

株式会社ウィッシュポケット

Web サイト制作規定 ガイドライン

作成日：2010 年 7 月 9 日

最終改定日：2011 年 5 月 30 日

この文書は、弊社（株式会社ウィッシュポケット）がWebサイトを実制作するときのガイドラインです。

※技術的なガイドではなく、コーディング時のルールに関する記載を中心としています。

本ガイドラインの構成（目次）

1.基本ガイドライン（全般）

- 1.1.閲覧環境
- 1.2.コーディングルール

2.(X)HTML ガイドライン

- 2.1.基本ガイドライン
- 2.2.書式に関するルール
- 2.3.<head 要素内>に関する共通ルール
- 2.4.<body 要素内>に関する共通ルール/主なタグ

3.CSS ガイドライン

- 3.1.基本ガイドライン
- 3.2.CSS 設計

4.Javascript ガイドライン

- 4.1.初期設定 Javascript
- 4.2.追加設定 Javascript
- 4.3.Google Maps API
- 4.4.Google Analytics

5.命名規則ガイドライン

- 5.1.ファイル名基本ガイド
- 5.2.ディレクトリに関する命名
- 5.3.HTML および CSS における命名
- 5.4.画像ファイルの命名

6.制作管理ガイドライン

- 6.1.案件の管理
- 6.2.ディレクトリの構成

1.基本ガイドライン（全般）

1.1.閲覧環境

1.1.1.出力メディア

PC スクリーンとプリントを対象とする。携帯電話などのメディアに関しては都度別途。
印刷対応にする場合は別途印刷用の CSS を用意する。

1.1.2.ディスプレイ

1024×768px のディスプレイを最小対象とする。
コンテンツ幅は 800～1000px を基本とする。

閲覧環境のトレンドに合わせ、
Yahoo!Japan を参照（ブラウザにより若干差異有）とし、
横サイズは 900 ～1000px、
縦サイズ 500px までに重要コンテンツを収めることを推奨する

1

1.1.3.対象ブラウザ

制作時点での日本国内ブラウザシェアを考慮するものとする。

[参照元 StatCounter] <http://gs.statcounter.com/>

【Windows】
InternetExplorer6.0（以下、IE6）以上、
FireFoxs 最新版（4.0）、GoogleChrome 最新版(11.0)
【Macintosh】
Safari 最新版（5.0）、FireFoxs 最新版（4.0）、GoogleChrome 最新版(11.0)

確認を行うブラウザは初期設定で行うものとする。

1.2.基本コーディングルール

1.2.1.文字コード

原則 UTF-8 とする。CSS や Javascript も HTML に準拠する。

[理由] CMS 等スクリプト要件にあわせるため。案件に応じ Shift-JIS や EUC-JP に。

1.2.2.改行とインデント

原則 CR+LF（Windows）とする。

インデントは『tab』を使用し、全角・半角スペースは使用しない。

ソースの改行は見易さ管理のため適宜行う。※場合による

2.(X)HTML ガイドライン

2.1.基本ガイドライン

2.1.1.DOCTYPE(文書型)

原則 XHTML1.0 transitional を採用

〔理由〕 a 要素での target 属性を使用、font 要素の使用、(iframe)が {strict} では認められていないため。

2.1.2.XML 宣言に関して

XML 宣言はつけない 〔理由〕 IE6.0 と Opera7.0 を後方互換モードに変えてしまうため。

2.1.3.マークアップ全般

文書内容に適したマークアップを行う。

W3Cが勧告する XHTML1.0 の仕様書に準拠したマークアップを行う。

W3C によるマークアップ検証サービス。

「Markup validator」→ <http://validator.w3.org/>

をパス（クリア）すること。

2.1.4.パスの記述に関して

サイト内リンクおよび画像ともに指定が無い場合は原則『相対パス』とする。

SSI や CMS の都合上『ルート相対パス』で記述する場合は組込み担当者と相談する。

2.1.4.id 属性、class 属性に関して

全頁に共通して記述されるものには『id 属性』 それ以外は原則『class 属性』を使う。

Id 属性と class 属性は同一の要素に関して設定可能であることを考慮する。

class 属性は同一の要素内に複数設定可能であることを考慮する。

<div id="main" class="faq">
<div id=class="box alignleft">

2.2.書式に関するルール

原則以下のとおりの書式で統一するが案件や場合により適宜対応する。

- インデントによる階層表示は行わず、左詰めとする。
- スペーサーや
は極力使用せず CSS で間隔余白を設定する。
- コメントは可能な限りソース内に分かりやすく記述する。適宜幅なども記載する。

※その際タグの内側に記述する。

(例)

<pre><ul id="globalNavi"><!-- ▼glovalNavi/start --> XXXX XXXX XXXX <!-- ▲glovalNavi/end --></pre>
--

```

<div id="main"><!-- ▼main/start/width-700px -->
<h2>見出し</h2>
<p>テキストテキストテキストテキストテキストテキストテキストテキストテキスト、
テキストテキストテキストテキストテキストテキストテキストテキストテキスト、
テキストテキストテキストテキストテキストテキストテキストテキストテキスト、</p>
<!-- ▼main/start/width-700px --></div>

<div id="side"><!-- ▼side/start/width-200px -->

```

2.3.<head 要素内>に関する共通ルール

2.3.1.<meta http-equiv=" " > title 要素の前に記述。

『Content-Type』『Content-Language』『Content-Style-Type』『Content-Script-Type』

2.3.2.<title>タグ (ページのタイトル)

SEO を意識して各ページ以下のように記述する。※全角 25 字以内が理想。

【トップページ】 提供する主要サービスワード 法人（サイト）名 地域など
【第二階層】 製品情報 提供する主要サービスワード 法人（サイト）名 地域など
【第三階層】 製品情報 製品名 提供する主要サービスワード 法人（サイト）名 地域など

『そのページで一番検索してほしいキーワード』 + 『サイト名』 + 「地域など」を基本とする。

全角『|』で区切るもしくは、『○○○○の○○○○』と助詞などでつなぐ。

2.3.3.<meta name=" description" > (ページの概要文)

SEO を意識した、ページの内容に即した簡単な説明文。

全角 100 字目安。2～3 のキーワードを 1 回以上含むようにする。

(例) 太字はキーワード

```

<meta name="description" content="ウエディングエステ専門、エステサロン セレブリティ。短期間で
の痩身プログラムで美しい身体を実現。東京都、六本木" />

```

Google では検索結果時の概要文になるのでクリック率に影響するライティングが必要。

2.3.4.<meta name=" keyword" > (ページのキーワード)

SEO を意識した、ページの内容に即したキーワード。2～7 個を『、』区切りで。

2.3.5.link 要素や script 要素

<head>要素内の一番最後に記述する。

2.4.<body 要素内>に関する共通ルール/主なタグ

2.4.1.<body>要素

最も大きくコンテンツを内包する要素であるため、body に id や class を振って、ページ毎にスタイルを切り替える。この手法を『CSS シグネイチャ』という。

このため初期テンプレートの段階で下記のように記述し、任意の属性名を割当てておくこととする。

`<body class="DIRNAME" id="PAGENAME">...`

複数のページで共通のレイアウトやスタイルを指定したい場合に便利。

2.4.2.<h●>要素 (見出し) <h1> <h2> <h3> <h4> <h5> <h6>

<h1>から順に出現させること。順番をとばしてはいけない。

(例) <h2>の次に<h4>が出現するのは NG!

<h1>は SEO の観点からページ最上部にリード文として出現させることを基本とする。

<h1>は場合によりサイトロゴに記述する。

各要素になるだけキーワードを盛り込むようにする。※画像の場合は alt 属性に記述する。

2.4.3.<div>要素 ブロック要素を包括する要素 (ブロックレベル要素)

レイアウト上もっとも重宝する要素であるが、多用を避け、入れ子構造などが煩雑にならないようにする。

div が多くなるとソースの概要が把握しづらくなるため適宜コメントや改行を挿入する。

2.4.4.要素 画像 (インライン要素)

必ずサイズ属性 (width、height) を指定する。[理由] ページ読み込みを早くするため

必ず『alt 属性』にて代替テキストを記述する。※意味の無い画像は『alt=" "』にする。

2.4.5.<p>要素 段落 (ブロックレベル要素)

主に本文に使用する。頻出タグなので『class 属性』を使うなど子孫セクタをうまく利用してレイアウトする。

尚、文字色・サイズなどを部分的に変更する場合は要素で括り、直接 style を指示する。

(例)

<p>大きなノッポの古時計、おじいさんの時計。</p>

2.4.6.<table>要素 表組み (ブロックレベル要素)

汎用性が高いが、サイズおよび寄せ (align) が変動する場合は多いタグのため、CSS での管理を考え、各々に『class 属性』をつけるようにする。

2.4.7.、要素 リスト (ブロックレベル要素)

サイト内で共通するものに関しては各々に『id 属性』をつけるが、場合により子孫セクタとして上位の<div>の『id 属性』で管理する。

汎用性が高いが、サイズおよび寄せ (align) が変動する場合は多いタグのため、CSS での管理を考え、各々に『class 属性』をつけるようにする。

2.4.8.<a>要素 アンカー（インライン要素）

サイト内リンクに関しては特に指定なし。サイト外へのリンクは（target 属性を_blank にし別タブ・窓で開く）指定をする。

- ページ内アンカーリンクは『id 属性』と『name 属性』をつける。
- 『a: hover と a: active』の擬似クラスの設定でマウスオンの指定を行う。

3.CSS ガイドライン

現在広く利用・ブラウザサポートされている CSS (Cascading Style Sheets、カスケーディングスタイルシート) のバージョン「CSS2.1」を採用します。

3.1.基本ガイドライン

3.1.1.CSS の初期化

リセット用の CSS を用意し、マージンなどの設定を初期化する。(別記)

基本的にコーディングの際に変更しないものとし、編集権限は弊社ディレクターのみにあるものとし編集した際には更新したことを記述する。

3.1.2.書式

一行目に必ず『@charset』の記述をする。HTML にあわせ基本は utf-8 にする。

インデントなどは Dreamweaver で CSS タブより記述したものに準ずる。

適用対象ごとに後の管理を考え、区別しやすいようにコメントで区切る。

```
/* Header  
-----*/
```

3.1.3.記述の場所

基本的に外部 CSS に記述するものとするが、部分的な文字装飾は直接 style で指定する。

その際、メンテナンス性を損なわないようにする。

3.1.4.子孫セレクトの使用

id や class を直接使う場合は限定し、子孫セレクトを積極的に使用する。

【悪い例】

```
p.items {  
    ...;  
}
```

【よい例】

```
div.items p {  
    ...;  
}
```

3.1.5.子孫セレクトの書き方

子孫セレクトで示すツリー構造は不必要な表現 (セレクト) を省く。

【悪い例】

```
div#stage div#contents div#wrap div#main div.box a  
    ...;  
}
```

【よい例】

```
div#main div.box a {  
    ...;  
}
```

CSS シグネイチャを使用する場合は body 要素の『id,class 属性』→必要な上位セレクト→

対象のタグの順で書く。Dreamweaver の記入ボックスから記述する場合によく考えて編集する。書き方次第で優先順位が変わるので注意が必要。

3.1.6. id/class 名の規則

id や class 名はアルファベット小文字からはじめ、2 語以上続く場合は 2 語目の頭文字を大文字にする。

(例) 『searchBox』『headerLogo』『footerLink』など

類似した識別名の場合は連番をふる。この際番号は 2 桁にする

(例) 『arrowLink01』『leadText03』など

3.1.7 フォントに関して

特に指定が無ければ以下のようにする

"メイリオ", "Meiryō", "ヒラギノ角ゴ Pro W3", "Hiragino Kaku Gothic Pro", Osaka, "MS P Gothic", "MS P Gothic", Arial, Helvetica, Verdana, sans-serif

文字の大きさを可変にするセレクトをつける場合は『%』でサイズ指定をする。

「CSS 文字サイズ」と検索したページを参照する。

それ以外は body 要素に 11~16px を指定し、要素ごとに CSS で『px 指定』する。

3.2.CSS 設計

3.2.1.共通 (テンプレート用) CSS 【common】

CSS は分割して管理し、以下のようなファイル構造設計にする。

ファイル名	内容	主な対象要素・属性
import.css	下記の共通する CSS を読み込むファイル 読み込む優先順位も指定する。	
reset.css	共通するレイアウトに対し、タグのリセットなどの初期化、およびハックを指定するファイル。 基本的に編集しない。	要素タグ全般 clearfix など
cmn-layout.css	ヘッダーやフッター、レイアウト段組、包括ブロックなど共通レイアウトを管理するファイル ※場合によって下記『cmn_○○○○.css』を統合。	div id=" stage" div id=" content" div id=" header" div id=" wrap" div id=" main" div id=" side" div id=" footer" など
cmn-header.css	ヘッダー部分に内包されるコンテンツすべて ※グローバルナビが水平な場合はこの中に含む。	h 1 div id=" siteLogo"

		div id=" helpLink" div id=" searchArea" ul id=" globalNavi" など
cmn-footer.css	フッターに内包される共通出現コンテンツすべて。	ul id=" footerNavi access など
cmn-main.css	メインエリア内で共通して必要なコンテンツ。 ※ パンくずリスト、見出し、ページ送り よくあるブロックなど	h2 div id=" topiPath" div class=" pager" など
cmn-side.css	サイドエリア内で共通して必要なコンテンツ。 ※グローバルナビが垂直な場合はこの中に含む。	ul id=" localNavi" など
cmn-style.css	サイト制作時にテンプレート的に頻出する class を 設定した CSS。 ※ 数値を編集することのみのもの。 ※ テキストのフォントやテキストリンクも	★別記
table.css	テーブル要素に関する設定。	table th,td
list.css	汎用リスト要素に関する設定。	ul,ol,li
parts.css	汎用的なコンテンツパーツに関する設定	dl,dt,dd div class=" ○○○"
other.css	その他上記以外で特例的なものに関する設定。	

cmn-style.css の詳細（頻出する class）

class 名	対象タグ（要素）	内容
.mgx00	全般	マージンの設定 00 は数値
.pad00	全般	パディングの設定 00 は数値
.inlineX	全般	Text-align XはL(左)、C(中央)、R(右)
.floatX		フロートの設定 XはL(左)、R(右)
.clearfix	<div>	クリアフィックス
.imgBoxX		画像とテキストのフロート設定
.bar0X	h2、h3、h4 など	背景にバー 0X は連番
.section0X	<div>	見出しと本文のまとめ
.box0X	<div>	囲み 0X は連番
.head	<div>ほか	背景用に上部が必要な場合など
.bottom	<div>ほか	背景用に底部が必要な場合など

※IE6 以下では class の複数指定は、CSS 側のセレクタの記述は対応していないので単体指定する。

3.2.2.各コンテンツ別 CSS 【each】

コンテンツのタイプに応じてその都度読み込む CSS。

ファイル名	内容	主な対象要素・属性
home.css	トップページのみに使用されるコンテンツのレイアウト設定 CSS ファイル。	div class=" newsArea"
news.css	ニュース一覧、ニュース詳細のレイアウト設定 CSS。	
faq.css	よくあるご質問全般に関するレイアウト設定 CSS。	
pageIndex.css	階層トップ、カテゴリーページなどの一覧ページのレイアウト設定 CSS。	
pageDetail.css	商品や実績の詳細ページのレイアウトを設定するときの CSS。	
document.css	個人情報保護方針および利用規約などテキスト文書中心のテンプレートページに関するレイアウト設定 CSS。	
form.css	フォームおよび問い合わせフォームなどのレイアウトに関する設定をする CSS。	Input ほか
sitemap.css	サイトマップページ専用の CSS。	ul,li

4. Javascript ガイドライン

4.1. 初期設定 Javascript

4.1.1. JQuery.js

JavaScriptのライブラリ。 <http://jquery.com/> 最新のものを使用する。

4.1.2. common.js

以下の挙動をあらかじめ設定する共通の Javascript

- ロールオーバー
要素に class=" over" をつけるとマウスオーバー時にファイル名「_on」のものに切り替える。
- linkBox.js
リンク範囲を Box まで広げるスクリプト
指定した class の付いた div 要素内の最初に出現するリンクの値を取得し、div 全体にリンクを広げるスクリプトです。
class 名は複数指定が可能（初期値：linkBox01, linkBox02）。また、オーバー時の class 名（初期値：linkBoxOver）を指定することで、CSS で見た目を変更することができます。
- smoothScroll.js
スムーズスクロール
a 要素の href 属性に「#」が含まれていると、id 属性が付いているフラグメントまでスクロールします。

4.2. 追加設定 Javascript／必要に応じ使用

※使用時は毎回最新のものをダウンロードすること。

4.2.1. DD_belatedPNG.js

IE6 で透過 png を適用させる JavaScript 要素に class="png"を付与
http://www.dillerdesign.com/experiment/DD_belatedPNG/

4.2.2. lightbox.js

画像を拡大表示する JavaScript ギャラリーなどのインターフェースに使用
<http://leandrovieira.com/projects/jquery/lightbox/>

4.2.3. swfobject.js

Flash を読み込ませる JavaScript
<http://blog.deconcept.com/swfobject/>

4.2.4. heightLine.js

ブロックレベル要素の高さ揃えをする JavaScript

<http://blog.webcreativepark.net/2007/07/26-010338.html>

4.3.Google Maps API

詳細は後ほど別記述

4.4.Google Analytics

</body>直前に記述する。コードは対象ドメインのみ有効。

5.命名規則ガイドライン

5.1.ファイル名基本ガイド

5.1.1.ファイル名使用文字

『a～z』アルファベット小文字、『0～9』での数字で連番 2 桁表記 例)『05』

『-』(ハイフン)と『_』(アンダースコア)

拡張子を含み半角 30 文字以内とする。

5.1.2.基本命名方針

- ユーザーに分かりやすい名前をつける。
- 原則英単語を使用する。(×…seihin ○…product)
- 下記の語に関しては表記を統一するにする

内容	英単語	統一表記
背景	background	bg
バナー	banner	bnr
ヘッダー	header	hdr
フッター	footer	ftr
アイコン	icon	ico
画像イメージ	image	img
写真	picture	pic
文章・テキスト	text	txt
矢印	arrow	arrow
線	line	line

5.2.ディレクトリに関する命名

ディレクトリ名は略さず最も適した英単語を用いる。大文字は使用せず半角 8 文字以内を推奨。

5.3.HTML および CSS における命名

ファイル名に大文字は使用しない。コンテンツのトップに当たるページには『index.html』とし、それを除き同一ファイル名は複数存在してはならない。

5.4.画像ファイルの命名

基本的に以下のような法則で命名する。

『部位名』+『_(アンダースコア)』+『ページ名』+『_(アンダースコア)』+『連番 2 桁』+.拡張子
例)『h2_faq.jpg』『pic_gallery_04.jpg』『bg_service_01.png』『arrow_topagetop.png』など

6.制作管理ガイドライン

6.1.案件の管理

案件の管理フォルダを以下のように設定する（下記図+表 参照）



フォルダ名	内容	中に含まれるもの
_html	制作中の最新 HTML+CSS	『img』『css』『js』 各ディレクトリおよび html ファイル
_html_bk	『_html』のバックアップ	年月日付ごとに管理
00_案件管理総合	案件に関する概要を管理するファイルなどを格納。	要件定義書類など
01_デザインカンブ	カンブデザイン（初稿）の html	Index.html など
02_デザイン PSD	PSD などデザインデータ	Top-design.psd など
03_画像+テキスト素材	制作に関係する素材	提供された画像およびテキスト素材。弊社用意の素材なども
04_ディレクトリマップ	サイトマップ・ディレクトリマップ	ディレクトリマップ
05_進行管理	進行管理に関係する書類など	納品までのスケジュール表や進行管理表など またやり取りのメールなど
06_契約書関連	受託制作契約書など	
07_指示書・ガイドライン	制作における規定など	コーディングガイドラインなど
08_企画提案書	企画提案書	提案書 PDF など
09_納品関連	納品時の仕様書など	マニュアルおよび納品仕様書や ID/PASS など

『_html』ファイルを編集する場合は必ずバックアップをとり、保存した上で作業すること。

6.2.ディレクトリの構成

■画像に関しては『img』フォルダ内に格納する。

さらに『img』内に『ディレクトリ名』を作り、ディレクトリごとの画像は管理すること。

例) トップページのみで使用する画像は『top』にわけ、img/top/img_mainvisual.jpg とする。

場合によって際と全体に共通する画像は『common』に格納する。

[理由]画像の管理領域を一元化することと、パスの構造を分かりやすくするため。

よってなるべく複雑な階層化は避けるようにする。