となるオブジェクトは、fileとdocumentのほかにもselection、paragraph、lineなども指定できます。

```
tell application "JChecker3"
    print file "Macintosh HD:mySite:index.html"
end tell
```

選択領域の3行目から5行目までを印刷する

```
tell application "JChecker3"
    tell document 1
        print lines 3 thru 5 of selection
    end tell
end tell
```

save 命令

```
save reference -- 保存するオブジェクト
[in alias] -- 保存先のエイリアス(ファイル)名
```

referenceで指定したオブジェクトをディスクに保存します。

inパラメータで保存先ファイルを指定できます。inパラメータを省略すると、そのdocumentの指定ファイルに保存されます。新規文書などファイルが未指定のときは保存ダイアログが表示され保存先ファイルを聞いてきます。

document 1の全体をfile "HD:mySite:new.html" に保存する。

```
save document 1 in file "HD:mySite:new.html"
```

revert 命令

revert reference -- 対象となるdocument名

指定のdocumentを編集前の状態にもどします。document以外のオブジェクトには使用できません。

```
revert document 1
```

close 命令

close reference -- クローズするオブジェクト

[saving yes/no/ask] -- yes=無条件で保存、ask=そのつど聞く、no=保存しない

[in alias] -- 保存先のファイル名

documentまたはwindowを閉じます。savingパラメータをなにも指定しないときは、クローズボックスをクリックしたときと同じで、documentに変更があれば保存するかどうか聞いてきます。inパラメータで、別名で保存することもできます。

```
close document "index.html" saving yes in file "index2.html"
```

quit 命令

quit

JCheckerのメニュー「ファイル」「終了」を選択したときと同じ動作をします。パラメータはありません。

data size 命令

data size reference -- データサイズを調べるオブジェクト

Result: integer -- そのオブジェクトのバイトサイズ

referenceで指定したオブジェクトのデータサイズをバイトで返します。オブジェクトがlineまたはparagraphのときは、末尾のリターンコードも含めたバイトサイズを返します。

```
data size of paragraph 3 of document 1
```

count bytes 命令も同じ値を返します。

get 命令

```
get reference -- オブジェクトまたはプロパティ  
Result: anything -- その内容
```

result変数に、referenceで指定したオブジェクトの内容やプロパティを返します。

```
get word 1 of paragraph3 of document 1  
get fontSize of selection of document 1  
get document 1
```

set 命令

```
set reference -- 対象となるオブジェクト名  
to anything -- あたらしい設定値
```

set命令は、オブジェクトやプロパティに値をセットしたり、カーソル位置や選択領域を操作したりすることができる大変応用範囲の広いコマンドです。

document1のparagraph 3の3番目のwordを、文字列"abcd"でおきかえる。

```
set word 3 of paragraph 3 of document 1 to "abcd"
```

document1のデフォルトフォントを"Chicago"に変更する。

```
set font of selection of document 1 to "Chicago"
```

カーソルをparagraph 4のword 1の直前にセットする。

```
set selection to before word 1 of paragraph 4
```

挿入位置を示す前置詞には、before、after、end ofが指定できます。

カーソルをparagraph 4の最後(リターンコードの前)にセットする。

```
set selection to end of paragraph 4
```

カーソルをparagraph 4の直後(リターンコードの後)にセットする。

```
set selection to after paragraph 4
```

paragraph 4の全体を選択する。

```
set selection to paragraph 4
```

line 1からline3までを選択する。

```
set selection to line 1 thru 3
```

insert 命令

```
insert reference -- 挿入するオブジェクト  
at location reference -- 挿入位置
```

テキストを操作する上で一番強力で利用範囲の広い命令がこのinsert命令です。

さまざまなオブジェクトを希望の位置に挿入/置き換えできます。オブジェクトにスタイル情報があるときはそれらも含めて挿入されます。

位置を指定する前置詞には、before/beginning of/end of/after が使用できます。前置詞を指定せず位置にオブジェクトそのものを指定した場合は、そのオブジェクトがreferenceで指定したオブジェクトと置き替わります。

ファイル "Macintosh HD:mySite:new.html" の内容をdocument 1の最後に挿入します。

```
insert file "Macintosh HD:mySite:new.html" at end of document 1
```

文字列"abcdedg"をdocument 1のはじめに挿入します。

```
insert "abcdedg" at beginning of document 1
```

word 3を文字列"abcdedg"で置き換えます。

```
insert "abcdedg" at word 3
```

document 1の内容全部を、document 2の最後に追加します。

```
insert document 1 at end of document 2
```

document 1のline3のword 2を、document 1の最後の行に追加します。

```
insert word 2 of line 3 of document 1 at end of last line of document 2
```

paragraph 2 から 3 までを最後のparagraphの後に追加します。

```
insert ( paragraphs 2 thru 3 ) at after last paragraph
```

document1の選択領域を、document2の最後に追加します。

```
insert selection of document 1 at end of document 2
```

delete 命令

delete reference -- 削除するオブジェクト

referenceで指定したオブジェクトを削除します。

document 1のline 2を削除する。

```
delete line 2 of document 1
```

move 命令

move reference -- moveするオブジェクト
to location reference -- 異動先の位置

referenceで指定したオブジェクトを、指定の位置に移動します。move命令は同一のdocument内でのみ有効です。

選択領域の内容を、line 5の最後へ移動する。以下の2つは同じ動作をします。

```
move selection to line 5  
move selection to end of line 5
```

paragraph 3のword 5を、paragraph 4の先頭に移動する。

```
move word 5 of paragraph 3 to before paragraph 4
```

選択領域の内容を、document 1の最後に移動する。

```
move selection of document 1 to end of document 1
```

または、

```
tell document 1  
  move selection to end of it  
end tell
```

find 命令

```
find string -- 検索文字列
  in location reference -- 検索先のdocumentを指定する。
  [case sensitive boolean] -- 大文字、小文字を区別するかどうか?
  [zenkaku sensitive boolean] -- 全角、半角を区別するかどうか?
  [entire word boolean] -- ワード単位で検索するかどうか?
  [wrap around boolean] -- 最後まで行って見つからなければ、
                          先頭にもどって検索する。
  [start at top boolean] -- documentの先頭から検索を開始する。
  [reverse search boolean] -- 逆方向(文末から文頭)へ検索する。
  [grep boolean] -- 検査文字列を正規表現として解釈する。
  [batch boolean] -- 一括検索するかどうか。
Result変数: boolean -- 見つかったときはtrue, そうでないときはfalseをかえす。
```

指定の文字列を in パラメータで指定したオブジェクトのなかで検索します。

in パラメータでdocumentを指定すると現在のカーソル位置から検索します。inパラメータでdocument以外のselection、paragraph、lineなどのオブジェクトを指定すると、そのオブジェクトの範囲内で検索を行います。

オプションパラメータを指定を省略すると、JCheckerの既定値で検索が行われます。

batchパラメータを指定すると、一括検索が行われ結果はマークリストにマークされます。

document 1について、文字列"JChecker"をワード単位で大文字小文字を区別せずに検索します。

```
find "JChecker" in document 1 with word without case sensitive
```

または、

```
tell document 1
  find "JChecker" in it with word without case sensitive
end tell
```

tell documentのなかでは「it」でdocumentを指定することに注意してください。

document 1の選択領域のなかで、文字列"JChecker"を検索します。

```
tell document 1
  find "JChecker" in selection with entire word without case sensitive
end tell
```

document 1の選択領域のなかで、文字列"JChecker"を一括検索します。

```
tell document 1
  find "JChecker" in selection with batch
end tell
```

replace 命令

```
replace string -- 検索文字列
to string-- 置換文字列
in reference -- 検索対象となるオブジェクト
[case sensitive boolean] -- 大文字、小文字を区別するかどうか?
[zenkaku sensitive boolean] -- 全角、半角を区別するかどうか?
[entire word boolean] -- ワード単位で検索するかどうか?
[grep boolean] -- 検査文字列を正規表現として解釈する。
Result: small integer -- 置換した文字列数
```

in パラメータのオブジェクトを対象として、文字列の置き換えを行います。in パラメータのオブジェクトとしては、document、selection、paragraph、lineなどを指定できます。

オプションパラメータを指定を省略すると、JCheckerの既定値で検索が行われます。

document 1の選択領域内の文字列"JChecker"を"JChecker3.0"に置き換えます。

```
tell document 1
  replace "JChecker" to "JChecker3.0" in selection
end tell
```

document 1内の文字列"JChecker"のすべてを"JChecker3.0"に置き換えます。

```
tell document 1
  replace "JChecker" to "JChecker3.0" in it
end tell
```

mark 命令

```
mark reference -- bookmark listへマークするオブジェクト
```

referenceで指定したオブジェクトをブックマークリストへマークします。

line 10 をマークします。

```
mark line 10
```

count 命令

```
count reference -- カウントの対象となるオブジェクト
each type class -- カウントするときの単位
Result: integer -- カウント結果
```

指定したオブジェクトのサイズを指定の単位でResult変数にかえします。

line1のワード数をかえします。以下の2行は同じ動作をします。

```
count words of line 1
```

```
count line 1 each word
```

選択領域の文字数をカウントします。

```
count characters of selection of document 1
```

exists 命令

exists reference -- 問い合わせするオブジェクト

Result: boolean -- 存在するときはtrue,しないときはfalseが返る

指定のオブジェクトが存在しているかどうかを問い合わせます。存在していればresult変数にtureを、そうでないときはfalseをかえします。

文書 1 が存在するかどうか問い合わせる。

```
exists document 1
```

(注) existsコマンドを、byte, character, word, line, paragraph, style run, picture, movie, aliasFile の各オブジェクトについて行くと、オブジェクトの存在の有無にかかわらず、常にtrueが帰ってきますので使用しても意味がありません。

cut 命令

cut reference -- カットの対象となるオブジェクト名

選択領域のカットし、その内容をクリップボードへコピーします。

オブジェクトを指定しているときは、そのオブジェクトを削除し、クリップボードへコピーします。スタイル情報もコピーされます。

(注) JChecker がバックグラウンドのときは、削除は行われますがクリップボードへのコピーは行われません。JCheckerをactivateしてから行ってください。

document 1の選択領域をカットします。

```
cut document 1
```

document 1のline 4をカットする。

```
cut line 4 of document 1
```

copy 命令

copy reference -- コピーの対象となるオブジェクト

選択領域の内容をクリップボードへコピーします。

(注)JChecker がバックグラウンドのときはコピーできません。JCheckerをactivateしてから行ってください。

document 1 の選択領域をクリップボードへコピーします。

```
copy document 1
```

オブジェクトにテキストデータを指定しているときは、toで指定しているコピー先へコピーします。

line 5 の word 3 を文字列"abcdef"で置き換える。

```
copy "abcdef" to word 3 of line 5
```

文字列"abcdef"を、選択領域へコピーする。

```
copy "abcdef" to selection
```

paste 命令

paste reference

クリップボードの内容を、選択領域へペーストします。referenceには、document以外のオブジェクトは指定できません。

```
paste document 1
```

select 命令

select reference -- 選択するオブジェクト

指定のオブジェクトを選択します。

1行目から5行目までを選択する。

```
select line 1 thru 5
```

カーソル移動を行うときは、select命令ではなくset selection 命令で行います。

```
set selection to word 3 of line 10
```

undo / redo 命令

```
undo reference
```

```
redo reference
```

前の操作のundo(取り消し)またはredo(やり直し)を行います。referenceには、document以外のオブジェクトは指定できません。

```
undo document 1
```

change case 命令

```
change case reference  
to upper/lower/capitalize
```

referenceで指定したオブジェクト内の半角アルファベットを大文字化または小文字化します。パラメータにcapitalizeを指定すると、語の先頭だけを大文字にします。

文書1内の半角アルファベットをすべて大文字にする

```
change case document 1 to upper
```

文書2の選択範囲の半角アルファベットをすべて小文字にする

```
change case selection of document 2 to lower
```

preDialog / postDialog 命令

preDialog/postDialog -- パラメータはなし

マクロのなかでファイルダイアログなどを表示させた後JCheckerに戻ったとき、JCheckerのウィンドウ選択が乱れてしまう場合があります。このようなときは、ダイアログを表示させるスクリプトの前にpreDialog命令を実行し、ダイアログのスクリプトの後でpostDialog命令を実行すると、マクロ終了後のウィンドウ選択の乱れを防ぐことができます。

フォルダ選択ダイアログの前後でpreDialog/postDialog命令を実行する

```
preDialog
try
  copy (choose folder with prompt "調べるフォルダは?") to checkFolder
on error
  postDialog
  error number -128
end try
postDialog
```

上記のスクリプトでは、try文を使用して選択ダイアログでキャンセルボタンを押したとき（エラーのとき）も、postDialog命令を実行し終了するようにしています。

suspend 命令

suspend -- パラメータはなし

マクロから他のアプリケーションを起動し作業をした後、JCheckerに戻ったときに日本語入力ができなくなってしまったり、間接変換モードになってしまう場合があります。このような不具合を防ぐため、他のアプリケーションを起動するスクリプトの直前で、この命令を実行してください。JCheckerへ帰ったときの日本語入力の乱れを防ぐことができます。

JCheckerの書類1で編集集中のHTMLをブラウザに表示させるためのスクリプト。10行目でブラウザを起動するまえにsuspendを実行している。

```
tell application "JChecker3"
  activate
  save document 1
  copy fileSpec of document 1 to FileName
  set NewName to FileName as text
  set text item delimiters of AppleScript to ":"
  set NewName to every text item of NewName
  set text item delimiters of AppleScript to "/"
  set fileToView to "file:/// " & NewName
  suspend
  OpenURL fileToView
end tell
```

updateMacroMenu 命令

updateMacroMenu -- パラメータはなし

マクロメニューの再構築を行います。

スクリプトでマクロメニューフォルダ内のマクロファイルを追加・削除したときに、このコマンドを実行すると新しいマクロメニューフォルダの内容がマクロメニューに反映されます。

inhibit redraw / allow redraw 命令

inhibit redraw -- パラメータに対象ドキュメントを指定する

allow redraw -- パラメータに対象ドキュメントを指定する

指定の文書オブジェクトのウィンドウの書き換えを禁止 (inhibit redraw)、または禁止した書き換えを再開 (allow redraw) する。頻繁に画面書き換えが発生するようなスクリプトを書き換えを禁止するとその実行速度が 2 倍近く速くなります。

たとえば、サンプルマクロ「出現数のカウント」のrepeat命令部分を以下のように書き換え禁止にすると、find命令で文字列が見つかるたびに画面が書き変わることがなくなるので処理がたいへん速くなります。

```
.....
inhibit redraw
repeat while (find theString in it without wrap around) = true
    copy xxx + 1 to xxx
end repeat
allow redraw
.....
```

openURL 命令

openURL reference -- URL string

openURL命令は、指定のURLのリモートファイルを開きます。

```
tell application "JChecker3"
    openURL "http://www.artman21.net"
end tell
```

check 命令

```
check reference --文法チェックするドキュメント
    chime boolean -- 結果をサウンドで知らせるかどうか
Result: integer -- エラー数
```

指定のHTMLドキュメントの文法チェックを行い、そのエラー数を返します。chime パラメータで結果をサウンドで知らせるかどうかも指定できます。

document 1 を、サウンドなしで文法チェックする。

```
tell application "JChecker3 "
    check document 1 without chime
end tell
```

linkCheck 命令

```
linkCheck reference --リンクチェックするドキュメント
    chime boolean -- 結果をサウンドで知らせるかどうか
    local boolean -- ローカルチェックをするかどうか
    remote boolean -- リモートチェックをするかどうか
Result: なし
```

指定のHTMLドキュメントのリンクチェックを行います。オプションパラメータとして、chime, local, remoteを指定できます。check 命令と異なりエラー数は返りません。

document 1 のリンクをチェックする。

```
tell application "JChecker3 "
    linkCheck document 1
end tell
```

minimize 命令

```
minimize reference --最小化するドキュメント
```

指定のHTMLドキュメントの最小化します。

document 1 を、最小化する。

```
tell application "JChecker3 "  
    minimize document 1  
end tell
```

beautify 命令

beautify reference --整形するドキュメント

指定のHTMLドキュメントの整形します。

document 1 を、整形する。

```
tell application "JChecker3 "  
    beautify document 1  
end tell
```

correct 命令

correct reference --修正するドキュメント

Result: integer -- 修正箇所数

指定のHTMLドキュメントのエラーを自動修正します。

document 1 を、自動修正する。

```
tell application "JChecker3 "  
    correct document 1  
end tell
```

checkAll 命令

checkAll reference --修正するドキュメント

[linkDepth integer] -- リンクしている書類の深さ

[validation boolean] -- 文法チェックを同時するときはtrueを指定する

指定のHTMLドキュメントとそのリンク先のHTMLを含めてチェックします。

document 1とそのリンク先 を文法とリンクの両方をチェックする。

```
tell application "JChecker3 "  
    checkAll document 1 with validation  
end tell
```

recolor 命令

```
recolor reference --再カラーリングするドキュメント
```

指定のHTMLドキュメントを再カラーリングします。

document 1 を、再カラーリングする。

```
tell application "JChecker3 "  
    recolor document 1  
end tell
```

GetTagName 命令

```
GetTagName reference --調べるドキュメント  
Result: string -- エlement名
```

指定のHTMLドキュメントのカーソル位置のタグのElement名を返します。カーソルがタグのなか
にないときは、空の文字列を返します。

GetAtrbValue 命令

```
GetAtrbName reference --調べるドキュメント  
    atrbName string -- 属性名  
Result: string -- 属性値
```

指定のHTMLドキュメントのカーソル位置のタグについて、指定の属性の属性値を返します。該当の
属性がないときは、空の文字列を返します。

SelectElement 命令

```
SelectTag reference --対象ドキュメント
```

指定のHTMLドキュメントのカーソル位置のエレメント全体（開始タグから終了タグまで）を選択します。

SelectTag 命令

```
SelectTag reference --対象ドキュメント
    Result: integer -- タグでないとき 0、開始タグのとき 1、終了タグのとき 2
```

指定のHTMLドキュメントのカーソル位置のタグを選択します。選択すべきタグがないときは0を、選択したタグが開始タグのときは1を、終了タグのときは2を返します。

SelectAttrb 命令

```
SelectAttrb reference --対象ドキュメント
    attrbName string -- 属性名
    [valueOnly boolean] -- 属性値だけ選択するかどうか
    Result: integer -- 0、1、2のいずれかの値
```

指定のHTMLドキュメントのカーソル位置のタグについて、指定された属性を選択します。指定の属性がなかったときは0を、属性名と属性値が選択されたときは1を、属性が省略されていて属性値だけが選択されたときは2を返します。

GetColor 命令

```
GetColor
    [oldColor string] -- 現在のHTMLカラー文字列 ( e.g. red, #C0C0C0 ... )
    Result: string -- the HTML RGB string
```

カラーパレットを表示し選択結果を、HTMLの16進文字列（#ccooなど）として返します。パラメータoldColorに現在の設定色を渡すこともできます。

GetRelativePath 命令

```
GetRelativePath
    target file specification -- ファイルのfile Spec
    Result: international text -- documentの位置から計算したtargetファイルの相対パス (URL)
```

targetに求めたいファイルのfileSpecを指定すると、targetファイルの相対パスをURLで返します。documentにファイルが割り当てられていないときは、targetファイルの絶対パスを返します。

SetAtrb 命令

```
SetAtrb reference --対象ドキュメント
    atrbName string -- 属性名
    atrbValue string -- 属性値
```

指定のHTMLドキュメントのカーソル位置のタグについて、指定された属性とその値をセットします。すでに同じ属性値があるときは値だけ修正します。属性がなかったときは、属性とその値をタグに追加します。

HTMLColorToRGB 命令

```
HTMLColorToRGB string --HTMLカラー文字列
    Result: RGB color -- RGB値
```

HTMLカラー文字列 (red, #C0C0C0 ...など)をRGB値に変換して返します。

GetDtdName 命令

```
GetDtdName reference --対象ドキュメント
    Result: string -- 辞書名
```

指定されたHTMLドキュメントの辞書名を返します。まだ辞書名が解析されていないときは、HTMLのDOCTYPE宣言を解析してから、辞書名を返します。