



デザイナー兼フロントエンジニアを目指す

猫好きWEB屋の情報発信ブログ

カテゴリー一覧

WEBデザインコーディングWEB制作全般 その他

photoshop HTML5 GoogleAnalytics英語

CSS 制作Tips

Javascript

検索...

WordPress

concrete5

トップ > コーディング > HTML5 >

Google HTML/CSS Style Guideを全部日本語に訳してみた【HTML編】

Google HTML/CSS Style Guideを全部日本語に訳してみた【HTML編】



HTML5

コーディング

2015年3月23日

「[Google HTML/CSS Style Guide](#)」は、Googleが出している様々な言語のスタイルガイド「[google-styleguide](#)」の中の一つです。コードの現実的な書き方と理想的な書き方、どちらもできる限り満たすよう工夫されています。今後のコーディングに取り入れるため、そして英語の学習のため、細目まで全訳してみました。ものすごく長くなったので、【HTML編】と【CSS編】に分けます。というわけでまずはHTML編。

すでに多くの方が翻訳されており、私もこちらの記事を大いに参考にさせて頂きました。キレイに要点がまとまっているので、とりあえずルールだけ把握したい！という方にはおすすめです。

「Googleが推薦するHTMLとCSSのコーディング方法」

※私の英語力はまだまだ心許ない上に意識も交じってます、ご了承ください(^^;)ではいきます。

背景

このドキュメントは、HTMLとCSSの書式やスタイルのルールを明確にします。その狙いは、共同開発やコードの品質を向上させること、そして基本構造を強化することです。HTMLやCSSファイルの（GSSも含む）基本的な部分に適用できます。コードの品質が保たれている限り、WEBツールを複雑化したり簡略化したりコンパイルしたり、といったことがやりやすくなります。

基本のスタイルルール

プロトコル

埋め込みリソースからプロトコルを省く

どうしても2つのプロトコル(http:/ https:)をまたがって使わざるを得ない限り、画像や他のメディアファイル、スタイルシート、スクリプトのURLからプロトコル部分を省いてください。

関連URLを作り得るプロトコルを省くことは、コンテンツが混ざってしまう問題を防ぐ上に、ファイルサイズをより小さく保つことにも繋がります。

```
<!-- NG -->
<script src="http://www.google.com/js/gweb/analytics/autotrack.js"></script>
<!-- OK -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>
```

```
/* NG */
div {
  background: url(http://www.google.com/images/example);
}
/* OK */
div {
  background: url(//www.google.com/images/example);
}
```

基本の書式ルール

インデント

インデントはスペース2つ

タブや、タブとスペースを混ぜて使ってはいけません。

```
<ul>
  <li>Fantastic
  <li>Great
</ul>
```

```
div {
  color: blue;
}
```

大文字と小文字

小文字のみ使う

すべてのコードは小文字で書きます。HTML要素、属性、（text/CDATAでなければ）属性値、CSSセレクタ、プロパティ、（文字列による例外はありますが）プロパティ値、に適用されます。

```
<!-- NG -->
<A HREF="/">Home</A>
<!-- OK -->

```

```
/* NG */
```

```
color: #E5E5E5;  
/* OK */  
color: #e5e5e5;
```

末尾のスペース

末尾のスペースは消す

末尾のスペースは不必要ですし、差異をわかりづらくします。

```
<!-- NG -->  
<p>What?_  
<!-- OK -->  
<p>Yes please.
```

基本のメタルール

エンコーディング

UTF-8（BOM無し）を使う

お使いのエディタが、BOM無しのUTF-8になっているか確認してください。
HTMLのテンプレートやドキュメントに<meta charset="utf-8">とエンコーディングを明記してください。スタイルシートにエンコーディングの明記は必要ありません。

(エンコーディングについてもっと正確に知りたい方は、[Handling character encodings in HTML and CSS](#)をご参照ください。)

コメント

可能な限り必要に応じてコードを説明する

コードを説明するコメントを書きましょう：どこを司るのか、書かれた目的は何か、それぞれのソリューションがなぜ使われたのか？

常にすべて文章化されたコードが現場で必要とされない場合、この項目は任意です。プロジェクトの複雑さによって、コメントの有用性は大いに変化するでしょう。

アクション項目

アクション項目には「TODO」マークをつける

todoには「TODO」というキーワードのみでハイライトします。「@@」のような他の共通書式を用いてはいけません。

「TODO(contact)」という形で、括弧内にcontact（ユーザーネームやメーリングリスト）を追記します。

「TODO:action item」という形で、コロンの後にaction itemを追記します。

```
{# TODO(john.doe): revisit centering #}  
<center>Test</center>  
<!-- TODO: remove optional tags -->  
<ul>  
  <li>Apples</li>  
  <li>Oranges</li>  
</ul>
```

HTMLスタイルルール

ドキュメントタイプ

HTML5を使う

HTMLドキュメントでは、HTML5（HTML syntax）が推奨されています：

（text/htmlとしてのHTMLを使うことが推奨されています。XHTMLを使わないように。application/xhtml+xmlであるXHTMLは、ブラウザとインフラストラクチャどちらのサポートも十分ではありません。HTMLと比べ、容量の最適化という点でも劣ります。）

HTMLとしては正しいとしても、void要素は閉じないように。すなわち、
ではなく
を使います。

正しいHTML

可能な限り正しいHTMLを使う

ファイルサイズのパフォーマンスゴールなど、それを不可能とする目的が他にない限り、正しいHTMLを使ってください。

[W3C HTML validator](#)のようなバリデーションツールを使いましょう。

正しいHTMLを使うことで、コードの基本的な質を保てますし、技術的な要求と制約を学ぶことにもつながります。また、あるべきHTMLの使い方を確かにします。

```
<!-- NG -->
<title>Test</title>
<article>This is only a test.
<!-- OK -->
<!DOCTYPE html>
<meta charset="utf-8">
<title>Test</title>
<article>This is only a test.</article>
</ul>
```

セマンティック

要素の目的に沿ったHTMLを使う

要素は（しばしば「タグ」と呼ばれますが、間違いです）それらが作られた目的通りに使います。例えばheader要素はheaderに、p要素は段落に、a要素はアンカーリンクに、など。

本来の目的に沿ってHTMLを使うことは、アクセシビリティ・再利用性・効率化といった点で重要です。

```
<!-- NG -->
<div onclick="goToRecommendations();">All recommendations</div>
<!-- OK -->
<a href="recommendations/">All recommendations</a>
```

マルチメディアの代替

マルチメディアには代替内容を用意すること

画像、動画、canvasを使ったアニメーションをいったマルチメディアには、別のアクセス方法を確保すること。可能であれば、画像には意味のある代替テキスト（alt）、動画と音声にはtranscriptとcaption、といったふうに。

```
<!-- NG -->

<!-- OK -->

```

構造の分離

ストラクチャから、プレゼンテーションと振る舞いを分ける

プレゼンテーション（スタイル）と振る舞い（スクリプト）は、ストラクチャ（マークアップ）から厳密に分けます。3者間の相互作用は、絶対的に少なく保つよう努めてください。

HTMLのみをドキュメントやテンプレートに含めます。このHTMLは、単純にストラクチャを構成するためのものです。プレゼンテーションはスタイルシートに、振る舞いはスクリプトに、すべて分けましょう。

ドキュメントやテンプレートにおいて、リンクによるスタイルシートやスクリプトとの関わりはできるだけ少なくします。

構造の分離は、メンテナンス性において重要です。

HTMLドキュメントやテンプレートを書き換えることは、いつだってスタイルシートやスクリプトを更新するより大変なんですから。

```
<!-- NG -->
<!DOCTYPE html>
<title>HTML sucks</title>
<link rel="stylesheet" href="base.css" media="screen">
<link rel="stylesheet" href="grid.css" media="screen">
<link rel="stylesheet" href="print.css" media="print">
<h1 style="font-size: 1em;">HTML sucks</h1>
<p>I've read about this on a few sites but now I'm sure:
  <u>HTML is stupid!!1</u>
<center>I can't believe there's no way to control the styling of
  my website without doing everything all over again!</center>
<!-- OK -->
<!DOCTYPE html>
<title>My first CSS-only redesign</title>
<link rel="stylesheet" href="default.css">
<h1>My first CSS-only redesign</h1>
<p>I've read about this on a few sites but today I'm actually
  doing it: separating concerns and avoiding anything in the HTML of
  my website that is presentational.
<p>It's awesome!
```

文字参照

文字参照は使わない

ファイルやエディタではもちろんチーム間でも同じエンコーディング（UTF-8）を使っていれば、一、”、😄のような文字参照は必要ありません。

「<」や「&」のようにHTMLで特別な意味を持つものや、特殊スペースのような「見

えないもの」に限っては特別にOKです。

```
<!-- NG -->
```

The currency symbol for the Euro is “&eur;”.

```
<!-- OK -->
```

The currency symbol for the Euro is “€”.

任意のタグ

任意のタグは省略する（オプション）

ファイルサイズの最適化やコードの見易さのために、任意タグの省略を検討してください。

[HTML5 specification](#)には、どのタグを省けるのか明示されています。

（このアプローチは、WEBデザイナーが適切としてきた基準とはっきり異なるため、一般的なガイドラインとして成立するのに猶予期間を必要とします。矛盾を無くし単一にするため、一部だけ実施することはせず、すべての任意タグを省いてください。）

```
<!-- NG -->
```

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Spending money, spending bytes</title>
```

```
</head>
```

```
<body>
```

```
<p>Sic.</p>
```

```
</body>
```

```
</html>
```

```
<!-- OK -->
```

```
<!DOCTYPE html>
```

```
<title>Saving money, saving bytes</title>
```

```
<p>Qed.
```

タイプ属性

スタイルシートとスクリプトのタイプ属性を省く

CSS以外のスタイルシート、またはjavascript以外のスクリプトを使わない限り、type属性を省いてください。

HTML5では、text/cssとtext/javascriptがデフォルトになっているので、type属性

は必要ありません。これは、古いブラウザでも問題ありません。

```
<!-- NG -->
<link rel="stylesheet" href="//www.google.com/css/maia.css"
  type="text/css">
<!-- OK -->
<link rel="stylesheet" href="//www.google.com/css/maia.css">
<!-- NG -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js"
  type="text/javascript"></script>
<!-- OK -->
<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>
```

HTMLの書式ルール

一般的な書式

それぞれのブロック/リスト/テーブル要素ごとに新しい行にし、子要素にはインデントをつける

(CSSでは、要素にdisplayプロパティごとの違う役割を想定しているので) 要素はスタイリングから独立させ、すべてのブロック/リスト/テーブル要素は新しい行で書き出します。

また、ブロック/リスト/テーブル要素の子要素にはインデントをつけます。

(もし、リスト要素間のスペース問題に対処する必要がある場合、を一行にまとめても構いません。リンターでは、エラーの代わりに警告を出すことが推奨されています。)

```
<blockquote>
  <p><em>Space</em>, the final frontier.</p>
</blockquote>
<ul>
  <li>Moe
  <li>Larry
  <li>Curly
</ul>
<table>
  <thead>
    <tr>
      <th scope="col">Income
      <th scope="col">Taxes
    </tr>
  </thead>
  <tbody>
```

```
<tr>
  <td>$ 5.00
  <td>$ 4.50
</table>
```

HTMLクオテーションマーク

属性値にはダブルクオテーションを使う

属性値に使うクオテーションは、シングル (') よりもダブル (") が好ましいです。

```
<!-- NG -->
<a class='maia-button maia-button-secondary'>Sign in</a>
<!-- OK -->
<a class="maia-button maia-button-secondary">Sign in</a>
```

訳してみて感じたこと

数か所、うまく訳せてないところがあります…。すみません。

原文は項目ごとの解説がめちゃくちゃ丁寧なんですね。そもそもなぜHTML5を使うのかなど、「理由はよくわからないけどルールになってるから当然だと思ってた」事柄が腑に落ちました。

ルールは作ることより、浸透させることのほうが難しいものです。コーディングルールなんて一人で書いていても破綻しやすい。関係者皆で共有するには、このドキュメントのように項目一つ一つの認識を明文化することが大事なんですね。勉強になりました。

また、このドキュメントは「背景」で始まり、「終わりの言葉」で締めくくられています。後者もまたとても良い文章でした。こちらは、CSS編の方に載せたいと思います。

[いいね!](#) [シェア](#) 60

« 少しの工夫でクリック率アップ! GIFアニメで作るバナーの動き 8 パターン
Google HTML/CSS Style Guideを全部日本語に訳してみた【CSS編】 »

カテゴリー

WEBデザイン

Photoshop

コーディング

HTML5

CSS

Javascript

WordPress

concrete5

WEB制作全般

GoogleAnalytics

制作Tips

その他

英語

プロフィール



なみ

1986年生まれのWEBデザイナー。3年間某喫茶店チェーンで働いたのち、WEBの世界に転身。

以来デザインの他コーディング・プログラミングにも興味を持ち、制作業務をこなしています。WEB制作は天職です。これまで多くのWEB屋さんブログに助けられ、困難を乗り越えてきました。ので、

自分も技術の研鑽に励む人たちへ、役に立つ情報を発信したいです。

TAGS

作ってみた

書評

行ってきた

MESSAGE

お名前（必須）

メールアドレス（必須）

コメント（必須）

送信する

© 2015 ぶちねこどうえぶ All Rights Reserved.