

株式会社サイブリッジ  
CSSガイドライン



CYBRIDGE



CYBRIDGE

# 目次

---

## 目次

- 目次
- 制作ガイドラインの利用方法
- 基本仕様1
- 基本仕様2
- 基本仕様3
- 基本スタイル1
- 基本スタイル2
- テキスト・フォント
- コメント
- CSSコーディングの確認事項1
- CSSコーディングの確認事項2



CYBRIDGE

## 制作ガイドラインの利用方法

---

### 見方

- 必須** . . . 必ず守っていただきたいルールです
- 推薦** . . . 適用していただくことをオススメするルールです。
- 任意** . . . 任意で適用していただくルールです

### ご質問・お問い合わせ

ご不明な点・疑問点、お困りになった事などがございましたら、ご連絡ください。

株式会社サイブリッジ  
担当者 平山 恵子 TEL 03-6381-5900[代表] 03-6381-5904[直通]



## CSS基本仕様

バージョン：CSS2.1 または CSS3

CSSは原則外部ファイルで管理する。

※CSS3を利用する場合、IEでもhtcやjsで適用されるようにするか、プログレッシブエンハンスメントにするか案件によって判断すること。

## 外部CSSファイル一覧

ファイル名	内容	格納フォルダ
common.css	CSSリセット&デフォルトCSS (OOCSSやレスポンシブ用のデフォルトスタイルを管理)	/css/
style.css	全ページ共通+メインレイアウトに関するスタイルシートファイル	
xxx.css (例：home.css)	各ページの固有スタイルシートファイル	

## セレクタ名の命名ルール

ID・class名は原則として関連する半角英数字を用い、小文字で記述する。

略した単語が一般的ではなく分かりづらい場合は略せずに記載すること。

命名の際、関連する単語が2語以上になる場合は、**2語目以降の先頭文字を大文字で記述する（キャメルケース）**。

※ただし、プラグインの利用などによる命名規則の差異は多少許容する

### 記述例

```
#adHead
```



### エリアに適用するスタイル

原則、各エリアには以下のID・class名をつける。

ファイル名	適用エリア
#wrapper	サイト全体
#container	サイト全体（#wrapperだけで問題ない場合は不要）
#header	ヘッダー
#navi	グローバルナビゲーション
.breadcrumbs	パンくずナビ
#contents	コンテンツエリアとサイドバー
#main	コンテンツエリア
#side	サイドバー
#footer	フッター
.section	メインのコンテンツエリアやサイドバーで各セクション毎に区切る場合
.item	各セクションの中でさらにエリア分けが必要な場合
.inner	さらに一階層下へエリア分けが必要な場合



## OOCSS(Object-Oriented CSS)

デザイン毎のclassを作成してOOCSS(Object-Oriented CSS)でコーディングする。  
再利用可能なモジュールで構成すればコーディングにかかる工数が減らせる他、ページの読み込みスピードの対策にもなる。

## <body>タグでのID振り分け

各ページの<body>タグにIDをつけてスタイルを振り分ける。  
OOCSSの考え方に基づいて、子孫セレクタは短く、可能な限り汎用的なスタイルの記述を心がける。

ファイル名	適用エリア
#pageHome	トップページ
#pageContact	お問い合わせ
#pageCompany	企業概要
#pageSearch	検索結果
#pageDetail	検索詳細

## CSSハック

CSSハックは最低限の使用で抑えるようにし、W3Cの定めるスタイルシートの構文から逸脱しないよう徹底する。

### W3C CSS 検証サービス

<http://jigsaw.w3.org/css-validator/>



## CSSの基本スタイル

### スタイルの適用について

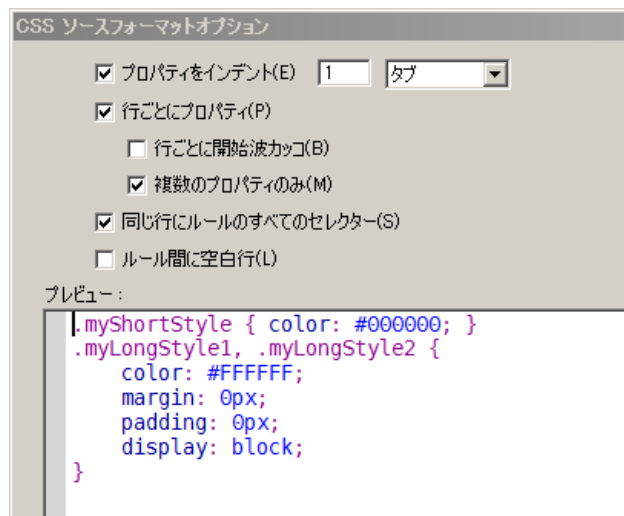
- ・ CSSをわかりやすく記述するために、tabキーによるインデントと改行を使用する。
- ・ Dreamweaverの「CSS ソースフォーマットオプション」を設定し、その記述を基に読みやすくカスタマイズする

### Dreamweaverでの設定方法

1. [編集]から[環境設定]を開く
- ↓
2. [コードフォーマット]をクリック
- ↓
3. [高度なフォーマット]項目の「CSS」をクリック

### 記述例

```
body {  
  font-size: 82%;  
  color: #444;  
  line-height: 137%;  
}  
#wrapper { width: 100%; }
```





## 省略形 (ショートハンド)

プロパティの省略形が可能な場合には省略形を利用し、読み込み速度を意識して短いソースコードを心がける。

### 色指定 (ガラケーサイトの場合は6桁きちんと記述する)

```
.hoge { color:#000; }
```

### 背景画像

```
.hoge { background:#FFF url("/img/common/bg.gif") repeat-x 0 0; }
```

### ボーダー

```
.hoge { border:1px solid #CCC; }
```

### マージン・パディング ※この場合は padding-bottom: 5px; でも構わない。

```
.hoge {  
  margin:12px 0;  
  padding:0 0 5px;  
}
```

## 単位について

各プロパティ毎に推奨される単位は異なる。下記のテーブルを参考にすること。

プロパティ	単位	備考
font-size	% または em	文字サイズはユーザーが指定できるよう原則可変にする。デザイン的に固定する場合はpxも可。 ※9ページも参照。
line-height	単位なし または px	文字サイズに合わせて適用されるように単位ナシで指定する。 emで指定する場合は子要素への影響が異なるので注意。 画像置換のLIR方式やデザイン的に固定する場合はpx指定。
background-position	em または % または px	アイコンを表示している場合など文字サイズを変更した際にbackground-positionの指定を変えなくてもよいのでem指定。デザイン的に必要な場合は%やpxも可。



## 文字の大きさについて

アクセシビリティを考慮しフォントサイズは可変を基本とする。  
 CSSでフォントサイズを指定する場合でも px や pt などのサイズ固定ではなく  
 em や % の形でユーザー可変となるフォントサイズ指定を行う。

必須

フォントサイズの相対指定するため「YUI Fonts CSS」を採用している。  
 ※CB-STANDARDの「common.css」を必ず参照すること。

## 文字の色について

アクセシビリティを考慮して背景色とのコントラストに注意する。  
 16進数表記で3桁に省略できるものは省略する（例：#FF0000 → #F00）

任意

文字状態	CSSセレクタ	色	16進数表記	デコレーション
通常	body	黒	#333 or #444	---
リンク	a:link	青	#06C or デフォルト	デフォルト
訪問済み	a:visited	紫	デフォルト	デフォルト
マウスオーバー	a:hover	青	デフォルト	デフォルト
アクティブ	a:active	青	デフォルト	デフォルト

## CSSのコメント

各カテゴリやページのCSSを記述する際に、どこに対してのCSSかをコメントで記述する。ヘッダーやフッターなどコンテンツの部位に応じて段階毎に下記のような見出しをつけ、CSSの可読性を向上させる。CSSファイルの先頭には索引覧を設置し、見出し（大）、見出し（中）に見出し番号を併記しておくことが望ましい。

※詳細はCB-STANDARDの [common.css](#) 及び [style.css](#) を参照。

### 見出し（大）

```
/*=====
 1. style setting
=====*/
```

### 見出し（中）

```
/* -----
 1-1. Font color styles
-----*/
```

### 見出し（小）

```
/*
 page top styles
-----*/
```



## 基本的な確認事項

実際にコーディングする前に、必ずCB-STANDARDの「common.css」「style.css」を確認すること。  
common.cssには汎用的に使用するclassが準備されているので、**OOCSS(Object-Oriented CSS)の考え方に基づき**、適宜利用する。  
CSSコーディングの際は、style.cssに予め記述されているスタイルを改変していく。  
サイトの規模がある程度大きい場合は、style.cssにサイト共通のスタイルを記述し、各ページごとのCSSを準備すること。

## 画像置換

CSSによる画像置換はtext-indent:-9999pxを採用。理由としてはブラウザで文字検索した際に、隠した文字が表示されてしまうのを防げる手法であるということ。**※Googleは非推奨としている以上、画像置換は必要最小限で利用する。**

### 画像置換の記述例

```
#navi li a {
  display: block;
  overflow: hidden;
  width: 136px;
  height: 30px;
  background: #FFF url("/img/common/gn.gif") no-repeat;
  text-indent:-9999px;
}
```

## CSSでの透過

リンク有りの画像やボタンなどを簡単にCSSでロールオーバーしたい場合は、class="hover"を指定すると、Opacityにより画像を透過できる。**ただし原則ロゴ画像は透過させないこと。**  
※あくまで透過させているので、透過させたいボタン画像の後ろに背景画像が重なった場合は、該当の画像を囲んでいるタグに背景色を指定するか、ロールオーバー時の画像を用意して回避する。



## カラム指定

common.cssの中にclearfix、leftBox、rightBoxというclassが準備されている。  
float:Left/Rightを指定したボックスの上の階層にclearfixをかけることで、floatがclearできる。  
下記のように記述すると、簡単に2カラムレイアウトの作成が可能。

### 2カラムのサンプル

```
<div class="photoBox clearfix">  
  <div class="leftBox"></div>  
  <div class="rightBox"></div>  
</div>
```

## floatの解除

### ■clearfixを使用

common.cssのclearfixを親要素に指定することで、floatの解除ができる。

### ■overflow:hiddenを使用

親要素にoverflow:hiddenでfloatを解除できる。便利だが、表示が途切れてしまうなどの問題が起きるケースもあるので、使用には注意する。

## IEへの対応

IEハックは極力最小限で済むようなコーディングを心がける。一部のCSS3は「PIE.htc」で対応可能。  
IE6へ透過PNG対策には「PIE.htc」では不具合もあるため、「DD\_belatedPNG.js」を使用する。  
※対応バージョンについては、クライアント毎に要確認。