



coding guideline

コーディングガイドライン

RWD

目次

I. はじめに	3
II. 基本方針	4
III. HTML 制作ルール	6
IV. CSS 制作ルール	13
V. JavaScript 制作ルール	20
VI. ディレクトリ作成ルール	21
VII. ファイル作成ルール	23
VIII.コーディング後チェック	27

1 はじめに

1. ガイドラインの活用

原則ガイドラインの内容に沿って制作をすすめること。

ただし、本書とは別にガイドラインや指定がある場合はそちらを順守すること。

2. 推奨設定

推奨 マークがついている内容は、サイトの特性を考慮し、指定があった場合のみ対応する。

II 基本方針

1. ガイドラインの目的

本ガイドラインは、レスポンシブ Web デザイン（以下、RWD）サイトのコーディングにあたって、以下にあげる項目を実現させるとともに、ガイドラインを策定・導入・配布することで、広く標準化を促すことを目的とした、Web 制作者のための手引きです。

1) クオリティの向上

最適だと思われるルールをガイドラインとして制定することで、一定以上のクオリティを担保する。

2) クオリティの均一化

制作・運営に複数人が関わる場合、担当者による制作物のバラつきを防ぐ。

3) 制作の効率化

ルールがあることで、制作にかかるタイムロスを防ぐ。

ルールを決めておくことで、無駄なチェック・修正の工数を削減する。

上記目的で最適化されたガイドラインの品質を保つために、必要に応じて積極的に見直し・更新を行い、共有・周知を図る。

2. コーディング6大原則

1. Web サイト制作者は本ガイドラインをコーディングの基準とし、順守すること。
2. サイトの内容や構成を十分に考慮し、適切な文書構造のマークアップを行う。
3. RWD の特性を十分に考慮し、適切かつ必要最小限の記述でコーディングを行う。
4. 運用/更新が想定される箇所においては、内容の増減に対応できるよう可能な限り配慮する。
5. サイトの構成要素は、運用のしやすさと複数人での制作を考慮した作りにする。
6. ユーザーの閲覧環境と最新の技術を考慮した作りにする。

※ただし、本書とは別にガイドラインや指定がある場合はそちらを順守すること。

3. レスポンシブ仕様基本方針

1) メディアクエリについて

RWD を再現する手法は様々あるが、HTML や CSS など基本的な要素への記述で実現できるメディアクエリを採用する。

2) ブレークポイント

CSS でレイアウトを切替えるブレークポイントは、下記の分岐点 (px) を推奨する。

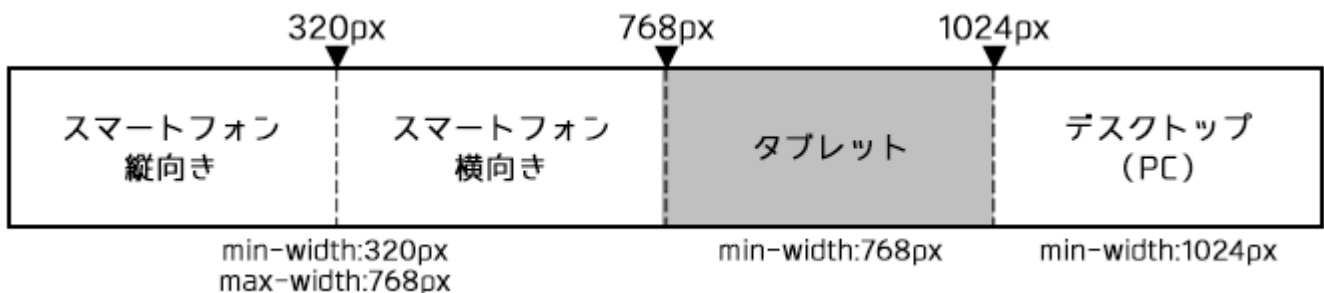
※ただし、制作するサイトで指定がある場合は、その仕様に合わせることを。

【最小サイズとブレークポイント】

スマホ /min-width: 320px; or (max-width: 768px)

タブレット /min-width: 768px;

デスクトップ/min-width: 1024px;



3) レイアウト設定 (フルード/アダプティブ)

制作するサイト要件に応じて最適なレイアウト設定を採用する。

4) ワンソース対応

メンテナンス性・安全性を考慮し、ワンソース記述を原則とする。

5) 高解像度ディスプレイ対応

原則として、高解像度ディスプレイに適切に表示される作りにする。

6) プログレッシブ・エンハンスメント

原則として「プログレッシブ・エンハンスメント」の概念に基づいた制作を行う。

III HTML 制作ルール

1. 文字コード

原則 UTF-8 とする。

※システム要件/言語/サーバによるリクエストなどから Shift-JIS や EUC-JP などとする場合もある。

2. 改行コード

原則 CR+LF (Windows) とする。

3. DOCTYPE

基本的に HTML5 で制作を行う。

例)

■HTML5 定義テンプレート

```
<!DOCTYPE html>
<html lang="ja">
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width,minimum-scale=1.0,maximum-scale=10.0,
user-scalable=yes">
<meta name="description" content="">
<meta name="keywords" content="">
<title>PAGETITLE | SITENAME</title>
<link rel="stylesheet" href=" common/css/import.css">
<script src="common/js/jquery.js"></script>
<script src="common/js/common.js"></script>
</head>
<body>
中略
</body>
</html>
```

4. viewport

viewport の設定は原則下記とする。

```
<meta name="viewport" content="width=device-width,minimum-scale=1.0,maximum-scale=10.0,
user-scalable=yes">
```

※仕様によって表示が崩れる可能性がある場合は user-scalable を固定とする。

推奨

PC 向けサイトでモバイルフレンドリー対応をする場合は、viewport は以下のように設定する。その際、画面幅の設定値はコンテンツ幅とするが、必要な場合はコンテンツの左右に余白を追加する。

```
<meta name="viewport" content="width=コンテンツ幅 (+左右余白分) ">
```

5. 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 の提唱する HTML5 の仕様に準拠したマークアップを行う。

6. 改行とインデント

- ソースの改行は、ブロック要素の終了タグの後等、適宜行う。
- body 内の要素には、構造に合わせて適切にインデントを施し、視認性の高いソースコードを実現する。
- インデントには tab (半角スペース 4 つ分) を使用し、全角・半角スペースを使用しない。

7. パスの記述ルール

7.1. 一般ルール

- サイト内リンク/画像のリンク共に、指定がない場合は原則として相対パス (ドキュメント) とする。
 - SSI や CMS の都合上、ルート相対パス (サイトルート) で記述する必要がある場合は、事前に双方にて確認を行う。
- ※ルート相対パスの場合は、XAMPP や MAMP など Apache を使用して制作・確認を行う。

例)

- 相対パス (ドキュメント)

```
<a href="../company/index.html">企業情報</a>
```

- ルート相対パス (サイトルート)

```
<a href="/company/index.html">企業情報</a>
```

- 絶対パス

```
<a href="http://coding-factory.com/company/index.html">企業情報</a>
```

7.2.インクルードファイルの場合

■SSI

インクルードファイルは「.html」で作成し、「/common/inc/」内に下記の名前で保存する。

/common/inc/header.html

footer.html

gnavi.html

lnavi.html

例)

```
<!--#include virtual="/common/inc/header.html"-->
```

■PHP

- ・ 相対パス (ドキュメント)

```
<?php include_once(dirname(__FILE__)."/相対パス"); ?>
```

例)

```
<?php include_once(dirname(__FILE__)."/../header.html"); ?>
```

- ・ ルート相対パス (サイトルート)

```
<?php include($_SERVER['DOCUMENT_ROOT'].'ルート相対パス'); ?>
```

例)

```
<?php include($_SERVER['DOCUMENT_ROOT'].'common/inc/header.html'); ?>
```


8. タグの記述ルール

8.1. HTML5 の使用タグについて

■以下のタグについては、適宜使用する。

<section> …… 一つのセクションであることを示す
<nav> …… ナビゲーションであることを示す
<article> …… 記事であることを示す
<aside> …… 余談・補足情報のセクションであることを示す
<header> ……ヘッダーであることを示す
<footer> …… フッターであることを示す
<figure> …… 図表であることを示す
<figcaption> …… 図表のキャプションであることを示す

8.2. 主要なタグの共通ルール

■<title>タグ

・ ページ名、親カテゴリ名、サイト名の順に、「|」（全角縦棒）で区切る。
親カテゴリがない場合： ページ名 | サイト名
親カテゴリにあたるカテゴリが多い場合： ページ名 | カテゴリ名 | サイト名

■<meta>タグ

・ keywords は、キーワード毎に半角カンマ「,」で区切ること。
※半角カンマ以外で区切ると、区切りとならずに最初から最後までがひとつのキーワードとなってしまう為。

■<section>タグ

・ <section>タグ内には、最初に<hX>を入れること。
※レイアウト目的のみで<section>タグを使用しない。

■<header>タグ

・ <header>タグ内には、サイトタイトル・ロゴ及びコンテンツの目次や検索フォームなどを含むものとする。

■<footer>タグ

・ <footer>タグ内には、サイトまたは直近のコンテンツ内容に関するリンク集、著作権データなどの情報を含むものとする。

■<nav>タグ

・ グローバルナビとローカルナビは、<nav>タグで囲む。

■<hX>タグ

- ・<h1>は、サイトタイトル・ロゴ及び、ページ固有のタイトルにあたるテキスト、もしくは img 要素に対して使用する。ただし、<h1>は 1 ページに対して、最大 2 つまでとする。
- ・ページ内では文書の論理構造に合わせて、必ず見出し要素を使用すること。

■<figure>タグ

- ・コンテンツ内で参照される図表等は<figure>タグで囲む。
- ・<figure>タグ内でのキャプション等の簡易説明は<figcaption>タグで囲む。

例)

```
<figure>

  <figcaption>図 1</figcaption>
</figure>
```

■タグ

- ・alt 属性にて画像の代替となるテキストを設定する。
- ※特に指定が無い場合は、その画像の簡易説明を alt の内容とする。
- ※装飾用の画像など、相応しい代替テキストが存在しない画像の場合には、「alt=""」と設定する。

推奨

- ・レンダリング時に画像エリアのレイアウト確保が必要な場合は、画像のサイズ属性と属性値を指定する。

8.3. フォーム関連タグの記述ルール

- <input>タグ、<textarea>タグ、<select>タグの id 属性に対してスタイルを適用しない。

※id 属性は変更されることを考慮し、class 属性を使用しスタイルを適用する

- チェックボックスとラジオボタンには<label for="任意値">と対応した id 属性を必須とする。

"label" + ["_"] + [連番]

例)

```
<label for="label_01"><input type="checkbox" id="label_01">ラベル</label>
```

- ラジオボタンには name 属性を必須とする。

"radio" + ["_"] + [連番]

例)

```
<label for="label_01"><input type="radio" name="radio_01" id="label_01">コーディング</label>
<label for="label_02"><input type="radio" name="radio_01" id="label_02">ファクトリー</label>
```

■<input>タグの checked と、<select>タグの selected 指定はデザイン（ワイヤー）に合わせる。

■フォーム全体を form タグで囲み、属性に method="post" と action="（遷移先 URL）" を指定する。
例)

```
<form method="post" action="遷移先 URL">
  ～～ フォーム ～～
</form>
```

■HTML5 のフォーム関連タグについて

・以下のフォームタグについては、適宜使用する。

```
<input type="search"> …… 検索テキストの入力欄を作成する
<input type="tel"> …… 電話番号の入力欄を作成する
<input type="url"> …… URL の入力欄を作成する
<input type="email"> …… メールアドレスの入力欄を作成する
<input type="datetime"> …… UTC（協定世界時）による日時の入力欄を作成する
<input type="date"> …… 日付の入力欄を作成する
<input type="month"> …… 月の入力欄を作成する
<input type="week"> …… 週の入力欄を作成する
<input type="time"> …… 時間の入力欄を作成する
<input type="number"> …… 数値の入力欄を作成する
<input placeholder> …… 入力欄に初期表示する内容を指定する
```

9. 重複したソースの記述について

■原則ワンソースで対応するが、レイアウト上やむを得ない場合は、デバイス毎にソースを記述する。
例)

```
<figure>
  
  
</figure>
```

```
@media screen and (max-width:767px){
  .onlyPC { display: none;}
  .onlySP { display: block;}
  img.onlySP { display: inline-block;}
}
```

10. パンくずリストの記述ルール

パンくずリストをより効果的にクローラーへ伝えるため構造化データで設置する。

このマークアップにより、そのページがどのようなカテゴリのコンテンツなのかが把握されやすくなる。

その構造化データでパンくずを設置するには、schema.org (<http://schema.org/BreadcrumbList>) に定義されているボキャブラリーをマークアップします。記述方法は Google から推奨されているシンタックス「microdata」でマークアップする。

- パンくずが示す URL を「itemprop="item"」で指定。
- パンくずに記載するページ名を「itemprop="name"」で指定。
- パンくずの階層構造は「itemprop="position"」で指定。パンくずの先頭から数えて、content="1" content="2"…とする。

例 1)

```
<ol itemscope itemtype="http://schema.org/BreadcrumbList">
<li itemprop="itemListElement" itemscope itemtype="http://schema.org/ListItem">
<a itemprop="item" href="トップページの URL"><span itemprop="name">トップ</span></a>
<meta itemprop="position" content="1"></li>
<li itemprop="itemListElement" itemscope itemtype="http://schema.org/ListItem">
<a itemprop="item" href="下層 1 の URL"><span itemprop="name">下層 1</span></a>
<meta itemprop="position" content="2"></li>
<li itemprop="itemListElement" itemscope itemtype="http://schema.org/ListItem">
<span itemprop="name">下層 2</span><meta itemprop="position" content="3"></li>
</ol>
```

例 2)

```
<div class="breadcrumbs" itemscope itemtype="http://schema.org/BreadcrumbList">
<span itemprop="itemListElement" itemscope itemtype="http://schema.org/ListItem">
<a itemprop="item" href="トップページの URL"><span itemprop="name">トップ</span></a>
<meta itemprop="position" content="1" /></span>
<span itemprop="itemListElement" itemscope itemtype="http://schema.org/ListItem">
<a itemprop="item" href="下層 1 の URL"><span itemprop="name">下層 1</span></a>
<meta property="position" content="2"></span>
<span itemprop="itemListElement" itemscope itemtype="http://schema.org/ListItem">
<span itemprop="name">下層 2</span><meta property="position" content="3"></span>
</div>
```

11. その他の記述ルール

■要素名および属性名はすべて小文字で記述する。

例)

要素 : div、p、input、img

属性 : id、class、type、src

■タグ内での属性の記述順は class="オリジナル 汎用"とする。

例)

```
<div class="sectionContact mt10">
```

■汎用クラスは多用しない。

■終了タグは省略しない。

例)

</p>、</div>、</tr>、、</option>

■CSS での再現が困難な場合を除き、必然性のない（レイアウトを目的とした）table コーディングは行わない。

■半角カタカナは原則として使用しない。

■メールアドレスはエンティティ文字に変換し記述する。

■機種依存文字の記述には、下記のようなエンティティ文字を使用する。

主なエンティティ文字の例)

記号	意味	エンティティ文字（文字参照）
<	小なり不等号	<
>	大なり不等号	>
&	アンパサンド	&
	半角スペース(改行禁止)	
®	登録商標	®
©	著作権	©
™	トレードマーク	™
¥	円	¥

■コメントのルール

・ソース内容が把握しやすいよう、class、id 付きの閉じ</div>の前に必ずコメントを入れる。

※コメントを挿入することで、ソースの視認性を上げ人的ミスの発生を抑えることを目的とする。

例)

```
<!-- /contents --></div>
```

IV CSS 制作ルール

1. バージョン

原則、CSS3 を使用する。

※CSS3 に対応していないブラウザの対応については、プログレッシブ・エンハンスメントの項(III.7)を参照。

2. 文字コード

HTML に合わせる。

3. フォントファミリー

フォントファミリーは以下の順序で記述する。

"ヒラギノ角ゴ Pro W3", "Hiragino Kaku Gothic Pro", Osaka, "MS Pゴシック", "MS PGothic", Arial, Helvetica, Verdana, sans-serif;

4. CSS 記述の統一

4.1. id 名

id 名は以下のルールに沿って命名する。

■文書構造上、意味のある名前にすること。

■ローワーキャメル方式で命名すること。

例)

wrapper

header

headerIn

headerUtility

headerLinks

headerLogo

gNavi

contents

main

sub

INavi

footer

footerIn

copyright

4.2. class 名

class 名は原則、以下のルールに沿って命名する。

- 文書構造上、意味のある名前にすること。
- 2 語以上のローワーキャメル方式で命名すること。
- 末尾に 2 桁の連番を付けること。

例)

```
categoryList01  
secTutorial01  
btnTweet01
```

上記の中で、バリエーションのあるパーツの場合は、Dreamweaver のコードヒント機能を活用するために以下の命名ルールを採用する。

[タグ名] + ["_"] + [識別名] + [連番] (+ ["_"] + [詳細])

- [識別名]と[詳細]は「_(アンダースコア)」で連結すること。

例)

```
ul_arrow01, ul_circle01, ul_square01  
h2_basic01, h2_basic01_in, h2_basic02, h2_basic02_in  
table_spec01, table_spec02
```

- 継承必須 class の命名

他の class からセレクトアの継承をしている場合に限り、1 単語の class 名を付けてよい。

例)

```
ul_entry01 {}  
  ul_entry01 li {}  
    ul_entry01 li .time {}  
    ul_entry01 li .name {}  
    ul_entry01 li .data {}  
    ul_entry01 li .detail {}
```

- 特殊 class 名について

以下の用途については class 名を統一する。

- ・ アクティブ時の class 名 : "current"
- ・ オーバー時の class 名 : "over"
- ・ div, section 要素の入れ子

基準となる要素の親要素 : "sectionXXXWrap"

基準となる要素の子要素 : "sectionXXXIn"

4.3. 一般 css 記述

■セレクトタについて

- ・ 1 行 1 プロパティとし、各プロパティはタブインデントを付ける。
- ・ プロパティの値の前には、半角スペースを入れる。
- ・ 継承させる場合はセレクトタにインデントを付ける。
- ・ 連続して同種のセレクトタに同じようなプロパティをかける場合は、1 行でまとめてもよい。

例)

```
.example {
    display: block;
    margin: 0 auto;
    width: 900px;
}

.example01 a { margin-left: -100px;}
.example02 a { margin-left: -200px;}
.example03 a { margin-left: -300px;}
```

※SASS や CMS 等で CSS が自動で出力される場合、また、読み込み速度向上を目的としたファイル容量軽減を行う場合は、上記の限りではない。

■ショートハンドプロパティについて

- ・ 一括指定可能なプロパティ (margin 等) の個別指定 (margin-top 等) は、1 つまでとする。
- ・ 2 つ以上の個別指定は一括指定にまとめる。

誤)

```
margin-top: 4px;
margin-right: 10px;
margin-bottom: 8px;
margin-left: 20px;
```

正)

```
margin: 4px 10px 8px 20px;
```


■色の指定

色は以下の2通りの記述方法で適宜指定をする。

i) 16進数6桁で記述する。省略できるものは3桁で記述する。

※英字は小文字を使用する

例)

```
#fde96c #d6e0f4 #000 #999 #f00
```

ii) RGBA値で記述する。

例)

```
rgba(200,100,255,0.5)
```

■コメントアウト

コメントアウトは以下のように記述する。

・大見出し

```
/* =====  
Comment  
===== */
```

・中見出し

```
/* Comment  
----- */
```

・小見出し、注記

```
/* comment */
```

■CSS hackについて

CSS hackはできるだけ使用を避けるものとし、やむを得ず使用する場合はIEに対してhackを使用すること。FirefoxやGoogle Chrome、Safariでレイアウトが崩れていたとしても、通常の記述でそれらの表示に合わせ、IE用にhackの記述を行うこと。

例) IE8以下のみ背景を緑に、それ以外では背景を赤にする場合

①で赤色を指定 → ②でIE10以下に緑色を指定 → ③でIE9とIE10に赤色を指定

```
セレクトタ {  
  background-color: red;    - ①  
  background-color: green\9; - ②  
}  
セレクトタ:not(:target) {  
  background-color: red \9;  - ③  
}
```

5. 初期設定 CSS

5.1. cmn_layout.css

「タグのリセット」と「ヘッダーやフッターなど、ページのレイアウトを管理する」CSS。
案件毎に設定する。

基本項目	説明
HTML5 elements	HTML5 要素のリセット
Reseting	タグのリセット
Reset Module	モジュールのリセット
Clearfix	Clearfix
Layout	大枠のレイアウト
header	ヘッダー
gNavi	グローバルナビ
breadcrumb	パンくず
mainimg	メインイメージ
contents	メインコンテンツ
pageTop	ページトップへボタン
footer	フッター
h(n)eading	見出し
Listing	リスト
Table	テーブル
Section	汎用性のあるボックス

5.2. cmn_style.css

頻出する class を設定した CSS。案件によって必要なものがある場合は都度編集する。

クラス名	説明
fwX	font-weight の設定
fzX	font-size の設定
taX	text-align の設定
vaX	vertical-align の設定
blockX	block 要素の中央・右配置
wNNN	px 単位での幅の設定
wHalf, wTri, wQuart, wFull, wMax	%単位での幅の設定。基本的にカラム分けするもの
wNNper	%単位での幅の設定。3~50%できりのいい数値
flL(R)	float 指定 (IE6 のマージンバグに対応する為、display: inline 付き※IE6 対応サイト制作時にのみ適用する。)
flmgL(R)	float する画像の設定。margin の値は適時変更。
ftBox	回り込みを禁止する。基本的に、使用する場合は flmgL(R)とセットで使う。
clear, block, hide, bgN, tdU, tdN	特定の CSS のみの設定
over, png, js_popup_width_height, js_window_close	JavaScript 呼び出し用 class
atode	後で制作するパーツに付ける目印の設定。
mX00	メディア幅を問わないマージンの設定(例: mb15)
pX00	メディア幅を問わないパディングの設定(例: pb15)

6. メディアクエリ記述ルール

原則、@media 規則で css ファイル内でメディアごとのスタイルを記述する。

例)

```
body{
    . . . . .
}
@media(max-width:768px){
body{
    . . . . .
}
}
```

7. プログレッシブ・エンハンスメントについて

原則、CSS3 でデザインを実現する。

ただし、プログレッシブ・エンハンスメントの概念に則り、CSS3 に対応していないブラウザにおいては、デフォルトの仕様に準拠したデザイン実現度を許容する。その際は、コンテンツの可読性は損なわない事を前提とする。

例)

テキストにドロップシャドウの装飾を施す場合、Firefox/chrome では text-shadow プロパティでドロップシャドウの装飾を施すが、text-shadow プロパティは IE9 に対応していないので、ドロップシャドウの装飾はされない。

8. ベンダープレフィックスについて

Firefox および Chrome における CSS3 非対応のプロパティについては、ベンダープレフィックスを用いて、デザインを実現する。

例)

```
-moz-border-image: url("bg_dot.png") 15 round; // Firefox 向け  
-webkit-border-image: url("bg_dot.png") 15 round; // Google Chrome、Safari 向け
```

V JavaScript 制作ルール

1. 記述場所の前提

原則、複数ページで共通に使用するスクリプトは外部ファイル化する。

2. 文字コード

HTML に合わせる。

ただし、ダウンロードしてくるものは原則変更しない。

※HTML ファイルと JavaScript ファイルの文字コードが異なる場合、キャラクターセットの指定を行うこと。

```
<script type="text/javascript" src="/common/js/common.js" charset="UTF-8"></script>
```

3. 初期設定 JavaScript

3.1. jQuery.js

原則最新(2.xxx)を使用する。ただし、対応内容によっては適宜適切なバージョンのものを選択する。

JavaScript ライブラリ： <http://jquery.com/>

※使用時は毎回ダウンロードすること。

3.2. common.js

共通設定の JavaScript。挙動/設定については下記参照。

■ロールオーバー

要素に class="over"を付与すると、

「_no」のファイル名をマウスオーバー時「_on」に切り替える。

■スムーズスクロール

<a>要素を内包するブロック要素に class="scroll"を付与すると、指定の箇所までスクロールする。

VI ディレクトリ作成ルール

1. 基本的なディレクトリ構成

```
root/
|
├ index.html
|
├ common/ (共通フォルダ)
|   ├── css/
|   |   ├── cmn_style.css
|   |   └── cmn_layout.css
|   ├── img/
|   └── js/
|       ├── jquery.js
|       └── common.js
|
├ css/ (TOP ページの CSS)
|   └── style.css
|
├ img/ (TOP ページの画像)
|
├ js/ (TOP ページの JavaScript)
|
└ category/
    ├── css/ (カテゴリー毎の CSS)
    ├── img/ (カテゴリー毎の画像)
    ├── js/ (カテゴリー毎の JavaScript)
    └── index.html
```

2. ディレクトリ名使用文字

ディレクトリ名に使用できる文字は以下のとおり。

- 「a」～「z」までの小文字のアルファベット (1 バイト)
- 「0」～「9」までの英数字 (1 バイト)
- 「- (ハイフン)」と「_ (アンダースコア)」 (いずれも 1 バイト)
- 単語が複数連続する場合、単語間に「_ (アンダースコア)」を使用する。

※ただし、システム等からの自動で出力されるファイルに関しては上記の限りではない。

下記のものにはディレクトリ名には使用しない。

- 先頭には「- (ハイフン)」 「_ (アンダースコア)」を使用しない。
- スペースは使用しない。

3. ディレクトリ文字数

ディレクトリ名は半角 31 文字以内で収めるようにすること。

4. ディレクトリルール

■共通画像とディレクトリ別画像のディレクトリは分ける。

共有画像：common/img/

ディレクトリAのみで使用する画像：ディレクトリ名A/img/

ディレクトリBのみで使用する画像：ディレクトリ名B/img/

■基本的に英単語表記にし、なるべく内容が分かるような命名をすること。

誤) 「sv001/」 「usrsup/」 「contfrms/」

正) 「services/」 「support/」 「contact/」

■数字を用いる必要がある場合は、01,02...09,10,11 など、原則 2 桁の連番にすること。

ただし、2 桁では足りない場合に限り 3 桁の連番を使用してもよい。

VII ファイル作成ルール

1. ファイル名使用文字

ファイル名に使用できる文字は以下のとおり。（ファイル名には拡張子必須）

- 「a」～「z」までの小文字のアルファベット（1バイト）
- 「0」～「9」までの英数字（1バイト）
- 「-（ハイフン）」と「_（アンダースコア）」（いずれも1バイト）

※ただし、システム等からの自動で出力されるファイルに関しては上記の限りではない。

下記のものはファイル名には使用しない。

- 先頭には「-（ハイフン）」「_（アンダースコア）」を使用しない。
- 2バイト文字とスペースは使用しない。

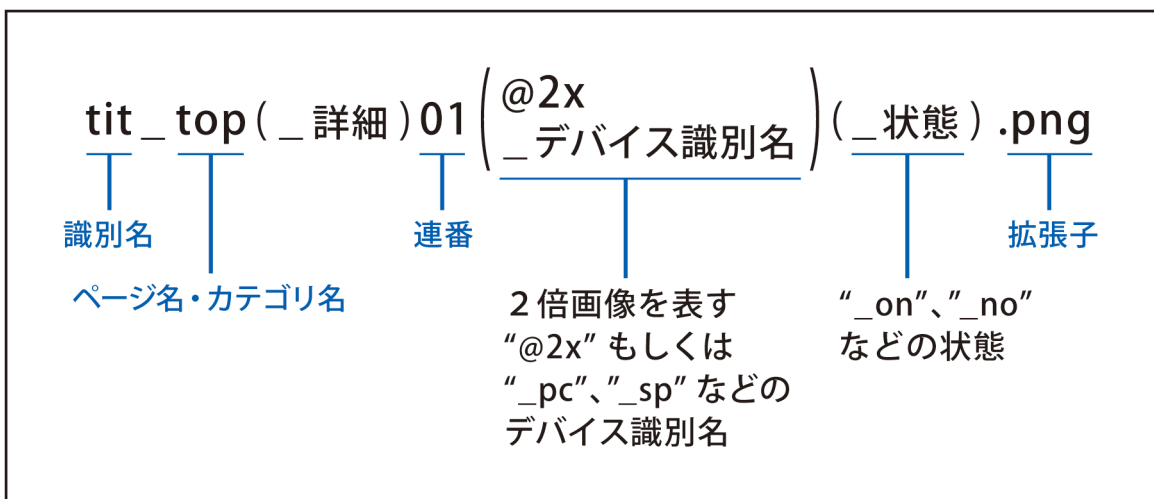
2. ファイル名文字数

- 拡張子を含む半角小文字 31 文字以内とする。
- ページ名が長い場合は省略できるものとする。

3. 画像ファイル作成ルール

3.1. 画像の命名ルール

- 画像名は以下の通りとする。



※ 「ページ名・カテゴリ名」 / 「詳細」 / 「デバイス識別名」 / 「状態」 は不要だと判断される場合は省略可。

※同種の画像が他にない場合でも、必ず2桁の連番をつける。

例)

tit_top01.png

bg_lineup_form01_sp.png

pic_service01@2x.png

btn_contact01_on.png

■識別名一覧（サイト内に制作する主な画像の命名ルールは下記を参照）

識別名	用途	命名サンプル
bg	背景や、ポスター用にリピートさせて使われる画像	bg_top01.png bg_dotx01.png bg_dotx_3x1_01.png
btn	ボタンとして使われる画像	btn_contact01_no.png btn_contact01_on.png
ico	アイコンとして使われる画像	ico_arrow01.png ico_square01.png
bnr	バナーとして使われる画像	bnr_sub01.jpg
h1～h6	<h1>タグなどの論理タグで囲まれる見出し画像	h2_business01.png h3_service01.png
tit	h1～h6 等に用途が限られない見出し画像	tit_topic01.png
txt	デザイン上、画像化するテキストの画像	txt_corporate01.png
pic	商品や人物、風景などの写真画像	pic_about01.jpg
thumb	写真と連動するサムネイル（小さい）画像。	thumb_about01.jpg thumb_about02.jpg
img	写真以外のイラストや図表などの画像	img_about01.jpg
mainimg	ページのメイン画像	mainimg_about01.jpg

■特殊識別名一覧

識別名	用途	命名サンプル
logo	サイトのシンボル画像（必ず common/img/に入れる）	logo01.png
gnavi	グローバルナビゲーションとして使われる画像（必ず common/img/に入れる）	gnavi01.png
lnavi	ローカルナビゲーションボタン（必ず common/img/に入れる）	lnavi01.png
接尾名		
_pc / _sp / _tb	PC とスマートフォンとタブレットで画像を出し分ける際、PC 専用画像、スマートフォン専用画像、タブレット専用画像の最後に付ける。	logo01_pc.png logo01_sp.png logo01_tb.png
@2x	Retina ディスプレイ用に画像サイズを2倍でスライスした画像に付ける。	logo01@2x.png

■状態一覧（画像の状態を表す命名ルール。画像名の最後に付ける。）

状態名	用途	命名サンプル
_on / _no	マウスオーバーで画像を切替える場合に、マウスオーバー用画像に「_on」、通常時画像に「_no」を追加する。	btn_contact01_no.png btn_contact01_on.png
_cr	カレント時用の画像に追加する。	lnavi01_cr.png
_top / _bottom	複数画像を組み合わせ一つの画像を形成する場合に、画像の詳細説明を追加する。	bg_faq_form01_bottom.png

3.2. 画像の書き出しルール

■jpg

色数が多く、サイズの大きい写真などの画像で使用する。

書き出す際は、原則として以下のように圧縮するが、クオリティに配慮しながら、極力ファイルサイズは軽減すること。

Fireworks : 100%

Photoshop : 80~90%

■gif

原則、使用しない。主にアイコンや簡単なテキスト画像など、色数の少ない、あるいは透明を含む画像などの場合は、png-8を用いる。

■png

透明を含む画像、スプライト用画像などに使用する。

pngを使用する場合は軽量化の処理を行うこと。

png-8：主にアイコンや簡単なテキスト画像など、色数の少ない、あるいは透明を含む画像などに使用する。

png-24/png-32：半透明の画像、色数が多く透明を含む画像など。

※png-24/png-32はjpgやgifと比べてファイルサイズが大きいため、透過の用途以外では使用しない。

3.3. 高解像度ディスプレイ対応

高解像度ディスプレイ対応のため、原則として表示サイズの2倍で書き出しをする。

3.4. デバイスを考慮したスライス

2倍画像で書き出しをする際の画像サイズは、偶数ピクセルサイズとする。

4. 不要ファイル削除ルール

納品時には下記の不要ファイルを削除する。

■Thumbs.db

■_notes

■.bak

■.DS_Store および 「_(ドットアンダースコア)」から始まるファイル名のデータ

5. CMS テンプレート作成ルール

各ファイルの H1～H3、meta タグ (keyword、description、title)、SEO 対策テキスト (P タグ) の呼び出しについては以下のように記述する。

※制作資料に特記がある場合はそれに準ずる。

5.1 共通

投稿記事のテキスト箇所は div タグを使用。

※システムが生成するタグが div タグのため p タグの使用は不可。

CMS アーカイブ一覧ページと CMS カテゴリー一覧ページは、重複ページとなるため noindex とする。

5.2 H1～H3 タグの設定

■CMS 一覧ページ (index.html)

ページタイトル + 「新着一覧」となる見出しを h1、詳細記事タイトルを h2。

■CMS アーカイブ一覧ページ (archive.html) / CMS カテゴリー一覧ページ (category.html)

「〇〇〇〇年〇〇月の記事一覧」 / 「〇〇カテゴリー名〇〇の記事一覧」を h1、詳細記事タイトルを h2。

■CMS 記事詳細ページ (detail.html)

記事タイトルを h1。

5.3 META タグテキストおよび SEO 対策テキストに関して

META および SEO 対策テキスト箇所は共通の CSV ファイル(meta.csv)を一部読み込む仕様とする。

※以下、記事タイトル等を代入するような箇所は [読み込む内容] として記載。

■CMS 一覧ページ

他の静的ページ同様に CSV ファイル(meta.csv)をそのまま読み込む。

例)

```
<!--#include virtual="/common/inc/meta.html"-->
<!--#include virtual="/common/inc/seo.php?p=meta"-->
```

■CMS アーカイブ一覧ページ／CMS カテゴリー一覧ページ

- ・ <head>内 meta タグ (keyword、description、title)

```
<title>[アーカイブ名/カテゴリー名] [metaCSV ファイル内 title 行のテキスト]</title>
<meta name="keywords" content="[アーカイブ名/カテゴリー名],[metaCSV ファイル内 keyword 行のテキスト]">
<meta name="description" content="[アーカイブ名/カテゴリー名]。 [metaCSV ファイル内 description 行のテキスト]">
```

- ・ <body>内の SEO 対策テキスト

```
<p id="*">[アーカイブ名/カテゴリー名] [metaCSV ファイル内 SEO 用テキスト (P)行のテキスト]</p>
```

■CMS 記事詳細ページ

- ・ <head>内 meta タグ (keyword、description、title)

```
<title>[ブログ記事タイトル] [metaCSV ファイル内 title 行のテキスト]</title>
<meta name="keywords" content="[metaCSV ファイル内 keyword 行のテキスト]">
<meta name="description" content="[ブログ記事本文の冒頭 100 文字]">
```

- ・ <body>内の SEO 対策テキスト

```
<p id="*">[ブログ記事タイトル (すべて) ] [metaCSV ファイル内 SEO 用テキスト (P)行のテキスト]</p>
```

6. on click タグの指定

各ファイルの H1～H3、meta タグ (keyword、description、title) 、SEO 対策テキスト (P タグ) の呼び出しについては以下のように記述する。

※制作資料に特記がある場合はそれに準ずる。

6.1 グローバルサイトタグ (gtag) の場合

用途	a タグの中に記述する on click タグ
スマホ表示に出る電話番号	onClick="gtag('event', location.href, {'event_category': this.href,'event_label': 'TEL'});"
予約、お問い合わせボタン	onClick="gtag('event', location.href, {'event_category': this.href,'event_label': 'CONTACT'});"
外部の予約システム等へのボタン	onClick="gtag('event', location.href, {'event_category': this.href,'event_label': 'contact_OUT'});"
ランディングページにおける電話番号箇所	onClick="gtag('event', location.href, {'event_category': this.href,'event_label': 'LP_tel'});"
ランディングページにおける全お問い合わせ箇所	onClick="gtag('event', location.href, {'event_category': this.href,'event_label': 'LP_contact'});"

ランディングページにおける全外部の予約システム等へのボタン	onClick="gtag('event', location.href, {'event_category': this.href,'event_label': 'LP_OUTcontact'});"
-------------------------------	---

6.2 旧ユニバーサルタグ (ga) の場合

用途	a タグの中に記述する on click タグ
スマホ表示に出る電話番号	onclick="ga('send', 'event', this.href, location.href, 'TEL');"
予約、お問い合わせボタン	onclick="ga('send', 'event', this.href, location.href, 'CONTACT');"
外部の予約システム等へのボタン	onclick="ga('send', 'event', this.href, location.href, 'contact_OUT');"
ランディングページにおける電話番号箇所	onclick="ga('send', 'event', this.href, location.href, 'LP_tel');"
ランディングページにおける全お問い合わせ箇所	onclick="ga('send', 'event', this.href, location.href, 'LP_contact');"
ランディングページにおける全外部の予約システム等へのボタン	onclick="ga('send', 'event', this.href, location.href, 'LP_OUTcontact');"

VIII コーディング後チェック

1. 文法チェック

HTML5-Lint（文法チェックツール）を用いて検証テストを行う。

<http://www.htmlhint.net/html-lint/htmlhint.html>

※6 点以上のエラーを無くす。ただし、仕様上やむを得ずエラーの記述が含まれる場合は、該当箇所のエラーについては許容する。

2. ブラウザ目視チェック

表示のズレ、リンク先のチェック等、目視でのチェックが必要な項目をブラウザごとに検証する。

※下記に記載の端末でチェックが行えない場合は、実際にチェックを行なった端末・ブラウザ名を記載し報告する。

■検証ブラウザ（標準対応ブラウザ）

対象ブラウザ		
デスクトップ	スマートフォン	タブレット
<ul style="list-style-type: none"> ■Windows 10 <ul style="list-style-type: none"> ・ Firefox 最新 ・ Google Chrome 最新 ・ Edge 最新 ■Windows 7 <ul style="list-style-type: none"> ・ Internet Explorer11（純正） ■Mac OS X 10.13 <ul style="list-style-type: none"> ・ Safari 11 	<ul style="list-style-type: none"> ■iOS10.x iPhoneSE <ul style="list-style-type: none"> ・ Safari ■iOS11.x iPhone7 <ul style="list-style-type: none"> ・ Safari ■iOS12.x iPhoneXS <ul style="list-style-type: none"> ・ Safari ■ AndroidOS 8.x Xperia X Compact <ul style="list-style-type: none"> ・ Chrome 	<ul style="list-style-type: none"> ■iOS11.x iPad(6th) <ul style="list-style-type: none"> ・ Safari ■ AndroidOS 7.x HUAWEI dtab Compact <ul style="list-style-type: none"> ・ Chrome

3. パフォーマンス・表示速度チェック

1. 【パフォーマンスチェック】トップページスコア（基準スコア）

Google サービスである「PageSpeed Insights」で計測をする。

【PageSpeed Insights】

<https://developers.google.com/speed/pagespeed/insights/>

- ・スコアは Medium 以上を基準（モバイル、パソコン共）
- ・Low の場合は「最適化についての提案」の項目が実施されているか確認

「最適化についての提案」で実施する項目は以下。

■圧縮を有効にする

- ・gzip 圧縮はベーシック認証に記述

■ブラウザのキャッシュを活用する

- ・ベーシック認証に記述

■画像を最適化する

- ・画像の最適化は以下のようなツールを用いる。

PNG : <https://tinypng.com/>

JPG : <https://kraken.io/web-interface>

2. 【ページ表示速度】速度テスト（秒数上限）

Google サービスである「testmysite」で計測をする。

【testmysite】

<https://testmysite.withgoogle.com/intl/ja-jp>

- ・表示速度は「5 秒」をしきい値とする
- ・5 秒を越える場合は PageSpeed Insights の「最適化についての提案」が実施されているか見直す。