

株式会社クロノドライブ コーディング仕様書

改訂履歴

2009/10/01 初版

2010/12/06 一部修正

2011/05/02 対応ブラウザ、画像ファイルの形式を修正、補足事項追記

はじめに

本書の目的は、制作の効率化、クオリティの向上、各作業者間のスタイルの統一を図る目安になることで、常にブラッシュアップを繰り返し、現時点での最適解の共有に活用するものである

■ 目次

□ 基本仕様

- [ファイル・ディレクトリ構成](#)
- [画像ファイル](#)
- [対応ブラウザ](#)
- [品質管理](#)

□ (X)HTML制作

- [文字コード](#)
- [テンプレート](#)
- [インデント](#)
- [パス指定](#)
- [title要素](#)
- [img要素](#)
- [表記規則](#)
- [その他のルール](#)

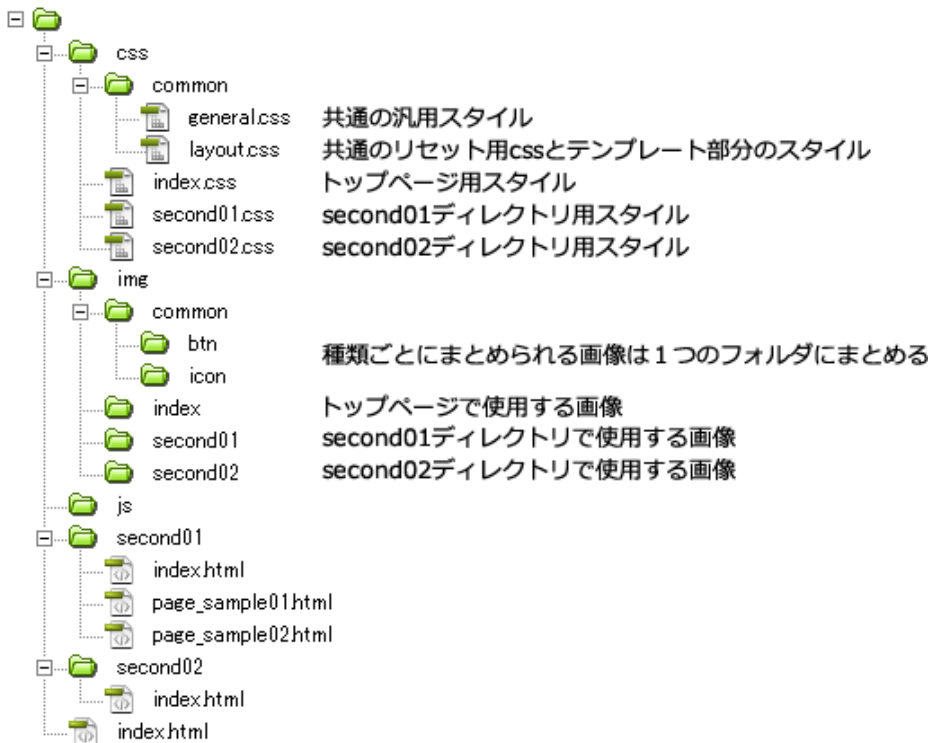
□ CSS制作

- [共通CSS](#)
- [読み込み方法](#)
- [ID名、CLASS名の命名規則](#)
- [ID名サンプル](#)
- [CLASS名サンプル](#)
- [コードフォーマット](#)
- [コメントの記述](#)
- [ショートハンドプロパティの使用](#)



基本仕様 1

■ ファイル・ディレクトリ構成



□ ファイル命名規則

- ディレクトリ名、ファイル名に空白(スペース)は使用しない
- ディレクトリ名、ファイル名は半角の小文字で最大31文字
- 単語が2つ以上続く場合はアンダースコア「_」で繋げる
- 違う意味の単語を区切る場合はハイフン「-」を使用する
- ディレクトリ名、ファイル名は短く、分かりやすい名前にする
- 連番を使用する場合は拡張子の直前に付け、連番の直前にはアンダースコア「_」、ハイフン「-」を付けない

基本仕様 2

■ 画像ファイル

□ 画像ファイル形式

画像の種類に応じて適切なファイル形式を選択する

● JPEG

写真等、色数の多い画像はJPEGを使用する

圧縮率: Fireworks 90% Photoshop 80%

● PNG

写真以外の画像は基本的にはPNG 8を使用する

半透明の画像は、Fireworksの場合はPNG 32、Photoshopの場合はPNG 24で書き出す

● GIF

アイコン等の小さい画像やアニメーションはGIFを使用する

□ 画像ファイル名一例

基本的な命名規則は2頁のファイル命名規則と同様

画像ファイル名は「場所名 + 用途・種類 + 連番 + .拡張子」とする

● 画像ファイル名一例

- 見出し: h2_img01.png h2_img02.png
- ボタン: btn01.png btn02.png
- アイコン: icon01.gif icon02.gif
- グローバルナビ: gnavi01.png gnavi02.png
- 背景: header_bg.png footer_bg.png

基本仕様 3

■ 対応ブラウザ

特に指定のない限り、以下のブラウザをチェック対象として制作する

● Windows

- Internet Explorer 6 / 7 / 8 / 9
- Firefox 最新版
- Google Chrome 最新版

● Mac

- Safari 最新版
- Firefox 最新版
- Google Chrome 最新版

□ 印刷対応について(別途料金)

- レイアウトやブラウザの組み合わせによって対応が難しい場合があるが、できる限りブラウザで表示されている状態に近づける

基本仕様 4

■ 品質管理

デザインとの相違がないか目視によるチェックに加えて、以下のチェックツールを使用し、HTMLファイルの文法等のエラーをできる限りクリアする

- **Web Developer (Firefox拡張)**

- <https://addons.mozilla.org/ja/firefox/addon/60>
- ALT、TITLEテキストのチェックを行う

- **HTML Validator (Firefox拡張)**

- <https://addons.mozilla.org/ja/firefox/addon/249>
- W3CのTidyエンジンを積んだHTML構文検証ツール。該当のページを開き [ソースを表示] し使用する。

- **HTML-lint**

- HTML-lint で90点以上を目指す
- ただし、リンク先が未定の箇所に対して「#」を指定していることによるエラーや、同名のテキストリンクでリンク先が違うことによるエラーなど、回避できないエラーは無視する

(X)HTML制作 1

■ 文字コード

UTF-8を使用

※言語やサーバの仕様により変更する場合がある

■ テンプレート

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ja" lang="ja">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta http-equiv="content-script-type" content="text/javascript" />
<meta http-equiv="content-style-type" content="text/css" />
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE7" />
<meta name="description" content="" />
<meta name="keywords" content="" />
<title></title>
<link href="css/common/layout.css" rel="stylesheet" type="text/css" />
<link href="css/common/general.css" rel="stylesheet" type="text/css" />
<link href="css/index.css" rel="stylesheet" type="text/css" />
</head>
<body>

</body>
</html>
```

(X)HTML制作 2

■ インデント

要素のセクション毎に適切に改行、タブインデントを設けて、ソースを読みやすくする

● インデントの例

```
<div>
  <p>テキストテキストテキスト</p>
  <ul>
    <li>テキストテキストテキスト</li>
    <li>テキストテキストテキスト</li>
    <li>テキストテキストテキスト</li>
  </ul>
</div>
```

■ パス指定

特に指定がなく、特殊な仕様がない場合は相対パスを使用する

● 相対パス記述例

- `<p>同ディレクトリのindexファイルへのリンク</p>`
- `<p>同ディレクトリのindexファイルへのリンク</p>`
※index.htmlを省略するかどうかを選択していただき、対応する
- `<p>1階層上のページへのリンク</p>`

● サイトルート相対パス記述例

- `<p>トップページへのリンク</p>`
- `<p>2階目のページへのリンク</p>`

● 絶対パス記述例

- `<p>ページへのリンク</p>`

(X)HTML制作 3

■ title要素

titleタグの内容の指定がない場合は、該当するページタイトルを先頭に、属する親カテゴリ一名を記述し、最後にサイト名称を、全角パイプ「|」で区切り明記する

● <title>の例

```
<title>ページ名称 | 大カテゴリ | サイト名称</title>
```

```
<title>ページ名称 | サイト名称</title>
```

■ img要素

- 全てのimgタグにalt属性を用いて代替テキストを付加する。
- 文字要素のない写真等は「○○の写真」など、適切な代替テキストを指定する。
- title属性は指定しない。
- width、height を指定する

■ 表記規則

- 要素名、属性名は全て半角、小文字で記述する
- 属性値はダブルクォーテーション「"」で囲む
- 特殊文字は実体参照で記述し、ローマ数字の場合はIとVの組み合わせで表現する
- メールアドレスはSPAM防止のため、16進数でエンコードして記述する
- 英数字は半角で統一する
- ナカグロ「・」は全角で統一する
- コロン「:」は全角で統一する
- 丸括弧()は全角で統一する

(X)HTML制作 4

■ その他のルール

- 一連の文章として、見出し、段落、箇条書きなどのどれに該当するのかを考え、ふさわしい要素でマークアップを行うよう心がける。
- メールアドレスはSPAM防止のため、16進数でエンコードして記述する
- 不要ファイル、バックアップファイルは削除もしくは別フォルダに移動するように努める。

CSS制作 1

■ 共通CSS

- **layout.css**
 - 共通のリセット用cssとテンプレート部分のスタイルを記述
- **general.css**
 - 共通で使用する、汎用的なスタイルを記述

■ 読み込み方法

各HTMLファイルから、上記共通CSSと各ページ固有のcssを読み込む

● CSS読み込み方法の例

```
<link href="css/common/layout.css" rel="stylesheet" type="text/css" />  
<link href="css/common/general.css" rel="stylesheet" type="text/css" />  
<link href="css/index.css" rel="stylesheet" type="text/css" />
```

■ ID名、CLASS名の命名規則

- キャメル方式で命名する(2語目以降の頭文字を大文字にする)
 - 使用可能な文字は、半角英数字、アンダーバー「_」
 - 頭文字に数字は使用できない
 - スタイルや見た目を表す名前ではなく、文書の要素としての命名をするように心がける
- **ID名、CLASS名の例**
 - × id="SideNavi" ○ id="sideNavi"
 - × class="fontRed" ○ class="notice"

CSS制作 2

■ ID名サンプル

- container
- header
- footer
- sideBar
- main
- conts
- gNavi

■ CLASS名サンプル

- section
- inner
- sideMenu
- newsBox
- photoBox
- bannerList
- mailForm

CSS制作 3

■ コードフォーマット

- 1行につき1プロパティを記述する
- それぞれのプロパティをタブインデントする
- プロパティ名の後、半角コロン「:」をスペース無しに入力し、半角コロンと値の間には半角スペースを1つ空ける
- }の前後はスペースやタブを設けない

● 記述方法

- 単一のセレクトタを指定する場合

```
.selector {  
    property: value;  
    property: value;  
    property: value;  
}
```

- 複数のセレクトタを指定する場合

```
.selector01,  
.selector02,  
.selector03 {  
    property: value;  
    property: value;  
    property: value;  
}
```

CSS制作 4

■ コメントの記述

● 記述方法

```
/*-----  
    [要素名、大カテゴリ名]  
-----*/  
/*---- クラス名、小カテゴリ名 ----*/
```

● 記述例

```
/*-----  
products  
-----*/  
/*--- /products/ ---*/  
#products h5 {  

```

■ ショートハンドプロパティの使用

一括指定できるプロパティはまとめて記述する

● 悪い例

```
margin-top: 10px;  
margin-right: 15px;  
margin-bottom: 20px;  
margin-left: 15px;
```

● 良い例

```
margin: 10px 15px 20px;
```