



HTML5

MD31

#1

01 Webアクセシビリティ

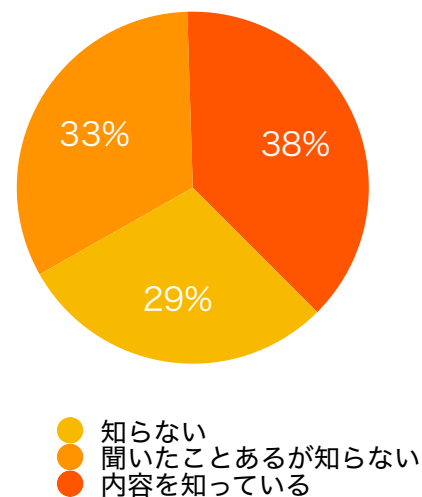
制作現場だけでなく、企業が取り組んでいるWebアクセシビリティは、いまや知らないと恥ずかしいレベル

アクセシビリティとは、「**アクセスのしやすさ**」を意味し、Webアクセシビリティは「**Webページにある情報や機能の利用のしやすさ**」のことを指します。少し具体的にいうと「**高齢者や障害者など年齢的・身体的条件にかかわらず、Webで提供されている情報にアクセスできること**」という意味になります。視覚・聴覚、あるいは機器の操作にハンデを持つ人でも、**問題なくWebにページにある情報を利用できること**を目指したガイドラインになります。



Web現場でもの6割超が知らない状態

2020年9月にFreee・サイバーエージェント・サイボウズの3社がWeb関連企業従事者807人に行った「Webアクセシビリティ」に関するアンケート調査の結果では、Webアクセシビリティの認知について「内容を知っている」38%、「聞いたことはあるが内容は知らない」32.7%、「知らない」29.3%と、**6割超が内容を知らない状態**でした。また、「Webアクセシビリティ向上に向けて活動している人・部署が存在するか」の質問では、「存在する」21.9%、



「存在しない」49.4%、「わからない」29.6%という状況です。さらに、「提供しているWebサービスに、Webアクセシビリティの向上が必要か」の質問では、「とても必要だと思う」21.0%、「どちらかといえば必要だと思う」32.9%となり、**過半数が必要性を認識する一方で必要と感じない認識も過半数**いるということになります。

Webアクセシビリティとは

「Access（アクセス）+ billity（可能性）＝利用できること」

世界中の誰もが問題や障害に関係なく利用できることになります。

これは、WWWの生みの親であるティム・バーナーズ・リーの有名な言葉と同じです。

The power of the Web is in its universality. Access by everyone regardless of disability is an essential aspect.

Tim Berners-Lee, W3C Director and inventor of the World Wide Web



「Web」のパワーはそのユニバーサル性にある。障害に関係なく誰もがアクセスできることは、「Web」の根幹をなす要素の一つである。」

W3Cが1999年に**WCAG（Web Contents Accessibility Guidelines）**を策定したものがWebアクセシビリティになります。2008年にはWCAG2.0を公開し、2012年にISO/IEC国際規格として承認され、2018年に**WCAG2.1**に拡張しました。

日本では2004年に**JIS（日本工業規格）でx8341-3（高齢者・障害者等配慮設計指針 第3部ウェブコンテンツ）**が公表され、2010年に**WCAG2.0に沿って改定**されました。この時に「高齢者・障害者等」は「様々な能力を持つ最も幅広い層の人々」と定義しています。



WCAGではウェブデザイナーやエンジニアなど
様々な人を対象とし、どのようにすればウェブコ
ンテンツがアクセシブル（利用しやすく）になる
のかを提供しています。 ▶ [WCAG 2.1](#)

Webメディアと他のメディアの違い

テレビやラジオのほか、新聞や雑誌などのメディアとWebは何が違うのかを
知ることがアクセシビリティ を押さえるポイントになります。

新聞 / 雑誌（印刷物）



- ▶ 視覚への情報伝達に優れたメディア
- ▶ 文字（言葉）と色・形で情報を伝達→印象・雰囲気がある程度伝わる
- ▶ 緊急性のある情報の発信にある程度対応できる
- ▶ 視覚障害を抱える人へは点字印刷である程度対応できる

ラジオ（音声）



- ▶ 聴覚への情報伝達に優れたメディア
- ▶ 声（言葉）と音だけで情報を伝達→印刷物より表情豊かに伝わる
- ▶ 緊急性のある情報の発信に優れている
- ▶ 聴覚障害を抱える人は骨伝導機器などで利用可能

テレビ（音声・映像）



- ▶ 視覚・聴覚への情報伝達に優れたメディア
- ▶ 文字・色・形・音声・動きで情報を伝達→表現・表情が豊かに伝わる
- ▶ 緊急性のある情報の発信に優れている
- ▶ 視覚・聴覚障害のどちらかであればカバーできる

WEB



- ▶ 視覚・聴覚・触覚への情報伝達に優れたメディア
- ▶ 文字・色・形・音声・動きで情報を伝達→表現・表情が豊かに伝わる
- ▶ 緊急性のある情報の発信に最も優れている
- ▶ 視覚・聴覚障害の両方を抱えていても点字ディスプレイなどで利用可能

視覚障害者のインターネット利用率は91.7%、聴覚障害で93.4%、肢体不自由者で82.7%、障害者全体では
53%、高齢者による利用率も年々増加傾向（総務省：平成24年調査研究結果より）

Webメディアの特徴

他のメディアと比較しても最強そうなWebですが、「利用のしにくさ」がWebの特
徴でもあるため「Webメディアの特徴」を理解して、よりアクセシビリティ への対
応が求められます。

1. 情報の一覧性が低い
表示画面内に情報が収まらない場合のレイアウトやアナウンスに配慮
2. 情報の視認性・可読性が低い
小さい文字や文書構造、行間や文字数、背景と文字色などへの配慮
3. 情報の信憑性が低い
情報の作成・発信が容易なため、情報の正確性が求められる
4. 情報が断片的、直感的な判断が求められる
何ができるのか、何があるのか、どこにあるのか、どうすれば良いか など親
切・丁寧な対応が必要
5. 双方向性や対話性が求められる
「どうすれば良い」のかを対話性を生かして誘導し、「わからない」ことなど
へは双方向性を生かしたコミュニケーション機能を活用

アクセシビリティ の基本10項目

WCAGのガイドラインを全て網羅できなくても、最低限必要な10項目は覚えておく
と良いでしょう。

1. ページの内容がわかる「ページタイトル」を記述
2. 見出し・リストなどの文書構造を適切にマークアップ
3. リンクテキストはリンク先がわかる文言にする
4. 情報を伝えている画像にalt（代替）テキストを提供
5. 情報を伝える配色に注意する
6. ユーザーがコンテンツを拡大表示できるようにする
7. キーボードだけでも操作可能にする
8. フォームのラベルや説明をマークアップ
9. エラーメッセージはエラー箇所と修正方法を明示
10. 動画はキャプション（字幕）を提供

02 HTML5

アクセシビリティとも関係が深いHTMLはver.5以降、よりアクセシビリティへの対応が進んでいる

HTML5は、それまでW3Cによって管理されブラウザの進化とともにHTMLもバージョンアップしてきました。1993年にHTML1.0の仕様書が作られてから、1999年のHTML4.1の勧告まで順調にバージョンアップが進みました。2000年にXHTML1.0が登場し、その後HTML4.1とXHTML1.0の共存時代が14年も続きます。HTMLのバージョンアップに興味を示さないW3Cに対し、2004年ウェブアプリケーションのためのHTMLと各種APIを整備し発展させる組織「WHATWG（Apple, Mozilla, Operaの元開発者が設立）」がHTML5の開発を開始したのをきっかけに、2007年W3CもWHATWGと共同開発をしていたのですが、2012年に共同開発を中止し個別で開発を進め、2014年HTML5がW3Cより正式に勧告されました。

この14年間でHTML5の情報がリークし、Webの信憑性の低さも相まって間違った（古い）HTML5の知識・技術が拡散され、まだその事実気づかない制作者や2000年で知識で止まったままの制作者が、現場で大暴れしている状態です。

現在のHTML5は2017年12月勧告のver.5.2で、W3Cが進めるver.5.3への準備は2019年に中止となり、以降WHATWGがHTML Living StandardでHTMLの最新仕様が確認できるように更新している状態です。HTML5.1や5.2を総称してHTML5と呼ぶため、柔軟に最新情報へ対応するスキルが必要です。

HTML5 & Friends

HTML5は従来のHTMLのバージョンと比較して、新しい要素や属性に加え動作が追加・変更された新しいHTMLとしてバージョンアップしました。これにより、より多彩でパワフルなWebサイトやアプリケーションを構築する大きな一連の技術です。

HTML5 & Friends



SEMANTICS

コンテンツが何であるかよりはっきりと表現可能



STYLING

さらに洗練されたCSSデザインが可能



MULTIMEDIA

ビデオやオーディオの使用や操作が可能



OFFLINE & STORAGE

オフラインでもよりよい動作をするよう最適化が可能



CONNECTIVITY

新しい方法でサーバーと通信することが可能



DEVICE ACCESS

より多くの入力・出力デバイスを使うことが可能



3D.GRAPHICS & EFFECTS

グラフやオブジェクトの表示や描画が可能



PERFORMANCE & INTEGRATION

さらにスピーディなWebサイトやアプリケーションの表示が可能

Webサイト制作に必要なHTML5は主に「SEMANTICS」と

「STYLING」の部分で、「SEMANTICS」は従来のHTMLと同じで、文章の内容をより伝えやすくするために、タグの意味や役割を細分化し追加・変更され、「STYLING」は今までJavascriptを使った動的な表現もデザインできるCSS Level3を中心としたデザインの部分です。

HTML5からの変更点

2017年12月に勧告されたHTML5.2は、大幅な変更はなかったもののタグや属性の細かな仕様変更がされ、さらにHTML Living Standardで新たな仕様が更新されています。HTML5以降で変更された全てのタグは、最新バージョンのブラウザで実装されることが多いため、全てのタグを覚える必要はありません。使用頻度の高い、注意の必要な変更などは知っておきましょう。

追加・変更・削除された主なタグ

| タグの追加・変更・削除 | HTML4.1 | HTML5 | HTML5.1 | HTML5.2 | HTML Living Standard |
|--|---------|-------|---------|---------|----------------------|
| <article> <aside> <audio> <canvas> <figure> <footer> <footer> <header> <main> <nav> <section> <source> <summary> <video> | × | ○ | ○ | ○ | ○ |
| <details> <picture> | × | × | ○ | ○ | ○ |
| <dialog> | × | × | × | ○ | ○ |
| <hgroup> | × | × | × | × | ○ |
| <I> <s> <small> <u> | △ | ○ | ○ | ○ | ○ |
| <a> <address> <blockquote> <hr> <i> | ○ | ○ | ○ | ○ | ○ |
| <div> <style> | ○ | ○ | ○ | ○ | ○ |
| <center> <frame> <frameset> | △ | × | × | × | × |

- 特に赤文字タグと以下の内容は覚えておくとい良いでしょう。
- <div>は<dl>内で使用可能
 - <style>は<body>内で使用不可
 - <hr>は装飾目的から話が切替わる区切り線として仕様変更
 - <small>は注釈文に仕様変更
 - <address>は住所から連絡先に仕様変更

追加・変更された主な属性

| 属性の追加・変更・削除 | Ver | 内容 |
|--|-------|-----------------|
| <input> <textarea> placeholder属性 | 5 | 入力のヒント |
| <input> <textarea> <select> require属性 | | 必須入力 |
| reversed属性 | | 逆順序リスト |
| <source> size属性 | 5.1 | 画像を表示するサイズ |
| <source> srcset属性 | | 使用可能な画像ファイルのセット |
| alt属性 | | 省略不可 |
| loading属性 | H L S | 画像の読み込み時間を指定する |

この他にも以下のような変更点は覚えておく必要があります。

- のように要素名を大文字で記述できない
- スタイリング（装飾など）はCSSを使い、HTMLではない
- 文字エンコーディングはUTF-8のみ
- インライン/ブロック要素の分類が廃止

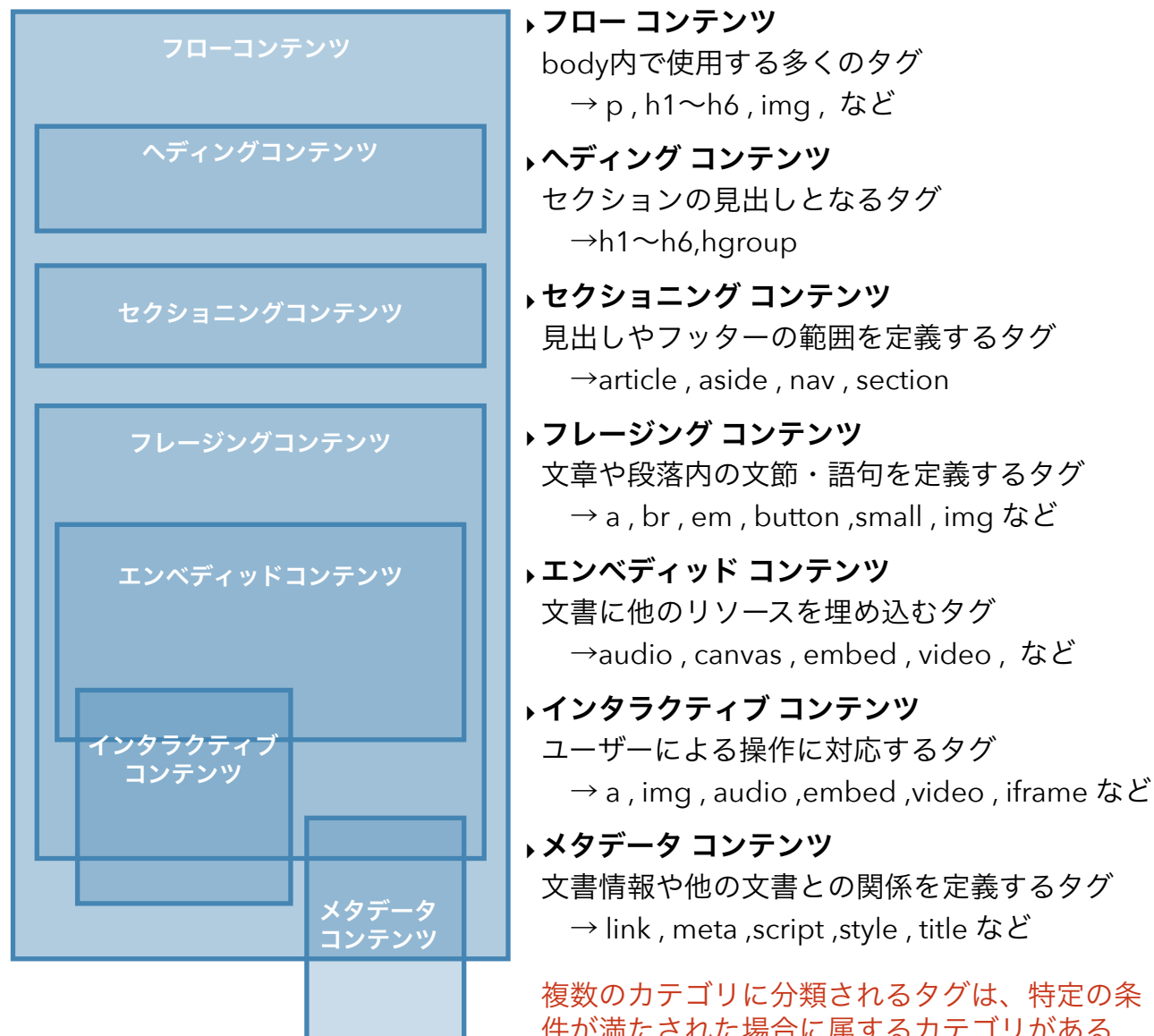
HTML5でよりアクセシブルに

常に正しい目的でHTMLタグを使用しているか確認するだけで、多くのWebページがアクセシブルになります。これはPlan Old Semantic HTML（簡潔な昔ながらの意味的なHTML）と呼ばれ、できる限り相応しいHTMLタグを、相応しい文章で使うということで、使用するタグの正確さがアクセシビリティを高めます。障害者・高齢者だけでなく、Webページにアクセスしやすくなるということは、検索サイトの検索結果からもアクセスしやすくなるため、SEOにも関係します。

03 コンテンツモデル

インライン/ブロックの分類が廃止され、文書構造を表す意味に基づいた新たな分類（セマンティックス）

コンテンツモデルは「それぞれのタグで、どんなタグを使う（含む）ことができるか」を定義した配置ルールのこと、HTML5の各タグは以下の7つのカテゴリ（種類）で分類されています。



フレーズングコンテンツである<small>は注釈文で使うタグで、具体的な使用場所は「※」を付けるような補足や、小さく赤字で付け加える文、フッターに表記するcopyright（著作権は©で©）表記などでマークアップします。

HTML

```
<small> &copy; 2021 HAL OSAKA </small>
```

左の図を見ると、フレーズングコンテンツはフローコンテンツに含まれているので「文章に適切なフローコンテンツのタグが、<small>の親タグで配置できますよ」というルールです。

HTML

```
<p>
・ ・ <small> &copy; 2021 HAL OSAKA </small>
</p>
```

下のソースコードのように<small>を親、<p>を子のタグとしてマークアップするのは間違いということになります。

HTML

```
<small>
・ ・ <p> &copy; 2021 HAL OSAKA </p>
</small>
```

また<p>や<a>は少し複雑なルールがあるタグですが、「段落」「リンク」という意味に従ってマークアップすれば問題ありません。例えば<p>の親に<a>をマークアップしても良いとなっています。

HTML

```
<a href="#">
・ ・ <p> ここをクリック </p>
</a>
```

04 ページの基本構造とレイアウト構造

ページ内の情報が散乱しないように、情報を正しく整理分類することが、柔軟なレイアウトに対応する

Webページに掲載される情報には、**ファイルの先頭や最後に定型として付け加えられる部分**をヘッダー<header>・フッター<footer>、サイト内の**移動を補助**するナビゲーション<nav>、**ページの本文**である<main>の4つに分類することができます。ページの**基本構造**となる<header> <nav> <main> <footer>はレイアウトの骨子になり、いくつかの**基本レイアウトのパターン**として定着しています。

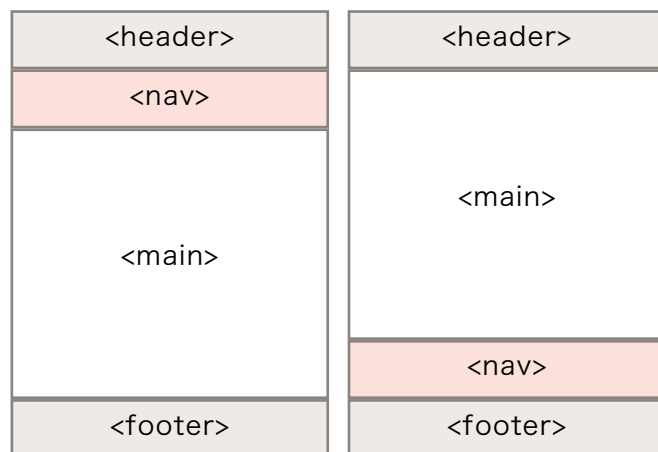
一般的な**基本レイアウト**には、**1カラム・2カラム・3カラム**のレイアウトがあります。主に、**1カラムレイアウト**はイメージ中心、**2カラムレイアウト**は文章が中心、**3カラムレイアウト**はページ数が多いサイトに向いています。さらに、サイトの特性に合わせて「**上or下部メニュー**」「**左or右or両サイドメニュー**」「**逆L字メニュー**」など、ナビゲーションの位置で操作性が変わります。

1カラムレイアウト

色や画像などでイメージを伝えるのが中心のサイトに向いている。

上部メニューは、左から順に見つけやすく、自由に選びやすい

下部メニューは、左から順に見つけやすく、順番に選びやすい



上部メニュー

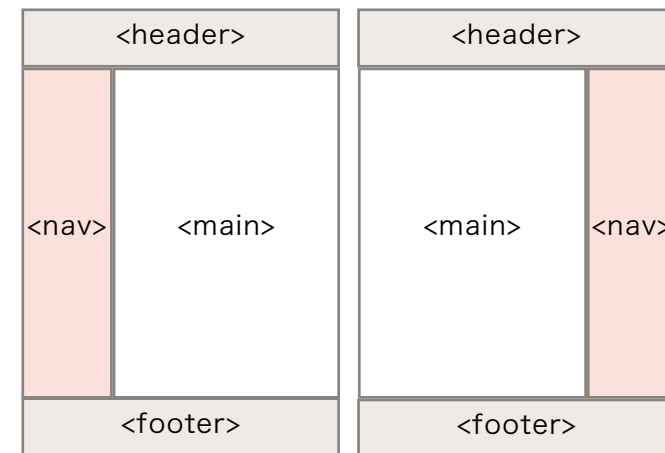
下部メニュー

2カラムレイアウト

文章で情報を伝えるのが中心のサイトに向いている。

左サイドメニューは、上から順に見つけやすく、自由に選びやすい

右サイドメニューは、上から順に見つけやすく、順番に選びやすい



左サイドメニュー

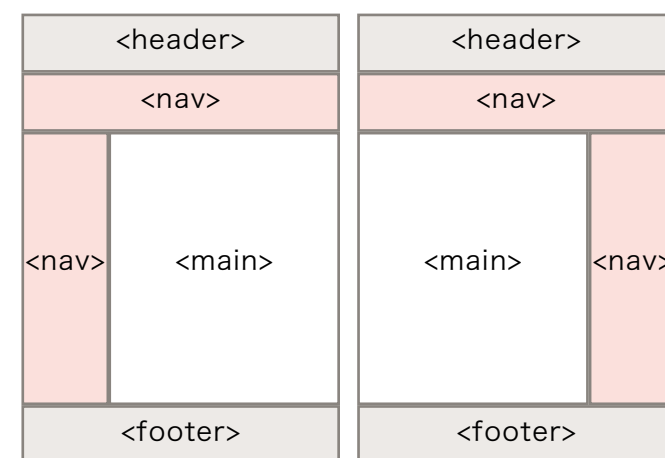
右サイドメニュー

ページ数の多い大規模なサイトで使用するレイアウト。

上部をメイン、サイドをサブとするナビが定番

ショッピング系では左サイドがメインメニューになり、上部がサブメニュー

パンフレットのように次のページを案内・誘導する必要がある場合は、上部がメインメニューになり、右サイドがサブメニューになる

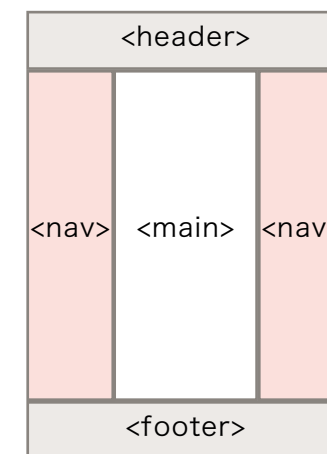


逆L字メニュー

3カラムレイアウト

ページ数の多い大規模なサイトや、ブログのように更新頻度が高いサイトに向いている。

左サイドをメイン、右サイドをサブとすることが多い
さらに上・下メニューを追加した進化系も考えられる



両サイドメニュー

サイトの目的や特徴に合わせてレイアウトを選択することで、操作性や快適性が向上します。

05 ページの見出し アウトライン

ページのコンテンツは目次のようにアウトライン（階層構造）を持たせ、目次のようにページ内の移動を助ける機能を持つ

<h1>...<h6>の見出しタグは、文章の要約文でありタイトルの意味があり、**HTML Living Standard**では下記のように書かれています。

これらの要素は、そのセクションの見出しを表す。
セマンティックスとこれらの要素の意味は見出しとセクションの節で定義される。
これらの要素は、要素名の数字で与えられたランクを持つ。h1要素が最も高いランクを持つと言われ、h6要素が最も低いランクを持ち、同じ名前をもつ2つの要素は等しいランクを持つ。

- セクション：何らかの内容について書かれた文章の一区切り
- セマンティックス：言葉のもつ意味の構造
- ランク：レベル・順位・重要度

文章を読解すると、「<h1>～<h6>のタグは、そこに書かれた文章の要約文を表す。言葉が持つ意味と見出しタグの意味は【見出しとセクション（に書かれた項目）】で定義されている。見出しタグは、1～6の数字で重要度を表している。<h1>が最も高いレベルで<h6>が最も低いレベルになり、<h2>と<h2>のように同じ数字を持つ見出しタグは同じレベルとなり<h2>と<h3>が同じレベルにはならない。」となります。この仕様書の説明から、見出しタグはアウトライン（階層構造）を持っていると確認できます。

<h1> 最も高いレベル
├─<h2> h1よりも低いレベル
│ ├─<h3> h2よりも低いレベル
│ └─<h2> 他のh2と同じレベル
└─<h3> 他のh3と同じレベル
 └─<h3> 他のh3と同じレベル

見出しのアウトライン（階層構造）

1. HTML5
├─ 1. HTMLの歴史
│ └─ 1 ブラウザとHTML
└─ 2. 他のバージョンとの違い
 ├─ 1. 追加されたタグ
 └─ 2. 追加された属性

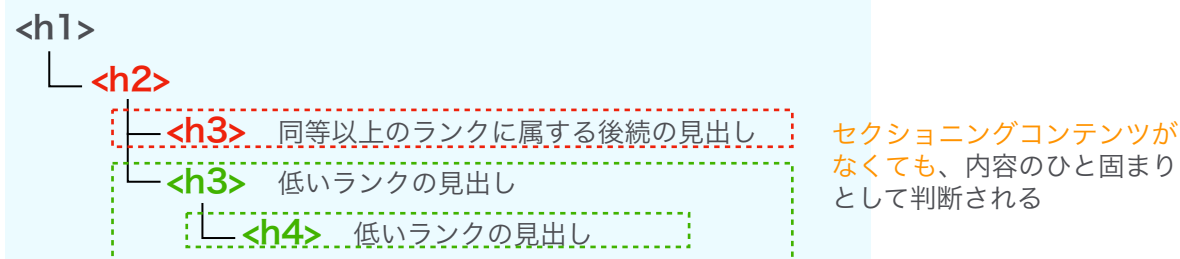
アウトラインをリスト化表示

見出しとセクションの節

セクショニングコンテンツの要素においてヘディングコンテンツの最初の要素は、そのセクションの見出しを表す。同等以上のランクに属する後続の見出しは、新しい（暗黙の）セクションを開始し、低いランクの見出しは、前のセクションの一部である暗黙のサブセクションを開始する。どちらの場合も、要素は暗黙のセクションの見出しを表す。

- 暗黙：何もなくても見出しが存在する文章の固まり
- サブセクション：小見出しが含まれる小さな文章の固まり

文章を読解すると、「<section>や<article>などで使用する<h1>～<h6>は、その文章の要約文を表す。例えば<h3>または<h2>の次に使う<h4>または<h3>は、<section>や<article>がなくても同様に判断でき、<h3>や<h4>は、その前に使われた<h2>や<h3>に関連した内容の一部として判断できる。どちらの場合も、見出しタグは文章の内容をひと固まりを示す要約文を表す。」となります。



セクショニングコンテンツがなくても、文章に付け加えられた要約文は「見出し」「段落」でセットとして判断し、「見出し」はレベルに合わせて階層化され、文章の関係性がわかるようになっている

アクセシビリティ 機能

音声ブラウザなどの利用者は、ページ内の内容を手早く特定するために、見出しから見出しへスキップすることができます。この時見出しレベルが適切でないと、内容が適切に伝わらず利用者の混乱を招き、アクセシビリティの低下につながり「利用しにくい」ページとなります。

見出しタグはページ内の「目次」の役割を持っていることを認識すべきです。

06 セクショニング コンテンツ

見出しのアウトラインさえ理解すれば、セクショニングコンテンツを適切に使用することができる

どの書籍やブログなどの情報を探っても、セクショニングコンテンツの扱い方はどこか不透明でわかりにくい部分が多くスッキリしません。そこまで深く考えずにマークアップする人が多いのも原因で、多くのサイトでいい加減なマークアップが行われ、音声ブラウザなどを利用する人に混乱を与えています。各セクショニングコンテンツのタグは、潜在的に見出しと段落などがセットになったアウトラインを持ちます。

潜在的に：そこに見出しタグがあってもなくても存在する

アウトラインを持つ：セクショニングコンテンツ内でアウトラインを構成する

<section>：汎用セクション要素

文書またはアプリケーションの一般的セクションを表す。この文脈において、セクションは、典型的な見出しを伴う、主題を表すコンテンツのグループである。

つまり「どこでも使える見出しと文章をグループ化できるタグ」ということです。しかし、そこに見出しのアウトラインも関係してくるため、多くのサイトでいい加減なマークアップが完成する原因になっています。さらに、

要素のコンテンツを配給することが理にかなう場合、著者はsection要素の代わりにarticle要素を使用するよう推奨される。

「セクショニングコンテンツを使った範囲の文章が、ページにとって重要な内容になる場合は、<section>ではなく<article>を使いましょう」となっています。

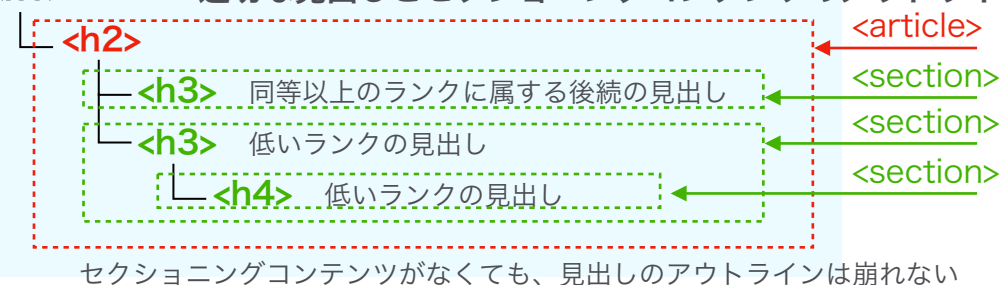
Section要素は包括的なコンテナ要素ではない。要素がスタイル付け目的またはスクリプティングの便宜のために必要になった場合のみ、著者は、代わりにdiv要素を使用するよう推奨される。要素の内容が文書のアウトラインで明示的に記載されるだろう場合のみ、section要素が適切であるというのが一般的なルールである。

「どこでも使える」訳ではなく、「デザイン目的であれば<div>を使いなさ

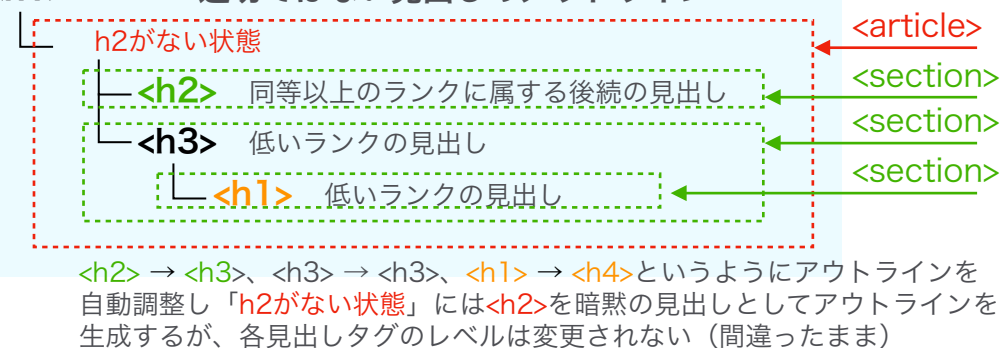
い」と書かれています。また、「<section>で囲まれた文章の内容が、アウトラインを崩すのであれば、<section>で囲まない方がいい」というのが一般的なルールとしています。

セクショニングコンテンツ内では、適切なアウトラインになるように自動調整する働きがあります。セクショニングコンテンツや見出しタグが適切でない場合、この自動調整がアウトラインを崩す原因となっています。

<h1> 適切な見出しとセクショニングコンテンツのアウトライン



<h1> 適切ではない見出しのアウトライン



<h1> 適切ではない見出しとセクショニングコンテンツのアウトライン

