－EZweb 仕様書－

【EZweb 全般】Web ページ記述ガイド

Version-1.0

2012-03-30
KDDI Corporation

ーはじめにー

本書は、お客様が EZweb コンテンツを制作し、KDDI 株式会社（以下「当社」といいます。）が提供するネットワークサービスである「EZweb」において当該コンテンツに係るサービスを提供するための仕様、必要な事項、留意すべき事項等を記載したものです。以下の注意事項に同意のうえ本書をご利用下さい。

＜注意事項＞

- 本書は、その内容の正確性、有用性、完全性、確実性、瑕疵の不存在等を保証するものではありません。
- 本書は、必要に応じて、予告なく変更される場合がありますので、予めご了承下さい。
- 当社の許諾を得ることなく、本書の内容の全部または一部を複製、譲渡、貸与等することはできません。
- お客様が本書を利用し、または利用できなかったことにより発生したお客様および第三者の損害（データの破損、業務の中断、営業情報の損失などによる損害を含みます。）について、当社は一切責任を負いませんので、予めご了承下さい。
- お客様が、本書の利用に関連して第三者に対して損害を与え、または当該第三者との間で紛争を生ぜしめた場合、お客様は、自己の責任と費用負担でこれを解決し、当社には何らの迷惑をかけないものとします。万一、当社が当該紛争等により損害を被った場合には、お客様は、当社に対し、その損害を賠償するものとします。

ー関連仕様書ー

・『EZweb 仕様書ー【EZweb 全般】EZweb コンテンツ制作ガイド』
・『EZweb 仕様書ー【位置情報サービス】技術仕様書[EZ ナビ_簡易位置情報]』

ー目次ー

1. Web ページ制作全般 10
1.1. Web サーバの準備 10
1.2. ブラウザの種類と記述言語 10
1.3. ブラウザによる表示の差異 10
1.4. Web ページのサイズ 11
(1) MAX-PDU サイズ 11
(2) 1 ページあたりのデータサイズの目安 11
(3) 1 ページあたりの表示領域及び画像の多用に関する注意 12
1.5. 文字コード 12
1.6. 有効表示領域 12
1.7. 取り扱い可能な URI 長 13
1.8. POST 可能なデータサイズ 13
1.9. マルチメディアデータのダウンロード 14
1.10. マルチメディアデータのインライン再生 14
1.11. 各種サービスとの連携 15
(1) 位置情報の取得 15
1.12. <object>要素を利用したダウンロード動作 15
2. XHTML Basic による記述のポイント 16
2.1. XHTML とは 16
2.2. XHTML 記述の規則 16
2.3. XHTML による記述上のテクニック 17
(1) ハイパーリンク選択時のソフトキーラベル文字列の指定方法 17
(2) ソフトキーの機能割り当てと表示文字列の設定方法 17
(3) 音声通話連携 18
(4) 指定したメールアドレスへのメールを作成するハイパーリンク記述方法 18
(5) お気に入り登録 20
(6) 絵文字の利用 20
(7) 特殊文字の表示 20
(8) format 属性による入力文字種の指定 21
(9) istyle 属性 22
(10) アクティビティの利用方法 22
3. XHTML Basic リファレンス 23

3.1. 宣言文 23
(1) XML 宣言 23
(2) DOCTYPE 宣言 23
3.2. 汎用属性 23
(1) class="クラス名" 【WAP2.0 対応ブラウザ差異あり】 23
(2) id="名前" 【WAP2.0 対応ブラウザ差異あり】 24
(3) lang="言語コード" 24
(4) style="スタイルシート" 24
(5) title="補足情報" 24
3.3. 色指定 24
3.4. 基本要素 25
(1) <body> 25
(2) <head> 25
(3) <html> 26
(4) <title> 26
(5) <meta> 26
(6) <!-- --> 27
3.5. 文書構造/フォント関連要素 27
(1) <blockquote> 27
(2)
 【WAP2.0 対応ブラウザ差異あり】 27
(3) 28
(4) <h1>,<h2>,<h3>,<h4>,<h5>,<h6> 29
(5) <p> 29
(6) <pre> 30
(7) 30
3.6. リスト関連要素 31
(1) <dd> 【WAP2.0 対応ブラウザ差異あり】 31
(2) <dl> 32
(3) <dt> 32
(4) 33
(5) 【WAP2.0 対応ブラウザ差異あり】 34
(6) 36
3.7. 罫線/テーブル関連要素 36
(1) <caption> 36
(2) <table> 【WAP2.0 対応ブラウザ差異あり】 37
(3) <td> 38

(4) <th> 39
(5) <tr> 40
3.8. リンク関連要素 41
(1) <a> 【仕様制限あり/WAP2.0 対応ブラウザ差異あり】 41
(2) <link> 43
3.9. マルチメディア関連要素 44
(1) <img> 【WAP2.0 対応ブラウザ差異あり】 44
(2) <object> 【WAP2.0 対応ブラウザ差異あり】 45
(3) <param> 46
3.10. フォームデータ入力関連要素 47
(1) <input> 【仕様制限あり/WAP2.0 対応ブラウザ差異あり】 47
(2) <form> 【WAP2.0 対応ブラウザ差異あり】 49
(3) <label> 【仕様制限あり/WAP2.0 対応ブラウザ差異あり】 52
(4) <option> 53
(5) <select> 54
(6) <textarea> 55
3.11. その他の要素 56
(1) <q> 【WAP2.0 対応ブラウザ差異あり】 56
4. EZ ブラウザがサポートするその他の要素 57
4.1. 基本要素 57
(1) <style> 57
4.2. 文書構造/フォント関連要素 57
(1) <blink> 【WAP2.0 対応ブラウザ差異あり】 57
(2) 58
(3) <font> 58
(4) <marquee> 【WAP2.0 対応ブラウザ差異あり】 59
(5) <u> 60
(6) <del> 【WAP2.0 対応ブラウザ差異あり】 61
(7) <s> 【WAP2.0 対応ブラウザ差異あり】 61
(8) <ins> 【WAP2.0 対応ブラウザ差異あり】 62
(9) <strike> 【WAP2.0 対応ブラウザ差異あり】 62
4.3. リスト関連要素 63
(1) <dir> 【WAP2.0 対応ブラウザ差異あり】 63
(2) <menu> 64
4.4. 罫線/テーブル関連要素 65

(1) <hr> 【WAP2.0 対応ブラウザ差異あり】 65
4.5. マルチメディア関連要素 66
(1) <bgsound> 66
4.6. フォームデータ入力関連要素 67
(1) <optgroup> 67
4.7. その他の要素 67
(1) <frameset> 【WAP2.0 対応ブラウザ差異あり】 67
(2) <frame> 【WAP2.0 対応ブラウザ差異あり】 68
(3) <noframes> 68
5. CSS リファレンス 69
5.1. CSS の利用方法 69
(1) <link>要素により外部 CSS ファイルを参照する方法 69
(2) ヘッダで宣言する方法 69
(3) 適用したい要素に直接書き込む方法 69
5.2. セレクタの種類 70
(1) Universal セレクタ 70
(2) Class セレクタ 70
(3) ID セレクタ 70
(4) Type セレクタ 71
(5) Link Pseudo Class セレクタ 71
(6) Descendent セレクタ 72
(7) Child セレクタ 72
5.3. 単位の設定 72
5.4. 背景に関連するプロパティ 73
(1) background-attachment 【WAP2.0 対応ブラウザ差異あり】 73
(2) background-color 【WAP2.0 対応ブラウザ差異あり】 73
(3) background-image 73
(4) background-position 【WAP2.0 対応ブラウザ差異あり】 73
(5) background-repeat 74
5.5. 文字体裁に関連する属性 74
(1) line-height 74
(2) text-align 【WAP2.0 対応ブラウザ差異あり】 74
(3) text-decoration 75
(4) text-indent 75
(5) white-space 75

5.6. サイズやレイアウトに関連する属性 76
(1) height 76
(2) vertical-align 76
(3) width 【WAP2.0 対応ブラウザ差異あり】 77
5.7. リストに関連する属性 77
(1) list-style-image 【WAP2.0 対応ブラウザ差異あり】 77
(2) list-style-position 【WAP2.0 対応ブラウザ差異あり】 78
(3) list-style-type 【WAP2.0 対応ブラウザ差異あり】 79
5.8. テキスト 80
(1) word-spacing 【WAP2.0 対応ブラウザ差異あり】 80
(2) letter-spacing【WAP2.0 対応ブラウザ差異あり】 80
(3) text-indent【WAP2.0 対応ブラウザ差異あり】 80
5.9. フォント 80
(1) font-size 【WAP2.0 対応ブラウザ差異あり】 80
5.10. その他の属性 81
(1) border-top/-bottom/-left/-right 【仕様制限あり/WAP2.0 対応ブラウザ差異あり】 81
(2) color 【WAP2.0 対応ブラウザ差異あり】 82
(3) display 【WAP2.0 対応ブラウザ差異あり】 82
(4) margin-top/-bottom/-left/-right 82

6. HDML による記述のポイント 83

6.1. HDML による記述上のテクニック 83
(1) ブラウザの自動折り返し機能 83
(2) 音声通話連携 83
(3) E メール連携 84
(4) 絵文字の利用 85
(5) FORMAT 属性による入力文字種の指定 85
(6) PUBLIC 属性によるデッキへのアクセス制限 86
(7) お気に入り登録 86
(8) カードの名前 88
6.2. WAP2.0 対応ブラウザに向けて HDML ページを記述する際の注意事項 88
(1) <A>要素の直前の改行処理 88
(2) <A>要素内への<IMG>要素挿入によるリンクの分割 89
(3) <A>要素における NOOP タスク指定時の文字列リンクの表示 89
(4) <ACTION>要素における NOOP タスク指定時のソフトキーラベルの表示 89
(5) RETURN/CANCEL タスクにおける CLEAR＝TRUE の動作 89

(6) RETURN/CANCEL タスクにおける Referer フィールドの送信 89
(7) RETURN/CANCEL タスクにおける POST メソッド指定 90
(8) RETURN タスクによる変数受け渡し時の「VARS 属性」の利用 90
(9) 「RETVALS 属性」を利用した変数受け渡し動作における「;」使用 90
(10) <ACTION>要素におけるタスク無指定時のデフォルト動作 90
(11) <NODISPLAY>カードでの「CANCEL 属性」省略時の遷移先の相違 91
(12) <NODISPLAY>でのデッキ全体指定を用いた遷移先指定 91
(13) キャッシュオペレーション機能の利用 91
(14) HDML3.1 のサポート範囲 91
(15) DEST 属性に同一デッキ内カードを指定した場合の POST メソッドの利用 91
(16) <LINE>要素内での<TAB>利用時における文字列の折り返し 92
(17) 複数行に跨る文字列の先頭に<TAB>を挿入した場合の表示 92
(18) 「画像データ＋言語」のダイジェスト送信要求時のメソッド 92
(19) ダイジェスト送信における構成要素の順番 92
(20) <CHOICE>カード内への<A>要素挿入時における選択リスト表示 92
(21) <CHOICE>カードにおける「IDEFAULT=0」指定時の初期選択値 93
(22) <CE>要素「VALUE 属性」におけるエンコード済日本語文字列設定後の表示 93
(23) 変数名に利用可能な文字 93
(24) 位置情報取得時におけるアクティビティ利用と変数の引渡し 93
(25) <RIGHT>、<CENTER>要素を利用時のスペースの付加 93
(26) 「device:～」実行後のアクセスページにおける「PUBLIC 属性」の設定 94
(27) <A>要素内における「TASK 属性」の記述順序 94
(28) SSL サイトにおける Nodisplay カードの利用 94
(29) 「CHOICE カード」での<ACTION TYPE=ACCEPT～>と<ACTION TYPE=SOFT1～> 94
(30) Web サーバへのデータの送信 96
(31) NODISPLAY カードの利用時における NAME 属性の指定 97
(32) E メールから NODISPLAY カードによるダウンロードを実行する場合の動作 97
(33) ACCEPT キーに"noop"を指定した際の動作 97
(34) <LINE>要素利用時の動作 97

7. 【参考】WML による記述上の注意事項 98

7.1. XHTML における WML の記述方法 98
(1) <wml:anchor> 98
(2) <wml:exit> 98
(3) <wml:spawn> 99
7.2. WAP2.0 対応ブラウザでの動作差異 99

(1) <do>,<wml:do>,<p:do>.....99
(2) <reset>,<wml:reset>,<p:reset>.....99
(3) mode="nowrap"指定.....100
(4) type="onpick" 101
(5) 変数参照 102

1. Web ページ制作全般

1.1. Web サーバの準備

MIME メディアタイプ設定など Web コンテンツ提供のための環境の準備について、『【EZweb 全般】EZweb コンテンツ制作ガイド』でご確認ください。

1.2. ブラウザの種類と記述言語

現在、EZweb 対応端末に搭載されているブラウザの種類と、各ブラウザで閲覧可能な Web ページ記述言語は以下のようになります。

ブラウザ種類		バージョン	Web ページ記述言語の対応状況				
			XHTML Basic	CSS	WML 1.3	HTML 4.01	HDML(※1)
WAP2.0 対応ブラウザ	既存 EZ ブラウザ	1.0x	○	○	○	△	△
		1.1x	○	○	○	△	△
		1.13	○	○	○	△	△
		7.2	○	○	○	△	△
	新 EZ ブラウザ(※2)	7.2	○	○	○	△	-

○：サポート
△：ベストエフォートで閲覧可能
－：仕様制限動作により閲覧不可

※1： HDML で記述された Web ページは EZ サーバによる WML への変換を通じて閲覧可能となります。
※2： 新 EZ ブラウザの詳細は『【EZweb 全般】EZweb コンテンツ制作ガイド』をご参照ください。

1.3. ブラウザによる表示の差異

異なるバージョンのブラウザであっても基本的に同様の表示や動作となりますが、一部、若干の差異がある場合があります。

それらブラウザの表示/動作差異に関する情報については、以下の場所に記載されています。

Web ページの記述言語	閲覧するブラウザ	参照すべき情報
XHTML	WAP2.0 対応ブラウザ（既存 EZ ブラウザ、新 EZ ブラウザ）	・本書「2.XHTML Basic による記述のポイント」 ・本書「3.XHTML Basic リファレンス」 ・本書「4. EZ ブラウザがサポートするその他の要素」 ・本書「5.CSS リファレンス」 ※：バージョンにより仕様差分がある場合には、レファレンス各標題に「WAP2.0 対応ブラウザ差異あり」と記載しています。
HDML	既存 EZ ブラウザ	・本書「6.HDML による記述のポイント」

1.4. Web ページのサイズ

(1) MAX-PDU サイズ

Web ページのデータサイズ、及び、インライン再生するデータのサイズは、端末が 1 回の HTTP レスポンスで受信することができるデータ容量（＝MAX-PDU サイズ）の範囲内である必要があります。
MAX-PDU サイズは機種によって異なります。HTTP リクエストヘッダの「x-up-devcap-max-pdu フィールド」から取得できます。

> Web ページに画像や音声などのデータが貼り付けられている場合、それぞれのデータは別の HTTP レスポンスで受信されますので、それぞれの（言語変換後の）ファイルサイズが MAX-PDU サイズ内に収まっていれば受信することができます。
> 例えば、[HTML データ 5KB]に[PNG イメージ 5KB]＋[SMAF データ 5KB]といった貼付も可能です。

※:1 ページに複数の画像データを貼り付けた場合、端末によってはそのすべてを表示できない場合があります。
※：インライン再生可能な画像サイズ（横×縦ピクセル数）も機種によって異なります。

(2) 1 ページあたりのデータサイズの目安

言語変換時のページサイズの増加を考慮すると、1 ページあたりの最大容量の目安は以下のようになります。

配信する端末	Web ページ記述言語	
	XHTML	HDML
WAP2.0 対応端末	約 9KB	約 7.5KB

しかしながら、言語変換に際し、ページサイズが増加する割合は要素の種類や組合せによって大幅に異なります。
上記値はあくまで目安であり、範囲内のページの表示を保証するものではありません。
なお、HDML で記述する場合には 1 デッキあたりのカード数を減らし、余裕を持たせることがサイズオーバーの回避策となります。

(3) 1ページあたりの表示領域及び画像の多用に関する注意

１ページのデータサイズが MAX-PDU サイズの範囲以内の場合でも、アイコンやバナーなど画像を多用した場合には端末のメモリが大量に消費されることによるメモリ不足により、一部の画像が表示されなかったり、ページが最後まで表示されなくなったりする場合があります。
また、利用する文字数が多い場合にも、画面の表示領域が広くなることによりメモリが消費されページの一部が表示されなくなる場合があります。
なお、文字数に関しては、表示するフォントサイズによってもメモリの消費量が変化します。
標準サイズでは正常に表示されている場合でも、大きいサイズのフォントに変更した場合、文章の改行などにより画面の表示領域が広くなることによりメモリが多く消費されページが最後まで表示されなくなる場合があります。

端末の出荷状態である、標準サイズのフォントでは必ずコンテンツが表示されるようにして下さい。 なお、あらかじめ大きいサイズのフォントで表示されない事象が確認されている場合には、コンテンツ内に推奨フォントサイズを明記するなどの対応をお願いいたします。

メモリ不足によりページや画像が表示できなくなる現象は、画面の表示領域や画像の利用数などの複合的な条件が重なることで発生するため、定量的な推奨サイズや利用数を提示することができません。
従って、このような現象を回避するためには、文字数が多い場合はページを分割するなどして１ページあたりの表示領域を小さくします。また、画像が多い場合（特にアイコンの利用が多い場合）には、記号で代替するなどして画像の数を少なくします。

1.5. 文字コード

Web ページ記述には文字コード「Shift-JIS」を使用して下さい。
また、改行コードは「CR+LF」を使用して下さい。

※：HTTP リクエストヘッダで文字コードを指定する場合には「Content Type フィールド」を利用して下さい。

※：文字コード指定を省略している場合において、コンテンツの最初に現れる 2 バイト系コードが半角カタカナのとき、文字化けすることがあります。

1.6. 有効表示領域

EZweb ブラウジング中の画面領域は、機種、表示モードにより異なります。各機種の画面表示領域を把握したい場合には『技術情報 > 機種別情報一覧』をご参照ください。

1.7. 取り扱い可能な URI 長

取り扱える「URI の長さ」は、端末及び取り扱う場面で異なります。
なお、下記に示す「URI の長さ」は「http://」を含む、URL エンコード後の長さとなります。

		WAP2.0 対応端末
リクエスト可能な最大の URI の長さ （Get メソッドで送信可能な URI サイズ）		HTML･･･1024 バイト(※1) HDML････980 バイト
ユーザが URI のダイレクト入力によりサイトへアクセスする時に入力することができる最大の URI の長さ		256 バイト
EZ トップメニューの「お気に入り」	登録可能な URI の長さ	―
	登録可能なタイトルの長さ	―
ブラウザメニューの「マイ URL リスト」	登録可能な URI の長さ	―
	登録可能なタイトルの長さ	―
ブラウザメニューの「お気に入りリスト」	登録可能な URI の長さ	1,024 バイト（※2）
	登録可能なタイトルの長さ	40 バイト

※1：WAP2.0 対応ブラウザ 7.2 から 1024byte を超える URI を送信する場合、エラーメッセージが表示されます。
※2：ただし、Get メソッドで送信可能な URI サイズが上限となります。

1.8. POST 可能なデータサイズ

POST メソッドで BODY として Web サーバに送信可能な最大データサイズは以下の通りです。

WAP2.0 対応端末	
XHTML 記述の場合	HDML 記述の場合
65,536 バイト（※）	1,998 バイト

※：「WAP2.0 対応ブラウザ 1.1x」において、EZ メニューの「画面メモ」機能で XHTML のフォームを利用したページが保存された場合、当該ページから post メソッドで渡せるデータサイズは「1,000 バイト以下」となります。

1.9. マルチメディアデータのダウンロード

画像、動画、アプリなどの各種マルチメディアデータは、Web ページの記述に従ってダウンロードすることができます。EZweb におけるマルチメディアデータのダウンロードスキームには「EZget 方式」と「オブジェクトダウンロード方式」の 2 種類の方式があります。

なお、記述言語により利用可能なダウンロード方式が異なります。

Web ページ記述言語	記述可能な（利用可能な）ダウンロード方式
XHTML	オブジェクトダウンロード方式 EZget 方式
HDML	EZget 方式

ダウンロード可能なデータの種類やダウンロード記述方法については『技術情報 > ダウンロード CGI』をご参照ください。

1.10. マルチメディアデータのインライン再生

XHTML では<img>要素、HDML では<IMG>要素を利用して、Web ページ内で画像データや音声データなどを再生（インライン再生）することができます。ただし、ユーザ操作によって「添付データ再生＝不可」に設定されている場合には再生することができません。

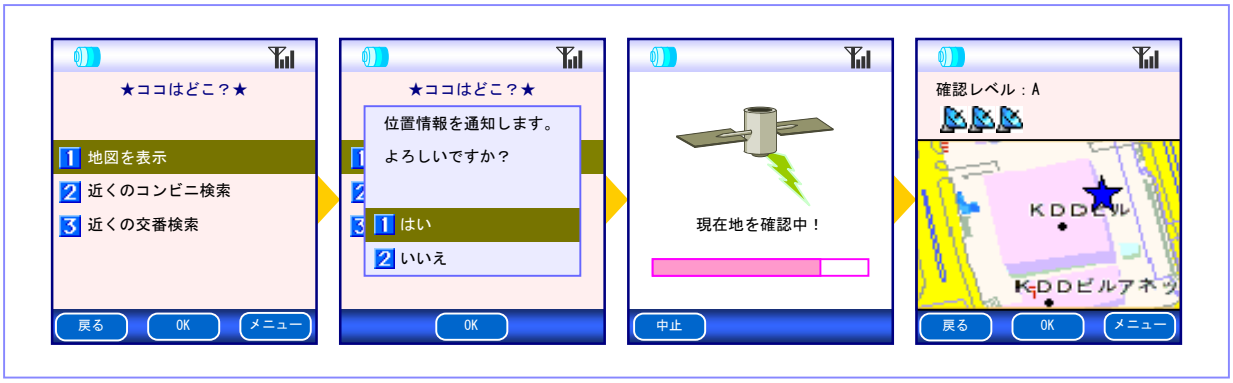
また、WAP2.0 対応ブラウザでは XHTML の<img>要素の copyright 属性により、インライン再生中データの「保存」を禁止することもできます。

※：インライン再生可能なデータサイズは MAX-PDU サイズに依存します。
※：1 ページに複数の画像データを貼り付けた場合、端末によってはそのすべてを表示できない場合があります。
※：インライン再生可能な画像サイズ（横×縦ピクセル数）も機種によって異なります。
※：インライン再生するデータには CRC チェックを付加しないで下さい。

1.11. 各種サービスとの連携

(1) 位置情報の取得

Web ページの記述により、Web ページから端末に対して EZ ナビによる位置情報の取得要求を行い、結果として端末が取得した位置情報を Web サイトに通知させることができます（記述方法は、『技術情報 > 位置情報』参照）。



また、Web ページの記述により、簡易位置情報機能によって端末が保持している位置情報を、Web サイトに通知させることもできます（記述方法は、『技術情報 > 位置情報』参照）。

1.12. <object>要素を利用したダウンロード動作

Web ページ上に<object>要素が記述されていた場合、WAP2.0 対応ブラウザ 1.0x/1.1x では、オブジェクトダウンロードと認識しますが、WAP2.0 対応ブラウザ 1.13 からは、オブジェクトダウンロードに加え、メディアデータ（画像／音声）再生（インライン再生／インタラクティブ再生）の要素としても認識します。

WAP2.0 対応ブラウザ 1.13 からは、<object>要素内に width/height 属性が指定されている場合は「インライン再生」、declare 属性が指定されている場合は「インタラクティブ再生」と判断します。width/height 及び declare の属性の記述がない場合、ダウンロードと判断します。
なお、width/height 属性及び declare 属性は、WAP2.0 対応ブラウザ 1.13 以降で有効な属性です。

<object>要素の属性の有無		ブラウザの動作	
declare 属性	width/height 属性※注	WAP2.0 対応ブラウザ 1.0x/1.1x	WAP2.0 対応ブラウザ 1.13 以降
有	無	動作保証外	インタラクティブ再生
無	有		インライン再生
無	無	オブジェクトダウンロード	オブジェクトダウンロード

※注：インタラクティブ再生及びインライン再生を指定する場合は、width/height 属性を必ずセットで指定します。
（片方のみの属性の指定は不可）

オブジェクトダウンロード及び Flash Lite コンテンツ再生の記述方法については『技術情報 > ダウンロード CGI』をご参照ください。

2. XHTML Basic による記述のポイント

2.1. XHTML とは

XHTML1.0 は HTML4.01 で使用できる要素をそのまま利用して、XML の規則に従って再定義した記述言語です。記述方法に多少の差異はありますが、基本的には HTML4.01 と同様に記述することができます。
EZweb 対応端末では携帯電話向けに用意された XHTML のサブセットである XHTML Basic をサポートしています。

2.2. XHTML 記述の規則

§ 要素名と属性名は必ず小文字で記述する

XML では大文字と小文字が区別されます。
XHTML1.0 の DTD では、要素名と属性名はすべて小文字で定義されており、必ず小文字で記述する必要があります。

§ タグは省略できない

開始タグと終了タグを記述する必要があります。

§ 属性値は省略できない

属性は「属性名＝"値"」の形式で指定します。
「checked」や「selected」などの属性でも「checked="checked"」「selected="selected"」のように指定する必要があります。

§ 属性値は必ず引用符「" "」で囲む

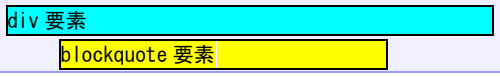
属性値は引用符「" "」で囲む必要があります。

§ タグのオーバーハングはできない

次のようにオーバーハングをしてはいけません。

× : <blockquote> あいう</blockquote>えお

○ : <blockquote>あいう</blockquote>えお



§ 空要素の記述

空要素も明示的に閉じる必要があります。

 …「<br」と「/>」の間に半角のスペースを挿入

2.3. XHTML による記述上のテクニック

(1) ハイパーリンク選択時のソフトキーラベル文字列の指定方法

ハイパーリンクが選択されている時にソフトキーラベルに表示される文字列を指定するには、リンク記述部分で title 属性を利用します。

```
<a href="next.html" title="次へ">次のページはこちら</a>
```

(2) ソフトキーの機能割り当てと表示文字列の設定方法

ソフトキーに機能を割り当て、表示文字列を設定するには、<body>～</body>内において<wml:～>記述を利用します（※本書「7. 【参考】WML による記述上の注意事項」もご参照ください）。

■ SOFT1 キーの定義例

```
<wml:do type="options" label="前へ">
  <wml:go href="before.html" />
</wml:do>
```

■ ACCEPT キーの定義例

```
<wml:do type="accept" label="次へ">
  <wml:go href="next.html" />
</wml:do>
```

※：ACCEPT キーに"noop"を指定した際の動作に関する注意
「WAP2.0 対応ブラウザ 1.1x/1.13」では、ACCEPT キーの動作として「noop」を指定(task=noop)した場合に、ページの上下にカーソルアウトする（フォーカスを外す）ことができません（詳細は本書「6.2.(33) ACCEPT キーに"noop"を指定した際の動作」参照）。

(3) 音声通話連携

以下のようなリンク記述から指定の電話番号へ音声通話の発信へ誘導することができます。
なお、この場合、EZweb の接続は一旦切断されます。

■ 電話番号として利用可能な文字

半角数字、-、#、*、P　（※：すべて半角）

※：「P」はポーズ機能で使用します。「P」の位置で一旦ポーズし、発信キーを押下することで続く番号を発信します。

※：電話番号に「クイックダイヤル」（＃ダイヤル）を指定する場合、その番号がすべての地域会社で利用可能とは限らないのでご注意ください。利用できない地域会社がある場合は、その地域会社を Web ページに明記して下さい。

■ 記述例

```
<a href="tel:XXXXXXX">電話はこちら</a>
```

(4) 指定したメールアドレスへのメールを作成するハイパーリンク記述方法

Web ページのリンク記述により、メーラーを起動し、指定のメールアドレスに対するメールの作成画面へ移行させることができます。

XHTML による記述では、To、Cc、Bcc、Subject、Body をパラメータとして指定することができます。

※：ハイパーリンク記述の最大長については、本書「1.7. 取り扱い可能な URI 長」をご参照ください。

■ 基本的な記述例

```
<a href="mailto:xxx@example.com">メールはこちら</a>
```

■ 複数送信先/件名/本文を指定する場合の記述例

```
<a href="mailto:xxx@example.com,yyy@example.com?subject=foo&cc=abc@example.com&bcc=xyz@example.com&body=foobar">メールはこちら</a><br />
```

■ 注意事項（URL エンコード）

「Subject」及び「Body」内の「日本語（2 バイトコード）」と「Unsafe Character（※）」は、メーラーで読み込む際にデコード処理されますので、あらかじめエンコードされていなければなりません。また、「半角カナ」は利用できません。

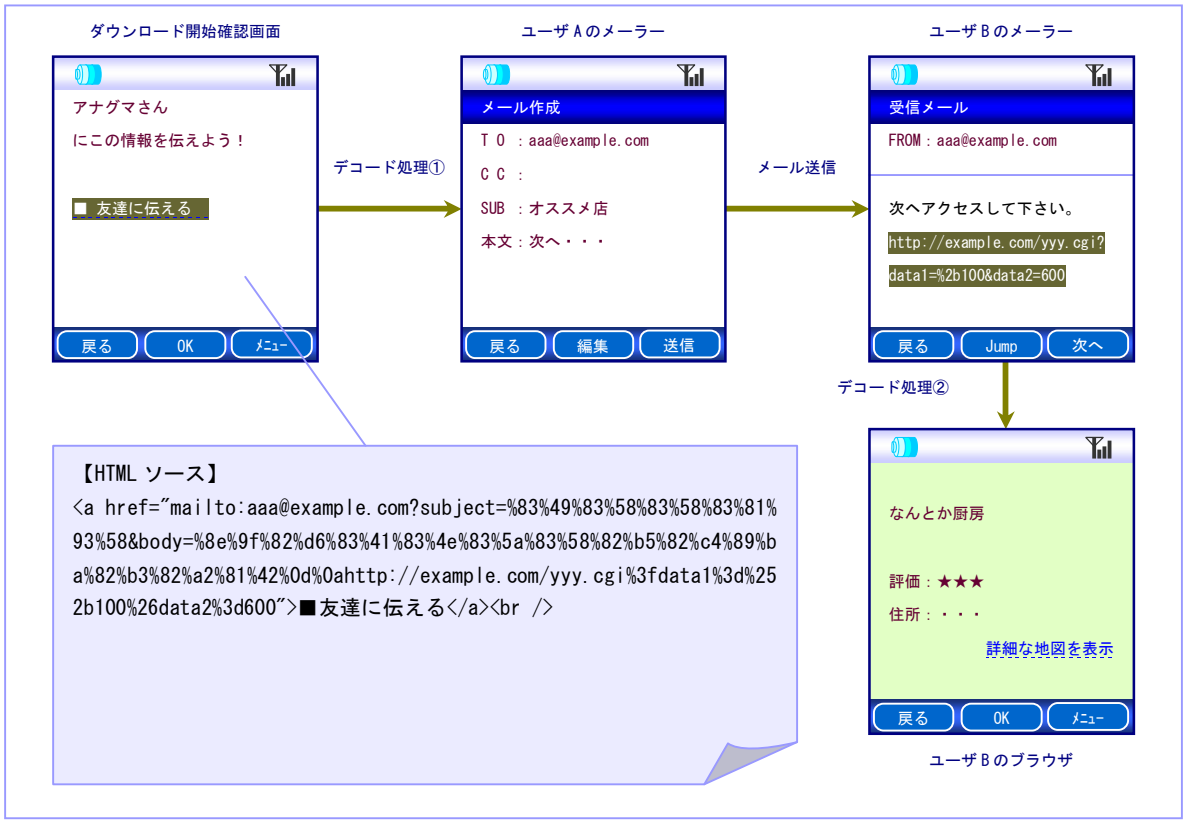
※：Unsafe Character：次の 18 キャラクタを Unsafe Character とします。

%、<、>、″、#、{、}、[、]、¥、|、^、`、（半角スペース）、+、?、&、=

例えば下記図のように「data1=+100」、「data2=600」をパラメータとして渡す URI を「body」に指定したい場合、その URI には「Unsafe Character」である「?」「&」「=」が含まれますので、これらをあらかじめコンテンツソース上でエンコードしておく必要があります。

さらに、「Unsafe Character」は、Web サーバ側でもデコード処理されます。つまり、下記例においてパラメータ付 URI の記述されたメールを受け取ったユーザ B が、URI へアクセスする際には、接続先の Web サーバへは GET メソッドにより「data1=%2b100&data2=600」が渡され、WEB サーバ側では、「data1=+100」、「data2=600」と認識されます。

従って mailto において指定する URI パラメータ内の「+」等の「Unsafe-Character」はコンテンツソース上で二重にエンコードしておく必要があります。（下記例では、「+」→「%2b」→「%252b」）



(5) お気に入り登録

■ リンク記述によるお気に入り登録への誘導

下記のような「device:」で始まる URI を用いた記述により、指定した URI を「お気に入り」へ登録するよう誘導することができます。ただし、パラメータ付きの URI のお気に入り登録については動作を保証できません。

```
<a href="device:home/bookmark?url=http://example.com/&title=日本語タイトル">マーク！</a>
```

■ お気に入り登録の禁止

明示的に禁止しない限り XHTML で記述された Web ページはお気に入りに登録することができます。
お気に入り登録を禁止するには<head>～</head>内において、以下のように<meta>要素を記述します（※本書「7.【参考】WML による記述上の注意事項」もご参照ください）。

```
<meta name="vnd.up.markable" wml:forua="true" content="false" />
```

なお、Web サーバに各種データを Post して表示されるページについては、上記記述を用いてお気に入り登録を禁止するようにして下さい。

■ 別の URI でお気に入り登録させる方法

ブラウザで表示されている Web ページとは別の URI をお気に入りとして登録させたい場合には、<head>～</head>内において、以下のように<meta>要素を記述します（※本書「7.【参考】WML による記述上の注意事項」もご参照ください）。

```
<meta name="vnd.up.bookmark" wml:forua="true" content="http://www.example.com" />
```

(6) 絵文字の利用

<img>要素の「localsrc」属性に絵文字の「絵文字番号」を指定することで、Web ページ上で絵文字を表示させることができます。絵文字は端末に実装しているため画像データよりも速く表示できるというメリットがあります。『技術情報 > 絵文字』を参照し、挿入したい絵文字を「絵文字番号」で指定して下さい。

■ 絵文字番号による指定例

```
<div> <img localsrc="191" alt="ここに画像" /> </div>
```

(7) 特殊文字の表示

「<」「>」「&」などの記述により特殊文字を表示させる場合には、本書別紙『特殊文字一覧』をご参照ください。

(8) format 属性による入力文字種の指定

XHTML 記述における<textarea>要素、type 属性が「text」の<input>要素では、「format 属性」が指定されていた場合は下記表の文字が入力できます。

■ 文字種の指定

format 属性の値として指定可能な値と、対応する開始入力モードと入力可能文字種は以下の通りです。

format 属性値	開始入力モード（※）	入力可能文字種
a(小文字)	半角英小文字	半角英小文字、記号/句読点
A	半角英大文字	半角英大文字、記号/句読点
N	半角数字	半角数字
x(小文字)	半角英小文字	半角英小文字、半角英大文字、数字、記号/句読点
X	半角英大文字	半角英小文字、半角英大文字、数字、記号/句読点
m(小文字)	半角英小文字	すべての文字種
M	全角かな	すべての文字種

※：端末の文字入力方法の実装により、入力可能となる文字種が異なる場合もありますので、実際の動作は実機にてご確認ください。

■ 入力桁数の制限

桁数を制限しない場合には、「*N」のように制御文字の前に「*」をつけます。
桁数を制限する場合は、桁数分だけ書式文字を書くか、書式文字の前に桁数をつけます。

【例】半角数字を 4 桁に制限する場合
① format="NNNN"
② format="4N"

■ format 属性におけるエスケープ処理

format 属性の指定において、「エスケープ処理（直前に"¥"を挿入)」を行うことで、「a」などの文字種を表す文字以外に、区切り文字などをセットすることができます。

format 属性値	入力値	表示
NNNN¥年 NN¥月 NN¥日	20030401	2003 年 04 月 01 日
2N¥:2N	1730	17:30

※：端末の文字入力方法の実装により、入力可能文字種や桁数の制限が無効となる場合もありますので、実際の動作は実機にてご確認ください。

エスケープ処理を行わない場合には「WAP2.0 対応ブラウザ 1.1x」及び「WAP2.0 対応ブラウザ 1.13」において、format 属性で指定した文字種は一切無効となり、「*M」（桁数制限のない全角かな）が設定されますのでご注意ください。

(9) istyle 属性

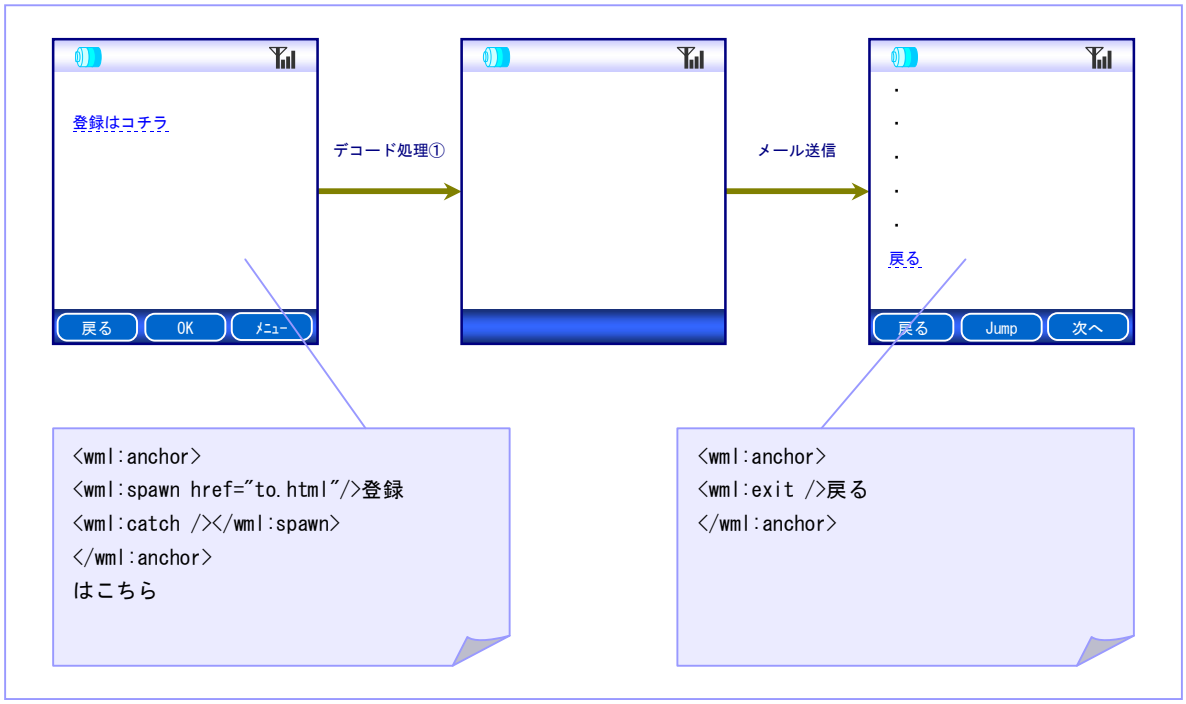
XHTML/HTML 記述において「istyle 属性」が指定されていた場合は下記表の文字が入力できます。

istyle 属性値	開始入力モード	入力可能文字種
1	全角かな	すべての文字種
2	半角カナ	すべての文字種
3	半角英小文字	すべての文字種
4	半角数字	すべての文字種

※：端末の文字入力方法の実装により、開始入力モードが異なる場合もありますので、実際の動作は実機にてご確認ください。

(10) アクティビティの利用方法

<WML:>要素を利用することで、HDML/WML のアクティビティの概念を踏襲することができます。ただし、リンク元への回帰が可能となるだけで、アクティビティ内での変数の保持/設定/参照といった機能は実現できません。



XHTML 記述において、HDML 記述のようにダウンロード完了後にアクティビティ先頭(GOSUB 発行カード)へ戻るような動作をさせる場合には、<wml:>要素を利用して以下のような記述を行って下さい。

```
<wml:anchor>
<wml:spawn href="device:data/dnld?url=http://example.com/cgi-bin/dl.cgi&name=data/sw.pmd&size=28997
&disposition=dev16trp&title=songs" />Title=ASCII<wml:catch /></wml:spawn>
</wml:anchor>
```

3. XHTML Basic リファレンス

以下に、XHTML Basic 仕様のうち、EZ ブラウザがサポートするもののリファレンスを記します。

3.1. 宣言文

EZweb における XHTML Basic 記述では宣言文は以下のようになりますが、省略することができます。

(1) XML 宣言

XML 文書では文書の先頭で以下のような XML 宣言を行い、XML のバージョンと使用する文字セットを宣言することができます。一般的には文字セットが国際符号化文字集合の「UTF-8」と「UTF-16」以外の場合には、必ず XML 宣言を行って使用している文字セットを示す必要がありますが、EZweb 対応端末では XML 宣言文を省略しても問題ありません。

```
<?xml version="1.0" encoding="Shift_JIS" ?>
```

(2) DOCTYPE 宣言

XHTML Basic では、Basic DTD と呼ばれる文書型定義を記述します。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
 "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
```

3.2. 汎用属性

(1) class="クラス名" 【WAP2.0 対応ブラウザ差異あり】

class 属性は、要素に対してクラス名をつけるための属性であり、主にスタイルシートのセレクタとして利用されます。id 属性とは異なり、ひとつの文書中の複数要素に対して同じクラス名をつけることができます。

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ 複数のクラス名を指定

「7.2」から、クラス名をスペースで区切り、複数指定することで、ひとつの要素に対して複数のクラス名を指定することができます。
「1.13」以前では、ひとつの要素に複数のクラス名を指定できません。

§ 大文字と小文字の区別

「7.2」から、クラス名では大文字と小文字が区別されます。
「1.13」以前では、大文字と小文字の違いは無視されます。

(2) id="名前" 【WAP2.0 対応ブラウザ差異あり】

id 属性は、要素に対して固有の名前（識別子）をつけるための属性であり、スタイルシートのセレクタや、リンクの対象、スクリプトからの参照などに利用されます。ひとつの文書中の複数要素に対して同じ名前をつけることはできません。

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ 大文字と小文字の区別

「7.2」から、名前では大文字と小文字が区別されます。
「1.13」以前では、大文字と小文字の違いは無視されます。

(3) lang="言語コード"

lang 属性は、要素の内容と他の属性値の言語を示すための属性であり、指定された言語の習慣に従って、内容をより正しく表現するために利用されます。日本語の場合は「ja」、英語の場合は「en」のように指定します。

(4) style="スタイルシート"

style 属性は、要素に対してスタイルシートの完全な記述なしに直接スタイルを指定することができる属性です。

(5) title="補足情報"

title 属性は、要素に対して補足的な情報を与えるための属性です。適用される要素によってはここで与えられた情報が、ソフトキーに割り当てられます。

3.3. 色指定

色の指定には以下の 2 通りの方法があります。なお、大文字と小文字は区別されません。

① 「#」に続く 16 進数（RGB の順に 2 桁ずつ、計 6 桁で表したもの）による指定

② 次の 16 種類の色名による指定
Black, Green, Silver, Lime, Gray, Olive, White, Yellow, Maroon, Navy, Red, Blue, Teal, Purple, Fuchsia, Aqua

3.4. 基本要素

(1) <body>

文書の内容となる要素を記述します。

属性	
text = "色"	文書全体の文字色を指定します。
link = "色"	キャッシュされていないページへリンクしている部分の色を指定します。
vlink = "色"	キャッシュされているページへリンクしている部分の色を指定します。
bgcolor = "色"	文書全体の背景色を指定します。
background = "URI"	文書全体の背景として、タイル状に隙間なく表示させる画像の URI を指定します。
汎用属性	class, id, style

記述例

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja">
<head>
<title>test</title>
</head>
<body>
...
</body>
</html>
```

(2) <head>

文書のタイトルや利用するスタイルシート、サーチエンジン向けのキーワードや説明など、その文書に関する情報を記述します。

属性	
profile = "URI"	<meta>要素などで指定するメタデータプロファイルの URI を指定します。メタデータプロファイルとは、メタ情報のプロパティの意味とその値を定義されている辞書のようなものです。

記述例

```
<head>
  document description elements
</head>
```

(3) <html>

文書の種類を規定します。XHTML Basic の要素の階層構造で最も上に位置する要素です。

属性	
version = "文字列"	その文書がどの DTD に従って書かれているかを示す属性です。DTD 内ですでに固定値としてバージョンが指定されているため、特にこの属性を指定する必要はありません。
xmlns = "XML 名前空間"	XML 名前空間を指定します。 ・http://www.w3.org/1999/xhtml
xml:lang = "言語"	要素で使用される言語を指定します。 ・ja

記述例

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="ja" xml:lang="ja">
......
</html>
```

(4) <title>

文書の内容を表すタイトルを指定します。各文書の<head>要素内に必ず 1 つ配置する必要があります。
お気に入りリストの登録名に使用されます。

記述例

```
<head>
<title>タイトル</title>
</head>
```

(5) <meta>

文書に関する情報である「メタデータ」を指定するための要素です。基本的にはプロパティと値によって表します。
プロパティは、制作者（author）やキーワード（keywords）などのことであり、値が実際の情報となります。

属性	
content = "プロパティの値"	メタデータのプロパティに対する値を指定します。
name = "プロパティ名"	メタデータのプロパティ名を指定します。
http-equiv = "HTTP ヘッダ用プロパティ名"	メタデータを HTTP ヘッダとして、ブラウザに認識させるために利用されます。 ※：「http-equiv = "refresh"」を利用することはできません。
scheme = "値の形式"	プロパティの値を正しく解釈するための補助的な情報として、値の形式を指定します。例えば、プロパティの値が「4-3-99」の場合に、値の形式として「Month-Day-Year」と指定しておくことで、それが 3 月 4 日ではなく 4 月 3 日であることを明確に示すことができます。

記述例

```
<meta http-equiv="Content-Type" content="text/html;charset=Shift-JIS" />
```

(6) <!-- -->

コメントであることを表します。

記述例
<!--ここはコメントです。-->

3.5. 文書構造/フォント関連要素

(1) <blockquote>

引用された文章であることを示します。
引用部分が長くブロックレベルで引用したい場合に使用します。ブラウザではインデントされて表示されます。

属性	
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例
ここはふつうの文章で <blockquote> ここから引用された文章ということになります。 </blockquote>

(2)
 【WAP2.0 対応ブラウザ差異あり】

要素の位置で改行します。

属性	
clear = "回り込みの解除指定"	・all ：左右両側の画像への回り込みを解除 ・left ：左側にある画像への回り込みを解除 ・right ：右側にある画像への回り込みを解除 ・none ：回り込んだ状態のままで改行（デフォルト）
汎用属性	class, id, style

記述例
 ここに書かれた文章は画像の左側に回り込みます。 <br clear="right" /> ここに書かれた文章は画像の下に続いて表示されます。

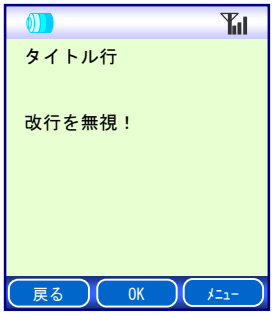
■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ <body>要素の開始タグ直後の
の動作

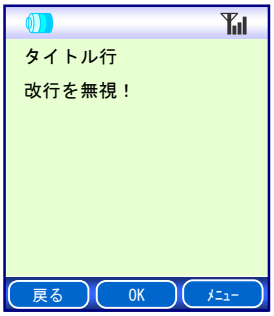
ブラウザのバージョンにより、<body>要素の開始タグ直後の
（
の場合も同様）の動作が異なります。
・7.2 /1.0x ：<body>要素の開始タグ直後の
は無視されず、改行されます。（※1）
・1.13/1.1x ：<body>要素の開始タグ直後の
は無視されます。

※1）1.0x は、ユーザが EZ 設定で「タイトル行非表示」に設定している場合、
を無視します。

```
<head>
<title>タイトル行</title>
</head>
<body>
<br />
<h3>改行を無視！</h3>
</body>
```



WAP2.0 対応ブラウザ 7.2 / 1.0x



WAP2.0 対応ブラウザ 1.1x / 1.13

(3)

範囲をブロック要素にします。class 属性や id 属性などと共に利用することにより、任意の範囲へのスタイル設定が可能となります。なお、 では１度しか改行されません。

属性	
align = "行揃え"	・right ：右寄せ ・center ：中央寄せ ・left ：左寄せ
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例

```
<div>
<a href="index.html">Home</a>
<a href="wtnew.html">What' s New</a>
<a href="prdct.html">Products</a>
<a href="cmpny.html">Companiy</a>
<a href="stmap.html">Site map</a>
</div>
```

(4) <h1>, <h2>, <h3>, <h4>, <h5>, <h6>

見出しをあらわす要素です。
"h"の後に続く数字は、見出しレベル（重要度）を表しており、<h1>要素がもっとも上位の見出しで<h6>がもっとも下位の見出しとなります。重要度の高い見出しほど大きく表示されます。ただし、端末によって実装するフォントサイズが異なるため、大きさは必ずしも保証されません。

属性	
align = "行揃え"	・left　：左寄せ ・center　：中央寄せ ・right　：右寄せ
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例
ここが見出しとして扱われる

(5)

1 つの段落であることをあらわす要素です。なお、 では 1 度しか改行されません。

属性	
align = "行揃え"	・right　：右寄せ ・center　：中央寄せ ・left　：左寄せ
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例
ここに書かれたものが段落として認識されます

(6) <pre>

空白文字や改行を入力されている通りに表示させます。
ただし、1 行の長さがウインドウの画面幅よりも長くなった場合には、自動的に改行されます。

属性	
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例

<pre>
この中に書かれた内容は
改行や
　　スペースなどは
そのまま表示されます。
</pre>

(7)

さまざまな目的に使われるインライン要素です。
要素自体は指定された範囲をインライン要素として設定するだけですが、class 属性や id 属性などと共に利用することによって、任意の範囲にスタイルを設定することができます。

属性	
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例
ここがインライン要素

3.6. リスト関連要素

(1) <dd> 【WAP2.0 対応ブラウザ差異あり】

<dt>要素に記述された用語に対する説明文を示します。

属性	
bgcolor = "色"	要素内の背景色を指定します。
汎用属性	class, id, style

記述例

```
<dl>
<dt>
    用語名
</dt>
<dd>
    その用語の内容
</dd>
</dl>
```

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ bgcolor 属性の適用領域

「1.13」以前では親要素の左辺から bgcolor 属性が適用されますが、「7.2」からは dd 要素自体の左辺から適用されます。

```
<dl>
    <dt>
        用語名
    </dt>
    <dd bgcolor="red">
        その用語の内容
    </dd>
</dl>
```

(2) <dl>

要素の範囲内が定義リストであることを示します。
定義リストとは、定義する用語とそれに対する説明文を対にした形式のリストです。

属性	
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例

```
<dl>
<dt>
    用語名
</dt>
<dd>
    その用語の内容
</dd>
</dl>
```

(3) <dt>

定義リストの定義する用語を示します。この要素に対する説明文には<dd>要素を使用します。

属性	
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例

```
<dl>
<dt>
    定義する用語名
</dt>
<dd>
    その用語の内容
</dd>
</dl>
```

(4)

要素や<ol>要素の内容であるリスト項目となります。

属性	
bgcolor = "色"	背景色を指定します。
type = "マークの種類"	各リスト項目の先頭につくマークの種類を指定します（※： 要素内のみで有効となります）。 ・disc　：塗りつぶされた丸（●） ・square　：線で描かれた四角（□） ・circle　：線で描かれた丸（○） 各リスト項目の先頭につく番号の種類を指定します（※： 要素内のみで有効となります）。 ・1　：算用数字 ・a　：小文字のアルファベット ・A　：大文字のアルファベット
value = "番号"	項目の番号を数字で指定します。以降の項目は、ここで指定した番号に続く番号になります。

記述例

```
<ul>
<li>ここに書いた文章が</li>
<li>リスト化されて
    <ul>
    <li>ここに書いた文章が</li>
    <li>リスト化されて</li>
    <li>表示されます。</li>
    </ul>
</li>
<li>表示されます。</li>
</ul>
```

(5) 【WAP2.0 対応ブラウザ差異あり】

要素の範囲内が各項目の先頭に番号（連番）をつけた形式のリストであることを示します。
リストの各項目は<li>要素によって記述されます。全体がインデントされた状態で表示されますが、リストを入れ子にすると階層に応じてさらにインデントされます。

属性	
bgcolor = "色"	背景色を指定します。
type = "番号の種類"	各リスト項目の先頭につく番号の種類を指定します。 ・1 ：算用数字 ・a ：小文字のアルファベット ・A ：大文字のアルファベット ・I ：ローマ字大文字（Ⅰ,Ⅱ,Ⅲ…） ※ WAP2.0 対応ブラウザ 1.1x からサポート ・i ：ローマ字小文字（ⅰ,ⅱ,ⅲ…） ※ WAP2.0 対応ブラウザ 1.1x からサポート
start = "開始番号"	項目の連番が何番から開始されるかを、数字で指定します。
汎用属性	class, id, style

記述例

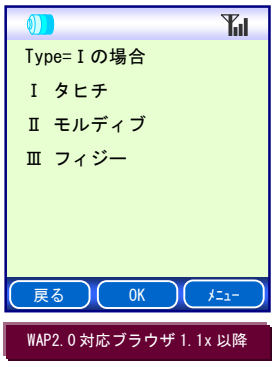
```
<ol>
<li>ここに書いた文章が</li>
<li>リスト化されて</li>
<li>表示されます。</li>
</ol>
```

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

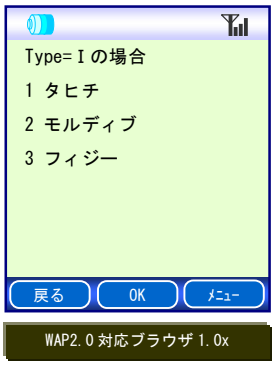
§ type 属性の"Ⅰ"と"ⅰ"のサポート

「1.1x」から、<ol>要素の type 属性の値として、「I」（ローマ字大文字：Ⅰ,Ⅱ,Ⅲ…）と「ⅰ」（ローマ字小文字：ⅰ,ⅱ,ⅲ…）がサポートされます。

```
type=I の場合<br />
<ol type="I">
<li>タヒチ
<li>モルディブ
<li>フィジー
</ol>
```

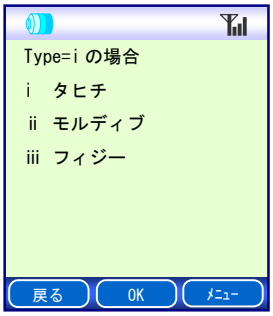


WAP2.0 対応ブラウザ 1.1x 以降

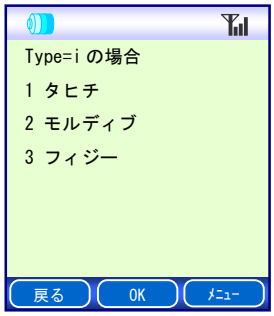


WAP2.0 対応ブラウザ 1.0x

```
type=i の場合<br />
<ol type="i">
<li>タヒチ
<li>モルディブ
<li>フィジー
</ol>
```



WAP2.0 対応ブラウザ 1.1x 以降

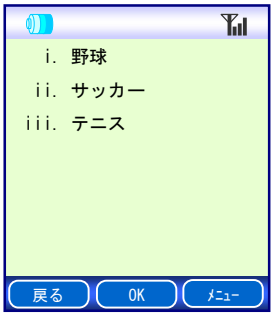


WAP2.0 対応ブラウザ 1.0x

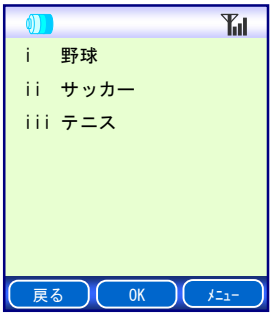
§表示位置

ブラウザのバージョンにより、リスト番号の表示方法が異なります。
・7.2 以降　：ol リストは右寄せで表示され、数字の後にピリオドがつきます。
　　また、WAP2.0 対応ブラウザ 7.2 は、英字の表示は WAP2.0 ブラウザ 1.13 と変更はありませんが、数字と同様に、英字の後にピリオドがつきます。
・1.13 以前　：左寄せで表示され、数字の後にはピリオドがつきません。

```
<ol type="i">
<li>野球
<li>サッカー
<li>テニス
</ol>
```



WAP2.0 対応ブラウザ 7.2 以降



WAP2.0 対応ブラウザ 1.13 以前

(6)

要素の範囲内が項目の先頭に番号（連番）のつかない形式のリストであることを示します。各リスト項目は<li>要素によって記述されますが、その先頭には丸や四角などのマークが付加されます。
全体がインデントされた状態で表示されますが、リストを入れ子にすると階層に応じてさらにインデントされます。

属性	
bgcolor = "色"	要素内の背景色を指定します。
type = "マークの種類"	リストの各項目の先頭につくマークの種類を指定します。 disc　：塗りつぶされた丸（●） square　：線で描かれた四角（□） circle　：線で描かれた丸（○）
汎用属性	class, id, style

記述例
  - ここに書いた文章が   - リスト化されて   - 表示されます。

3.7. 罫線/テーブル関連要素

(1) <caption>

表に対してタイトルや説明文をつけます。<table>要素の開始直後に１つだけ配置することができます。

属性	
align = "配置位置"	表に対するタイトルの位置を指定します。 ・top　：表の上（デフォルト） ・bottom　：表の下 ・left　：表の左 ・right　：表の右
汎用属性	class, id, style

記述例
<table border="1"> <caption align="bottom"> ここにタイトル </caption> ～ </table>

(2) <table> 【WAP2.0 対応ブラウザ差異あり】

要素の範囲内が表であることを示します。表のタイトルやセルなどの表を構成する要素は、すべて<table>要素内に配置します。なお、<table>要素内に<table>要素を記述することはできません。

属性	
bgcolor = "色"	背景色を指定します。
border = "外枠の太さ"	表全体を囲う外枠の太さを「ピクセル数」で指定します。
cellspacing = "セルの間隔"	表全体を囲う外枠とセルの間隔、セルとセルの間隔を「ピクセル数」または、「%」で指定します。
cellpading = "セル内のマージン"	セルの内容とセルの間隔（セル内のマージン）を「ピクセル数」または、「%」で指定します。
width = "幅"	表全体の横幅を「ピクセル数」または、「%」で指定します。

記述例

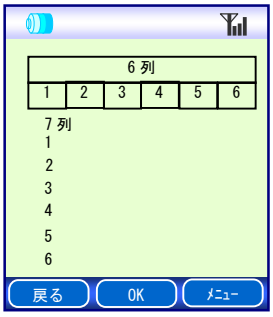
```
<table border="1">
～
</table>
```

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

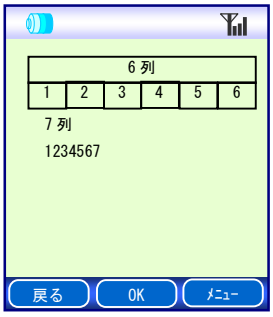
§ <table>要素を利用した表組み表示

「1.1x」から、テーブルの行が画面表示領域を超えるような場合、セルデータを取り出し境界線なしで表示します。また、この場合、セル(<td>content</td>)毎に改行されるため画面表示上セルは横に並ばずに縦一列に表示されます（※「1.0x」では画面表示を超えたセルは wrap されて表示されます)。
従って、<table>要素を利用し表組みを行う場合には画面表示内に収まるように表を組む必要があります。
テーブルを画面レイアウトとして利用している場合には特にご注意ください。

```
<table border="1">
<tr><th colspan="6">6 列</th></tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</table>
<table border="1">
<tr><th colspan="7">7 列</th></tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</table>
```



WAP2.0 対応ブラウザ 1.1x 以降



WAP2.0 対応ブラウザ 1.0x

(3) <td>

表の１セルを表します。セルの内容がデータである場合に<td>要素を使用します。

属性	
align = "行揃え"	セル内のデータの行揃えを指定します。 ・left　：左寄せ ・center　：中央寄せ ・right　：右寄せ
bgcolor = "色"	背景色を指定します。
colspan = "右方向の連結セル数" （デフォルト；1）	このセルを、右方向に指定したセル数分の領域を持った１つのセルにする場合のセル数を指定します。値として、"0"を指定すると、そのセルから右方向にあるすべてのセルの領域を持った１つのセルになります。
rowspan = "下方向の連結セル数" （デフォルト；1）	このセルを、下方向に指定したセル数分の領域を持った１つのセルにする場合のセル数を指定します。値として、"0"を指定すると、そのセルから下方向にあるすべてのセルの領域を持った１つのセルになります。
valign = "縦方向の位置揃え"	各セル内データの縦方向の配置位置を指定します。 ・top　：上寄せ ・middle　：中央寄せ ・bottom　：下寄せ
汎用属性	class, id, style

記述例

```
<table border="1" cellpadding="4">
<tr>
    <th>見出し 1</th>
    <th>見出し 2</th>
    <th>見出し 3</th>
</tr>
<tr>
    <td>データ 1</td>
    <td>データ 2</td>
    <td>データ 3</td>
</tr>
</table>
```

(4) <th>

表のひとつのセルを表します。セルの内容が列（行）の見出しである場合に<th>要素を使用します。

属性	
align = "行揃え"	各セル内のデータの行揃えを指定します。 ・left　：左寄せ ・center　：中央寄せ ・right　：右寄せ
bgcolor = "色"	背景色を指定します。
colspan = "右方向の連結セル数" （デフォルト：1）	このセルを右方向に指定したセル数分の領域を持った１つのセルにする場合のセル数を指定します。値として、"0"を指定すると、そのセルから右方向にあるすべてのセルの領域を持った１つのセルになります。
rowspan = "下方向の連結セル数" （デフォルト：1）	このセルを下方向に指定したセル数分の領域を持った１つのセルにする場合のセル数を指定します。値として、"0"を指定すると、そのセルから下方向にあるすべてのセルの領域を持った１つのセルになります。
valign = "縦方向の位置揃え"	各セル内データの縦方向の配置位置を指定します。 ・top　：上寄せ ・middle　：中央寄せ ・bottom　：下寄せ
汎用属性	class, id, style

記述例

```
<table border="1" cellpadding="4">
<tr>
    <th>見出し 1</th>
    <th>見出し 2</th>
    <th>見出し 3</th>
</tr>
<tr>
    <td>データ 1</td>
    <td>データ 2</td>
    <td>データ 3</td>
</tr>
</table>
```

(5) <tr>

表の横一列分のセルを含む要素です。<th>要素と<td>要素のみを配置することができます。

属性	
align = "行揃え"	各セル内のデータの行揃えを指定します。 ・left　：左寄せ ・center　：中央寄せ ・right　：右寄せ
bgcolor = "色"	背景色を指定します。
valign = "縦方向の位置揃え"	各セル内データの縦方向の配置位置を指定します。 ・top　：上寄せ ・middle　：中央寄せ ・bottom　：下寄せ
汎用属性	class, id, style

記述例

```
<table border="1" cellpadding="4">
<tr>
    <th>見出し 1</th>
    <th>見出し 2</th>
    <th>見出し 3</th>
</tr>
<tr>
    <td>データ 1</td>
    <td>データ 2</td>
    <td>データ 3</td>
</tr>
</table>
```

3.8. リンク関連要素

(1) <a> 【仕様制限あり/WAP2.0 対応ブラウザ差異あり】

指定した範囲をリンクのアンカーにする要素です。アンカーは、ある文書の一部からある文書（またはその一部）へとリンクが設定されている場合の出発地点と到達地点のことで、<a>要素によりその両方を指定することができます。
リンクの出発地点とする場合は、href 属性でリンク先の URI を指定します。
また、到達地点としたい場合は、name 属性または id 属性を使用して識別子を付加します。このとき、href 属性で「URI#名前」と指定することで、特定の文書内の識別子の位置にリンクさせることができます。

属性	
accesskey = "アクセスキー"	この要素にショートカットキーを割り当てることができます。 0-9, #, *
href = "URI"	リンク先の URI を指定します。 ・URI ・mailto: メールアドレス ・tel: 電話番号
name = "名前"	ページ内リンクの飛び先に名前をつけます。
汎用属性	class, id, style

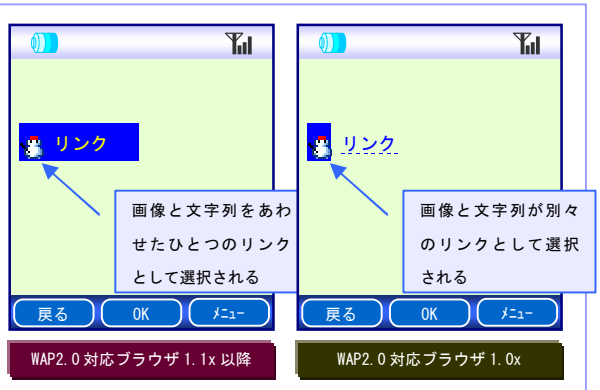
記述例

```
<a href="test1.html">通常のリンク</a><br />
<a href="#test2">ページ内リンク</a><br />
...
<a id="test2">ページ内飛び先</a><br />
```

■ WAP2.0 対応ブラウザ 1.0x の仕様制限事項

<a>要素内に<img>要素と文字列を挿入すると、<img>データと文字列が別々のリンクとして選択されてしまいます。なお、WAP2.0 ブラウザ 1.1x 以降では、本仕様制限事項が解消されています。

```
<a href="xxx.html"><img localsrc="191" />リンク
</a>
```



■ WAP2.0 対応ブラウザのバージョンによる動作の違い

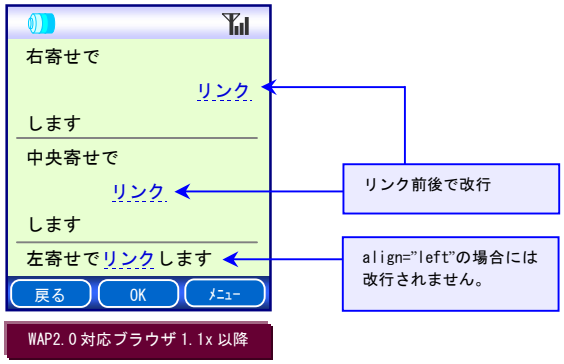
§ 直リンクによる音声データの再生

「1.1x」から、にて音声データ（QCELP/C-MIDI/SMAF）を指定した場合、音声データが連続再生されます。

§ align 属性利用時の改行

「1.1x」から、のように<a>要素に align 属性として「center」または「right」を指定した場合、<a>要素の開始タグ直後、及び<a>要素の終了タグ（）直前で改行されます。

```
右寄せで
<a href="xxx.html" align="right">リンク</a>します
<hr />

中央寄せで
<a href="xxx.html" align="center">リンク</a>します
<hr />

左寄せで
<a href="xxx.html" align="left">リンク</a>します
```



§ Flash Lite コンテンツの再生

「1.13」から、<a href=～>にて Flash Lite コンテンツを指定した場合、Flash Lite コンテンツをインタラクティブ再生します。

(2) <link>

<head>要素内でのみ使用し、関連する別の文書との関係を示します。例えば、文書の外部スタイルシートと出力対象メディアを示す場合などに利用されます。

属性	
rel = "リンクタイプ"	この文書から見た、href 属性で指定された文書との関係を指定します。
href = "URI"	関連する別の文書の URI を指定します。
type = "メディアタイプ"	関連する別の文書のメディアタイプを指定します。ブラウザが対応していない形式のデータタイプを無駄にロードしてしまうことを避けるなどの目的で利用されます。
汎用属性	class, id, style

記述例

```
<head>
<link rel="stylesheet" href="stylesheet.css" type="text/css" />
</head>
```

3.9. マルチメディア関連要素

(1) <img> 【WAP2.0 対応ブラウザ差異あり】

src 属性で指定した URI の画像データや音声データを表示（インライン再生）します。

属性	
align = "文字との位置関係"	画像を表示する場合の画像と文字との縦の位置関係、または文字を回り込ませる場合の画像の横位置を指定します。 ・top　：画像の上と文字の上を揃える。(デフォルト) ・middle　：画像の中心と文字のベースラインを揃える。 ・bottom　：画像の下と文字のベースラインを揃える。 ・left　：画像は左に配置し、その右側に文字を回り込ませます。(デフォルト) ・right　：画像は右に配置し、その左側に文字を回り込ませます。
alt = "代替テキスト"	画像を表示できない場合に、代わりに表示させるテキストを指定します。
border = "枠の太さ"	画像の周りに表示する枠の太さを「ピクセル数」で指定します。デフォルトは"0"（枠線を表示しない）ただし、リンクポイントである場合、デフォルトは"1"となります。
copyright = "保存可否情報"	・yes　：添付データ保存、画面メモを禁止します。 ・no　：添付データ保存、画面メモを許可します。(デフォルト)
height = "高さ"	画像の高さを「ピクセル数」で指定します。
hspace = "左右の空間"	画像の左右に設けられる空間の幅を「ピクセル数」で指定します。
localsrc = "絵文字番号"	表示させたい絵文字を"絵文字番号"で指定します。
src = "URI"	インライン再生させたいマルチメディアデータの URI を指定します。
vspace = "垂直方向の余白"	画像の上下に設けられる空間の幅を「ピクセル数」で指定します。
width = "幅"	画像の幅を「ピクセル数」または「%」で指定します。
汎用属性	class ,id ,style

記述例

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ alt 属性

「1.1x」から、alt 属性によって指定可能な文字列は画面表示上 4 行に制限されます。

§ Flash Lite コンテンツの再生

「1.13」から、<img src=~>にて Flash Lite コンテンツを指定した場合、Flash Lite コンテンツをインライン再生します。

(2) <object> 【WAP2.0 対応ブラウザ差異あり】

画像、動画、アプリなどの各種マルチメディアデータのダウンロードを行います。WAP2.0 対応ブラウザ 1.13 では、ダウンロードではなく再生（インライン再生／インタラクティブ再生）させることも可能です。
詳細は『技術情報 > ダウンロード CGI』をご参照ください。

属性	
data = "URI"	ダウンロードをしようとするデータの存在する URI を絶対パスで指定します。
type = "メディアタイプ"	data 属性で指定されるデータのメディアタイプを指定します。
copyright = "保存可否情報"	「yes」または「no」で、ダウンロードしたコンテンツの端末への保存可否を指定します。 ※本属性はダウンロードしたコンテンツの転送（他の端末へのメール添付による送信等）のコントロールを行うものではないのでご注意ください。
standby = "文字列"	data 属性で指定されるデータをダウンロードする際の、リンクとして表示される文字列を指定します。
height = "高さ"	data 属性で指定されるデータをインライン再生する時の高さを指定します。 ※本属性は、インライン再生する場合のみ記述します。 ※width 属性とセットで指定します。（片方のみの指定は不可）
width = "幅"	data 属性で指定されるデータをインライン再生する時の幅を指定します。 ※本属性は、インライン再生する場合のみ記述します。 ※height 属性とセットで指定します。（片方のみの指定は不可）
declare = "declare"	data 属性で指定されるデータをインタラクティブ再生する場合に指定します。 ※本属性は、インタラクティブ再生する場合のみ指定します。
id	インタラクティブ再生する場合、object 要素識別用 ID を指定します。

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ height 属性、width 属性、declare 属性

「1.13」から、height 属性、width 属性、declare 属性がサポートされます。

(3) <param>

object 要素と組み合わせて使用し、object 要素で指定されたデータに対するパラメータを指定します。
詳細は『技術情報 > ダウンロード CGI』をご参照ください。

属性	
name = "パラメータの名前"	パラメータの名前を指定します。
value = "パラメータの値"	パラメータの値を指定します。
valuetype = "値のタイプ"	value 属性で指定した値のタイプを指定します。

3.10. フォームデータ入力関連要素

(1) <input> 【仕様制限あり/WAP2.0 対応ブラウザ差異あり】

ユーザがデータの入力や選択、データの送信などを行うための部品となる要素です。フォームデータは、name 属性でつけた名前と組になった形でサーバに送信されます。

属性	
accesskey = "ショートカットキー"	この要素にショートカットキーを割り当てることができます。 ・0～9, *, #
checked = "checked"	type 属性の値が"checkbox"、"radio"の場合に、そのボタンが選択されている状態にします。
disabled = "disabled"	選択や変更などの操作をできないようにする場合に指定します。 この属性が指定されている要素のデータはサーバに送信されません。
emptyok = "空値送信"	空値送信の可否を設定します。type 属性が"text"、"password"の場合のみ有効です。 ・true　：空値送信可（デフォルト） ・false　：空値送信不可
format = "入力文字種"	ユーザの入力支援として、入力文字種を指定することができます。(「2.3.(8) format 属性による入力文字種の指定」参照)。
localsrc = "絵文字番号"	type 属性の値が"submit"もしくは"reset"の場合に、送信ボタンとして利用する絵文字を絵文字番号で指定します。
maxlength = "最大文字数"	type 属性の値が"text"、"password"の場合、入力可能最大文字数を指定します。デフォルトの状態では入力文字数の制限はありません。
name = "名前"	名前を指定します。フォームデータが送信される場合は、この名前と組になって送信されます。ラジオボタンやチェックボックスで共通する項目の選択肢として利用する場合には、同じ名前をつけて下さい。
readonly = "readonly"	要素に対して変更ができないようにする場合に指定します。ただし、選択することは可能でありデータはサーバに送信されます。type 属性が"text"と"password"の場合のみ有効です。
size = "幅"	この部品の幅を指定します。"text"、"password"の場合は文字数で、それ以外の場合はピクセル数で指定します。
src = "URI"	type 属性の値が"submit"もしくは"reset"の場合に、送信ボタンとして利用する画像の URI を指定します（設定可能なデータは画像データ gif/jpeg のみとなります)。
type = "形式"	・text　：１行のテキスト入力フィールド（デフォルト） ・password　：パスワード入力用フィールド ・checkbox　：チェックボックス（複数選択可） ・radio　：ラジオボタン（１つのみ選択可） ・submit　：送信ボタン ・reset　：リセットボタン（フォーム内の全部品を初期状態に戻す） ・hidden　：表示させずにサーバに送信するデータ
value = "値"	type 属性の値が"text"の場合は入力フィールドの初期値、type 属性の値が"submit"、"reset"の場合は、ボタンのラベルとして表示させる文字列を指定します。type 属性が"checkbox"、"radio"の場合は、その項目が選択されている時にサーバに送信される値となります。
汎用属性	class, id, style

記述例

```
<form action="/cgi-bin/test.cgi" method="post">
名前：<input type="text" name="test1" format="M*" emptyok="true" /><br />
※名前は必須入力項目です。<br />
住所：<input type="text" name="test2" istyle="1" /><br />
性別：<br />
<input type="radio" name="sex" value="none" checked="checked" />指定なし<br />
<input type="radio" name="sex" value="man" />男<br />
<input type="radio" name="sex" value="woman" />女<br />
年齢：<input type="text" name="age" maxlength="3" /><br />
趣味（複数選択可）:
<input type="checkbox" name="test" value="1" />スポーツ<br />
<input type="checkbox" name="test" value="2" />読書<br />
<input type="checkbox" name="test" value="4" />睡眠<br />
</form>
```

■ WAP2.0 ブラウザ 1.0x/1.1x/1.13 の仕様制限事項

<input>要素により「ラジオボタン」を利用する際、「name=」の値を同一にしても、ラジオボタンの各選択肢を<select>要素で分断すると、複数選択肢を同時選択できてしまいます。
なお、WAP2.0 ブラウザ 7.2 からは、本仕様制限事項が解消されています。

```
<input type="radio" name="SEIBETSU" value="AIRPLANE" />飛行機<br />

<select name="K-KAISYA">
    <option value="0">A 社</option>
    <option value="1">B 社</option>
</select><br />

<input type="radio" name="SEIBETSU" value="TRAIN" />電車><br />

<select name="T-KAISYA">
    <option value="0">A 社</option>
    <option value="1">B 社</option>
</select><br />
```

⇒本例の場合、「飛行機」と「電車」の双方選択が許容されてしまいます。

```
<input type="radio" name="SEIBETSU" value="AIRPLANE" />飛行機<br />
<input type="radio" name="SEIBETSU" value="TRAIN" />電車><br />
```

⇒本例の場合、「飛行機」か「電車」かのいずれか一方の選択のみ許容されます。

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ <input>の value として表示可能な文字(byte)数

「1.1x」から、<input>要素の value 属性により「初期値」として画面上に表示可能な文字（byte）数は、「1023byte（全角 511 文字）まで」となります。
「7.2」からは、byte 数ではなく文字数単位での制限となります。表示可能な文字数は「1024 文字まで」となります。
※：上限を超えて文字をセットした場合、上限以降の文字列は表示されません。

(2) <form> 【WAP2.0 対応ブラウザ差異あり】

入力フォームを構成するためのさまざまな項目（入力フィールド、ボタン、メニューなど）を含み、入力されたデータの取り扱いについて指定する要素です。ユーザが送信ボタンを押すと、フォームの内容は action 属性で指定された URI へ送信されます。

属性	
action = "URI"	フォームデータを送信する URI を指定します。
method = "HTTP メソッド"	フォームデータ送信の際に使用する HTTP メソッドを指定します。 ・get ：action 属性で指定された URI の後に「?」とフォームのデータを追加した URI 形式でデータを送信します。 ・post ：フォームのデータをフォームの BODY として送信します。
汎用属性	class, id, style

記述例

```
<form action="/cgi-bin/test.cgi" method="post">
名前：<input type="text" name="name" /><br />
<input type="radio" name="sex" />男<br />
<input type="radio" name="sex" />女<br />
<input type="submit" name="test" value="登録" />
<input type="reset" name="clear" value="クリア" />
</form>
```

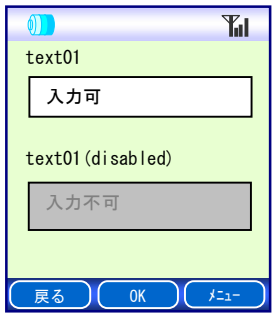
■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ disabled 属性の動作

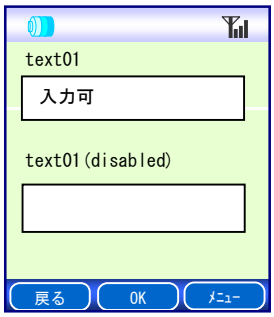
disabled 属性により要素が無効化されている場合（disabled="disabled"）、無効化されている要素の表示形式が、ブラウザバージョンによって異なります。

【表示例：type="text"の場合】

```
<form method="post" action="../post-get.cgi">

<div>text01<br />
<input type="text" name="text01_1" value="入力可"
 />
</div>

<div>text01(disabled)<br />
<input type="text" name="text01_2" value="入力不
可" disabled="disabled" />
</div>
```



WAP2.0 対応ブラウザ 7.2



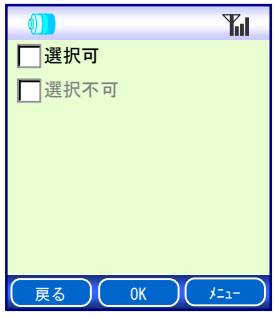
WAP2.0 対応ブラウザ 1.13

WAP2.0 対応ブラウザ 1.1x

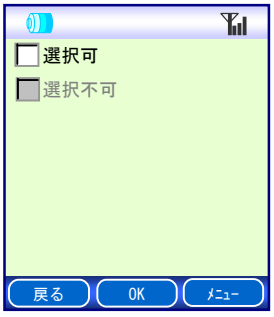
WAP2.0 対応ブラウザ 1.0x

【表示例：type="radio"の場合】

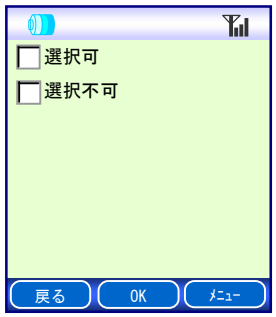
```
<form method="post" action="../post-get.cgi">

<input type="checkbox" name="CHECK1" value="xx"
 />選択可<br />

<input type="checkbox" name="CHECK2" value="yy"
disabled="disabled" />選択不可<br />

</form>
```



WAP2.0 対応ブラウザ 7.2 / 1.13



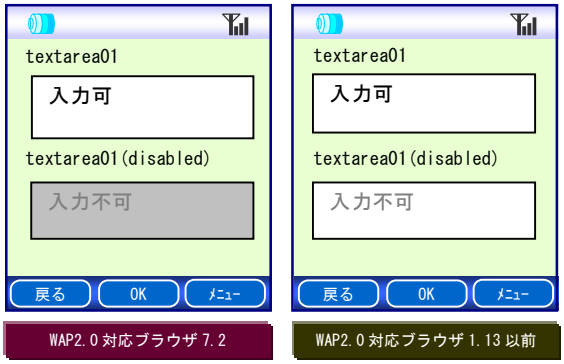
WAP2.0 対応ブラウザ 1.1x



WAP2.0 対応ブラウザ 1.0x

【表示例：textarea の場合】

```
<form method="post" action="../post-get.cgi">
<div>textarea01<br />
<textarea name="textarea01_1" cols="40" rows="4">
入力可</textarea>
</div>
<div>textarea01(disabled)<br />
<textarea name="testarea01_2"
 disabled="disabled" cols="40" rows="4">入力不可
</textarea>
</div>
</form>
```



■ ファイルアップロード

ブラウザバージョン「7.2.7.1.K.4（09 夏端末の一部）」から RFC1867 に規定される HTML における書式からのファイルアップロード機能が搭載されます。
EZweb ブラウザからのアップロード可能なファイルサイズは 500Kbyte 以下の著作権なしファイルとなります。
なお、アップロード可能なファイルは移動機のデータフォルダまたは外部メモリ(microSD)データフォルダ内のファイルが対象であり、動画ファイルは au の移動機で撮影されたもののみが対象です。
著作権ありのファイルは対象外となりますのでご注意ください。

```
<form method="post" action="../post-get.cgi"
enctype="multipart/form-data">
<input type="file" name="upfile" />
<input type="submit" value="アップ" />
</form>
```

（注 1）ファイルをアップロードするには、enctype に"multipart/form-data"を指定する必要があります。
enctype を指定しない、または"application/x-www-form-urlencoded"を指定した場合、type="file"で指定されたファイルの内容は POST されません。

（注 2）enctype に"multipart/form-data"が指定された場合、form 内のファイル以外の値（type="text"等）は、URL エンコードされてサーバへ POST されるため、POST データを受け取るプログラム側で URL デコードする必要があります。

(3) <label> 【仕様制限あり/WAP2.0 対応ブラウザ差異あり】

フォームの中でフォームの構成部品（一行テキストボックス・チェックボックス・ラジオボタン等）と、その項目名（ラベル）の関係付けを行う要素です。なお、フォームの構成部品として<select>は利用できません。

属性	
for = "部品の ID"	このラベルと関連付ける部品の id 属性と同じ値を指定し、ラベルと部品を関連付けます。
汎用属性	class, id, style

記述例

```
<form action="/cgi-bin/test.cgi" method="post">
<label>その 1<input type="text" name="test4" /><br /></label>
<label>その 2<input type="checkbox" name="test5" /><br /></label>
<label>その 3<input type="radio" name="test6" /><br /></label>
<label for="test7">その 4</label>
<input type="text" name="test7" id="test7" /><br />
<label for="test8">その 5</label>
<input type="checkbox" name="test8" id="test8" /><br />
<label for="test9">その 6</label>
<input type="radio" name="test9" id="test9" /><br />
</form>
```

■ 仕様制限事項

① <label>要素内に複数のフォーム構成部品（入力コントロール）を配置することはできません。

```
<label>
名前入力
<input type="text" name="firstname" />
<br />
<input type="text" name="lastname" />
</label>
```

② <label>要素内にフォームの構成部品(入力コントロール)として<select>要素を利用することはできません。<select><option>要素を利用する場合には、<label>要素を使用してはいけません。

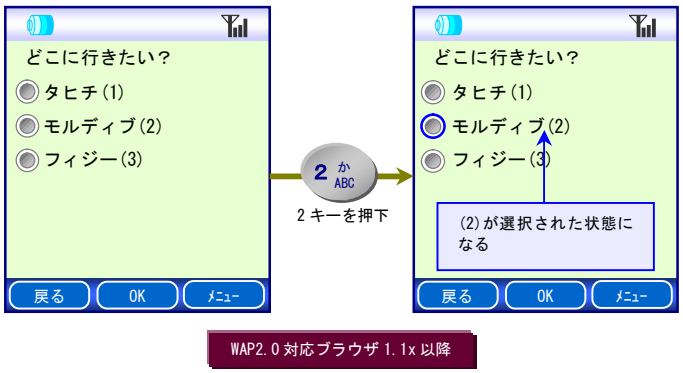
```
<label>
性別選択
  <select name="nameA">
    <option value="male">男</option>
    <option value="female">女</option>
  </select>
</label>
```

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ <label>要素のサポート

<label>要素は「1.1x」からサポートされます。

```
<div>
どこに行きたい？<br />
<label accesskey="1">
<input type="radio" name="foo1" value=
"a" />タヒチ(1)</label><br />
<label accesskey="2">
<input type="radio" name="foo2" value=
"b" />モルディブ(2)</label><br />
<label accesskey="3">
<input type="radio" name="foo3" value=
"c" />フィジー(3)</label>
</div>
```



WAP2.0 対応ブラウザ 1.1x 以降

(4) <option>

<select>要素で制作されるメニューの選択肢となる要素です。
なお、<label>要素と併用する場合には仕様制限事項があります。

属性	
disabled = "disabled"	要素に対して選択や変更などの操作をできないようにする場合に指定します。この属性が指定されている要素のデータはサーバに送信されません。
selected = "selected"	選択肢をあらかじめ選択された状態にします。
value = "値"	この選択肢が選択されている時に、サーバに送信される値を指定します。この属性が指定されていない場合には、option 要素の内容が送信されます。
汎用属性	class, id, style

記述例

```
<form action="test1" method="post">
<select name="test1">
<option value="1">その 1</option>
<option value="2">その 2</option>
<option value="3">その 3</option>
</select>
</form>
```

(5) <select>

メニューを構成します。メニュー内の選択肢は<select>要素の内容である<option>要素によって指定されます。なお、<label>要素と併用する場合には仕様制限事項があります。

属性	
disabled = "disabled"	要素に対して選択や変更などの操作をできないようにする場合に指定します。この属性が指定されている要素のデータはサーバに送信されません。
multiple = "multiple"	選択肢の中から複数の項目を選択できるようにする場合に指定します。
name = "名前"	名前を指定します。フォームデータが送信される場合にはこの名前と組になって送信されます。
size = "行数"	メニューの表示行数を指定します。
汎用属性	class, id, style

記述例

```
<form action="test1" method="post">
<select name="test1">
<option value="1">その 1</option>
<option value="2">その 2</option>
<option value="3">その 3</option>
</select>
</form>
```

(6) <textarea>

複数行の入力が可能なテキスト入力フィールドを構成します。この要素の内容として配置された文字はフィールドにあらかじめ入力された状態で表示されます。フォームのデータは name 属性でつけた名前と組になった形で送信されます。

属性	
accesskey = "ショートカットキー"	要素にショートカットキーを割り当てることができます。 ・0～9, *, #
cols = "幅"	入力フィールドの表示幅を文字数で指定します。 端末の表示可能な幅以上の値を設定した場合、表示可能な幅までに制限されます。
disabled = "disabled"	要素に対して選択や変更などの操作をできないようにする場合に指定します。この属性が指定されている要素のデータはサーバに送信されません。
format = "入力文字種"	ユーザの入力支援として入力文字種を指定することができます。(「2.3.(8) format 属性による入力文字種の指定」参照)。
maxlength = "最大文字数"	入力可能最大文字数を指定します。デフォルトの状態では入力文字数の制限はありません。
name = "名前"	名前を指定します。フォームデータ送信の際にはデータと組になって送信されます。
readonly = "readonly"	要素に対して変更ができないようにする場合に指定します。ただし、選択することは可能であり、データはサーバに送信されます。
rows = "行数"	入力フィールドの表示行数を指定します。ただし、この数値によって入力可能な行数が制限されるわけではありません。
汎用属性	class, id, style

記述例

```
<textarea name="kanso" rows="4" cols="10">
ここにご感想をお願いします
</textarea>
```

3.11. その他の要素

(1) <q> 【WAP2.0 対応ブラウザ差異あり】

短い引用文を記述します。「WAP2.0 対応ブラウザ 1.1x」からサポートされます。

属性	
cite = "uri"	引用元の URI を指定します。
汎用属性	class, id, style

記述例

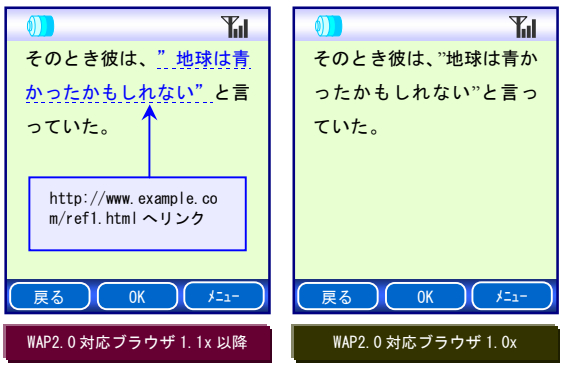
```
<p>そのとき彼は、
<q cite="http://www.example.com/ref1.html">
地球は青かったかもしれない
</q>
と言っていた。</p>
```

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ cite 属性のサポート

「1.1x」から cite 属性がサポートされます。
cite 属性は外部リソースへの参照を含んだ引用文を利用する場合に使用されます。cite 属性を指定すると<q>要素間はハイパーリンクとして扱われ、cite 属性の値に指定した URI がリンク先になります。

```
<p>
そのとき彼は、
<q cite="http://www.example.com/ref1.html">
地球は青かったかもしれない
</q>
と言っていた。
</p>
```



4. EZ ブラウザがサポートするその他の要素

以下に、XHTML Basic の仕様には含まれていない、EZ ブラウザがサポートする要素を記します。

4.1. 基本要素

(1) <style>

<head>要素内にスタイルシートを組み込みます。

属性	
type = "type" [必須属性]	要素内容として記述するスタイルシート言語のメディアタイプを指定します。 CSS のメディアタイプは"text/css"です。
type = "media"	スタイルシートを適用するメディアを指定します。 ・handheld‥‥携帯用端末 ・all‥‥‥‥‥すべてのメディア

記述例

```
<head>
<style type="text/css">
h1 {
font-size : small;
text-align : center;
}
</style>
</head>
```

4.2. 文書構造/フォント関連要素

(1) <blink> 【WAP2.0 対応ブラウザ差異あり】

要素の範囲内の文字列を点滅表示させることができます。

属性	
汎用属性	class, id, style

記述例

```
<blink>この部分が点滅します。</blink>
```

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ 点滅の間隔

「1.13」以前と「7.2」以降とで異なります。「1.13」より「7.2」の方が早くなっています。

(2)

要素の範囲の内容を中央揃えにして表示します。

属性	
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例
ここはセンタリングされない。   ここがセンタリングされる。 ここもセンタリングされない。

(3)

文字色を指定します。

属性	
color = "色"	文字色を指定します。
汎用属性	class, id, style

記述例
ここの文字が緑色になります。

(4) <marquee> 【WAP2.0 対応ブラウザ差異あり】

要素の範囲内の文字をスクロールします。
なお、<marquee>要素内に<p>、<h1>、<form>等のブロックレベル要素を記述してはいけません。

属性	
behavior = "振る舞い"	スクロールの動作を指定します。 ・scroll : direction 属性で指定されている方向へスクロールを繰り返します（デフォルト）。 ・slide : direction 属性で指定されている方向に一度スクロールして、動作を停止します。 ・alternate : 端まで進んだ段階で逆方向へと、スクロールする動作を繰り返します。
bgcolor = "色"	スクロールする範囲の背景色を指定します。
direction = "方向"	スクロールの方向を指定します。 ・left : 右から左へスクロールします。 ・right : 左から右へスクロールします。
loop = "繰り返し回数"	・1～20 (Default:1) ・負の値 ※ WAP2.0 対応ブラウザ 1.1x でサポート ・infinite（再生回数無限） ※ WAP2.0 対応ブラウザ 1.1x でサポート
scrollamount = "ピクセル"	スクロール幅単位を指定します（デフォルト；2 pixels)。
scrolldelay = "時間(msec)"	スクロール時間単位を指定します（デフォルト；36 msec)。
汎用属性	class, id, style

記述例
<marquee>この部分がスクロール表示します</marquee>

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ loop 属性の指定

「1.1x」から、loop 属性の値として再生回数無限を表す「負の値：-1」と「infinite」（再生回数無限）がサポートされます。

§

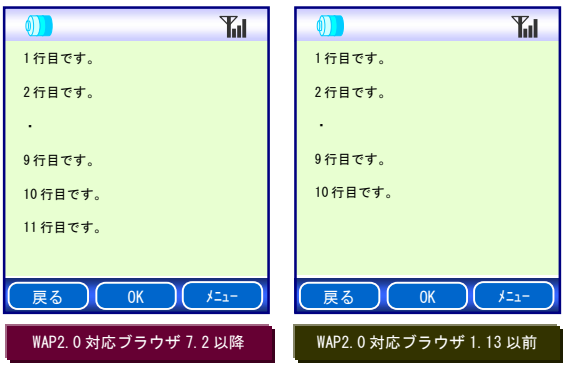
「1.13」以前と「7.2」以降とで、1 ページ中で使用可能な<marquee>要素の数が異なります。

1.13 以前　：10 個以内
7.2 以降　：10marquee 要素

よって、1 つの marquee 要素中に複数行が存在すれば、WAP2.0 対応ブラウザでは 10 行以上の marquee 要素が動作することになります。

また、7.2 以降では、回数が指定されていたコンテンツで、別ページに遷移した後、「クリアボタン」もしくは「左十字キー」等を押下して、ページに戻った場合、規定回数の動作が終わったマーキーは動作しません。

```
<marquee>1 行目です。</marquee>
<marquee>2 行目です。</marquee>
   ・
   ・
   ・
<marquee>9 行目です。</marquee>
<marquee>10 行目です。<br />
11 行目です。</marquee>
```



(5) <u>

この要素の範囲内を下線付きで表示します。

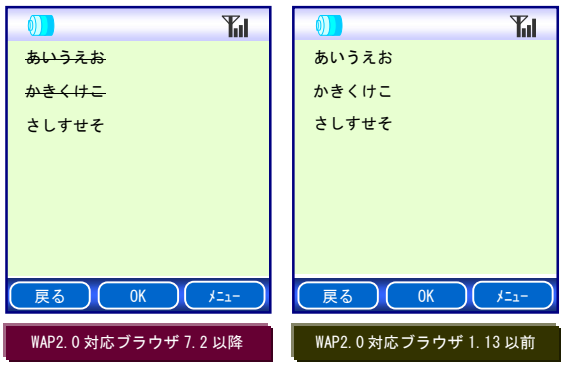
属性	
汎用属性	class, id, style

記述例
<u>ここの文字が下線付きで表示されます。</u>

(6) <del> 【WAP2.0 対応ブラウザ差異あり】

打ち消し線付きの文字を表示します。WAP2.0 対応ブラウザ 7.2 からのサポートとなります。

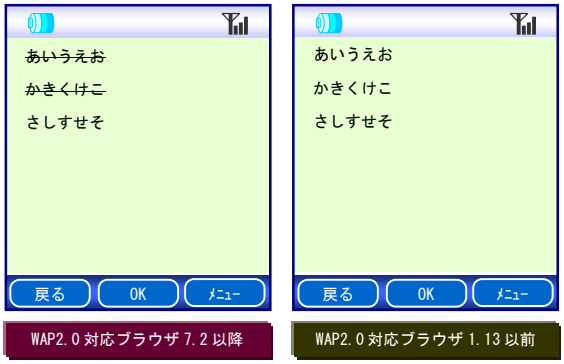
```
<del>あいうえお</del><br />
<del>かきくけこ</del><br />
さしすせそ
```



(7) <s> 【WAP2.0 対応ブラウザ差異あり】

打ち消し線付きの文字を表示します。WAP2.0 対応ブラウザ 7.2 からのサポートとなります。

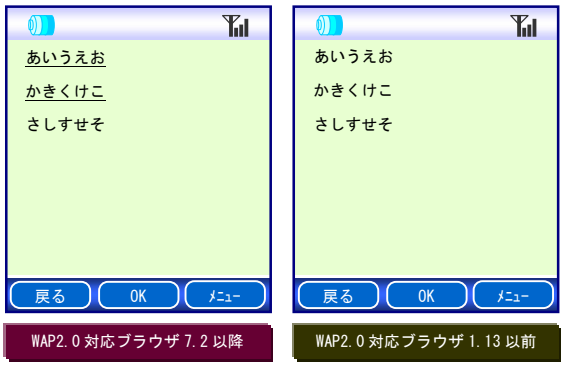
```
<s>あいうえお</s><br />
<s>かきくけこ</s><br />
さしすせそ
```



(8) <ins> 【WAP2.0 対応ブラウザ差異あり】

下線付きの文字を表示します。WAP2.0 対応ブラウザ 7.2 からのサポートとなります。

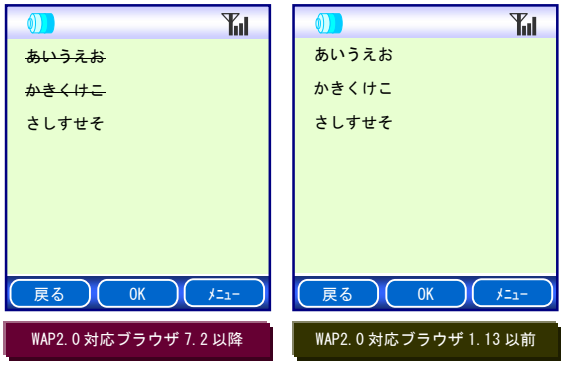
```
<ins>あいうえお</ins><br />
<ins>かきくけこ</ins><br />
さしすせそ
```



(9) <strike> 【WAP2.0 対応ブラウザ差異あり】

打ち消し線付きの文字を表示します。WAP2.0 対応ブラウザ 7.2 からのサポートとなります。

```
<strike>あいうえお</strike><br />
<strike>かきくけこ</strike><br />
さしすせそ
```



4.3. リスト関連要素

(1) <dir> 【WAP2.0 対応ブラウザ差異あり】

ディレクトリのように、複数の段組み形式で表示される番号のつかないリストを表します。
ブラウザでは、<ul>要素のように動作します。

属性	
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例

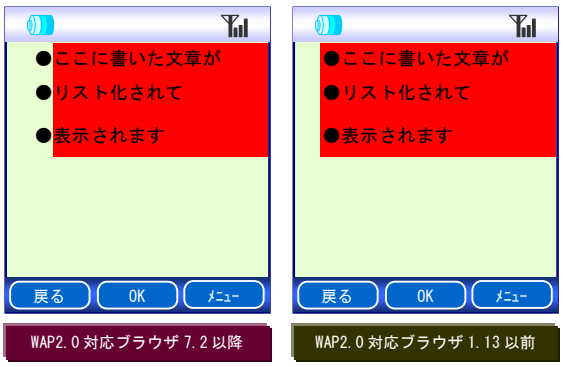
```
<dir>
<li>ここに書いた文章が</li>
<li>リスト化されて</li>
<li>表示されます。</li>
</dir>
```

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ bgcolor 属性の適用領域

「1.13」以前ではマーカー部分から bgcolor 属性が適用されますが、「7.2」からは文字部分から適用されます。

```
<dir bgcolor="red">
<li>ここに書いた文章が</li>
<li>リスト化されて</li>
<li>表示されます。</li>
</dir>
```



(2) <menu>

番号のつかないリストを表します。メニューのように各項目が比較的短いものに対して使用します。
ブラウザでは<ul>要素のように動作します。

属性	
bgcolor = "色"	背景色を指定します。
汎用属性	class, id, style

記述例

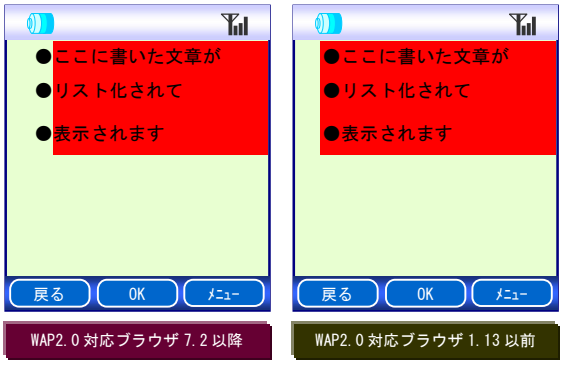
```
<menu>
<li>ここに書いた文章が</li>
<li>リスト化されて</li>
<li>表示されます。</li>
</menu>
```

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ bgcolor 属性の適用領域

「1.13」以前ではマーカー部分から bgcolor 属性が適用されますが、「7.2」からは文字部分から適用されます。

```
<menu bgcolor="red">
<li>ここに書いた文章が</li>
<li>リスト化されて</li>
<li>表示されます。</li>
</menu>
```



4.4. 罫線/テーブル関連要素

(1) <hr> 【WAP2.0 対応ブラウザ差異あり】

横罫線を表示します。

属性	
align = "配置位置"	罫線の横方向の配置位置を指定します。 ・left : 左寄せ ・center : 中央寄せ ・right : 右寄せ
size = "太さ"	罫線の太さを「ピクセル数」で指定します。
width = "長さ"	罫線の長さを「ピクセル数」または「%」で指定します。
汎用属性	class, id, style

記述例

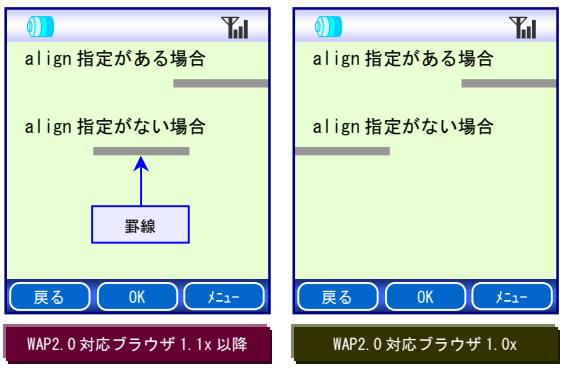
```
<hr />
<hr width="480" />
<hr width="90%" />
<hr size="5" />
```

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ 罫線のセンタリング

「1.1x」から、<hr />を利用した罫線が（align 属性による指定なし）センタリングされて表示されます。

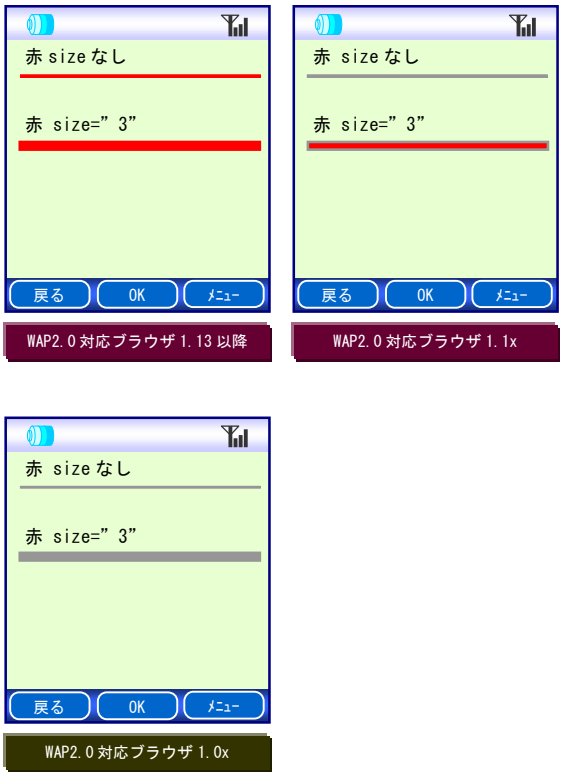
```
align 指定がある場合
<hr align="right" width="50" size="4" />

<br />

align 指定がない場合
<hr width="50" size="4" />
```



§ 罫線の色付け

「1.1x」から、<hr />を利用した罫線に対して color 属性による色付けが可能となります。
「1.1x」では size 属性が「3」以上の場合に色付けが可能となります。
「1.13」以降では size 属性にかかわらず色付けが可能となります。

```
赤 size なし
<hr color="red" />
赤 size="3"
<hr color="red" size="3" />
```



4.5. マルチメディア関連要素

(1) <bgsound>

効果音や BGM 等の音声データ（SMAF/C-MIDI/QCELP）を再生したい場合に使用します。
なお、SMAF/C-MIDI データに画像やテキストが含まれる場合であっても音声のみの再生となります。

属性	
src = "URI"	バックグラウンド再生させたいサウンドデータの URI を指定します。
loop = "回数"	N ：再生する任意の回数を指定します。 Infinite ：無限に再生を繰り返します。

記述例
<bgsound loop=infinite src="sound/sample.mmf" />

4.6. フォームデータ入力関連要素

(1) <optgroup>

メニューの選択肢をグループ化します。

属性	
disabled = "disabled"	要素に対して選択や変更などの操作をできないようにする場合に指定します。 この属性が指定されている要素のデータはサーバに送信されません。
label = "選択肢"	グループ化したグループ名を指定します。
汎用属性	class, id, style

記述例

```
<form>
<select name="test1">
<optgroup label="第一グループ">
<option value="1">第一グループの 1</option>
<option value="2">第一グループの 2</option>
<option value="3">第一グループの 3</option>
</optgroup>
<optgroup label="第ニグループ">
<option value="1">第ニグループの 1</option>
<option value="2">第ニグループの 2</option>
<option value="3">第ニグループの 3</option>
</optgroup>
</select>
</form>
```

4.7. その他の要素

(1) <frameset> 【WAP2.0 対応ブラウザ差異あり】

フレーム機能です。WAP2.0 対応ブラウザ 7.2 からのサポートとなります。

```
<frameset rows="100,*">
 <frame src="frame_ue.html" name="ue">
 <frameset cols="150,*">
   <frame src="frame_hidari.html" name="hidari">
   <frame src="frame_migi.html" name="migi">
 </frameset>
 <noframes>
  このページはフレームを使用しています。
 </noframes>
</frameset>
```

(2) <frame> 【WAP2.0 対応ブラウザ差異あり】

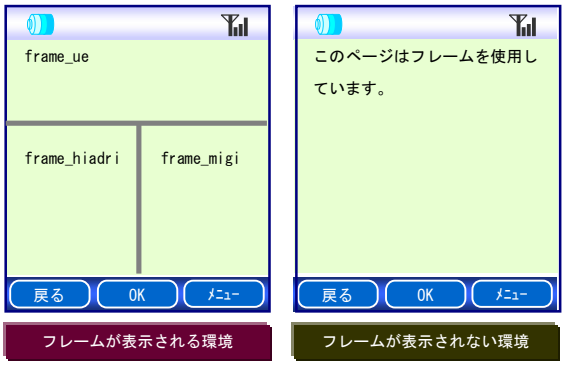
フレームの内容を指定します。WAP2.0 対応ブラウザ 7.2 からのサポートとなります。

```
<frameset rows="100,*">
 <frame src="frame_ue.html" name="ue">
 <frameset cols="150,*">
   <frame src="frame_hidari.html" name="hidari">
   <frame src="frame_migi.html" name="migi">
 </frameset>
 <noframes>
  このページはフレームを使用しています。
 </noframes>
</frameset>
```

(3) <noframes>

フレームが表示されない環境向けの内容を指定します。

```
<frameset rows="100,*">
 <frame src="frame_ue.html" name="ue">
 <frameset cols="150,*">
   <frame src="frame_hidari.html" name="hidari">
   <frame src="frame_migi.html" name="migi">
 </frameset>
 <noframes>
  このページはフレームを使用しています。
 </noframes>
</frameset>
```



5. CSS リファレンス

以下に、EZ ブラウザがサポートする CSS のリファレンスを記します。

5.1. CSS の利用方法

CSS は以下の 3 種類の指定方法により利用することができます。

(1) <link>要素により外部 CSS ファイルを参照する方法

<link>要素を使用して別のファイルに記述したスタイルシート(外部スタイルシート)を文書中に取り込むことができます。外部スタイルシートを利用することで、複数の XHTML 文書でスタイルシートを共有したり、XHTML 文書に手を加えずにスタイルを変更したりすることができます。

```
<head>
  <title>タイトル</title>
  <link rel="stylesheet" type="text/css" href="http://www.example.com/cool.css" title="cool!!" />
</head>
```

(2) ヘッダで宣言する方法

XHTML 文書中に<style>要素の内容としてスタイルシートを記述することができます。<style>要素は<head>要素の中に記述します。

```
<head>
<title>タイトル</title>
<style type="text/css">
h1{color:blue;}
</style>
</head>
```

(3) 適用したい要素に直接書き込む方法

スタイルを適用したい要素に style 属性を指定して、その値として直接スタイルシートを記述することができます。また、「;」で区切ることにより、一度に複数のスタイルを指定することもできます。

```
<body>
<p style="color:green">この章は緑で表示されます。</p>
</body>
```

5.2. セレクタの種類

セレクタはスタイルシートのプロパティの適用要素を示すためのものです。セレクタには以下の種類があります。

(1) Universal セレクタ

すべての要素に対してスタイルを適用させます。

```
<head>
<title>タイトル</title>
<style type="text/css">
*{color:red;}
</style>
</head>
<body>
<h1>ここは赤色表示されます。</h1>
<p>ここも赤色表示されます。</p>
</body>
```

(2) Class セレクタ

その要素の中で、指定した Class 属性を持つ要素に対してスタイルを適用させます。

```
<head>
<title>タイトル</title>
<style type="text/css">
h1.bar{color:green;}
</style>
</head>
<body>
<h1 class=bar>ここは緑色表示されます。</h1>
<h1>ここは緑色表示されません。</h1>
</body>
```

(3) ID セレクタ

その要素の中で、指定した id 属性の識別子を持つ要素に対してスタイルを適用させます。

```
<head>
<title>タイトル</title>
<style type="text/css">
#z98y{color:green;}
</style>
</head>
<body>
<h1 id="z98y">ここは緑色表示されます。</h1>
</body>
```

(4) Type セレクタ

指定した要素名の要素に対して、スタイルを適用させます。

```
<head>
<title>タイトル</title>
<style type="text/css">
h1{color:blue;}
</style>
</head>
<body>
<h1>ここは青色表示されます。</h1>
<p>ここは青色表示されません。</p>
</body>
```

(5) Link Pseudo Class セレクタ

link{ }
　<a>要素によるハイパーリンクの表示において、リンクしたことがないハイパーリンク（キャッシュされていない<a>要素）に対してスタイルを適用させます。

visited{ }
　<a>要素によるハイパーリンクの表示において、リンクしたことがあるハイパーリンク（キャッシュされている<a>要素）に対してスタイルを適用させます。

```
<head>
<title>タイトル</title>
<style type="text/css">
a:link{color:blue;}
a:visited{color:red;}
</style>
</head>
<body>
<a href="http://www.example.com/">テストサイトへのリンク</a>
</body>
```

(6) Descendent セレクタ

シンプルセレクタを半角スペースで区切った時には、前のシンプルセレクタに含まれる（子孫である）後のシンプルセレクタに対してセレクタを適用させることができます。

```
<head>
<title>タイトル</title>
<style type="text/css">
ol li{color:red;}
</style>
</head>
<body>
<ol>
<li>アイテム 1 赤色です。</li>
<li>アイテム 2 赤色です。</li>
</ol>
</body>
```

(7) Child セレクタ

シンプルセレクタを「>」で区切った時には、前のシンプルセレクタの直接の子要素であるシンプルセレクタに対してスタイルを適用させることができます。

```
<head>
<title>タイトル</title>
<style type="text/css">
*{color:green;}
ol > li {color:red;}
</style>
</head>
<body>
<p>
緑色です。
<ol>
<li>アイテム 1 赤色です。</li>
<li>アイテム 2 赤色です。</li>
</ol>
</p>
</body>
```

5.3. 単位の設定

高さ・幅を表す場合には、数値に単位をつけて指定します。

```
px：1 ピクセルを 1 とする単位です。
em：その要素の文字サイズを 1 とする単位です。
ex：その要素の「x」の高さを 1 とする単位です。
```

5.4. 背景に関連するプロパティ

(1) background-attachment 【WAP2.0 対応ブラウザ差異あり】

背景画像のスクロール方法を指定するプロパティです。WAP2.0 対応ブラウザ 1.1x からサポートされます。

引数	・scroll : 背景を一緒にスクロールする ・fixed : 背景はスクロールしない ・inherit : 継承する

(2) background-color 【WAP2.0 対応ブラウザ差異あり】

要素の背景を設定するプロパティです。
WAP2.0 対応ブラウザ 7.2 から、 要素、<p>要素において、background-color プロパティの適用が可能となりました。

引数	任意の色名

(3) background-image

要素の背景画像を設定するプロパティです。

引数	url("背景画像の URI")

(4) background-position 【WAP2.0 対応ブラウザ差異あり】

背景画像の開始位置を指定するプロパティです。WAP2.0 対応ブラウザ 1.1x からサポートされます。

引数	＜横方向の位置＞ ・left : 左寄せ ・right : 右寄せ ・center : 中央寄せ ・%値 ＜縦方向の位置＞ ・top : 上寄せ ・center : 中央寄せ ・bottom : 下寄せ ・%値

(5) background-repeat

背景画像が指定されている時に、画像の並び方を設定するプロパティです。

引数	・repeat ：縦横タイル状に並べられます。 ・repeat-x ：横方向にのみ繰り返して並べられます。 ・repeat-y ：縦方向にのみ繰り返して並べられます。 ・no-repeat ：繰り返されずに１つだけ表示されます。

5.5. 文字体裁に関連する属性

(1) line-height

行の高さ（インライン要素のボックスの高さ）を設定するプロパティです。インライン要素に対して設定した場合は、そのボックスの高さとなりますが、ブロックレベル要素に対して設定した場合は、そこに含まれるインライン要素のボックスの最小の高さを設定したことになります。

引数	・normal ：ブラウザが適切な高さを設定します。 ・実数値＋単位 ：数値に単位をつけて高さを指定します。 ・%値 ：この要素のフォントサイズに対する割合で指定します。

(2) text-align 【WAP2.0 対応ブラウザ差異あり】

指定したブロックレベル要素の中での行揃えを設定するプロパティです。

引数	・left ：左寄せに設定します。 ・right ：右寄せに設定します。 ・center ：中央寄せに設定します。

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ <input>要素、<textarea>要素への text-align プロパティの適用

「1.1x」から<input>要素、<textarea>要素に対して、text-align プロパティの適用が可能です。
このプロパティにより、テキスト領域の境界線（枠線）を基準として入力テキストの位置をあわせることができます。

(3) text-decoration

文字に対して下線をつけたり、点滅をさせたりすることのできるプロパティです。
スペースで区切って複数の値を同時に指定することもできます。このプロパティは、ブロックレベル要素に対して指定した場合は、そこに含まれるすべてのインライン要素にも摘要されます。文字に対してつけられる線の色は、文字色と同じになります。

引数	・none ：標準の状態にします。 ・underline ：要素の下に線が引かれた状態にします。 ・blink ：文字を点滅させます。

(4) text-indent

ブロックレベル要素に含まれる文章の１行目のインデントを設定するプロパティです。

引数	実数値＋単位（…数値に単位をつけて、インデントの幅を指定します）

(5) white-space

指定された要素の中の「半角スペース」「タブ」「改行」をどのように処理するかを設定するプロパティです。

引数	・normal ：連続する半角スペース・タブ・改行を１つの半角スペースに変換して表示します。 ・nowrap ：連続する半角スペース・タブ・改行を１つの半角スペースに変換して表示します。 ただし、自動改行は行いません。

5.6. サイズやレイアウトに関連する属性

(1) height

ボックスの内容領域の高さを設定するプロパティです。

引数	・auto ：内容領域の高さを自動的に設定します。 ・実数値＋単位 ：数値に単位をつけて内容領域の高さを指定します。 ・%値 ：内容領域の高さを、この要素を含むボックスの高さに対する割合で指定します。

(2) vertical-align

指定された要素が表示される行中での、縦方向の位置揃えを設定するプロパティです。

引数	・baseline（デフォルト） 指定された要素のボックスのベースラインと、親ボックスのベースラインを揃えます。指定された要素にベースラインがない場合は、ボックスの下の位置に揃えます。 th 要素と td 要素に対して指定した場合は、横列全体の各セル全体の最初の行のベースラインに揃えられます。 ・top 指定された要素のボックスの上と、その行全体をボックスとした場合の上を揃えます。 th 要素と td 要素に対して指定した場合は、セルボックスの上と横列全体での上を揃えます。 ・middle 指定された要素のボックスの中心と、親ボックスの中心を揃えます。 th 要素と td 要素に対して指定した場合は、セルのボックスの中心と横列全体での中心を揃えます。 ・bottom 指定された要素のボックスの下と、その行全体をボックスとした場合の下を揃えます。 th 要素と td 要素に対して指定した場合は、セルのボックスの下と横列全体での下を揃えます。 ・text-top 指定された要素のボックスの上と、親要素のフォントの上を揃えます。th 要素と td 要素の場合は、この指定は無効となります。 ・text-bottom 指定された要素のボックスの下と、親要素のフォントの下を揃えます。th 要素と td 要素の場合は、この指定は無効となります。

※：ページ内の構成によっては、意図した通りに表示されない場合もありますので、実際の動作は実機にてご確認ください。

(3) width 【WAP2.0 対応ブラウザ差異あり】

ボックスの内容領域の幅を設定するプロパティです。

引数	・auto	：内容領域の幅を自動的に設定します。
	・実数値＋単位	：数値に単位をつけて内容領域の幅を指定します。
	・%値	：内容領域の幅を、この要素を含むボックスの幅に対する割合で指定します。

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ <input>要素、<textarea>要素への width プロパティの適用

「1.1x」から、<input>要素、<textarea>要素に対して、width プロパティの適用が可能です。

5.7. リストに関連する属性

(1) list-style-image 【WAP2.0 対応ブラウザ差異あり】

リスト項目のマーカーとして使用する画像を指定するプロパティです。

引数	url（リスト項目のマーカーとして使用する画像の URI を指定します）

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ 要素が<ol>要素に包含されている場合の list-style-image プロパティの適用

「1.1x」から、<li>要素が<ol>要素に包含されている場合に、包含されている<li>要素に対して、list-style-image プロパティの適用が可能になります。

(2) list-style-position 【WAP2.0 対応ブラウザ差異あり】

リスト項目の要素内容の配置方法を指定するプロパティです。

引数	・outside ：マーカー分字下げした形で、要素内容を配置します。 ・inside ：要素内容をマーカーと同じ開始位置で配置します。

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ マーカーの配置位置

「7.2」からは、outside が指定された場合、マーカーをリスト項目のボックスの外側に配置します。
inside が指定された場合、マーカーをリスト項目のボックスの内側に配置します。

```
<ul>
    <li style="list-style-position:outside; bor
der:Gray 1px solid;">
    list-style-position<br />outside
    </li>
    <li style="list-style-position:inside; bord
er:Gray 1px solid;">
    list-style-position<br />inside
    </li>
</ul>
```

(3) list-style-type 【WAP2.0 対応ブラウザ差異あり】

リストの見出し記号を指定するプロパティです。

引数	＜順序あり＞ ・lower-roman : i, ii, iii... ・upper-roman : I, II, III... ・lower-alpha : a, b, c... ・upper-alpha : A, B, C... ・decimal : 1, 2, 3... ＜順序なし＞ ・disc : ● ・circle : ○ ・square : □ ・none : なし

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ 要素、<ol>要素が<ul>要素の子である場合の list-style-type プロパティの適用

「1.1x」から、<li>要素、または<ol>要素が<ul>要素に包含されている場合、包含されている<li>要素、または<ol>要素に対して、list-style-type プロパティの適用が可能になります。

§ 値「lower-roman」と「upper-roman」のサポート

「1.1x」から、list-style-type プロパティの値として「lower-roman」と「upper-roman」がサポートされます。

§ リストの見出し番号の表示

「1.0x」「1.1x」「1.13」は、左寄せ、「7.2」以降は、右寄せで表示されます。

5.8. テキスト

(1) word-spacing 【WAP2.0 対応ブラウザ差異あり】

WAP2.0 対応ブラウザ 7.2 から、word-spacing プロパティがサポートされました。

(2) letter-spacing【WAP2.0 対応ブラウザ差異あり】

WAP2.0 対応ブラウザ 7.2 から、letter-spacing プロパティがサポートされました。

(3) text-indent【WAP2.0 対応ブラウザ差異あり】

WAP2.0 対応ブラウザ 7.2 から、<table>要素、<tr>要素、<td>要素において、text-indent プロパティの適用が可能となりました。

5.9. フォント

(1) font-size 【WAP2.0 対応ブラウザ差異あり】

WAP2.0 対応ブラウザ 7.2 から、<input>要素の submit や reset のボタンにおいて、font-size プロパティの適用が可能となりました。

5.10. その他の属性

(1) border-top/-bottom/-left/-right 【仕様制限あり/WAP2.0 対応ブラウザ差異あり】

上下左右の境界線に対して、色、太さ、線種を一度に設定できるプロパティです。

引数	・border-width ：実数値＋単位：実数値に単位をつけた形式で太さを指定します。 thin ：細い境界線に設定します。 medium ：中くらいの境界線に設定します。 thick ：太い境界線に設定します。 ・border-color ：色を指定します。 ・border-style ：線種を指定します。

■ 仕様制限事項

境界線に対して色、太さ、線種のうち一部のみ指定した場合、境界線が意図したように表示されない場合があります。意図したように境界線を表示させるためには、色、太さ、線種をすべて指定して下さい。

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ <img>要素、<object>要素、<table>要素、<td>要素への border-*プロパティの適用

「1.1x」から、<img>要素、<object>要素、<table>要素、<td>要素の境界線（枠線）に対して、border-style プロパティ、border-width プロパティ、border-color プロパティの適用が可能になります。

§ border-style による線種の指定

「1.13」までは、border-style に対して、solid しか適用できません。
「7.2」から、solid に加えて、double、dashed、dotted、groove、ridge、inset、outset の各値の適用が可能になります。

(2) color 【WAP2.0 対応ブラウザ差異あり】

要素内の文字色（前景色）を設定するプロパティです。

引数	任意の色名

■ WAP2.0 対応ブラウザのバージョンによる動作の違い

§ <hr>要素、<input>要素、<textarea>要素、<li>要素への color プロパティの適用

「1.1x」から、<hr>要素、<input>要素(type 属性が text、password)、<textarea>要素、<li>要素に対して、color プロパティの適用が可能になります。
「7.2」から、type 属性が submit、reset の<input>要素に対しても、color プロパティの適用が可能になります。

§ <select>要素への color プロパティの適用

「7.2」から、<select>要素に対して、color プロパティの適用が可能になります。

(3) display 【WAP2.0 対応ブラウザ差異あり】

画面にどのように表示するかを指定します。
WAP2.0 対応ブラウザ 1.1x から、「none」のみサポートされます。

引数	none

(4) margin-top/-bottom/-left/-right

上下左右の各マージンを個別に設定されるプロパティです。

引数	・実数値＋単位 ：数値に単位をつけてマージンを指定します。 ・%値 ：マージンをこの要素を含むボックスの横幅に対する割合で指定します。 上下のマージンについても、高さではなく横幅を参照します。

6. HDML による記述のポイント

以下に、HDML を記述する上でのポイントを記します。

6.1. HDML による記述上のテクニック

(1) ブラウザの自動折り返し機能

文章がブラウザの 1 行に収まらない場合、画面の端で自動的に折り返されて表示されます（<WRAP>要素を使用した場合と同様の動作）。
段落の区切りなどの明示的に改行をほどこしたい部分以外には
要素を使用するようにすることで、端末毎のブラウジング画面領域の違いによるレイアウトの乱れを防ぐことができます。

(2) 音声通話連携

Web ページのリンク記述から指定の電話番号へ音声通話の発信へ誘導することができます。
なお、この場合、EZweb の接続は一旦切断されます。

■ 電話番号として利用可能な文字

半角数字、-、#、*、P （※：すべて半角）

※：「P」はポーズ機能で使用します。「P」の位置で一旦ポーズし、発信キーを押下することで続く番号を発信します。

※：電話番号に「クイックダイヤル」（＃ダイヤル）を指定する場合、その番号がすべての地域会社で利用可能とは限らないのでご注意ください。利用できない地域会社がある場合は、その地域会社を Web ページに明記して下さい。

■ 記述例

```
<CE TASK="CALL" NUMBER="03xxxxxxxx">電話連絡先へ CALL
```

```
<ACTION TYPE="ACCEPT" LABEL="Call" task="CALL" NUMBER="03xxxxxxxx">
```

(3) E メール連携

Web ページのリンク記述により、メーラーを起動し、指定のメールアドレスに対するメールの作成画面へ移行させることができます。

HDML による記述では、To、Cc、Bcc、Subject、Body をパラメータとして指定することができます。

■ 基本的な記述例

下記例のように「mailto:」以降にメールアドレスを記述したリンクを作成します。

【記述例①】

```
<A TASK="GOSUB" DEST="mailto:xxx@example.com">メール送信</A>
```

【記述例②】

```
<CE TASK="GOSUB" DEST="mailto:xxx@example.com" LABEL="Mail">メール送信
```

■ 複数送信先/件名/本文等の指定

下記のような記述により「複数送信先」、「件名」及び「本文」を指定することができます。

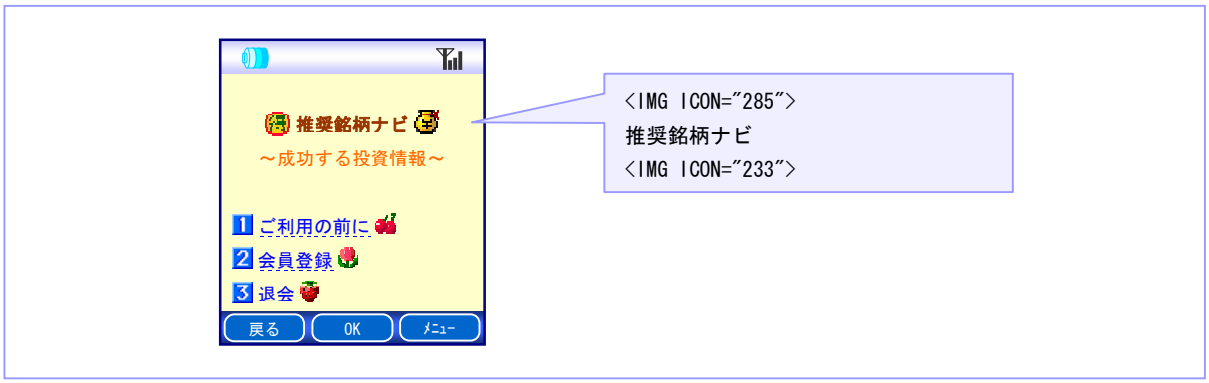
```
<A TASK="GOSUB" DEST="mailto:xx1@example.com,xx2@example.com?cc=yyy@example.com&bcc=zzz@example.com&subject=foo&body=foobar">メール送信</A>
```

※：上記例では TO に「xx1@example.com/xx2@example.com」、CC に「yyy@example.com」、BCC に「zzz@example.com」が指定され、さらに件名/本文にメッセージが指定された形で、メーラーが起動します。

※：複数の TO を設定するには「,」（カンマ）で区切ります。

(4) 絵文字の利用

<IMG>要素の「ICON 属性」記述により Web ページ上で絵文字を表示させることができます。絵文字は端末に実装しているため画像データよりも速く表示できるというメリットがあります。
『技術情報 > 絵文字』を参照し、挿入したい絵文字を「絵文字番号」で指定して下さい。



【例：ページ上で画像として利用する場合】

```
<IMG ICON="138">ベーシック認証
```

【例：ソフトキーに絵文字を表示させる場合】

```
<ACTION TYPE="ACCEPT" LABEL="Call" TASK="CALL" ICON="161" NUMBER="03xxxxxxxx">
```

(5) FORMAT 属性による入力文字種の指定

ENTRY カードによりユーザに対して文字の入力を促す場合、「FORMAT 属性」が指定されていた場合は下記表の文字が入力できます。

■ 文字種の指定

format 属性の値として指定可能な値と、対応する開始入力モードと入力可能文字種は以下の通りです。

format 属性値	開始入力モード（※）	入力可能文字種
a(小文字)	半角英小文字	半角英小文字、記号/句読点
A	半角英大文字	半角英大文字、記号/句読点
N	半角数字	半角数字
x(小文字)	半角英小文字	半角英小文字、半角英大文字、数字、記号/句読点
X	半角英大文字	半角英小文字、半角英大文字、数字、記号/句読点
m(小文字)	半角英小文字	すべての文字種
M	全角かな	すべての文字種

※：端末の文字入力方法の実装により、入力可能となる文字種が異なる場合もあります。

■ 入力桁数の制限

桁数を制限する場合は、桁数分だけ書式文字を書くか、書式文字の前に桁数をつけます。

【例】半角数字を 4 桁に制限する場合
① format="NNNN"
② format="4N"

桁数を制限しない場合には、「*N」のように制御文字の前に「*」をつけます。

(6) PUBLIC 属性によるデッキへのアクセス制限

デッキへのアクセスを許可するサイトの設定を行います。原則としてどのドメインからもリンクが可能となるように、各 HDML ページの冒頭に下記の記載を行って下さい。

```
<HDML VERSION ="3.0" PUBLIC="TRUE">
```

「PUBLIC="TRUE"」の記述を省略した場合には「PUBLIC="FALSE"」と同義となり、「ACCESSDOMAIN」などの属性で明示的に指定されたドメイン以外からのアクセスが行えなくなってしまいます。

(7) お気に入り登録

■ MARKABLE 属性

MARKABLE 属性はブラウザ機能による「お気に入りに追加」の制御を行います。デフォルトでは「MARKABLE="FALSE"」であり、デッキをお気に入りに登録することができません。
原則としてトップページとなる Web ページには、「MARKABLE="TRUE"」を指定して下さい。
なお、「MARKABLE="TRUE"」の指定は「PUBLIC="TRUE"」の指定を含んでいます。

■ お気に入り登録名称の指定

お気に入り登録時の名称を指定するには、各カード（DISPLAY,ENTRY,CHOICE など）の要素の属性の 1 つである「TITLE」を利用します。

```
<HDML VERSION="3.0" MARKABLE="TRUE">
<!-- 以下上から順に、お気に入りに登録の際の default の名称として、
       「EZ DISPLAY」「EZ ENTRY」「EZ CHOICE」が表示されます -->
<DISPLAY NAME="DSP" TITLE="EZ DISPLAY">
  …
</DISPLAY>
<ENTRY NAME="ENT" TITLE="EZ ENTRY">
  …
</ENTRY>
<CHOICE NAME="CHC" TITLE="EZ CHOICE">
  …
</CHOICE>
</HDML>
```

■ お気に入り登録への誘導

通常、お気に入りへの登録は、ユーザによるブラウザメニューの操作で行われますが、下記のような「device:」で始まるURIを用いた記述により、上記の操作なしにお気に入りの登録へ誘導することができます。
なお、WAP2.0対応ブラウザ7.2から、「vars=%3d」を外した指定となります。
「vars=%3d」を外した指定は、WAP2.0対応ブラウザ7.2より前のバージョンでも許容しております。

```
① device:home/bookmark?url=[URI]&vars=%3d&title=[TITLE]
② device:home/bookmark?url=[URI]&title=[TITLE]
```

※［URI］：登録させたいURIを指定します。2バイト文字、URLエンコードされた文字列は使えません。
※［TITLE］：登録させたいタイトルを指定します。日本語を指定したい場合には、URLエンコードを行って下さい。

```
<HDML VERSION="3.0" MARKABLE="TRUE">
<DISPLAY NAME="TOP" TITLE="EZ-TOP">
<action  type="soft1" task="gosub"
         dest="device:home/bookmark?url=http://example.com/top.hdml&vars=%3d&title=EZ-TOP"
         label="登録">
<action  type="accept" task="gosub" dest="http://example.com/menu.hdml" label="次へ">
<br>ようこそ<br><br>
このページを「お気に入り」に登録するならば「登録」ボタンを押してね。
</DISPLAY>
</HDML>
```

```
<HDML VERSION="3.0" MARKABLE="TRUE">
<DISPLAY NAME="TOP" TITLE="EZ-TOP">
<action  type="soft1" task="gosub"
         dest="device:home/bookmark?url=http://example.com/top.hdml&title=EZ-TOP"
         label="登録">
<action  type="accept" task="gosub" dest="http://example.com/menu.hdml" label="次へ">
<br>ようこそ<br><br>
このページを「お気に入り」に登録するならば「登録」ボタンを押してね。
</DISPLAY>
</HDML>
```

(8) カードの名前

各カード（DISPLAY,ENTRY,CHOICE）の要素の属性の 1 つである「NAME」によって、カードに任意の名前をつけることができます。
カードつけられた名前はリンク先の指定において「DEST=#名前」のように利用することができます
なお、「NAME」には「2 バイト文字」及び「URL エンコードした ASCII 文字列」を含めることはできません。

```
<HDML VERSION="3.0" MARKABLE="TRUE" TTL="120">
<DISPLAY NAME="DSP" TITLE="サンプル" ACTION TYPE="ACCEPT" TASK="GOSUB" DEST="#ENT">
<!--            ↑ここに 2 バイト文字は利用できません。  -->
  …
  …
</DISPLAY>
<ENTRY NAME="ENT" TITLE="入力画面" ACTION TYPE="ACCEPT" TASK="GOSUB" DEST=...>
<!--          ↑ここに 2 バイト文字は利用できません。  -->
  …
  …
</ENTRY>
</HDML>
```

6.2. WAP2.0 対応ブラウザに向けて HDML ページを記述する際の注意事項

WAP2.0 対応ブラウザ（WAP2.0 対応端末）で HDML 記述のコンテンツを閲覧する際、HDML は EZ サーバで WML に変換されます。そのため、下記のような注意事項があります。なお新 EZ ブラウザは、HDML をサポートしておりません。

(1) <A>要素の直前の改行処理

以下のように<A>要素記述を連続して行う場合、記述方法によって WAP2.0 対応ブラウザでは<A>要素の開始タグの直前で改行が入ります。

<A>要素の記述		WAP2.0 対応端末画面 での見え方例	
<A ～ >こちら</A><A ～ >そちら</A>	：<A>要素が連続	<A>要素の開始タグの直前で改行 ⇒	こちら そちら
<A ～ >こちら</A> <A ～ >そちら</A>	：<A>要素間に半角スペース	<A>要素の開始タグの直前で改行 ⇒	こちら そちら
<A ～ >こちら</A> <A ～ >そちら</A>	：<A>要素間に改行（CR/LF）	<A>要素の開始タグの直前で改行 ⇒	こちら そちら
<A ～ >こちら</A>へ<A ～ >そちら</A>へ	：<A>要素間に文字	<A>要素の開始タグの直前で改行 ⇒	こちらへ そちらへ

例えば、<A>リンク 1</A>テキスト<A>リンク 2</A>のような記述がある場合、HDML ブラウザでは「リンク 1」と「テキスト」の間で改行されます。
しかし、WAP2.0 対応ブラウザでは、上記の通り<A>要素の開始タグの直前に必ず改行処理を追加するため「テキスト」と「リンク 2」の間で改行がなされてしまいます。

「絵文字とリンクを一行表示させているケース」や、「テキストと“変数を用いた入力を促すリンク”を一行表示させているケース」（お名前[ ]等と表示されているケース）にはご注意ください。
また、アクセスキーを指定した場合は、アクセスキーはテキストの前ではなくリンクの直前に表示されます。

(2) <A>要素内への<IMG>要素挿入によるリンクの分割

<A>要素内に<IMG>要素と文字列を挿入した場合に、文字列と画像が別々のリンクとして選択されます。


```
<A DEST="XXXX">
  <IMG ICON="180">
  トナカイスキー場
  <IMG ICON="191">
</A>
```

⇒ 同一リンク先であっても複数リンクが生成されてしまう

(3) <A>要素における NOOP タスク指定時の文字列リンクの表示

HDML ブラウザでは、<A>要素にて NOOP タスクを指定した場合の文字列はリンク表示されます。
これに対し WAP2.0 対応ブラウザでは、文字列がリンク表示されません。

(4) <ACTION>要素における NOOP タスク指定時のソフトキーラベルの表示

HDML ブラウザでは、<ACTION>要素で NOOP タスクを指定した場合、ソフトキー領域に LABEL 属性で指定された文字列が表示されます。
これに対し WAP2.0 対応ブラウザでは、ソフトキー領域に何も表示されません。

(5) RETURN/CANCEL タスクにおける CLEAR＝TRUE の動作

GOSUB タスクにおいて FRIEND 属性を TRUE に設定しアクティビティを生成した場合、HDML ブラウザでは、RETURN/CANCEL タスクにおいて、CLEAR 属性を TRUE とすると変数をクリアすることが可能です。
これに対し、WAP2.0 対応ブラウザでは変数のクリアが行えません。
WAP2.0 対応ブラウザでの変数のクリアは、<NODISPLAY>カードへ遷移させ、その中で明示的に変数をクリアすることにより実現することができます。

(6) RETURN/CANCEL タスクにおける Referer フィールドの送信

HDML ブラウザでは、RETURN/CANCEL タスクにおいても SENDREFERER 属性を TRUE と設定することにで、HTTP リクエストヘッダ Referer フィールドを送信することができます。
これに対し WAP2.0 対応ブラウザでは､RETURN/CANCEL タスクではいかなる場合にも HTTP リクエストヘッダ Referer フィールドの送信することができません。
RETURN タスク/CANCEL タスクにおけるリンク先では、HTTP リクエストヘッダ Referer フィールドを利用しないようにして下さい。

(7) RETURN/CANCEL タスクにおける POST メソッド指定

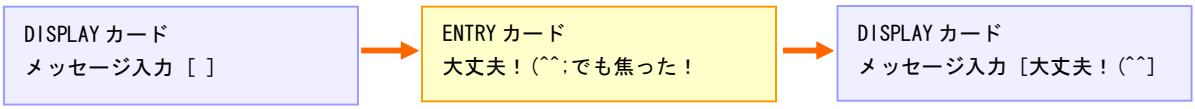
HDML ブラウザでは、RETURN/CANCEL タスクの Method 属性を POST と設定することで、HTTP リクエストのメソッドを POST に指定することができます。
これに対し WAP2.0 対応ブラウザでは、いかなる場合（POST 指定/省略時）においても HTTP リクエストのメソッドは GET となります。
POST データを受け取る Web サーバ側プログラムでは、メソッドが POST/GET のいずれであっても、対応可能となるようにして下さい。

(8) RETURN タスクによる変数受け渡し時の「VARS 属性」の利用

HDML ブラウザでは RETURN タスクでの変数の受け渡しにおいて、RETVALS 属性のかわりに VARS 属性を利用することもできます。
これに対し WAP2.0 対応ブラウザでは、VARS 属性を利用した場合には変数の受け渡しができません。
変数の受け渡しには、RETVALS 属性を利用するようにして下さい。

(9) 「RETVALS 属性」を利用した変数受け渡し動作における「;」使用

WAP2.0 対応ブラウザでは、HDML におけるカード間の変数受け渡し動作（RETVALS 属性の利用）において、変数として;（セミコロン）を文字列の間に利用すると、;（セミコロン）以降の文字列が破棄されます。



(10) <ACTION>要素におけるタスク無指定時のデフォルト動作

<ACTION>要素内にタスクを指定しない場合の各種カードのデフォルト動作が、HDML ブラウザと WAP2.0 対応ブラウザで一部異なります。
ブラウザ間で統一的な動作をさせるには<ACTION>要素内で確実にタスク指定をするようにして下さい。

条件		HDML ブラウザ搭載端末動作	WAP2.0 対応端末動作
<CHOICE> Card	<CE>要素あり <ACTION>要素 タスク指定なし	TASK=PREV	TASK=NOOP
	<CE>要素なし <ACTION>要素 タスク指定なし	TASK=NOOP	TASK=PREV
<ENTRY> Card	<ACTION>要素 タスク指定なし	TASK=PREV	TASK=PREV
<DISPLAY> Card	<A>要素 タスク指定なし	リンク選択不可	リンク選択不可
	<ACTION>要素なし	TASK=PREV	TASK=PREV
	<ACTION>要素 タスク指定なし	TASK=PREV	TASK=PREV

(11) <NODISPLAY>カードでの「CANCEL 属性」省略時の遷移先の相違

<NODISPLAY>カードで GOSUB タスクを指定し CANCEL 属性を省略した場合には、CANCEL タスク実行後に GOSUB タスク起動画面へ戻りますが、遷移先が指定されていないため表示する画面が存在しません。
このとき、HDML ブラウザでは、GOSUB 起動画面の前画面に遷移しますが、WAP2.0 対応ブラウザでは GOSUB 起動後の画面（DEST 属性指定画面）の表示を行います。
ブラウザ間で統一的な動作をさせる場合には、CANCEL タスク利用時に、必ず GOSUB タスクと共に CANCEL 属性を記述するようにして下さい。

(12) <NODISPLAY>でのデッキ全体指定を用いた遷移先指定

<NODISPLAY>カードでは必要のない場合は NAME 属性の省略することができます。
また、HDML ブラウザでは遷移先を示す<ACTION>要素の記述をデッキ全体に指定するように<NODISPLAY>～</NODISPLAY>の外に配置しても動作します。

これに対し WAP2.0 対応ブラウザでは、NAME 属性を省略した場合には、クリアキーや PREV タスクにより前ページに戻る時に空白ページが表示されます。
また、<ACTION>要素の記述を<NODISPLAY>～</NODISPLAY>の外に配置した場合には、画面遷移の途中に空白画面が表示されます。

(13) キャッシュオペレーション機能の利用

HDML ブラウザでは、ダイジェスト送信との併用で、ブラウザ内に存在するキャッシュを URI 単位で無効にしたり、アクティビティ全体のキャッシュを無効にしたりする「キャッシュオペレーション」と呼ばれる機能を利用することができます。
これに対し、WAP2.0 対応端末ではキャッシュオペレーション機能は利用できません。WAP2.0 対応端末に対するキャッシュの制御には TTL 属性の指定を利用して下さい。

(14) HDML3.1 のサポート範囲

HDML ブラウザでは HDML3.0 をサポートし、HDML3.1 の範疇までの機能を非公式にサポートしています。
これに対し WAP2.0 対応ブラウザ（EZ サーバでの変換機能）では HDML3.1 機能は一部しかサポートしていません。
未サポート機能に関しては、他の手法により同機能を実現して下さい。

```
※WAP2.0 対応ブラウザでのサポート機能
      ONTIMER       Value      <ACTION type="ontimer" ...>
      MULTIPLE      Attribute  <CHOICE multiple="true" ...
```

(15) DEST 属性に同一デッキ内カードを指定した場合の POST メソッドの利用

HDML ブラウザでは、<ACTION TASK ="GO">などの記述において、同一デッキ内のカードを指定しても POST Method を指定することができます（e.g. <ACTION TASK ="GO" DEST=#Card1 METHOD="POST" POSTDATA=・・・・>など）。
これに対し WAP2.0 対応端末では、DEST 属性に同一デッキ内のカードを指定した場合には、このカードへのリンクすることができません（Origin Server へリロードされてしまい、同様の動作となりません）。

(16) <LINE>要素内での<TAB>利用時における文字列の折り返し

HDML ブラウザでは、「<LINE> <TAB>WAP2.0 対応ブラウザの仕様制限事項」のような記述において、<TAB>以降の文字列が一行に含まれない場合でも改行することなく、選択された時のみスクロール動作を開始し全文字列表示を行います。
これに対し WAP2.0 対応ブラウザでは、改行されて表示されます。<LINE>内で文字列のインデントを行う場合には、<TAB>のかわりに を利用して下さい。

(17) 複数行に跨る文字列の先頭に<TAB>を挿入した場合の表示

HDML ブラウザでは、「<TAB>WAP2.0 対応ブラウザの仕様制限事項･･･」のような記述において、複数行表示される場合には第一行目のみがインデントされます。
これに対し WAP2.0 対応ブラウザでは、全行インデント表示されます。先頭行のみの文字列のインデントを行う場合には、<TAB>のかわりに を利用して下さい。

(18) 「画像データ＋言語」のダイジェスト送信要求時のメソッド

HDML ブラウザでは、「画像データ＋言語」のダイジェスト送信を要求する場合の HTTP リクエストに Post メソッドと Get メソッドを利用することができます。
これに対し WAP2.0 対応端末では、Get メソッドの利用しかできません。
ダイジェスト送信を要求するコンテンツへのアクセスには、Post ではなく Get メソッドを利用するようにして下さい。

```
<ACTION TYPE="ACCEPT" TASK="GO"  DEST="AAA.cgi" POSTDATA="ABC=$(abc:esc)&DEF=$(def:esc)">
という記述は以下のように変更することで対応できます。

⇒  <ACTION TYPE="ACCEPT" TASK="GO" DEST="AAA.cgi?ABC=$(abc:esc)&DEF=$(def:esc)">
```

(19) ダイジェスト送信における構成要素の順番

HDML ブラウザでは「画像データ＋言語」、「言語＋画像データ」のいずれであっても正常に表示することができます。
これに対し WAP2.0 対応ブラウザでは、「言語＋画像データ」という順序でダイジェストを構成しなければ正常に表示することができません。

(20) <CHOICE>カード内への<A>要素挿入時における選択リスト表示

HDML ブラウザでは、<CHOICE>カード内の<A>要素有無に関わらず選択リスト表示されます。
これに対し WAP2.0 対応端末では<A>要素が存在する場合にはラジオボタン表示になります。
選択リスト表示される場合、項目選択後に自動的に次のカードへの遷移が可能ですが、ラジオボタン表示では選択項目それぞれに遷移先 URI を記述(e.g.<CE ････DEST="#NEXT">等)しない限り、「選択⇒カーソルアウト⇒アクセプトキー押下」の操作が必要となります。
ユーザの操作性を確保するためには、「選択項目それぞれに遷移先 URI を記述する」、もしくは「<CHOICE>カード内では<A>要素を使用しない（ソフトキーを利用する。 e.g. <ACTION TYPE=SOFT1 TASK=PREV>)」などの対応を行ってください。

(21) <CHOICE>カードにおける「IDEFAULT=0」指定時の初期選択値

HDML ブラウザでは、<CHOICE>カードで「IDEFAULT=0」とした時に、<CE>要素より前の文章が短い場合、最初の選択行が初期選択されます。
これに対し WAP2.0 対応ブラウザでは、何も選択されていない状態となります。

(22) <CE>要素「VALUE 属性」におけるエンコード済日本語文字列設定後の表示

<CE>要素「VALUE 属性」にエンコード済みの日本語文字列を指定して変数設定し、その後その変数を表示させる場合、HDML ブラウザでは日本語文字列をデコードして表示することができます。
これに対し WAP2.0 対応端末では、エンコード済みの日本語文字列がそのまま表示されます。
ブラウザ間で共通の動作をさせるには VALUE 属性では日本語文字列をエンコードせずに設定して下さい。

条件	VALUE 値(例)	補足	HDML ブラウザ搭載端末	WAP2.0 対応端末
noesc/esc 指定なし	中性	エンコードなし	%92%86%90%ab	%92%86%90%ab
noesc 指定	中性	エンコードなし	中性	中性
esc 指定	中性	エンコードなし	%92%86%90%ab	%92%86%90%ab
noesc/esc 指定なし	%92%86%90%ab	エンコード済	%92%86%90%ab	%2592%2586%2590%25ab
noesc 指定	%92%86%90%ab	エンコード済	中性	%92%86%90%ab
esc 指定	%92%86%90%ab	エンコード済	%92%86%90%ab	%2592%2586%2590%25ab

(23) 変数名に利用可能な文字

WAP2.0 対応ブラウザでは、HDML/WML で利用可能な文字種は「アルファベット(a-z、A-Z)」、「数字(0-9)」「_(アンダーバー)」のみとなります。また、変数名の先頭には「数字」と「アンダーバー」は利用できません。

(24) 位置情報取得時におけるアクティビティ利用と変数の引渡し

WAP2.0 対応ブラウザでは、位置情報の取得前後においてアクティビティ機能の利用、及び変数の引渡しを行うことができません。

(25) <RIGHT>、<CENTER>要素を利用時のスペースの付加

WAP2.0 対応ブラウザでは、<RIGHT>あいうえお
や、<CENTER>あいうえお
のような記述をすると、文字列の最後尾にスペースが付加され右寄せ/中央寄せされます
<RIGHT>あいうえお

や、<CENTER>あいうえお

のように、
を重ねた場合には、スペースは付加されません。<RIGHT>要素、<CENTER>要素を複数並べる場合には、末尾の
の数によって表示位置が異なりますのでご注意ください。

(26) 「device:～」実行後のアクセスページにおける「PUBLIC 属性」の設定

端末によっては「device:～」で始まる URI へのリンク操作後に、直前にローカルで生成する画面を挟む場合があります。この場合アクセス先のページで、「PUBLIC 属性」が「FALSE」に設定されるとアクセスが拒否されることとなります。「device:～」実行後にアクセスされるページの「PUBLIC 属性」は必ず「TRUE」に設定するようにして下さい。

(27) <A>要素内における「TASK 属性」の記述順序

WAP2.0 対応ブラウザでは、<A>要素内で「TASK 属性」を記述する前に「RECEIVE 属性/DEST 属性」を記述する(例：<A DEST=～ TASK="GOSUB" RECEIVE=～>等)と、変数が正常に引き渡されない場合があります。
<A>要素内では必ず「TASK 属性」を最初に記述するようにして下さい。

(28) SSL サイトにおける Nodisplay カードの利用

非 SSL ページから SSL ページへのリンク時に「Nodisplay カード」を経由させている場合、「戻る」や「クリア」動作によって、SSL ページから、非 SSL ページへ遷移させようとすると、間を経由している「Nodisplay カード」の枚数分に加え、「遷移先の非 SSL ページ」の分のセキュリティエリア終了に関する確認メッセージが表示されます。確認メッセージの表示を回避するためには、非 SSL ページから SSL ページへのリンクの間に「Nodisplay カード」を経由させないようにして下さい。

(29) 「CHOICE カード」での<ACTION TYPE=ACCEPT～>と<ACTION TYPE=SOFT1～>

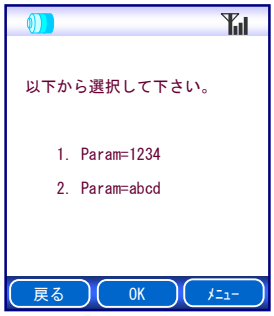
CHOICE カードで<ACTION TYPE=ACCEPT～>、<ACTION TYPE=SOFT1～>という記述において、HDML ブラウザでは<CE>要素の選択肢が選択されるまで、その指定は「無効」となっています。
これに対し WAP2.0 対応端末では、<CE>要素の選択肢が選択されていない状態でも、その指定が「有効」となっています。

```
<choice name="ch_01" key="param">
<action type="accept" task="go" dest="xxx.cgi?prm=$param" label="次へ">
<action type="soft1" task="go" dest="#menu" label="戻る">
<ce value="1234"><line>param=1234
<ce value="abcd"><line>param=abcd
<ce value="あいう"><line>param=あいう
</choice>
```

※：この例では、選択されていない場合、「$param=Null」とし、呼び出された「xxx.cgi」で、パラメータが Null であった場合に、適したコンテンツをブラウザへ返すようにすることを想定しています。

上記のようなCHOICEカードの場合、HDMLブラウザでは、選択肢が必ず選択されている状態、もしくは、「IDEFAULT=0」指定で選択肢が選択されていない状態になります。

一方、WAP2.0対応ブラウザでは、選択肢が選択されていない状態であっても、<ACTION>要素指定が有効なので、(下図) この例の場合、変数「$param」が決定されていない状態で、「xxx.cgi」が呼び出されてしまいます。
もし、この画面に遷移する前に別の画面で、同じ変数名「$param」に関係ない値をセットしていた場合があれば、その値がセットされて「xxx.cgi」が呼び出されてしまいます。



以上のような、意図しない動作を防ぐためには下記のような対応を行う必要があります。

【対応策①】: アクセプトキーに「NOOP」タスクを割り当てて、<CE>要素の中で「DEST」指定する。

```
<choice name="ch_01">
<action type="accept" task="noop">
<action type="soft1" task="go" dest="#menu" label="戻る">
<ce task="go" dest="xxx.cgi?prm=1234" label="次へ"><line>param=1234
<ce task="go" dest="xxx.cgi?prm=abcd" label="次へ"><line>param=abcd
<ce task="go" dest="xxx.cgi?prm=%82A0%82A2%82A4" label="次へ"><line>param=あいう
</choice>
```

【対応策②】:「NODISPLAY カード」を利用して変数を初期化する。

```
<nodisplay name="cl_03">
<action type="accept" task="go" dest="#ch_01" vars="" param="">
</nodisplay>

<choice name="ch_01" key="param">
<action type="accept" task="go" dest="xxx.cgi?prm=$param" label="次へ">
<action type="soft1" task="go" dest="#menu" label="戻る">
<ce value="1234"><line>param=1234
<ce value="abcd"><line>param=abcd
<ce value="あいう"><line>param=あいう
</choice>
```

※:EZget 方式によるダウンロードにおいて「TASK="GOSUB" DEST="device:data/dnld?url=$param"」とした場合、「$param」が「Null」であれば、端末にてエラーを検知しエラーメッセージを表示します。

(30) Web サーバへのデータの送信

<ENTRY>カード、<CHOICE>カードにより Web サーバにデータを送信する場合、ブラウザによって送信される文字列形式が異なります。

■ <ENTRY>カード

メソッド	条件	入力文字	HDML ブラウザ搭載端末	WAP2.0 対応端末
POST	noesc/esc 指定なし	あ	あ	%82%a0
POST	noesc 指定	あ	あ	あ
POST	esc 指定	あ	あ	%82%a0
GET	noesc/esc 指定なし	あ	%82a0	%82%a0
GET	noesc 指定	あ	あ	あ
GET	esc 指定	あ	%82a0	%82%a0

■ <CHOICE>カード

メソッド	条件	VALUE 値(例)	HDML ブラウザ搭載端末	WAP2.0 対応端末
POST	noesc/esc 指定なし	中性	中性	%92%86%90%ab
POST	noesc 指定	中性	中性	中性
POST	esc 指定	中性	中性	%92%86%90%ab
GET	noesc/esc 指定なし	中性	%92%86%90%ab	%92%86%90%ab
GET	noesc 指定	中性	中性	中性
GET	esc 指定	中性	%92%86%90%ab	%92%86%90%ab
POST	noesc/esc 指定なし	%92%86%90%ab	中性	%2592%2586%2590%25ab
POST	noesc 指定	%92%86%90%ab	中性	%92%86%90%ab
POST	esc 指定	%92%86%90%ab	中性	%2592%2586%2590%25ab
GET	noesc/esc 指定なし	%92%86%90%ab	%92%86%90%ab	%2592%2586%2590%25ab
GET	noesc 指定	%92%86%90%ab	中性	%92%86%90%ab
GET	esc 指定	%92%86%90%ab	%92%86%90%ab	%2592%2586%2590%25ab

(31) NODISPLAY カードの利用時における NAME 属性の指定

WAP2.0 対応ブラウザ「1.1x」では、ダウンロード実行ページに NODISPLAY カードを利用し NAME 属性を省略した場合、ダウンロードデータ再生後のクリアキーや PREV タスクの動作（前のページに戻る）で空白ページが表示されるといった現象が発生します。
ダウンロード実行ページでの利用に限らず NODISPLAY カードを利用する場合には必ず NAME 属性を指定して下さい。

(32) E メールから NODISPLAY カードによるダウンロードを実行する場合の動作

WAP2.0 対応ブラウザ「1.1x」では、E メール本文の URI=to 記述から NODISPLAY カードを呼び出して、直接ダウンロードを実行させようとした場合、ダウンロード直後のデータ再生を実行することができません。
本現象を回避するには、ダウンロード実行画面を実体のある DISPLAY カード等で表示をさせた上で、ダウンロードを実行させて下さい。

(33) ACCEPT キーに"noop"を指定した際の動作

WAP2.0 対応ブラウザ「1.13/1.1x」では、ACCEPT キーに"noop"を指定(task="noop")した場合、ページの上下にカーソルアウトする（フォーカスを外す）ことができません。

```
<HDML VERSION="3.0" MARKABLE="TRUE">
<ACTION TYPE="ACCEPT" TASK="NOOP">

<DISPLAY NAME="01">
旅日記（モルディブ編）は現在準備中です。
<A TASK="GO" DEST="http://example.com/top.hdml"
>
TOP へ</A>
</DISPLAY>
</HDML>
```

(34) <LINE>要素利用時の動作

WAP2.0 対応ブラウザでの閲覧において、HDML の<LINE>要素記述は EZ サーバにより WML の<p mode="nowrap">に変換されます。
このように「nowrap」が指定される場合には、WAP2.0 対応ブラウザのバージョンにより動作差異が生じますのでご注意ください。

※：本書「7.2.WAP2.0 対応ブラウザでの動作差異」参照。

7. 【参考】WML による記述上の注意事項

以下に、HDML と同等の機能を WML による記述で実現するための注意事項を記します。

7.1. XHTML における WML の記述方法

以下に示す<wml:>要素により XHTML ページ中において WML の記述を行うことができます。
<wml:>要素を利用することで、ソフトキーラベル表示文字の指定（※1）や、お気に入り登録の禁止（※2)、アクティビティ機能の実現（※3）などを行うことができます。

※：本書「2.3.(2) ソフトキーの機能割り当てと表示文字列の設定方法」参照。
※：本書「2.3.(5) お気に入り登録」参照。
※：本書「2.3.(10) アクティビティの利用方法」参照。

(1) <wml:anchor>

リンクの実行点を制作します。<a>要素と違いリンク先の URI を指定できませんが、<wml:spawn>要素などの他のリンク要素と組み合わせることで、同等の動作を実現させることができます。

属性	
title = "タイトル"	リンクにタイトルをつけます。ここに指定した文字列が、アクセプトキー欄に表示されます。
accesskey = "ショートカットキー"	この要素にショートカットキーを割り当てることができます。 ・0～9, *, #

記述例

```
<wml:anchor accesskey="0" title="LINK">
<wml:spawn href="test1.wml#anc"/>他のファイルへリンク<wml:catch /></wml:spawn>
</wml:anchor><br />
```

(2) <wml:exit>

直前に実行した<spawn>要素の含まれるカードへ戻る要素です(HDML の RETURN に相当/ただし変数の引き回しはできません)。

記述例

```
<wml:anchor accesskey="0" title="LINK">
<wml:spawn href="test1.wml#anc"/>他のファイルへリンク<wml:catch /></wml:spawn>
</wml:anchor><br />
:
<wml:anchor><wml:exit />戻る</wml:anchor>
```

(3) <wml:spawn>

指定した URI へリンクします。<wml:exit>要素を実行することで、最近に実行した<wml:spawn>要素のあるページまで戻すことができます(HDML の GOSUB に相当)。

属性	
href = "URI"	リンク先の URI やカード名を指定します。

記述例

```
<wml:anchor accesskey="0" title="LINK">
<wml:spawn href="test1.wml#anc"/>他のファイルへリンク<wml:catch /></wml:spawn>
</wml:anchor><br />
```

7.2. WAP2.0 対応ブラウザでの動作差異

WML により記述する場合、XHTML ページ内における<wml:>や<p:>により WML 記述をする場合、EZ サーバにより HDML→WML へ変換される場合において、以下の WML 要素では WAP2.0 対応ブラウザのバージョンによって動作が異なりますのでご注意ください。

(1) <do>,<wml:do>,<p:do>

§ <do type="prev">指定をしたページでのカーソルアウト時の「戻る」の動作

「1.1x」以降は、<do type="prev">指定をしたページにおいて、カーソルアウトした時に表示される「戻る」を実行すると、ひとつ前のページへ戻る動作ではなく、タスクで指定する動作(go、prev、noop、refresh)が実行されます。

(2) <reset>,<wml:reset>,<p:reset>

§ <reset>要素の動作の無効

「1.1x」以降は、<reset>要素は無効となるため、利用することができません。

(3) mode="nowrap"指定

§ <p mode="nowrap">指定の動作

<p mode="nowrap">指定をした場合、「7.2」「1.0x」では nowrap 指定した段落のみ nowrap が有効（次の段落では無効）となります。「1.13」「1.1x」では、次の段落でも有効となります（nowrap 指定が継承されます）。
nowrap 指定の継承を解除するためには段落で明示的に<p mode="wrap">指定する必要があります。

```
<p align="center">nowrap の継承</p>
<p mode="nowrap">この段落は mode="nowrap"です。
</p>
<p>この段落は何も指定していません。</p>
<p mode="nowrap">この段落は mode="nowrap"です。
</p>
<p mode="wrap">この段落は mode="wrap"です。</p>
```

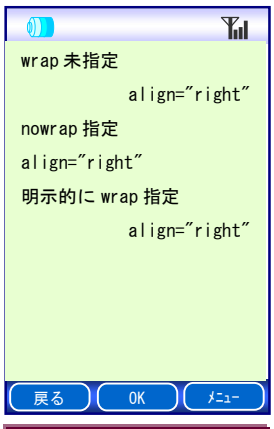
WAP2.0 対応ブラウザ 7.2 / 1.0x

WAP2.0 対応ブラウザ 1.13 / 1.1x

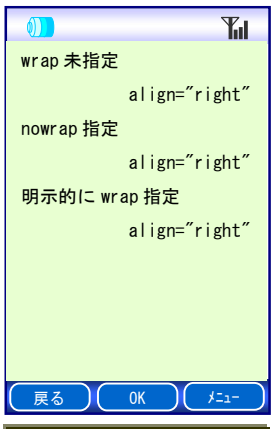
§ nowrap 指定時の"align"属性、<blink>要素の動作

「1.1x」、「1.13」では、"nowrap"指定の有効範囲では"align"属性や<blink>要素が無効となります。
「7.2」以降では、同様の場合に<blink>要素のみ無効となります。("align"属性は有効)
これらの動作を有効とするためには、明示的に"wrap"指定をして"nowrap"を解除しなければなりません。
※：「1.1x」、「1.13」では、HDML の<LINE>要素は WML の<p mode="nowrap">に変換されます。

```
<p> wrap 未指定
  <p align="right">align=&quot;right&quot;</p>
</p>
<p mode="nowrap"> nowrap 指定
  <p align="right">align=&quot;right&quot;</p>
</p>
<p mode="nowrap">明示的に wrap 指定
  <p align="right" mode="wrap">align=&quot;right&quot;
</p>
</p>
```



WAP2.0 対応ブラウザ 1.1x、1.13



WAP2.0 対応ブラウザ 1.0x

§ の有効範囲内において<a>要素内に
要素を配置した場合の動作

の有効範囲内で<a>要素内に
要素を配置した場合、「1.0x」では nowrap 指定が有効となりますが、「1.1x」から nowrap 指定が無効となり、画面の表示領域外の文字列が表示されません。nowrap 指定を有効とするためには<a>要素内で
要素を利用しないようにします。
なお、この動作は HDML の<LINE>要素の有効範囲内で<A>要素の中に
を配置した場合も同様となります。

```
<p mode="nowrap">
<a href="./foo.html" title="OK">あああああああああああああああ<br />
</p>

<p mode="nowrap">
<a href="./foo.html" title="OK">かかかかかかかかかかかかかかかか
</p>
```

WAP2.0 対応ブラウザ 1.1x 以降

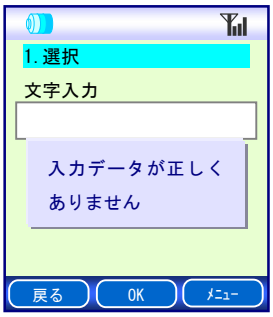
WAP2.0 対応ブラウザ 1.0x

(4) type="onpick"

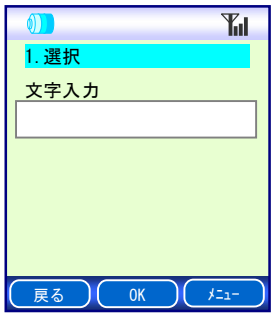
§ format 属性で指定する条件を満たさない状態での<type="onpick">の動作

「1.1x」以降では、<select>,<option>要素で<type="onpick">を利用し、<input>要素の format 属性で入力文字数（桁）を指定（例：format="MM"）した場合、入力文字が format 属性で指定する条件を満たさない状態で選択肢を選択すると、エラーメッセージ「入力データが正しくありません」が表示され、次ページ等へリンク（遷移）することができません。
これを回避するためには、format 属性で入力文字数（桁）を指定しないようにします。（例：format="*M"）

```
<p>
<select>
<option>
<onevent type="onpick">
<go href="./foo.wml" />
選択
</onevent>
</option>
</select>
文字入力
<input type="text" name="name" maxlength="200"
format="MM" />
</p>
```



WAP2.0 対応ブラウザ 1.1x 以降



WAP2.0 対応ブラウザ 1.0x

(5) 変数参照

§ <wml:>要素、<p:>要素による WML 変数の使用

「1.13」「1.1x」は、XHTML ページ内における<wml:>要素、<p:>要素による WML 記述をする場合、WML の変数参照が利用できません。

【例】:<wml:postfield name="var" value="$(var)" />

「1.13」「1.1x」は、変数の名前を変数の参照として扱わず、単に文字列として扱うため、変数参照を利用せず直接値をセットしなければなりません。

※ XHTML ページ内における<wml:>要素、<p:>要素による WML 記述ではなく、WML ページとして記述する場合は、変数参照が利用できます。

－更新履歴－

Version	日付	更新内容
1.0	2012/03/30	初版