

# コンテンツの空間的相対関係に基づく CSS の分析

## An Analysis of CSS Descriptions based on Spatial Relationship

岡重 匡俊<sup>†</sup>      福安 直樹<sup>‡</sup>      満田 成紀<sup>‡</sup>      松延 拓生<sup>‡</sup>      鯨坂 恒夫<sup>‡</sup>  
 Masatoshi Okashige   Naoki Fukuyasu   Naruki Mitsuda   Takuo Matsunobe   Tsuneo Ajisaka

### 1. はじめに

ウェブサイトの視覚的な表現を記述するための言語として、CSS (Cascading Style Sheets) が広く利用されている。CSS は文書の構造から分離されたレイアウトを記述するためのスタイルシートであり、ボックスモデルに基づいて中身、余白、境界線について要素ごとに記述する。それぞれの要素にはブロックレベル要素とインラインレベル要素の2つに大きく分かれ、他の要素と主従関係を持つことがある。その一方で、CSS ではどの要素を主とするかによって、同じ視覚的な表現に対して複数の記述方法が存在する。これは CSS の実装作業において、特に複数人で行う場合には異なる記述方法が混在し、後の保守に影響を与えてしまう可能性がある。そこで本研究では、ブロックレベル要素のボックスモデルを対象に、要素の空間的相対関係に基づく CSS の分析を行う。視覚的な表現を構成するデザイン要素の1つである空間に着目し、コンテンツの空間的相対関係から既存のウェブサイトで記述されている CSS を分類する。また、分類の結果からコンテンツの空間的相対関係に基づく視覚的な表現を抽出し、それぞれの記述方法について分析することで、どの記述方法がどのような影響を与えるかを明らかにする。

### 2. 空間的相対関係

文献 [1] では、視覚的な表現を構成する以下の7つのデザイン要素を紹介している。視覚的な表現はそれらを組み合わせることで実現される。

- Line (線)
- Color (色)
- Shape (形状)
- Texture (質感)
- Space (空間)
- Form (外形)
- Typography (タイポグラフィー)

またデザイン要素ごとに特徴があり、例えば線は太さや方向、長さを持つ。そのため、CSS では視覚的な表

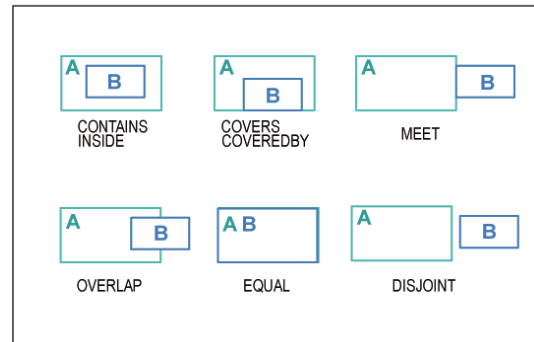


図 1: 空間的相対関係

現を実現するために数多くのプロパティや値が定義されている。フォントについて表現する場合は、サイズ、スタイル、ウェイトといったプロパティがあり、またサイズには px, small, large といった値を記述することで指定することが出来る。これによって、特徴を活かした多彩な表現がウェブサイト上で実現できるようになっている。

しかし、CSS を記述する場合において空間は他の要素とは異なる。線や色などと違い、空間は他のコンテンツとの相対関係によって表現されるため、CSS では主とするコンテンツと従とするコンテンツの相対関係から記述方法が異なる場合がある。そのため、CSS において空間が関わる視覚的な表現にはいくつかの記述方法が存在する。

空間の相対関係は図 1 に示すような 6 種類に分けられる [2]。これは二次元上にある 2 つのオブジェクトの関係について名称を与えたものである。

#### MEET

従とするコンテンツが主とするコンテンツの外側で接触する関係。

#### COVERS/COVEREDBY

従とするコンテンツが主とするコンテンツの内側で接触する関係。

#### CONTAINS/INSIDE

従とするコンテンツが内包し、主とするコンテンツが外延する関係。

#### OVERLAP

従とするコンテンツが主とするコンテンツの部分的

<sup>†</sup> 和歌山大学大学院システム工学研究科, Graduate School of Systems Engineering, Wakayama University

<sup>‡</sup> 和歌山大学システム工学部, Faculty of Systems Engineering, Wakayama University

表 1: MEET

方法	1	2	3
CSS記述	<pre>#content1{ float: left; } #content2{ float: left; } #content3{ float: left; }</pre>	<pre>#content1{ left: 0px; } #content2{ left: 192px; } #content3{ left: 384px; }</pre>	<pre>#content1{ display: table-cell; } #content2{ display: table-cell; } #content3{ display: table-cell; }</pre>
視覚的な表現	二段組, 三段組, 四段組 コンテンツの並び	コンテンツの並び	二段組, 三段組 コンテンツの並び

一致となる関係.

### EQUAL

従とするコンテンツが主とするコンテンツと重複する関係.

### DISJOINT

従とするコンテンツが主とするコンテンツと支離する関係.

## 3. CSS による空間的相対関係の記述 (事例)

図 1 で示した 6 種類の空間的相対関係から, 優良デザインサイト [3] で紹介された 30 個のウェブサイトから記述されている CSS を分類する.

### 3.1 MEET

- 主とするコンテンツを基準に従とするコンテンツを寄せて並べる.
- 各コンテンツの位置を直接指定して並べる.
- 表組からコンテンツを配置して並べる.

MEET に該当する記述方法は 3 つに分類され, 結果は表 1 となる. ウェブサイト上では MEET に該当するのはコンテンツを並べる場合であり, コンテンツの並べ方によって記述方法が異なる. その中で最も多く利用されていた記述方法は float を使用した方法であり, 該当した視覚的な表現に二段組, 三段組, 四段組, コンテンツの並びがある. また, 各コンテンツを直接指定して配置する記述方法はコンテンツの並び, 表組を指定してコンテンツを配置する記述方法は二段組, 三段組, コンテンツの並びが該当する.

### 3.2 COVERS/COVEREDBY

- 主とするコンテンツの中で 2 つのコンテンツを左右に分けて配置する.
- 主とするコンテンツの中で従とするコンテンツを真ん中に配置する.

COVERS/COVEREDBY に該当する記述方法は 2 つに分類され, 結果は表 2 となる. 主とするコンテンツの

表 2: COVERS/COVEREDBY

方法	1	2
CSS記述	<pre>#parentcon{ width: 940px; } #childcon-l{ float: left; width: 622px; } #childcon-r{ float: right; width: 300px; }</pre>	<pre>#parentcon{ display: block; } #childcon{ width: 980px; margin: 0 auto; }</pre>
視覚的な表現	二段組, 三段組, 四段組 コンテンツ間のホワイトスペース	センタリング

表 3: CONTAINS/INSIDE

方法	1
CSS記述	<pre>#content1{ margin: 0 302px 0 172px; } #content2{ float: left; margin-left: -100%; } #content3{ float: right; margin-left: -302px; }</pre>
視覚的な表現	二段組, 三段組

中で 2 つのコンテンツを左右に分けて配置する記述方法は, 該当した視覚的な表現に二段組と段とするコンテンツ間のホワイトスペースで記述される. これは 2 つのコンテンツを合わせた幅が主とするコンテンツの幅よりも小さい場合に余白を生み出すためである. また, 三段組や四段組でも記述されていて, これは二段組の段としたコンテンツの中でさらに二段組を形成することで三段組や四段組を表現するためである.

主とするコンテンツの中で従とするコンテンツを真ん中に配置する記述方法はセンタリングだけであり, 他の視覚的な表現では該当しない.

### 3.3 CONTAINS/INSIDE

- ネガティブマージンで従とするコンテンツを配置する.

COVERS/COVEREDBY に該当する記述方法は 1 つだけである. この記述方法は表 3 のようにコンテンツに負のマージンを指定し, 重ね合わせるようにしてコンテンツを配置する方法である. 一部のウェブサイトで用いられ, 該当した視覚的な表現に二段組, 三段組がある.

表 4: EQUAL

方法	1	2	3
CSS記述	<pre>#content{ display: block; z-index: 50; } #hiddencon{ display: none; z-index: 0; }</pre>	<pre>#content{ display: block; opacity: 1; } #hiddencon{ z-index: 2; display: block; opacity: 0; }</pre>	<pre>#content{ display: block; z-index: 1; } #hiddencon{ display: block; z-index: 0; }</pre>
視覚的な表現	コンテンツの重ね合わせ	コンテンツの重ね合わせ	コンテンツの重ね合わせ

### 3.4 OVERLAP

OVERLAP に該当する記述方法は存在しない。これは OVERLAP に該当する視覚的な表現はウェブサイト上ではレイアウトが崩れ、コンテンツが適切に配置されていない場合が多いためである。

### 3.5 EQUAL

1. 従とするコンテンツを消すことで主とするコンテンツを表示する。
2. 従とするコンテンツを透過させて主とするコンテンツを表示する。
3. 重ね順を入れ替えて主とするコンテンツを表示する。

EQUAL に該当する記述方法は 3 つに分類され、結果は表 4 となる。3 つの記述方法は視覚的な表現であるコンテンツの重ね合わせに該当し、コンテンツの重ね順を `z-index` で指定している上で、従とするコンテンツを表示のままか、どのような方法で非表示にするかで記述方法が異なる。

### 3.6 DISJOINT

1. 主とするコンテンツに余白を指定する。
2. 従とするコンテンツに余白を指定する。
3. どちらのコンテンツにも余白を指定する。
4. 余白用のコンテンツを HTML 上で作成し、余白の幅を指定する。
5. 横並びを指定してコンテンツを配置する。

DISJOINT に該当する記述方法は 5 つに分類され、結果は表 5 となる。横並びを指定してコンテンツを配置する記述方法以外は、視覚的な表現にコンテンツ間のホワイトスペースが該当する。MEET の関係であった並べられたコンテンツに余白を指定することで、コンテンツ間のホワイトスペースを表現する。主とするコンテンツに余白を指定する記述方法は、一番上のコンテンツから余白を指定していき、一番下となるコンテンツでは余白

表 5: DISJOINT

方法	1	2	3
CSS記述	<pre>#content1{ float: left; margin: 0 12px 0 0; } #content2{ float: left; }</pre>	<pre>#content1{ float: left; } #content2{ float: left; padding-left: 240px; }</pre>	<pre>#content1{ float: left; padding-right: 240px; } #content2{ float: left; padding-left: 240px; }</pre>
視覚的な表現	コンテンツ間のホワイトスペース	コンテンツ間のホワイトスペース	コンテンツ間のホワイトスペース

方法	4	5
CSS記述	<pre>#contant1{ } #spacecon{ height: 20px; } #content2{ }</pre>	<pre>#content1{ display: inline-block; } #content2{ display: inline-block; } #content3{ display: inline-block; }</pre>
視覚的な表現	コンテンツ間のホワイトスペース	コンテンツの並び

を指定しない方法である。従とするコンテンツに余白を指定する記述方法は、一番上のコンテンツでは余白を指定せず、次のコンテンツから余白を指定する方法である。どちらのコンテンツにも余白を指定する記述方法は、中身によって余白の指定を変更するか、しないかでさらに 2 つの方法に分かれる。余白用のコンテンツを HTML 上で作成し、余白の幅を指定する方法もある。一方で横並びを指定してコンテンツを配置する記述方法では余白を指定しない。これはコンテンツに `inline-block` を指定する場合、コンテンツ間に余白が生まれるからである。

## 4. コンテンツの空間的相対関係に基づく CSS の分析

コンテンツの空間的相対関係に基づく CSS の記述を分類した結果、複数の記述方法が存在した。また、コンテンツの空間的相対関係に基づく視覚的な表現が存在し、複数の記述方法がある視覚的な表現は段組、コンテンツ間のホワイトスペース、コンテンツの重ね合わせ、コンテンツの並びであることを明らかにした。この結果から、視覚的な表現に基づいてそれぞれの記述方法がどのような影響を与えるか分析する。

### 4.1 段組

段組の記述方法は大きく分けて、主とするコンテンツの中で 2 つのコンテンツを左右に分けて配置する方法、

主とするコンテンツを基準に従とするコンテンツを寄せて並べる、表組からコンテンツを配置して並べる方法、ネガティブマージンで配置する方法の4つが存在する。

まず、主とするコンテンツの中で2つのコンテンツを左右に分けて配置する方法は、段と段の間の余白を指定する必要がない。つまり `margin` と `padding` を記述する必要がなく、また保守性に優れるといえる。ただし他の記述方法と違い、必ず左右の幅を指定する必要があることや、段と段の間のホワイトスペースを考慮して幅を指定する必要がある。この記述方法では二段組からその段としたコンテンツを主とした上で、もう一度二段組を形成することで三段組を表現する記述方法や、本文部分、メニューリスト、他サイトのバナーリストで構成されていた場合に `float left`, `float right` を利用し、区別する記述方法も存在する。また、二段組では主とするコンテンツを `float left`, `float right` のどちらで記述するかで分かれ、理由として本文部分がどちらに配置しているかで異なっている。

次に、主とするコンテンツを基準に従とするコンテンツを寄せて並べる方法は、一方向に寄せてから余白を指定することで細かい表現や、段とするコンテンツの横幅をパーセントで指定することでリキッドレイアウトを実現することが出来る。しかし余白の指定を行うため柔軟性は他の記述方法よりも低く、変更の際に記述内容を変えないといけない場合がある。この記述方法では必ずホワイトスペースを組み合わせて表現している。

次に、表組からコンテンツを配置して並べる方法は、全てのブラウザに対応できる一般的な記述方法として、2000年代初頭あたりまで利用していた。しかし、記述内容が複雑になることや、本来の表組の用途とは外れた方法であるため、W3C (World Wide Web Consortium) では推奨されないやり方である [4]。

最後に、ネガティブマージンで配置する記述方法はどちらか一方の横幅を固定し、もう一方を可変する場合に利用できる。しかしこれも表組と同様に記述内容が複雑になることがある。また、古いブラウザに対応しない場合がある。

段組の記述方法が一番多く、主とするコンテンツの基準や中身の関連性によって記述方法が異なっていた。また、CSS3.1では新たに段組を生成する指定として、フレキシブルボックスレイアウトが定義している [5]。フレキシブルレイアウトを指定することで、`float` や `table` を記述せずに段組を表現することが出来るため、今後はさらに記述方法が増えると考えられる。

## 4.2 ホワイトスペース

ホワイトスペースの記述方法では大きく分けて、コンテンツの一方向に余白を指定する方法、それとも両方向

に余白を指定する方法の2つが存在し、他の視覚的な表現から記述方法は異なる。

まず、どちらか一方向に余白を指定する方法は指定や変更するのが容易である。またお互いに `margin` を持っているときに片方だけ余白を指定することで、大きい値の余白の指定が適用され、小さい余白の指定は無視されるようになるため、細かい指定がしやすい。しかし、上方向なら一番初め、下方向なら一番最後のコンテンツに余白を指定してしまうとレイアウトを崩してしまう恐れがあるため、他のコンテンツとは別に作成する必要がある。

両方向に余白を指定する場合は別のコンテンツを作成する必要はないが、余白の指定や変更が面倒である。また一番初めと一番最後のコンテンツの余分な方向にも余白を持つため、視覚的にレイアウトが崩れた表現になってしまう場合がある。

これは並べたコンテンツ間のホワイトスペースでも同様であり、よく利用されているのは片一方に余白を指定する場合である。二段組や三段組などの段と段の間のホワイトスペースで余白を指定する場合は片一方に余白を指定する。これはレイアウトが崩れるのを防ぐためである。

## 4.3 コンテンツの重ね合わせ

コンテンツの重ね合わせでは従とするコンテンツを消すことで主とするコンテンツを表示する方法、従とするコンテンツを透過させることで主とするコンテンツを表示する方法、重ね順を入れ替えて主とするコンテンツを表示する方法の3つが存在し、従とするコンテンツの指定によって記述方法が異なる。

3つの大きな違いに切り替えの演出方法がある。表示していたコンテンツを非表示にする場合に切り替わるだけのような演出か、少しずつ消える演出にするかで記述方法が異なる。少しずつ消える演出では不透明度を変化させる必要があるため、透過処理を行う `opacity` を記述していることが多く、これは CSS だけでは変更することが出来ない。

また、コンテンツの重ね合わせでは重ね合わせ順を変化させるために動的処理を行う場面が多く、またスライド式などのアニメーションを用いるため、CSS だけではなく、jQuery や Flash といった記述も関わる。

## 4.4 コンテンツの並び

コンテンツの並びでは主とするコンテンツを基準に従とするコンテンツを寄せて並べる方法、各コンテンツの位置を直接指定して並べる方法、横並びを指定してコンテンツを配置する方法、表組からコンテンツを配置して並べる方法の4つが存在し、主とするコンテンツによって記述方法が異なる。

まず、主とするコンテンツを基準に従とするコンテンツを寄せて並べる方法は上下左右に `margin` を指定することが出来るため、細かい表現が可能で、従とするコンテンツもそれに合わせて寄せることが出来る。しかし、並びのコンテンツ間のホワイトスペースを表現する場合は一方向に余白を指定し、左方向なら一番初め、右方向なら一番最後のコンテンツを他のコンテンツとは別に作成する必要がある。

次に、各コンテンツの位置を直接指定して並べる方法は `position` などから直接指定することで細かい指定や、ホワイトスペースを利用してグルーピングすることも出来る。しかし、他の記述方法と比べて柔軟性が低いため、変更する場合は他のコンテンツも変更する必要がある。

次に、横並びで指定して従とするコンテンツを配置する方法は、`display:inline-block` を指定することで表現できるため、他の記述方法と比べて記述が容易である。しかし、改行した場合に隙間が発生することやブラウザによっては対応しない [6]。

最後に、表組から従とするコンテンツを配置する方法は、`display:table-cell` を指定することで表現できるため、`display:inline-block` 同様、記述が容易である。また、改行しても隙間が出来ない。しかし、`margin` を持たないため、余白を指定する場合は `padding` で記述する必要がある。

#### 4.5 考察

それぞれの視覚的な表現から、記述方法がどのような影響を与えるのかを明らかにした。その中で段組やコンテンツの並びのようなコンテンツを配置する場合に記述方法が異なりやすく、それぞれの与える影響が違った。原因として主とするコンテンツの基準、従とするコンテンツの指定、中身の関連性の3つがある。

まず段組では主とするコンテンツの配置に HTML で記述した本文部分となるコンテンツが重要であり、コンテンツの並びであれば主とするコンテンツの形式から記述方法が異なる。このため、主とするコンテンツの基準が必要である。

次に従とするコンテンツの指定だが、段組とコンテンツの並びでは従とするコンテンツの余白によって記述方法が異なり、コンテンツの重ね合わせは表示するか、非表示にするかで記述方法が異なる。このため、従とするコンテンツの指定が必要である。

最後に中身の関連性だが、三段組では他サイトのリンク集などを切り離すために記述方法を変えていることがある。これは四段組でも見られ、中身の関連性からコンテンツを区別し、それによって記述方法が異なる。また、コンテンツ間のホワイトスペースでは中身によって余白の指定を変更している場合もあり、それによって記

述方法が異なる。このため、中身の関連性の把握が必要である。

それぞれの点に着目し、記述方法を決定することで、コンテンツの空間的相対関係に該当する視覚的な表現に基づく記述方法の混在はなくなると考えられる。

## 5. おわりに

本研究ではコンテンツの空間的相対関係に基づく CSS の分析を行った。

視覚的な表現を構成するデザイン要素の1つである空間から、コンテンツの相対関係によって記述方法は異なる点に着目し、ウェブサイトからコンテンツの空間的相対関係に基づく CSS の記述を分類した。また、分類した結果から複数の記述方法がある視覚的な表現を明らかにし、それぞれの記述方法がどのような影響を与えるのか分析した。その結果、コンテンツの空間的相対関係に基づく視覚的な表現の記述方法に、主とするコンテンツの基準、従とするコンテンツの指定、中身の関連性が関係していることを明らかにした。

## 参考文献

- [1] 感覚派デザイナーも知っておいて損はない「デザイン要素と原則」,  
“<http://webnaut.jp/design/620.html>”.
- [2] M. J. Egenhofer, A Formal Definition of Binary Topological Relationships, Foundations of Data Organization and Algorithms, Lecture Notes in Computer Science, Vol.367, pp.457–472, 1989.
- [3] HTML クイックリファレンス/優良デザインサイト,  
“<http://www.htmq.com/design/index.shtml>”.
- [4] Web Content Accessibility Guidelines 1.0, 5 May 1999,  
“<http://www.w3.org/TR/WAI-WEBCONTENT/>”.
- [5] CSS Flexible Box Layout Module Level 1, 14 May 2015,  
“<http://www.w3.org/TR/css-flexbox-1/>”.
- [6] 株式会社アंक 著, CSS 辞典 第5版 [CSS2/3/4 対応], 翔泳社, 2014.