

株式会社アイグローブ

Web サイト

制作ガイドライン

β version 2.2.4

(発効日: 2009 年 5 月 29 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 ユーザビリティ
 - 2.3 注意事項
- 3 製作
 - 3.1 案件の管理
 - 3.2 Web サイトの構成
- 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/属性一覧
- 5 CSS
 - 5.1 基本
 - 5.2 スタイルレイヤー
 - 5.3 セレクタ
 - 5.4 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)

- 2009/05/29 2.1.閲覧環境 (対応ブラウザを更新)
- 2008/10/09 2.1.閲覧環境 (対応ブラウザを更新)
3.2.Web サイトの構成の表内容 (common/css/print.css 追加)
- 2007/07/23 3.2.Web サイトの構成のイラスト及び表内容 (common/css/reset.css 追加)
4.1.基本 ■書式部分のソース (閉じタグのコメントを div の前に移動)
4.3.head 要素 ■含める内容のソース (head 内の記述を変更)
4.4.body 要素 ■隠しリンクのソース (記述方法を変更)
5.3.スタイルレイヤー ■HTML ファイルによる CSS 参照のイラスト
5.2.書式 のソース (CSS の記述方法を詳細に指定)
5.2.書式 ■デフォルトの記述を新規で追加

1. 目的

株式会社アイグローブが製作する 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。

印刷を前提とする場合は、印刷用の CSS を用意し、プレビューで必ず確認する。

■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、760～900 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。

OS	ブラウザ
Windows	Internet Explorer（以下 IE）6、7、8 Mozilla Firefox 3 Opera 9 Safari 4 Google Chrome 2
Mac OS	Safari 4 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. ユーザビリティ

■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。

(ディレクター指示により変更有り)

2.3. 注意事項

■ディレクトリ閲覧の防止

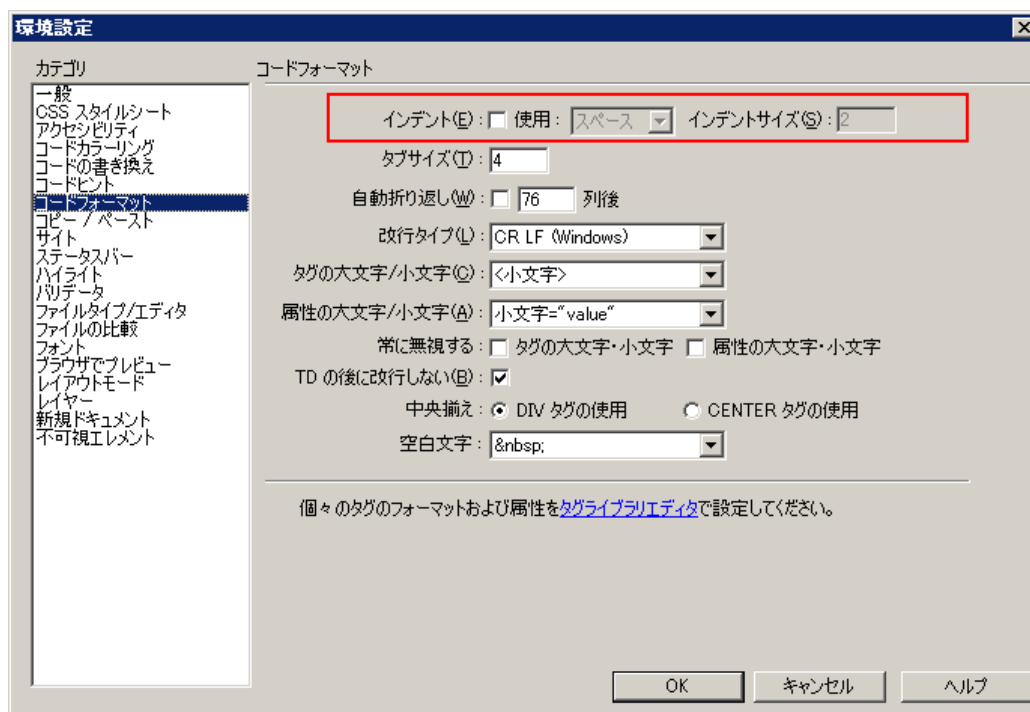
フォルダ内には必ず index.html を作成する。

■DreamWeaver の利用

Macromedia 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、外部ファイル化し (.js)、「/common/js/」フォルダに保存する。どのような目的でその JavaScript を利用しているかを説明するために、JavaScript ファイル内の最上部に

コメントとして記述する。

[例]

```
// ■2006/06/15 (TakahiroMashiko)
// 画像のプリロードのための JavaScript ファイル。
```

なお、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 には 1 度クリックしないと動作しないという「Eolas 問題」があるため、外部 JavaScript ファイルを読み込むかたちで埋め込む。

[JavaScript (任意の名前.js) ソース]

```
document.write('<embed src="FLASHのファイル名.swf" quality="high"
pluginspage="http://www.macromedia.com/go/getflashplayer"
type="application/x-shockwave-flash" width="幅" height="高さ"></embed>')
```

※ SWF ファイルを配置したときの<embed>～</embed>までを「document.write('...')」のシングルクォテーション内に記述する。

[XHTML ソース]

```
<html>
<body>
  <script type="text/javascript" src="任意の名前.js"></script>
</body>
</html>
```

※ フラッシュが入る場所に <script type="text/javascript" src="任意の名前.js"></script> を記述する。

■ノートン対策

ノートン・インターネット・セキュリティの広告ブロックが有効になっている場合、表示されるべき画像やページが表示されないことがある。広告ブロックを無効にすれば問題は発生しないが、広告ブロックを有効にしているユーザーを考慮して制作を進めること。

参考サイトは次のとおりである。

- <http://service1.symantec.com/SUPPORT/INTER/nisjapanesekb.nsf/3c0446b8d7bec4b385256c470047c0ed/b4fe248101f7440e49256c84004694b0?OpenDocument>
- http://www.skullysoft.com/zakki/20030509_8831.shtml

なお、ノートン対策のため、次のようなディレクトリ名は使用しないようにする。

/ad/	/ads/	/advertise/	/advertising/
/banner/	/banner_images/	/banners/	/promote/
/promotions/	/sponsor/	/sponsors/	

3. 製作

3.1. 案件の管理

■案件開始時

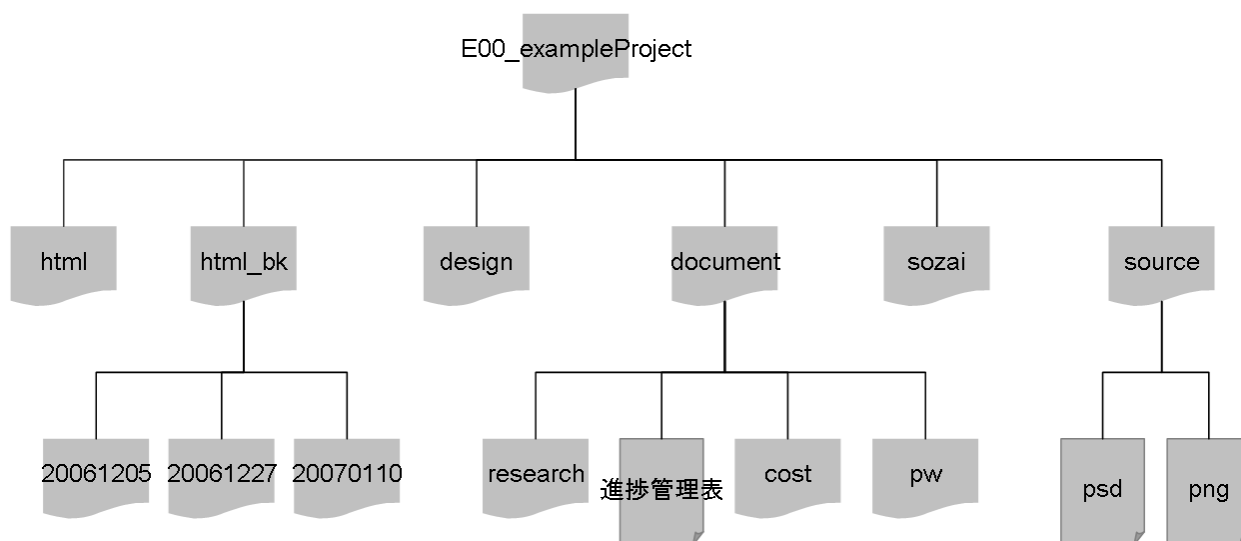
案件に関わることが確定している人を対象にミーティングを行う。

■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含む ファイルが多い場合は、png、psd などフォルダを作って保存
/design/	デザイン確認用に制作した png や html ファイルを含む

■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。

Webサイトのディレクトリ構成

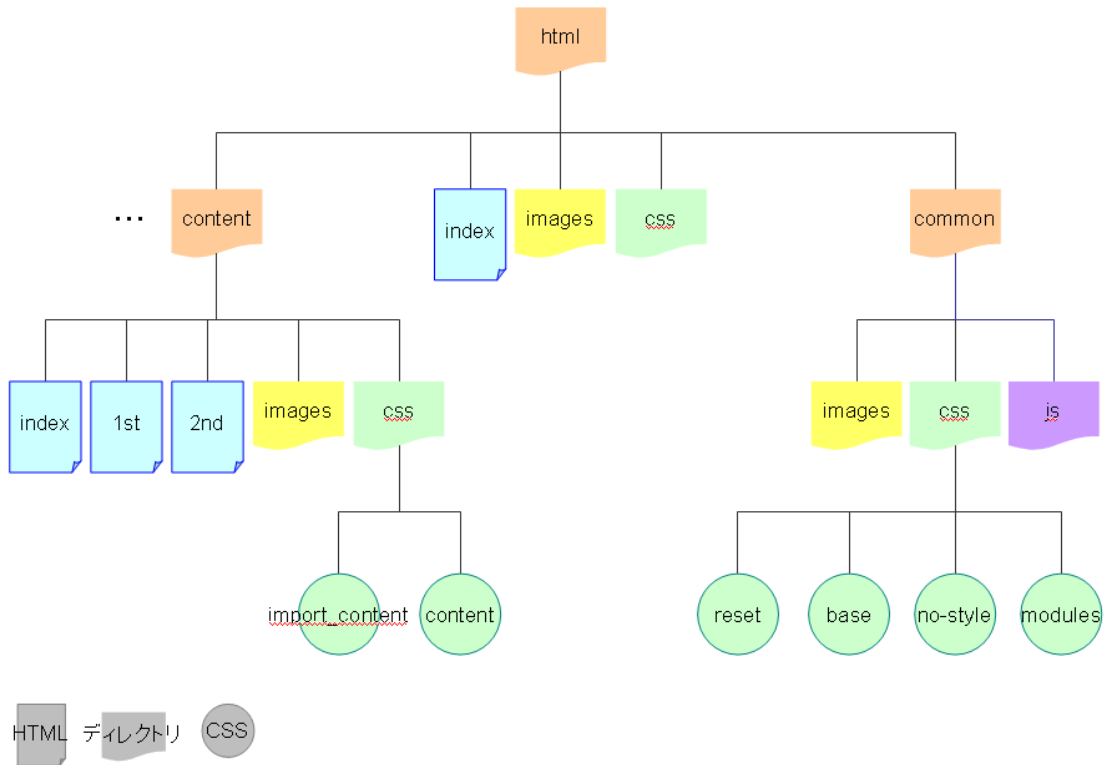


表.ディレクトリ役割表

/	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/import_content.css	コンテンツに使用する CSS ファイルを、まとめてインポートする CSS ファイル
/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包括
/common/images/	Web サイトで共通の画像ファイルを包括

/common/css/	Web サイトで共通の CSS ファイルを包括
/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/no-style.css	IE4.0 においてスタイルを適用しないための 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/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 を包括

■ディレクトリ/ファイル名の範例

sitemap	サイトマップ
privacypolicy	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
overview	会社概要
contact	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

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 strict を基本とし、案件によっては transitional とする。

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

```
<!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の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
- スペースは極力使用せず、CSS で間隔・余白は設定していく。
- 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="gNavi"> Rtn  
<ul class="contentList"> Rtn  
<li>...</li> Rtn  
<li>...</li> Rtn
```

```
</li>...</li> Rtn
</ul><!-- / class contentList --> Rtn
<!-- / id gNavi --></div>Rtn
Rtn
<div id="wrapper"> 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 Strict (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。

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	2 段組レイアウトの主要な要素
secondary	2 段組レイアウトの副次的な要素
gNavi	グローバルナビゲーション
lNavi	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
location	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-Language、Content-Style-Type、Content-Script-Type の 4 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。
- IE3 や NN4 のような古いブラウザでの誤動作を避けるため、「import_xxx.css」を用意して css ファイルを読み込む。IE3 と NN4 は @import に対応していないため、指定した css ファイルが読み込まれない。

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta http-equiv="Content-Language" content="ja" />
<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)" />
<meta name="robots" content="index, follow">
<link href="common/css/no-style.css" rel="stylesheet" type="text/css" />
<link href="css/import_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 language="JavaScript" type="text/javascript" src="common/js/script.js"></script>
</head>
```

なお、html 要素の開始タグは次のように記述する（属性の指定順序に注意）。

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja" dir="ltr">
```

■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。

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

ただし、ページ構成によっては、ある見出しレベルを使うのが難しい場合もあり、このような階層構造を逸脱しなければならない場合もありうる。

■h1 要素で定義すべき内容

h1 要素で定義すべき内容は、ページ最上部のリード文を基本とする。

リード文が無い場合、h1 要素内テキストを画面外に表示することで対応する。(text-indent:-9999px)

■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。

■隠しリンクの設定

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ーのユーザビリティを高める為、「本文」、「グローナビゲーション」、「その他の共通ナビゲーション」にジャンプ出来る隠しリンクを設定

する。

■初めに記述するソース（トップページ用）

```
<div class="naviHidden">
<p>多くのかたに当社ウェブサイトを利用していただくため、アクセシビリティに配慮したウェブサイ
トの制作をおこなっております。</p>
<p>本ウェブサイトでは、JavaScript およびスタイルシートを使用しております。
お客さまがご使用のブラウザではスタイルシートが未対応のため、本来とは異なった表示になってお
りますが、掲載している情報は問題なくご利用いただけます。</p>
<p>以下のリンクより、本文、メニュー、または共通メニューへジャンプができます。</p>
<a href="#content">本文へジャンプ</a>
<a href="#gNavi">メニューへジャンプ</a>
<a href="#guide">共通メニューへジャンプ</a>
<!-- / class naviHidden --></div>
```

■初めに記述するソース（コンテンツページ用）

```
<div class="naviHidden">
<p>以下のリンクより、本文、メニュー、または共通メニューへジャンプができます。</p>
<a href="#content">本文へジャンプ</a>
<a href="#gNavi">メニューへジャンプ</a>
<a href="#guide">共通メニューへジャンプ</a>
<!-- / class naviHidden --></div>
```

■飛び先に記述するソース（飛び先が「グローバルナビゲーション」の時の例）

```
<div class="naviHidden">ここからメニューです<!-- / class naviHidden --></div>
<div id="gNavi">
```

■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に対応しても良い。

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 属性は使用しないものとする。

4.5. 非推奨要素/属性一覧

以下は、XHTML の仕様上、非推奨とされている要素・属性の一覧である。

ただし、a 要素の target 属性や name 属性については、その使用を認める。

Object 要素を使用する場合、target 属性が正常に動作しないブラウザ（IE、FireFox など）があるので、そのときに限り iframe タグを使用しても良い。

他の要素・属性を使う必要がある場合、事前にスタッフ内で協議のうえで使用すること。

■非推奨要素と代替方法一覧

非推奨要素	代替方法
applet 要素	object 要素
basefont 要素	body 要素に font-size プロパティを指定。あるいは、ユニバーサルセレクタ (*) で font-size プロパティを指定
center 要素	ブロックレベル要素については、margin-right と margin-left の両プロパティで値に「auto」と指定。インライン要素やテキストについては、親要素のブロックレベル要素に text-align プロパティで値に「center」と指定
dir 要素	ul 要素
font 要素	color、font-size、font-family の各プロパティを指定
frame 要素	なし
frameset 要素	なし
iframe 要素	object 要素
isindex 要素	form 要素と input 要素（およびその必要な機能を備えたスクリプトなど）
menu 要素	ul 要素
noframes 要素	なし
s 要素	削除文を意味するのであれば del 要素。装飾目的であれば text-decoration プロパティで値に「line-through」と指定
strike 要素	削除文を意味するのであれば del 要素。装飾目的であれば text-decoration プロパティで値に「line-through」と指定
u 要素	強調を意味するのであれば em 要素や strong 要素。装飾目的であれば text-decoration プロパティで値に「underline」と指定

■非推奨属性と代替方法一覧

非推奨属性		代替方法
(全要素)	lang 属性	xml:lang 属性
(a 要素)	name 属性	id 属性
	target 属性	なし
(area 要素)	target 属性	なし
(body 要素)	text 属性	color プロパティ
	bgcolor 属性	background-color プロパティ
	background 属性	background-image プロパティ

	link 属性	a 要素に:link 擬似クラスで color プロパティを指定
	vlink 属性	a 要素に:visited 擬似クラスで color プロパティを指定
	alink 属性	a 要素に:active 擬似クラスで color プロパティを指定
(br 要素)	clear 属性	親要素であるブロックレベル要素に clear プロパティを指定
(caption 要素)	align 属性	caption-side プロパティ
(div 要素)	align 属性	<p>□値が「center」の場合 ボックス自体は margin-right と margin-left の両プロパティで値に「auto」と指定。インライン要素やテキストについては、親要素のブロックレベル要素に text-align プロパティで値に「center」と指定</p> <p>□値が「left」の場合 ボックス自体は margin-right プロパティで値に「auto」と指定。インライン要素やテキストについては、親要素のブロックレベル要素に text-align プロパティで値に「left」と指定</p> <p>□値が「right」の場合 ボックス自体は margin-left プロパティで値に「auto」と指定。インライン要素やテキストについては、親要素のブロックレベル要素に text-align プロパティで値に「right」と指定</p>
(form 要素)	name 属性	id 属性
(h1~h6 要素)	align 属性	text-align プロパティ
(hr 要素)	align 属性	<p>□値が「center」の場合 margin-right と margin-left の両プロパティで値に「auto」と指定</p> <p>□値が「left」の場合 margin-right プロパティで値に「auto」と指定</p> <p>□値が「right」の場合 margin-left プロパティで値に「auto」と指定</p>
	noshade 属性、 size 属性	ボーダー関連プロパティ
	width 属性	width プロパティ
(html 要素)	version 属性	文書型宣言
(img 要素)	align 属性	vertical-align プロパティまたは float プロパティ

	border 属性	ボーダー関連プロパティ
	hspace 属性、 vspace 属性	マージン関連プロパティ
	name 属性	id 属性
(input 要素)	align 属性	vertical-align プロパティまたは float プロパティ
(legend 要素)	align 属性	なし
(li 要素)	type 属性	list-style-type プロパティ
	value 属性	なし
(link 要素)	target 属性	なし
(map 要素)	name 属性	id 属性
(object 要素)	align 属性	vertical-align プロパティまたは float プロパティ
	border 属性	ボーダー関連プロパティ
	hspace 属性、 vspace 属性	マージン関連プロパティ
(ol 要素)	compact 属性、 start 属性	なし
	type 属性	list-style-type プロパティ
(p 要素)	align 属性	<p>□値が「center」の場合 ボックス自体は margin-right と margin-left の両プロパティで値に「auto」と指定。インライン要素やテキストについては、親要素のブロックレベル要素に text-align プロパティで値に「center」と指定</p> <p>□値が「left」の場合 ボックス自体は margin-right プロパティで値に「auto」と指定。インライン要素やテキストについては、親要素のブロックレベル要素に text-align プロパティで値に「left」と指定</p> <p>□値が「right」の場合 ボックス自体は margin-left プロパティで値に「auto」と指定。インライン要素やテキストについては、親要素のブロックレベル要素に text-align プロパティで値に「right」と指定</p>
(pre 要素)	width 属性	width プロパティ
(script 要素)	language 属性	type 属性
(table 要素)	align 属性	float プロパティまたはマージン関連プロパティ
	bcolor 属性	background-color プロパティ

(td 要素)	bgcolor 属性	background-color プロパティ
	width 属性	width プロパティ
	height 属性	height プロパティ
	nowrap	white-space プロパティで値に「nowrap」
(th 要素)	bgcolor 属性	background-color プロパティ
	width 属性	width プロパティ
	height 属性	height プロパティ
	nowrap	white-space プロパティで値に「nowrap」
(tr 要素)	bgcolor 属性	background-color プロパティ
(ul 要素)	compact 属性	なし
	type 属性	list-style-type プロパティ

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。

■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。その為コメントには日本語を使用しないようにする。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が、コメントには基本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。
- 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には、検索をし易くする為に「#01」など#つきで2桁の番号を記述する。

```
@charset "UTF-8";
```

```
/* *****
```

```
Title: base.css
```

```
Created: 2007-07-21
```

```
Last Modified: 2007-07-21
```

```
Editor(s): Hiroto Inukai
```

```
Last Editor: Hiroto Inukai
```

```
***** */
```

```
/* Table Of Contents
```

```
-----  
#01 General Elements
```

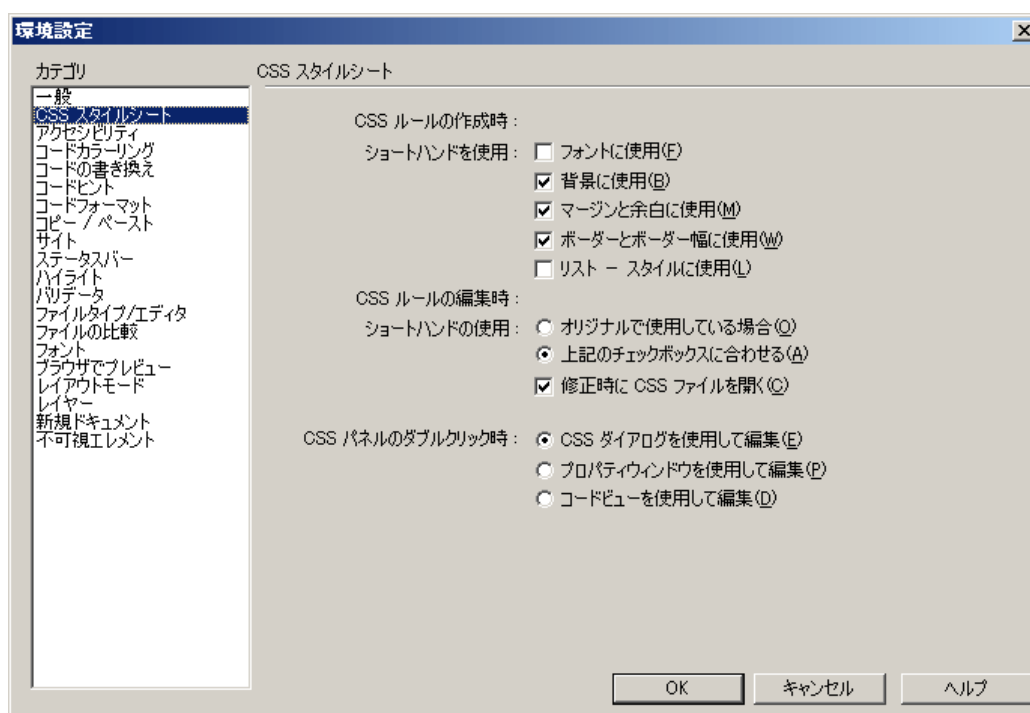
```
#02 body
```

#03 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/p_note_bcg.gif) no-repeat;  
  border: 1px solid #000;  
  font-family: Verdana, Arial, sans-serif;  
  font-size: 78%;  
  font-weight: bold;  
  line-height: 1.5;  
  margin: 10px 15px;  
  padding: 5px 0;  
}
```

■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 78%とする

■line-height プロパティ

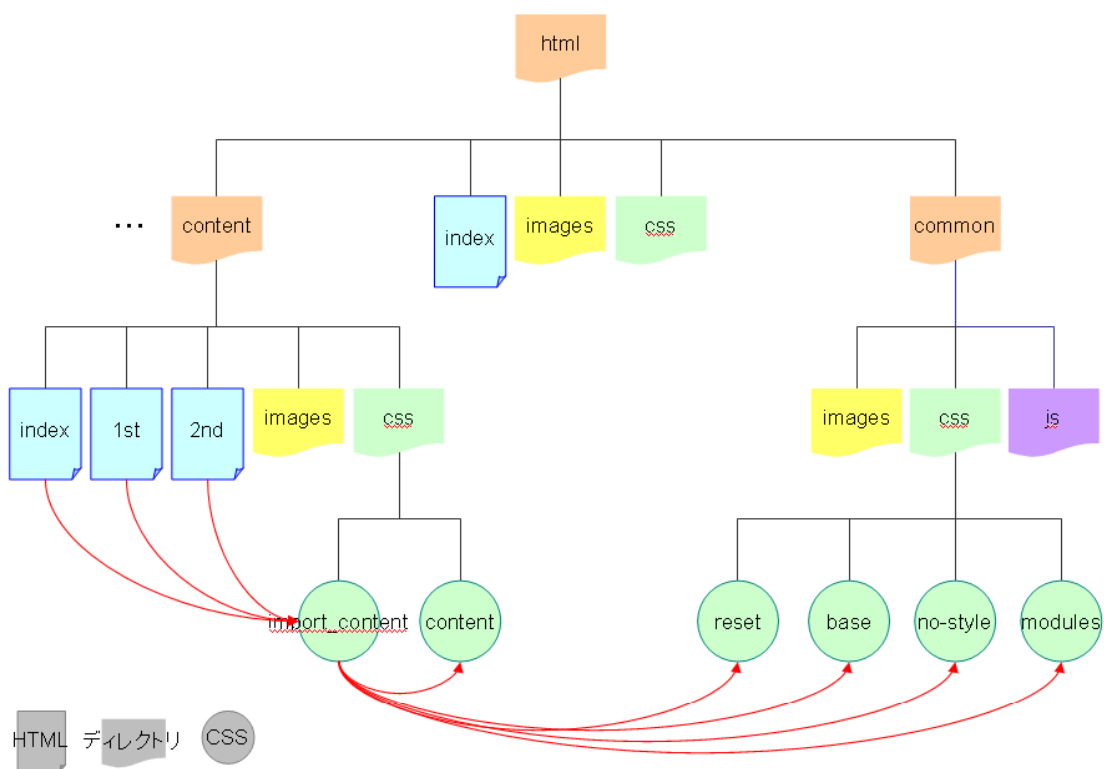
行高の指定は、値に単位をつけず、小数点以下まで記述する。(CSS Validator 対策)

例) line-height: 1.0

5.3. スタイルレイヤー

■HTML ファイルによる CSS 参照

CSSの参照



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

子孫セレクタを積極利用する。id/class は管理が煩雑なため、なるべく使わないようにする。

[悪い例]

```
li.items {  
  ...;  
}
```

[よい例]

```
div#global ul li {  
  ...;  
}
```

■子孫セクタは各部位からはじめる

たとえば「div#header」「div#gNavi」「div#content」「div#sideBar」「div#footer」などから子孫セクタをはじめる。

子孫セクタで示すツリー構造は、 unnecessary な表現を省く。

[悪い例]

```
div#container div#wrapper div#sidebar a {  
  ...;  
}
```

[よい例]

```
div#sidebar 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こと。

詳しくは 6 章を参照のこと。

[悪い例]

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

[よい例]

```
div#main ul.photoList {
  ...;
}
```

■id/class には要素タイプをつける

[悪い例]

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

[よい例]

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

■id/class 名の区切り記号

id/class 名が長くなる場合、「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、line-height、font-family、font-style および font-weight の各プロパティである。

一方、background-color、text-decoration および text-shadow の各プロパティは継承されない。ただし、text-decoration と text-shadow は、装飾は継承されるという例外がある。

■IE の width/height の誤解釈

IE6 の後方互換モードにおけるブロック要素の width/height の解釈にはバグがあり、他のブラウザと表示に差異が生じる。このバグの対処方法は、width/height と padding/border を同一の要素に指定しないことである。

以下に標準の解釈と IE の解釈の違いと、対処方法の具体例を示す。

図 1：正しい解釈

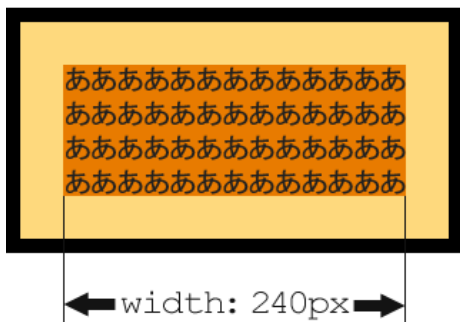


図 2：間違った解釈

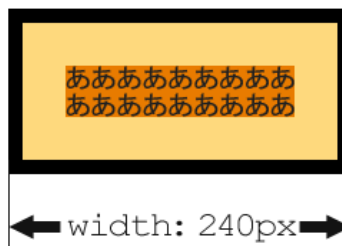
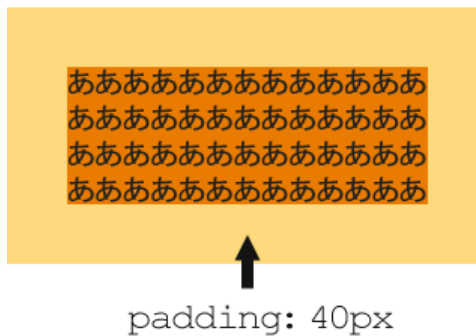


図 1 は標準の解釈である。図 2 は IE の後方互換モードにおける誤解釈であり、width の値に padding と border も含まれる。

図 3：外のブロックで width 指定



図 4：中のブロックで padding 指定



これは、ブロック要素を二重にし、外側のブロック要素で width/height を指定し、内側のブロック要素で padding(あるいは margin)を指定することで対処できる。このとき、内側のブロックの width/height は指定しない (auto にする)。

■margin の相殺 (collapsing margins)

隣接あるいは入れ子の関係にある複数のボックス要素において、間に padding/border の領域を挟まずに隣接する上下の margin は、結合して一つの margin になる。(=値の大きい方の margin をとる)

隣接する左右の margin は結合しないので注意。

■画像のロールオーバー（カーソルオン時の画像切り替え）

JavaScriptではなくCSSを利用する。background プロパティと: hover 擬似クラスを組み合わせるとロールオーバーを実現するのが一般的な方法である。

```
div#global ul li#shopping a {
  background: url(../images/shopping_off.gif) no-repeat left top;
  ...;
}
div#global ul li#shopping a: hover {
  background: url(../images/shopping_on.gif) no-repeat left top;
  ...;
}
```

■画像の疑似プリロード

対象となる要素の背景にロールオーバー後の画像を配置する。そうすることにより、ちらついた場合は背景に配置されたロールオーバーの画像が表示されるのでちらついているように見えない。

[css ソース]

```
div#global ul li#idname a{
  display:block;
  width:○○○px;
  height:○○px;
  background:url(../images/img.jpg) no-repeat;
  outline:none;
}
div#global ul li#idname a: hover {
  background:url(../images/img_a.jpg) no-repeat;
}
div#global ul li#idname {
  background:url(../images/img_a.jpg) no-repeat;
}
```

■特記事項

スタイル指定上、特記事項がある場合は、CSS ファイル内にコメントとして含める。これは特に CSS ハックを利用した場合に重要である。

[例]

```
/*-----  
■2006/06/20 (HirotoInukai)  
Opera 8 以前をターゲットにしたフォントサイズ調整のためのスタイル。  
-----*/
```

5.6. ハック

ハックは基本的に使用しない。

使用する場合は、アンダースコアハックを使用する。

■アンダースコアハック

プロパティの先頭にアンダースコア(_)をつけると IE のみを適用対象とすることができる。

```
p {  
  padding: 10px 0 0 0;  
  _padding: 5px 0 0 0;  
}
```

このように記述することにより、IE には padding-top に 5px が適用され、その他のブラウザには 10px が適用されるようになる。

■スターハック

スターハックは Windows 版 IE のみを適用対象とするためのハックである。

```
p {  
  font-size: 0.8em; /* [hack 1 before] */  
}  
* html p {  
  font-size: 0.80em; /* [hack 1 after for IE] */  
}
```

なお、その他のハックを使用した際は、その直前にコメントを書いておく。

[例]

```
/*-----  
■2006/06/20 (HirotoInukai)  
Opera 8 以前をターゲットにしたフォントサイズ調整のためのスタイル。  
-----*/
```

■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、「*+html」という記法を利用する。たとえば次のようにである。

```
*+html p {  
  ...;  
}
```

また、Windows 版 IE 7 を適用対象からはずす場合は「html:lang(ja)」という記法を利用する。たとえば次のようにである。

```
html:lang(ja) p {  
  ...;  
}
```

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
- 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	bu
footer	f
header	h
lcon	ico
image	lmg
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、レイアウトの部位名 (id/class)、画像の種類、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

例

共通の画像

「h_logo.jpg」

「g_navi_ico_about.gif」

「f_txt_copyright.gif」

「l_navi_allow.gif」

→ 例えば content 内でも使うときは、複製する。

各ページの画像

「h2_service.jpg」

「h3_solution.jpg」

「main_visual.jpg」

「service_commodity.jpg」

「service_bg.gif」

「service_line_dotted.gif」