L-029

HTMLにおける日本語の空白文字の扱い方に関する提案

臼井 義比古　　西田 泰伸
富山県立大学工学部情報システム工学科

1 はじめに

1.1 純粋な文章と配置情報を含んだ文章

配置の為の改行や空白やタブなどを含まない形で計算機に記憶された文章 (以下純粋な文章) は, そのまま, 表示範囲の横幅に応じて適切に折り返して表示したり, 正しい位置で区切って音声読み上げを行なったり, 或は検索の為などの単語への分割等が行ない易い等のメリットがある.

これらの情報をテキストエディタで編集する場合は純粋な文章に文章の配置を制御するための改行や空白等を挿入した文章 (以下配置情報を含んだ文章) の形で保存しそのまま利用することも多い.

1.2 HTML4.01 の配置情報を含んだ文章から純粋な文章への変換ルール

HTML4.01 でもテキストエディタを使って文章の入力を行なう可能性が多くある事をみこんで, 純粋な文章ではなく, 配置情報を含んだ文章で記述できるようになっており, インターネットブラウザ (以下ブラウザ) での出力時などに, 配置の為の文字を削除するルールを以下のように定めている.

- 文章中に連続した空白がある場合は空白を一個にする.
- 文章中の改行コードまたは改行コードに連続する空白文字も一個の空白にする.

このルールは, 英文の様に単語と単語の間に必ず空白が入る言語 (以下一空白スクリプト) ではうまく機能し, テキストエディタで原稿を編集するときに一行の長さを調整するための改行を単語を区切る空白の位置に入れても, ブラウザが表示を行なう際にはこのルールに基づいて単語と単語の間に一つだけの空白が入った正しい文章を表示することができる.

1.3 HTMLでの零空白スクリプトの扱いの問題点

文章中に空白を入れることができない日本語の様なスクリプト (以下零空白スクリプト) の場合は, テキストエディタで原稿を編集するときに改行を入れる事ができず, テキストエディタで原稿を編集するときに不便になる.

この問題について HTML4.01 の 9.1 節では「.. conventions for inter-word space vary from script to script. In Japanese and Chinise inter-word space is not typically renderd at all」と記している. テキストエディタで編集時に改行を挿入した場合に, その改行は上記のルールに基づき空白に置き換えられるが, 日本語や中国語の場合はその空白はブラウザで表示する段階で消されるべきだということである. これは一見問題なく機能するように思えるが, 現実には日本語文中の"inter-word space"は幅が 0 でない空白としてブラウザで表示されてしまう. 実際問題として HTML 中の日本語の文章をテキストエディタで編集する場合には改行を入れないのが業界の暗黙のルールとなっている.

つまり, HTML4.01 の 9.1 節のルールは英語のような一空白スクリプトだけを想定したルールであり, 日本語や中国語のような零空白スクリプトでは不十

分な部分が残ることがわかる.

さらに上記引用のやりかたでは, もし空白が表示されなかったとしても, 表示の為の空白が文章の中に残っていることになり, 純粋な文章では無いので音声読み上げソフトでは正しく処理されない可能性が残る.

以上の事を踏まえて本論文では一空白スクリプトでも零空白スクリプトでも, 更には混在していても, 配置情報を含んだ文章から純粋な文章を得るルールを提案する.

2 前提

2.1 空白が複数種類あるスクリプトと一空白スクリプトと零空白スクリプト–国際化

本来空白文字の扱い方は各言語固有の問題であるから地上の全ての言語について調べた上で議論する必要があるが, しかし全ての言語について詳細に調べるのは現実問題として不可能に近い様に思える.

そこで今回は一空白スクリプトと零空白スクリプトの二種類についてのみ考察する. 以下では今回の一空白スクリプトと零空白スクリプトの二種類についてのみの考察でほぼ十分であることを示し, 精度の高い確認は次の機会に行なう.

現在使われている言語にはいろいろな使い方をされる空白があるかもしれないが, 我々がテキストエディタで使う ASCII の空白と違う使われ方をされる空白は, 別のコードを割当てるしかない. 複数種類の空白を持つ言語も同様に考えて今回の問題と切り放せる事がわかる. 一例を示す. タイ語の場合は通常の空白と改行時に改行文字を挿入可能な場所を示す為の空白との二種類の空白がある. タイ語の場合は後者の空白は HTML ではあまり使われていないようではあるが, もし使われていても一般の空白 (ASCII の空白) と区別可能な文字コードで表すしかない. したがってタイ語の後者の空白の存在は今回の問題には関わらない.

空白が一種類しかない場合は, 計算機が編集のための改行を取り除く作業を行なう場合は, 改行時に空白を消すか一つ残すかの選択しかないことがわかる.

例えば文章中に入るべき空白の数が文章の意味的や文法的あわせて変化する言語に対応するのは現時点では不可能であり, 挿入される空白の数は固定となる. また, 空白文字 2 個以上を長さ固定で表示するのは結局空白の長さの問題であり, 必要であれば空白を長く表示するのは表示処理の時に行なえばよい.

以上の様な理由から配置情報を含んだ文章を純粋な文章に変換するルールを考えるには, 改行と連続する空白を全て消す零空白スクリプトか一つ残す一空白スクリプトを考慮すれば十分であることがわかる.

3 提案

3.1 ルールとしての純粋な文章への変換

HTML4.01 の 9.1 節では単純な改行や空白の置き換えに関するルールという形で記されているが, 本論文では「純粋な文章に変換するルール」という考え方でこのルールを捕える.

そして HTML4.01 の 9.1 節で記述されたのが一空白スクリプトに対して配置情報を含んだ文章から純粋な文章を得るためのルールであると考え, 零空白スクリプトに対しての配置情報を含んだ文章から純粋な文章を得るためのルールと一空白スクリプトと零空白スクリプトを切り替えるルールについて追加する.

3.2 純粋な文章への変換方法の提案

3.2.1 変換ルールの決定方法

改行を含む一連の空白の変換ルールを零空白スクリプト用のルールにするか一空白スクリプト用のルールにするかの決定は, その一連の空白の直前と直後が零空白スクリプトと一空白スクリプトのどちらであるかで決定でき, それより遠くにある文字が属するスクリプトの影響を受けない.

3.2.2 空白の付け方が同じスクリプトの場合

改行を含む一連の空白の前と後が一空白スクリプト同士や零空白スクリプト同士で同じ場合は，それぞれ，一個と零個の空白を挿入する．

3.2.3 空白の付け方が違うスクリプト間の改行の場合

改行の前と後が零空白スクリプトと一空白スクリプトにわかれる場合は，挿入する空白数は 0 にする．

なぜならば，零空白スクリプトと一空白スクリプトが違うという事はそもそもスクリプトが違うのであるから，それだけで内容に区切りがあることがわかるので，空白は必要がないからである．

必要がない場所にある空白は配置の為の空白になってしまい，純粋な文章には含めることができない．

区切りがある事がわかれば，ブラウザでの出力などの場合に必要に応じた配置方法を取ることができる．たとえば，日本語 Tex ではスクリプトの変り目で四分空けと呼ばれる全角の 1/4 のサイズの空白を出力するが [2]，この考え方により HTML でもスクリプトの変り目のスペースの開け方を文章全体で統一することができるようになる．必要に応じてどのスクリプトからどのスクリプトに変るかに応じてこのスペースの開け方を変更することも可能になる．

HTML は文章の構造を記述するための言語であり，ユーザが空白文字を挿入することなどにより表示の配置をコントロールすることになじまないから，この考え方はより理想の HTML に近づいたと言える．

こまかい調整を行ないたい場合は CSS[3][4] を用いる必要が出てくる．

3.2.4 改行を含まない空白の場合

改行を含まない一連の空白も同様の処理を行ない純粋な文章を得ることができる．

これにより，全てのスクリプトの場合に，文章構造上必要な空白はあるが，配置のための空白が入っていない純粋な文章を得ることができる．

4 まとめ

一空白スクリプトと零空白スクリプトが混在する場合において，HTML に記述された配置情報を含んだ文章から純粋な文章を作る方法についての提案を行なった．これにより日本語を含む零空白スクリプトにおいてもテキストエディタで改行を入力して編集できるようになる．

これは単にテキストエディタでの利用が便利になるというだけではない．配置情報を含んだ文章から純粋な文章情報を抽出するという考え方を持つことにより，文章の表示や読み上げや検索などの処理を考える場合のより一般的な考えの筋道が見えやすくなるという効果も考えられる．

参考文献

1 W3C, "HTML 4.01 Specification", http://www.w3c.org/TR/html401/", 1999

2 倉沢 良一,「TEX システムの日本語化」,1987

3 W3C,"Cascading Style Sheets, level2 CSS2 Specification",1998

4 W3C,"Cascading Style Sheets, level2 CSS2 Revision 1(CSS 2.1) Specification",2009