

## 対話型 HTML エディタについて

---

池田隆一

### ソフトフロント

当社は、HTML 言語の知識を持たないユーザーが HTML を利用した文書（そのほとんどはいわゆる「ホームページ」と呼ばれるもの）を作成するための HTML エディタ「こざいく」を開発した。本稿では主に HTML 文書のブロック構造、文書間・文書内のリンクの張り方に関して 1.HTML 言語をユーザーが意識しなくても良い。2.ブラウズ時のイメージと編集イメージの間に違和感を生じさせない の2点を両立させるために「こざいく」で採用した編集形態について述べる。

Commercial WYSIWYG HTML Editor, COSAIC

---

IKEDA, Ryuichi

### SOFT FRONT

*We developed HTML editor "COSAIC" for the user who did not have knowledge of HTML language to make the document which utilized HTML. This report describes the editing form that was adopted with "COSAIC" in order to let next two characteristic be compatible: 1.not need knowledge of HTML language.2.does not produce a difference on appearance in browsing and editing.*

# 1 背景

本章では、「こざいく」の最初のバージョンが開発された当時(1995年)のHTMLエディタのターゲットユーザーと求められていた機能から導かれた「こざいく」の編集機能に関する仕様の決定方針に関して述べる。

---

## 1.1 HTML

「こざいく」を開発時に、HTMLの特徴を以下のように捕らえた。これらは現時点においても今日までWWWでHTMLにより記述された文書が大量に公開されるようになった要因の大部分であると考えている。

### 1. 文書間でリンクを張ることができる

原点にして最大の利点、文書中の任意の語、画像と他の文書を関連付け、閲覧時に容易に移動することができるというこの特徴は、複数の文書—多くの場合は複数の人により書かれ、様々な場所で公開される—を作ることの障壁をなくした。

### 2. 構造化された文書の作成が可能

文字装飾や画像といった視覚上のデザインだけでなく文書構造のデザインをユーザーが記述することが可能。空白、改行によるインデントや行頭の文字による箇条書きと異なり、テキストと文書構造の記述が分離しているため、既存のテキストの流用や文書構造の変更が容易。

### 3. 文字装飾が可能

ワードプロセッサには及ばないもののある程度の文字装飾、色、大きさの設定が提供されており、意匠をこらした文書を作成することができる。

### 4. 画像の扱いが容易

文書中の任意の位置に画像を入れたり、背景として画像を利用することができる。このことによりHTML文書の用途が広がったという利点は大きい。加えて、デザイン面でもユーザーの創作意欲を刺激したという点も見逃せない。

HTMLの本来の意図であったかは別として、特に後の2つがユーザー層の拡大を進めた要因であると思われる。

---

## 1.2 ターゲットユーザー層

「こざいく」開発当時はWWWの利用は今日と異なり、企業、自治体等による情報発信が大部分を占め個人による情報発信の目的の比率は低かった。

これらのユーザーがWebサイトを開設する際はHTML文書を作成する担当者を決めるかは外部のデザイン会社に委託するかのいずれかであるが、HTMLに関する知識を取得する機会は一

般ユーザーにほとんどなかったこともあり、どちらのケースにおいても HTML を直接記述することができる担当者はほとんどなく、

1. テキストの作成
2. 文書構成
3. 視覚的なデザイン

といったスキルに長けている人が担当する場合がほとんどであった。

この事から「こざいく」では「HTML 言語に関する知識を一切持たない人」をターゲットユーザーとしてすることとした。

---

## 1.3 文字装飾・画像・文書構造

前2節により当社では「HTML 言語の知識を持たないユーザーが HTML の特徴を容易に利用できる」という方針でユーザーインターフェイスの設計を進めることとした。

1 節であげたそれぞれの特徴についての設計方針は以下のとおり。

- リンクはリンクを張る文字列や画像を選んでそこにリンク対象をドロップする。ドロップするリンク対象をわかりやすくする。リンクはブラウザアプリケーションを利用することなく直接確認することが可能。
- 文書構造を視覚的にわかりやすく表示し、文書構造単位での編集が可能。また現在の編集対象に対して設定可能な項目がユーザーに直接提示される。
- 装飾に関しては、一般的なワードプロセッサの文書作成アプリケーションの操作性を踏襲する。すなわち選択された文字列に対してメニューやボタン操作により装飾の設定解除を行う。現在の装飾状態はできるだけメニューやボタンに反映させるというもの
- 利用しようとする画像の閲覧に特別な操作を設けない。手持ちの画像が HTML で扱えない場合も多い事が予想されるので GIF, JPEG への自動変換を提供する。

「視覚的な表示」と「不要な選択操作を減らす」という2点を操作体系やユーザーインターフェイスの設計時の最優先項目とした。

## 2 リンクの設定方法

リンクの設定は HTML に慣れていないユーザーにとっては経験したことのない操作であるため、「こざいく」では操作体系と結果の反映が直接的であることに重点を置いた。

---

## 2.1 リンク操作

「こざいく」ではリンクを設定する文字列または画像を選択し、リンク対象をドロップするという方法でリンクを設定することができる。この操作方法により大部分のユーザーにとって不慣れな事柄である URL の記述やマーカーの設定といった入力を避けることができた。

一方この方法だけではリンク対象が何らかの形で画面上に現れている必要があり、習得の進んだユーザーが直接リンク対象を記述したいという要求も予想されたので、リンク設定のためのダイアログも用意した。

この操作体系は付属のクライアントサイドクリッカブルマップ作成ツールでも採用されている。

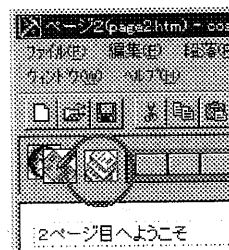
---

## 2.2 リンク対象

HTML の文法上は、単に URL がリンクの対象となり (<A HREF="(URL)"> ~</A>), リンク対象がマーカーを持つ場合はそのマーカー名を付加情報として持つことができる (<A HREF="(URL)#marker"> ) であるが、「こざいく」では、ドラッグ&ドロップで設定することが可能なリンク対象を次の3つに絞った。

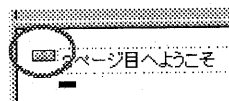
### 1. 編集している文書のトップに置かれたアイコン

文書の URL がリンクの対象となる。編集時のディレクトリ構成から相対パスで記述される。「こざいく」は Web ブラウザ機能も持っている所以他のサーバー上にあるファイルを開いてリンク対象とすることも可能。



### 2. 編集している文書内に表示されたマーカーアイコン

文書内の URL+マーカー名がリンクの対象となる。マーカーはリンク操作を行った時点で初めて設定される。またマーカー名は文書内で重複することなく自動的に決定される。



### 3. ファイル一覧に表示されたファイル名

リンクの対象が HTML ファイルでない場合も考慮して「こざいく」に付属するファイル一覧ツールからファイル名を選択してリンクの対象とすることも可能。

---

## 2.3 リンクの確認

リンク設定は文書の外見に対するフィードバックが少ないため、設定ミスに気づきにくいこともありリンク設定の確認方法が様々なフェイズで必要になると判断し、「こざいく」では次の方法を提供している。

- 編集中の文書をそのままブラウズモードに切り替えるとブラウザ同様にリンク文字列をクリックすることでリンク対象にジャンプする。ブラウズモードでは Web サーバー上のファイルもブラウズ可能なので確認できるリンク対象はユーザーの編集環境にあるファイルに限定されない。

- リンク文字列にカーソルを重ねるとステータスバー(文書窓の下部)にリンク先の URL が表示される。
- 前述のリンク設定ダイアログを開くと現在のリンク設定がデフォルト値となっている。
- リンク構造をたどって表示するツールを別途提供。

## 3 文書構造のブロック化

「こざいく」では文書構造を視覚的にわかりやすく表示することに重点を置いた。これは一般のワードプロセッサからの差別化を図ることと文書構造のデザインツールとしての用途を前面に押し出そうという意図によるものである。

本章では「こざいく」での HTML の文書構造に関する表現方法と操作体系に関して述べる。

---

### 3.1 ブロック

「こざいく」では次の HTML タグをブロックとして扱っている。

1. 段落 <P>
2. 見出し <Hn> ( $1 \leq n \leq 6$ )
3. 区切り線 <HR>
4. リスト <OL> <UL>
5. テーブル <TABLE>
6. フォーム <FORM>

この中で「リスト」「テーブル」「フォーム」は他のすべてのブロックを中に入れることができる(ただしフォームブロックの中にフォームブロックを入れることだけは出来ない)。

---

### 3.2 ブロックの入れ子構造

ブロック内部に他のブロックを入れることができる3種類のブロックはそれぞれ持ち方が異なる。逆に言えばその持ち方によりブロックが特徴付けられているともいえる。

1. リストブロック  
リストブロックは縦方向に並べられた複数個のリスト項目によってのみ構成される。リスト項目の中には1つ以上の縦方向に並べられたブロックのみを持つことができる。
2. テーブルブロック

テーブルブロックは縦横に並べられた複数個のセルによってのみ構成される。セルの中には 1 つ以上の縦方向に並べられたブロックのみを持つことができる。

### 3. フォームブロック

フォームブロックは 1 つ以上の縦方向に並べられたブロックのみを持つことができる。

---

## 3.3 ブロック構造の構成例

次のようなリストを例にブロックの構成を説明する。

```
<OL>
<LI><B>項目 1 </B>
<LI><B>項目 2 </B><HR><FONT SIZE="-1">項目 2 の補足</FONT>
<LI>項目 3 <TABLE> (中略) </TABLE>
</OL>
```

このリストは「3つのリスト項目を持つ番号付きリスト」というブロックとして特徴付けられる(図1)。

リストブロックの各項目内にそれぞれ、ブロックを持っている。第1項目は1つの段落ブロック、第2項目は段落ブロック-区切り線ブロック-段落ブロックの3つのブロック(図2)、第3項目は段落ブロックとテーブルブロックの2つのブロックを持つ。

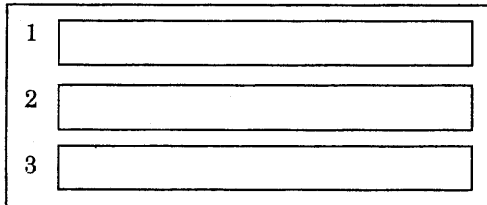


図1-リストブロック

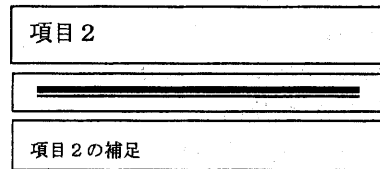


図2-項目2が持つブロック

項目内に置けるものはブロックのみであるため、「こさいく」ではプレーンなテキストエリアは存在することはない。テキストは段落ブロック(または見出しブロック)として存在する。他のブロックにテキストが存在することはない。

このようなブロック構成の利点として、

1. ブロック編集が容易に行える
2. ブロックの属性とブロック内のテキストが完全に別の物として扱われる

という点があげられる。

---

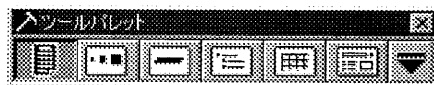
## 3.4 ブロック編集操作

ユーザーはブロックを選択してブロックの移動、複製をドラッグ&ドロップにより行うことができる。これらの操作はブロックのどの階層に対しても行うことができる。また文書間での操作も

ドラッグ&ドロップで可能。例えばリスト項目の中にある段落ブロックをドラッグしてテーブルのセルにドロップすると言ったようにリスト項目、テーブルのセル、フォームブロックといったブロックが並んでいる場所であればどこでもドロップ先とすることができる。

ブロック以外の枠線と区別するために全てのブロック枠は通常は細線で表示されるがマウスカーソルが枠上を通過する際に太線表示に変わる。また、ドラッグ時にはドロップ可能な位置を通過する際にドロップ可能であることを示すマークが現れるのでユーザーは各ブロックの階層を意識することなく文書構造のデザインを行うことができる。

ツールパレットからブロックを選んでドロップすることで新規ブロックの作成を行うこともできる。

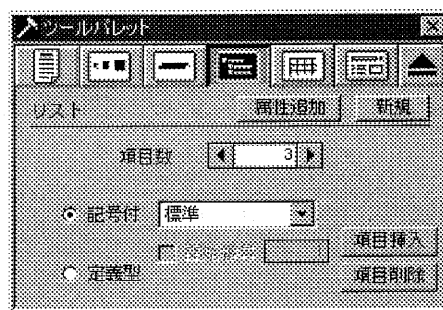


## 3.5 属性の操作

各ブロックの属性はすべてツールパレットから操作することができる。

文字入力カーソルのあるブロックまたは選択されているブロックに応じてツールパレットは表示が切り替わり表示内容も現在の選択が反映される。逆にツールパレットでパラメータの変更を行うとその場で一更新ボタンといった操作を介さずに一変更内容が文書に反映される。

例えば 3.3 で例にあげたリストが選択されている際にツールパレットは右の図のように表示される。ここで項目数を変更したり記号の種類を変更すると同時に文書内のリストブロックに変更内容が反映される。



## 4 今後の展開

今まで説明した仕様は HTML 初心者に対する配慮に重点を置いて設計されたものであるため、高度な使い方をするユーザーが細かい変更を行う際には適さない部分が現れてきた。特に新しいトピックが次々と現れる現時点においては、それらに柔軟に対応するための仕組みが要求されている。本章では現行仕様の利点を残したまま今後の展開をいかに行うかについて若干の説明を行う。

---

## 4.1 ソース編集

HTML ソース編集を可能にすることは、細かい設定ができるため便利なようではあるが一方では編集したい部分がわかり難く変更の対象を間違う可能性が高い。この問題を回避する為に次の2つの方法による提供を予定している。

1. ブロック単位でソース表示に切り替え編集を可能にする。
2. 選択されたブロックタグのオプション値の一覧を提供し、オプションの追加や値の変更を可能にする。

---

## 4.2 CSS 対応

CSS(Cascading Style Sheet)は HTML の表現力の幅が格段に広がり、再利用性も高いため今後普及が予想される。HTML エディタと独立したビジュアルな CSS エディットツールの提供と CSS を HTML 文書で利用する為の仕組みを用意する必要がある。

---

## 4.3 ブロックの部品化

HTML 文書の再利用性を高めるために、各ユーザーにとって使用頻度の高い表現を部品として保存し、保存した部品をドラッグ&ドロップにより他文書の任意の場所で利用することができるという機能の提供を予定している。