

株式会社
ユーザー・センタード・デザイン

Web サイト
制作ガイドライン

version 3.0.1

(発効日: 2010 年 07 月 02 日)

本ガイドラインの構成

- 1 目的
- 2 ターゲットユーザ
 - 2.1 閲覧環境
 - 2.2 注意事項
- 3 製作
 - 3.1 案件の管理
 - 3.2 Web サイトの構成
 - 3.3 アクセシビリティ・ユーザビリティ
- 4 XHTML
 - 4.1 基本
 - 4.2 レイアウト
 - 4.3 head 要素
 - 4.4 body 要素
 - 4.5 非推奨要素/属性
- 5 CSS
 - 5.1 基本
 - 5.2 スタイルレイヤー
 - 5.3 セレクタ
 - 5.4 予備知識
 - 5.5 ハック
- 6 命名規則
 - 6.1 基本
 - 6.2 ディレクトリ
 - 6.3 HTML/CSS ファイル
 - 6.4 レイアウトの部位
 - 6.5 画像ファイル

更新履歴 (赤字：最新更新箇所、青字：前回更新箇所)

2010/07/02 version3.0.1 に更新。

2010/01/07 version3 発行。

1. 目的

株式会社ユーザー・センタード・デザインが製作する Web サイトの品質向上を目指す。

本ガイドラインに準拠することで、Web 標準、SEO、ユーザビリティ、アクセシビリティ、メンテナンスに配慮した Web サイトを製作することができる。

2. ターゲットユーザ

2.1. 閲覧環境

■出力メディア

PC スクリーンとプリントを対象とする。

必要に応じて印刷用の CSS を用意し、プレビューで必ず確認する。

ただし、ユーザの選択無しに印刷用レイアウトが適用される場合、ユーザビリティを損なう点に配慮する。また、最低限の処置として、“用紙に合わせて縮小”機能の無い IE6 の印刷結果が用紙内に収まる様に配慮する。

■ディスプレイ

1024×768 ピクセルのディスプレイを対象とする。

幅は、760～960 ピクセルを基本とし、案件により適宜対応する。

■ブラウザ

以下のブラウザを対象とする。ただし、案件に応じて対応範囲を調整する。

OS	ブラウザ
Windows	Internet Explorer（以下 IE）6、7、8 Mozilla Firefox 3 Opera 10 Safari 5 Google Chrome 3
Mac OS	Safari 5 Mozilla Firefox 3

■文字コードセット

UTF-8 を基本とし、案件に合わせて適宜対応する。

HTML と CSS で使用する文字コードを統一する。

2.2. 注意事項

■ディレクトリ閲覧の防止

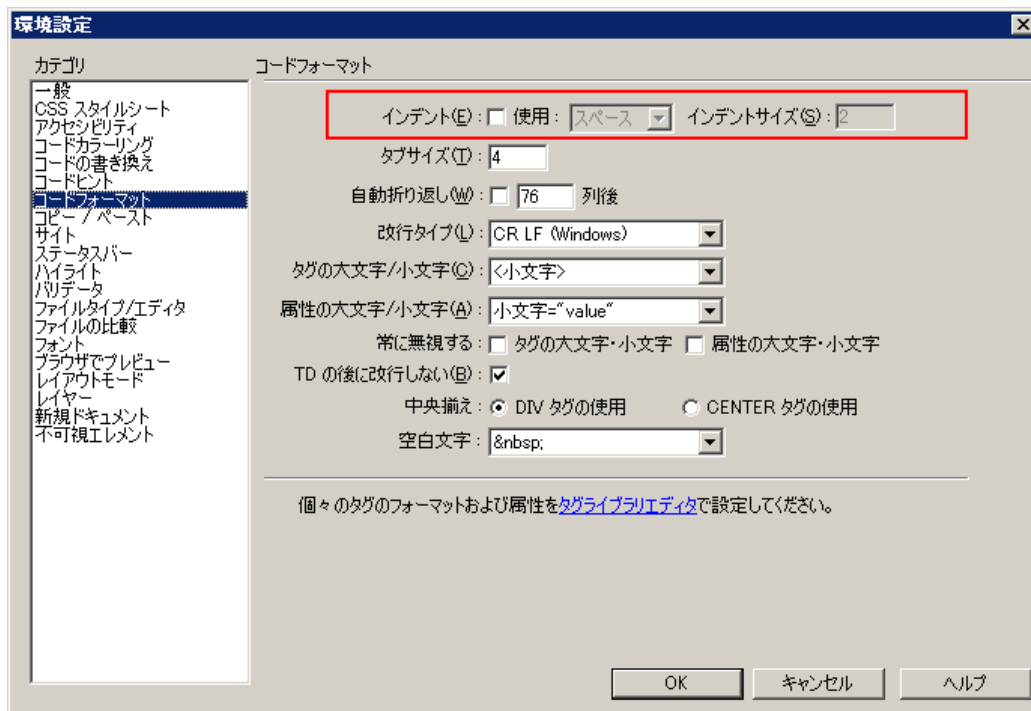
フォルダ内には必ず index.html を作成する。

■Dreamweaver の利用

Adobe 製品使用時に製作されてしまう「_note」フォルダは削除するようにする。

Dreamweaver 使用時は、サイトの定義で「デザインノートの保持」のチェックをはずせばフォルダが出来なくなるので、必ずその様に設定すること。出来てしまった場合は、今の場所で「クリーンアップ」を行えば、設定したローカルルートフォルダ内の「_note」が削除される。

また、HTML のソースを左に揃える為、環境設定でインデント設定からチェックを外すように設定すること。以下参照。



■JavaScript の利用

- ・ JavaScript を利用する場合は外部ファイル化する。html ファイル内への直接の書き込みは極力避けるようにする。
- ・ JavaScript エラーが発生する場合、ブラウザに実装されているデバッガなどを利用して修正作業を行なう。ただし、自力で解決できない場合はシステム担当者に相談すること。

■Flash の利用

Flash の利用はバリデーションへの配慮から外部 JavaScript ファイルを読み込む形で埋め込む。

Dreamweaver の機能による埋め込みは煩雑である事やバージョン違いを考慮し、原則として使用しないようにする。具体的には、以下の方法を推奨する。

・jQuery を使用する場合

jQuery Media Plugin(<http://malsup.com/jquery/media/>)を使用する。サーバへのリクエストを減らすため、jQuery の js ファイル内に追記するのが望ましい。使用法は URL 及びサンプル HTML を参照すること。

・jQuery を使用しない場合

swfobject v2(<http://code.google.com/p/swfobject/>) を使用する。読み込み方法（スタティック・ダイナミック）の設定は任意。使用法は適宜参照すること。

3. 制作

3.1. 案件の管理

■案件開始時

案件に関わることが確定している人を対象にミーティングを行う。

■ファイルサーバの使用について

以下のルールに従うこと。

- ファイルを変更する前に必ずバックアップをとる
- クライアントから受け取ったファイルは整理し、必ずチームで共有する
- ファイルの整理は、チームごとに適宜決定し、決めた方法に必ず則る

案件のディレクトリ構成と各ディレクトリの役割を図示する。



/	案件全体を包括
/html/	Web サイトを包括
/html_bk/	/html/のバックアップを保存する 命名規則は yyyyymmdd とする
/document/	案件に関する社内文書を包括
/document/cost/	クライアントに提出した見積書を包括
/document/pw/	サーバー情報のドキュメントファイルを包括
/sozai/	クライアントから受け取ったファイルを包括
/source/	Web サイトを作るためのファイルを包括 ファイルが多い場合は、png、psd などフォルダを作って保存

■納品と返却

- 納品および返却の際は、不要なデータ（テストページやテスト画像など）を削除した状態にすること。
- パートナーは、納品完了後、資料を納品書とともに返却すること。返却方法は郵送または持ち込みとする。

■作業の終了とチェック、文字/文章校正、交換校正、プログラムの動作確認

サイト制作作業の終了後、必ず自分で「制作物チェック.xls」をチェックし、修正作業を行う。終了後ディレクターに報告する。

■案件終了時

案件に参加した人でミーティングを行い、反省点をまとめる。重要な項目があった場合、全体会議の議題とする。

3.2. Web サイトの構成

HTML ファイルを木構造で構成する。

css ファイル、import 用 css ファイル、画像ファイルは、HTML に属した位置に配置する。

ただし、サイト全体で共通のファイルは common 以下で管理する。

以下にディレクトリ構成例を図示する。

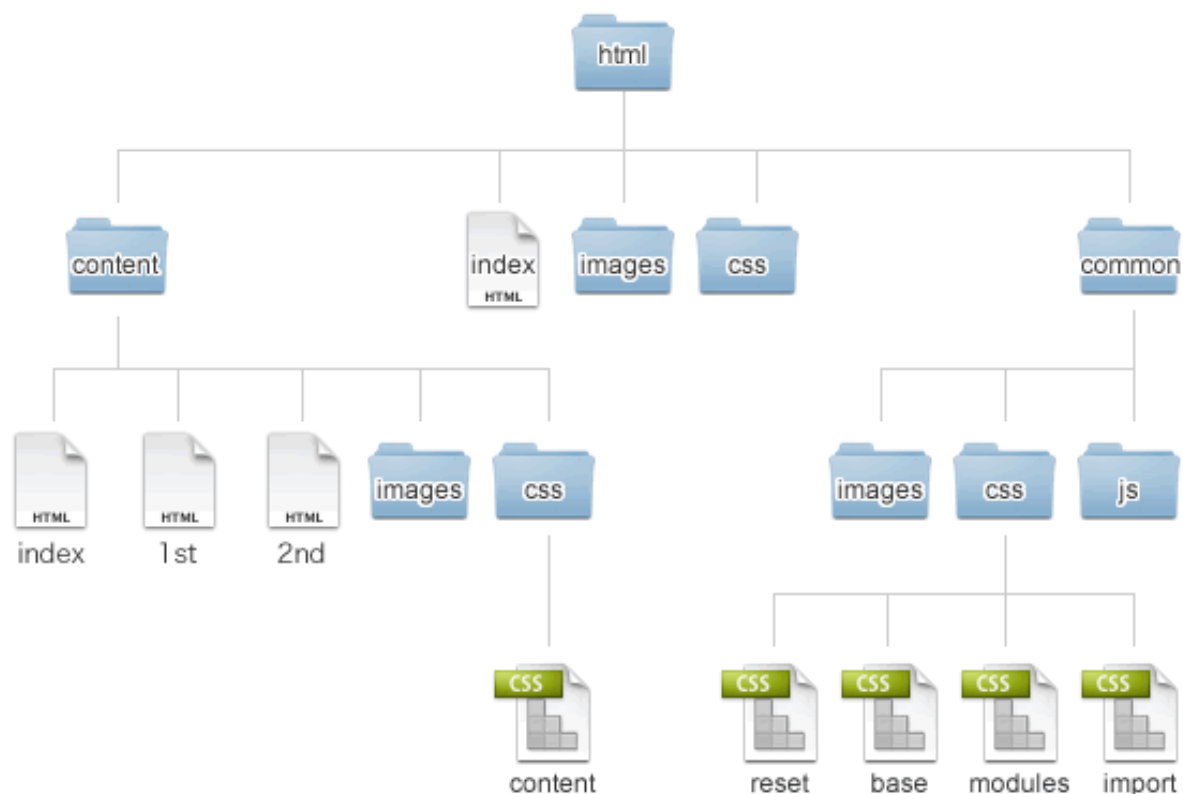


表.ディレクトリ役割表

/	Web サイト全体を包括
/index.html	Web サイトのトップページ
/images/	トップページに使用する画像ファイルを包括
/css/	トップページに使用する CSS ファイルを包括
/content/	コンテンツディレクトリ
/content/index.html	コンテンツのトップページ
/content/1st.html	コンテンツの第 1 ページ
/content/2nd.html	コンテンツの第 2 ページ
/content/images/	コンテンツに使用する画像ファイルを包括
/content/css/	コンテンツに使用する CSS ファイルを包括
/content/css/content.css	コンテンツの CSS ファイル

/common/	Web サイトで共通のファイルを含む
/common/images/	Web サイトで共通の画像ファイルを含む
/common/css/	Web サイトで共通の CSS ファイルを含む
/common/css/reset.css	ブラウザの初期設定をリセットし、再定義した CSS ファイル
/common/css/base.css	Web サイトで共通のスタイルをまとめた CSS ファイル
/common/css/modules.css	CSS で頻繁に使用する部分を汎用化し、まとめた CSS ファイル
/common/css/import.css	reset.css, base.css, module.css をまとめてインポートする CSS ファイル
/common/css/print.css	印刷用のスタイルをまとめた CSS ファイル（印刷対応時のみ）
/common/js/	Web サイトで使用する JavaScript を含む

■ディレクトリ/ファイル名の範例

sitemap	サイトマップ
privacy	個人情報の取り扱い、プライバシーポリシー
access	交通手段、地図、アクセス
company	会社概要
about	開催概要
contact / inquiry	お問い合わせ
link	当サイトへのリンク
faq	FAQ
recruit	採用情報
history	沿革

3.3. ユーザビリティ

■alt 属性

alt 属性を持つ要素は alt 属性を必ず付与すること。画像自体に意味がない、あるいは直前の文章と重複する内容の場合は alt="" と空にして対処する。

■トップページに戻るリンク

ヘッダ部分のロゴに「index.html」に戻れるリンクを設定すること。

4. XHTML ガイドライン

4.1. 基本

■文書型 (XHTML のバージョン)

XHTML 1.0 Transitional を基本とする。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml1-transitional.dtd">
```

■XML 宣言について

XML の仕様では、XML 宣言を記述することが強く推奨されているが、XML の標準仕様に則う事で省略が可能。つまり XHTML の文字コードが Unicode であれば問題ないので、前述の通り文字コードは UTF-8 を基本とする。それ以外の文字コードを使用する場合は担当ディレクターと相談の上決定する事。

■id 属性、class 属性

原則 class 属性を使用する。

要素が一意に特定できる場合 (レイアウトにおける header など) に限り、id を使用する。

id 属性と class 属性は、同一の要素に対して設定可能。

■書式

原則、以下の書式で統一するが、案件によって指定がある場合は適宜対応する。

- インデントによる階層表示は行わず、左詰めとする
- コメントは可能な限りソース内に分かりやすく記述していく。
- 終了タグに id (または class) の名前をコメントで記述する

書式は下記の例を参照し、スペースの有無などに気をつけること。

また、この記述は、レイアウトに使用しているタグ (div や table) においては原則とする。

```
Rtn
<div id="gNav">Rtn
<dl>Rtn
<dt>主なカテゴリ</dt>Rtn
<dd>Rtn
<ul>Rtn
<li>...</li>Rtn
<li>...</li>Rtn
```

```
</li>...</li>Rtn
</ul>Rtn
</dd>Rtn
<!-- / id gNav --></div>Rtn
```

■エラーチェックの実施

XHTML は W3C Markup Validation Service (<http://validator.w3.org/>) で必ずエラーチェックする。「This Page Is Valid XHTML 1.0 Transitional !」とならない場合は問題の箇所を修正し、必ずこの結果になるように調整する。

参考 :

Firefox 用アドオン「Html Validator」でエラーチェックを簡易化できる。

Html Validator

<https://addons.mozilla.jp/firefox/details/249>

4.2. レイアウト

div 要素（または table 要素）で段組を行い、適切な id を設定する。

原則、下記の表に従うものとする。

やむを得ない場合は他のスタッフにもわかりやすい名前を設定すること。

Id 名	備考
layout	ページ全体
header	ページのヘッダ
pageBody	ページのボディ
primary	content 内の主要な要素
secondary	content 内の副次的な要素
gNav	グローバルナビゲーション
lNav	ローカルナビゲーション
content	主要コンテンツ
sidebar	サイドバー
footer	ページのフッタ
topicPath	トップページからの階層構造を示したリンクリスト
banner	ページのバナー表示部
visual	大きめの画像や Flash

4.3. head 要素

■含める内容

head 要素に含める内容について、次の点に気をつける。

- <meta http-equiv="" /> は title 要素の前に記述する。指定するのは Content-Type、Content-Style-Type、Content-Script-Type の 3 つとする。
- <meta name="" /> は title 要素の直後に記述する。
- link 要素や script 要素は head 要素の中で最後に記述する。
- タイトル属性など日本語を使用する要素は、必ず charset の指定後に記述する。
- 各ページ間の移動がスムーズに出来るように、<link rel="contents" href="/" title=" " /><link rel="index" href="/sitemap/" title="サイトマップ" />などの記述を追加する。前後にページがある場合は、next、prev も追加する。
- 基本 CSS(base.css, module.css, reset.css)は import.css でまとめて読み込む。
- フォルダ毎の css は別途読み込む。

これらを踏まえ、以下を基本的な構成とする。

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja" dir="ltr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta http-equiv="Content-Style-Type" content="text/css" />
<meta http-equiv="Content-Script-Type" content="text/javascript" />
<title> (そのページのタイトル) </title>
<meta name="description" content="(そのページの簡単な説明文)" />
<meta name="keywords" content="(そのページのキーワード。半角カンマ区切りで列挙)" />
<link href="common/css/import.css" rel="stylesheet" type="text/css" media="screen, print" />
<link href="css/top.css" rel="stylesheet" type="text/css" media="screen, print" />
<link rel="contents" href="/" title="ホーム" />
<link rel="index" href="/sitemap/" title="サイトマップ" />
<script type="text/javascript" src="common/js/script_top.js"></script>
</head>
```

(参考)

携帯サイトへの対応として、「Mobile Link Discovery」(モバイル・リンク・ディスカバリー)がある。これを設定することで、Google モバイル/Yahoo モバイルに指定した URL が正しくインデックスされるようになる。

各 PC ページから対応する携帯ページへの URL を記述するのが理想だが、更新の手間などの理由で不可能な場合はトップページにのみ入れ込むようにする。

```
<link rel="alternate" media="handheld" href="(サイトの URL アドレス)" />
```

■title 要素

Title 要素は、SEO を意識して各ページ以下のように記述する。

また、以下 title タグ内のパイプ、及びスペースはそれぞれ半角とする。

株式会社ユーザー・センタード・デザインという会社のサイト：トップページの場合

```
<title>株式会社ユーザー・センタード・デザイン</title>
```

株式会社ユーザー・センタード・デザインという会社のサイト：第二階層の場合

```
<title>サービス | 株式会社ユーザー・センタード・デザイン</title>
```

株式会社ユーザー・センタード・デザインという会社のサイト：第三階層の場合

```
<title>WEB 制作 - サービス | 株式会社ユーザー・センタード・デザイン</title>
```

4.4. body 要素

■適切な見出し構造

- h1 要素から出現すること
- 途中の見出しレベルを飛ばさないこと。セクションにおいて、h1→h2→h3 と見出しの出現順序を厳守する。

■h1 要素で定義すべき内容

h1 要素で定義すべき内容は、ページタイトルを基本とする。

■img 要素の注意点

img 要素には必ず alt 属性で代替テキストを指定する。（意味のない画像には alt="" と記述）

また、更新の際の簡略化をはかるため width 属性と height 属性は入力しない。

意味のある画像は置換してはいけない。（例：グラフ、地図等）

■a 要素の注意点

title 属性（補足情報）は SEO 上の特定の指定があるときのみ使用する。（場合によっては、SEO スパムと認識される可能性があるため）

■div 要素の注意点

div 要素はブロック要素を包括する汎用要素なので、インライン要素（img 要素、strong 要素、a 要素 など）を直接含んではいけない。

■隠しテキストの設定

音声ブラウザ・テキストブラウザなどを使用しているユーザのユーザビリティを高める為に必要と考えられる要素は、module.css に記述されている offscreen クラス、あるいは offscreen クラスと同様の CSS プロパティを記述して対処する。

■body 要素の class の記述

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、body 要素に class 属性を設定することで対応しても良い。

また、一つのページに特定のレイアウトやスタイルを指定したい場合も body 要素に class 属性を設定することで対応し、id 属性は極力使用しないものとする。

4.5. 非推奨要素/属性

非推奨要素/属性を使用する事は原則禁止とする。原則外の対処の必要がある場合、事前にスタッフ内で協議のうえで使用すること。

5. CSS ガイドライン

5.1. 基本

■CSS のバージョン

CSS2 (CSS level 2) を採用する。

■エラーチェックの実施

CSS は W3C CSS Validation Service (<http://jigsaw.w3.org/css-validator/>) で必ずエラーチェックする。なお、チェック結果について、「エラー」は必ず修正しなければならないが、「警告」は無視してよい。

※例外として、zoom プロパティの使用を許可する。ただし、印刷対応や本ガイドラインにおけるブラウザ対応を円滑に行う場合のみの使用に限る。

■CSS オフ時でも情報が伝わるページに

CSS をオフにした状態でも情報がきちんと伝わるページにする。Firefox では「ツールバー」→「スタイルシート」→「スタイルシートを使用しない」を選択することで CSS をオフにできるので、この状態で情報の出現順序や流れなどをチェックする。

- style 属性（インラインスタイル）は使用せず、必ず外部 CSS に指定する。
- style 要素は使用せず、必ず外部 CSS に指定する。

5.2. 書式

- インデント（字下げ）は Tab キーで行う
- プロパティごとに改行し、インデントして記述する
- 適度に改行（Rtn キー [Enter キー]）を入れて見やすくする
- 1 行目には必ず @charset の記述をする。XHTML で指定した文字コードに合わせ、基本 UTF-8 とする。
XHTML の文字コードが UTF-8 以外の場合は、その文字コードに合わせる。
- コメントには基本的に日本語は使用しないようにする。
- 適用対象ごとに適切にコメントで区切り、他の部分と区別しやすくする。コメントの記述の注意点として、「/*」のあと、「*/」の前に必ず半角スペースを入れる。（Win IE での不具合回避の為）
- 各セレクタの間は必ず 1 行空け、区切りのコメント前は 2 行空けるようにする。
- 値が「0」の場合は、「px」や「%」などの単位はつけない。
- 原則としてプロパティの順序はアルファベット順とする。

```

Rtn
Rtn
/*----- Rtn
(各部位の id 名と同じ名前を書く) Rtn
(詳細説明がある場合はここに記述) Rtn
-----*/ Rtn

Rtn
div#header Space { Rtn
Tab プロパティ: Space 値: Rtn
Tab プロパティ: Space 値: Rtn
} Rtn
Rtn
div#header p{ Rtn
Tab プロパティ: Space 値: Rtn
Tab プロパティ: Space 値: Rtn
} Rtn
Rtn

```

■デフォルトの記述

CSS ファイルの冒頭には、その CSS のファイル名、作成日、更新日、更新者などが分かるようにコメントを記述する。またその続きとして、そのファイルの内容が分かる目次をコメントとして記述すること。その際に各目次行頭は#つきで記述する。

[例]

```
@charset "UTF-8";

/* *****
Title: base.css
Created: 2010-01-07
Last Modified: 2010-01-07
Editor(s): Hiroto Inukai
Last Editor: Hiroto Inukai
***** */

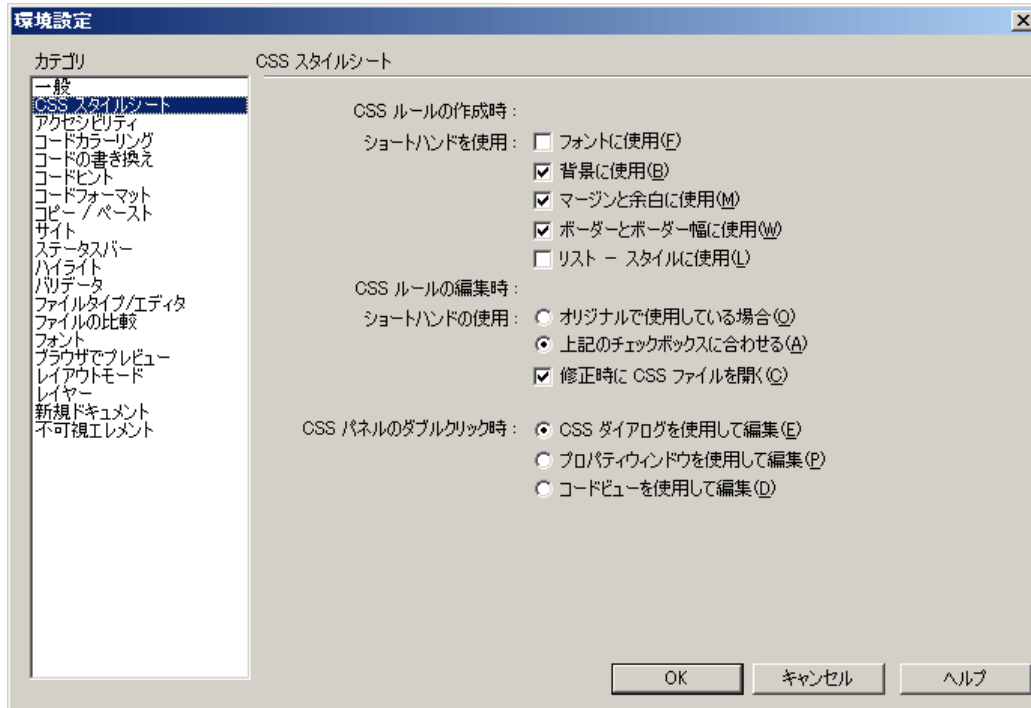
/* Table Of Contents
-----

# General Elements
# body
# layout

-----*/
```

■ショートハンドプロパティ

margin、border、padding、background プロパティはショートハンドで記述するのが望ましい。font プロパティはショートハンドで記述しない。Dreamweaver の環境設定も以下のように変更しておくこと。



```
p.note {
  background: #ddd url(..../images/ico_note.gif) no-repeat;
  border: 1px solid #000;
  font-family: Verdana, Arial, sans-serif;
  font-size: 83%;
  font-weight: bold;
  line-height: 1.5;
  margin: 10px 15px;
  padding: 5px 0;
}
```

■色関連プロパティ

background-color、color などの色関連プロパティの値は 16 進数の小文字表記にする。

(例: #000, #fc3360)

■font-size プロパティ

フォントサイズ指定は、次のルールに基づいて指定する。

- 単位は%を使用する
- 基本サイズは 75%(12px 相当)とする

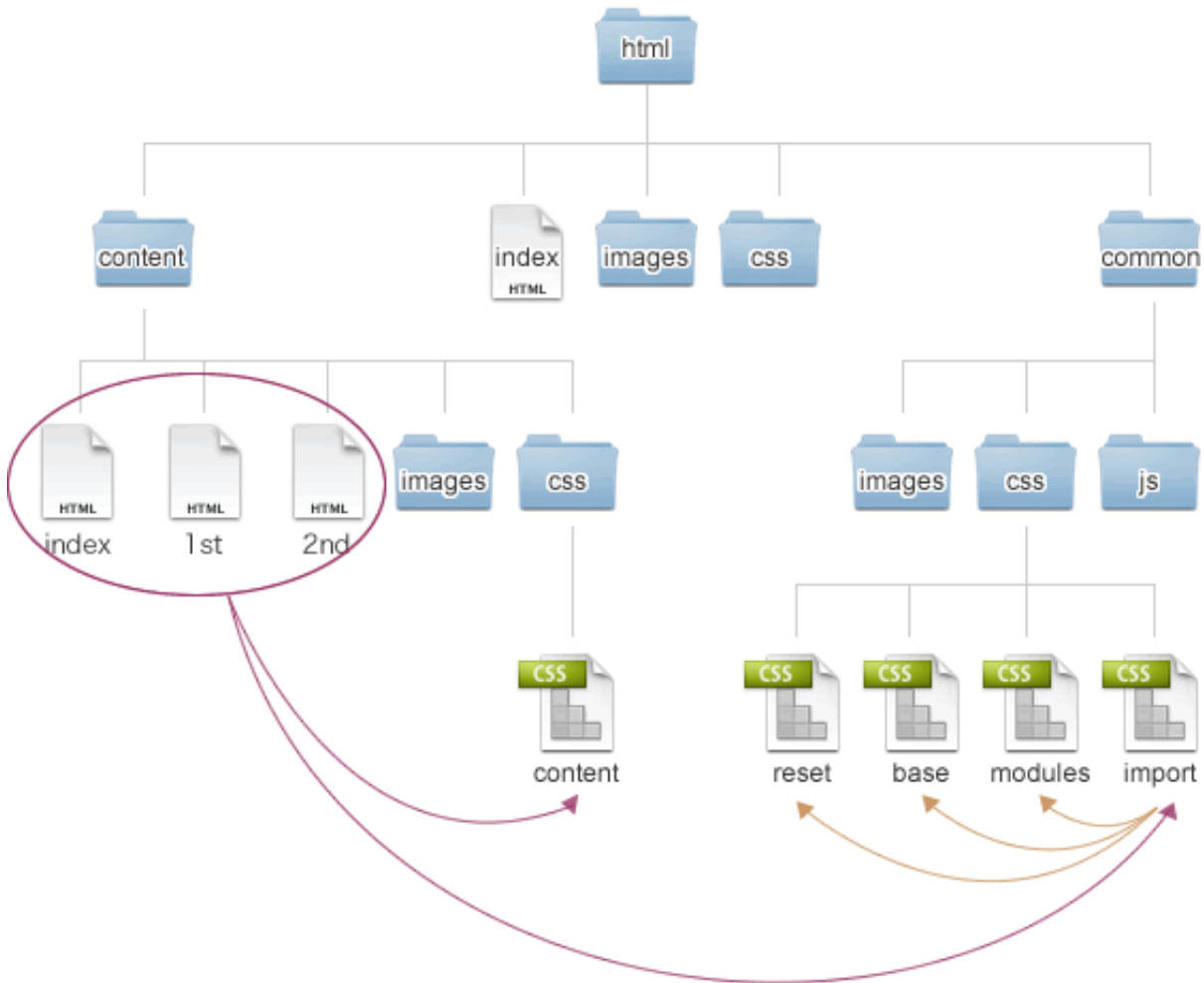
■line-height プロパティ

行高の指定は、値に単位をつけず、小数点以下まで記述する。

(例 : line-height: 1.5)

5.3. スタイルレイヤー

■HTML ファイルによる CSS 参照



5.4. セレクタ

■id/class はなるべく使わず、子孫セレクタを使う

以下のようにセレクタの影響範囲が広すぎる使用の仕方はしない。

以下の例では div#content 以下の全ての div に対してスタイルが影響してしまう。適宜、id/class を指定して対処すること。

[悪い例]

```
div#content div {  
  width: 300px;  
}
```

[良い例]

```
div#content div#news {  
  width: 300px;  
}
```

■子孫セレクタは各部位からはじめる

たとえば「div#header」「div#gNav」「div#content」「div#sidebar」「div#footer」などから子孫セレクタをはじめる。

子孫セレクタで示すツリー構造は、 unnecessary な表現を省く。

[悪い例]

```
div#pageBody div#sidebar ul#INav li a {  
  ...;  
}
```

[良い例]

```
div#sidebar ul#INav li a {  
  ...;  
}
```

■原則 class を使用、一意に特定できるときは id を使用

4.1 を参照。

■id/class には意味的な名前を

id/class には意味的な名前をつける。たとえば「red」などの視覚的な名前をつけない。

詳しくは 6 章を参照。

[悪い例]

```
div#content ul.red {  
  ...;  
}
```

[良い例]

```
div#content ul.photoList {  
  ...;
```

```
}
```

■id/class には要素タイプをつける

[悪い例]

```
#request {  
  ...;  
}  
.examples {  
  ...;  
}
```

[良い例]

```
form#request {  
  ...;  
}  
pre.examples {  
  ...;  
}
```

■id/class 名の区切り記号

id/class 名に単語の区切りが有る場合、「shopList」や「searchBox」のように後ろの単語を大文字ではじめる。

5.5. 予備知識

■プロパティの継承 (要画像差し替え)

明示的に値を宣言しなくても、プロパティには何らかの値がセットされる。

プロパティに対して宣言を行わなかった場合、そのプロパティの値はプロパティごとに次のいずれかの方法でセットされる。

- 親要素における値がそのまま使われる。
- プロパティごとに定められた初期値を使う。



前者をプロパティの値の継承という。継承されるプロパティは、color, font-size, text-align, text-indent,

■画像のロールオーバー（カーソルオン時の画像切り替え）

グローバルナビゲーション・ローカルナビゲーション等の主要パーツ以外は、基本的にJavaScriptを用いて対処する。例えば、以下の様にする。

[例]Jqueryを使用した例。class名にimgoverを指定する事で画像を切り替える

```
//script.js
$(function() {
  $('.imgover').each(function() {
    var osrc = $(this).attr('src');
    var hsrc = osrc.replace(/(¥.gif|¥.jpg|¥.png)/, '_o$1');
    $.data(this, 'osrc', osrc);
    $.data(this, 'hsrc', hsrc);
    $('<img>').attr('src', hsrc);
  }).hover(function() {
    $(this).attr('src', $.data(this, 'hsrc'));
  }, function() {
    $(this).attr('src', $.data(this, 'osrc'));
  });
});
```

5.6. ハック

ハックは基本的には使用しないよう努力する。止むを得ない場合に限り、以下のハックの使用を推奨する。

■Windows 版 IE6 への対応

Windows 版 IE6 のみを対象とするハックは、「*html」という記法を利用する。

[例]

```
p {
  font-size: 116%;
}
*html p {
  font-size: 116%;
}
```

■Windows 版 IE 7 への対応

Windows 版 IE 7 のみを適用対象とするハックは、「*:first-child+html」という記法を利用する。たと

えは次の様にする。

[例]

```
*:first-child+html p {  
  font-size: 116%;  
}
```

6. 命名規則

6.1. 基本

- ユーザにわかりやすい名前をつける
- 原則英単語を使用する（× koushin → ○ update）
- 同義の単語はサイト内で統一する（product≒manufacture≒goods≒commodity→goods）
- 下記の略語対応表にある単語については、略語表記で統一する
- 基本的に、下記の略語対応表以外の略語を使用しない。

下記に、単語の略語対応表を示す。

英単語	統一する表記
background	bg
banner	bnr
button	btn
footer	f
header	h
Icon	ico
image	img
source	src
text	txt

6.2. ディレクトリ

- ディレクトリ名は略さない
- コンテンツを包括するディレクトリの場合は、最も適した英単語を用いる
- 同一のディレクトリ名が複数存在してはいけない
- ディレクトリ名に大文字は使用しない

6.3. HTML/CSS ファイル

- ファイル名に大文字は使用しない
- コンテンツのトップページである場合は、「index.html」とする

- 上位階層の単語は重複になるため使用しない
- 原則として、同一のファイル名が複数存在してはいけない（「index.html」を除く）

6.4. レイアウトの部位

- Id/class にアンダーバーは使用しない
- 複数の単語が連なる場合、2つ目以降の単語の頭文字を大文字にする（例 userCenterdDesign）
- 上位階層の単語は重複になるため使用しない

6.5. 画像ファイル

- 画像ファイルに大文字を使用しない
- 画像ファイル名は使われている箇所がわかるように、画像用途、レイアウトの部位名 (id/class など)、名前、で構成する。
- レイアウトの部位名に大文字が使われている場合は、「アンダーバー+小文字」に変換する
- 命名にあたっては、5.7 の略語と以下の表を参考にする。

用途	参考表記
背景画像	bg_
バナー	bnr_
ボタン	btn_
アイコン	ico_
基本画像	img_
大見出し	bh_
中見出し	mh_
小見出し	sh_
小見出し以降	ssh_...(sを追加していく) or sh_を使用

例

「bh_service.jpg」

「mh_solution.jpg」

「sh_solution.jpg」

「bg_service_commodity.jpg」

「gnav_about_o.gif」

「ico_pdf.gif」