

## デジタルキッチン/HTML コーディングガイドライン

### 目的

有限会社デジタルキッチンが提供する HTML コーディングの品質の安定化と向上の為にこのガイドラインを定めます。

### ベースコーディング

デジタルキッチンの HTML コーディングは、HTML5 によって策定された仕様に従って記述されますが、可読性、既存 HTML との互換性を考慮し、下記のように定めます。

### HTML バージョン

特別な指示がない限りは HTML5 を使用します。

### 文字コード

特別な指示がない限りは UTF-8 を使用します。また、ファイルは UTF-8 (BOM 無し) で保存します。

### 機種依存文字

基本的に文字実体参照及び数値文字参照に変換して記述します。  
但し、前述の文字参照で表示できない文字については、代替文字で書換えいたします。

### 属性とタグの記述

すべての HTML タグと属性は英小文字で記述します。但し、「title」属性や「alt」属性などのウェブサイトの閲覧者に視認されるデータ等は、適切な文字で記述いたします。

### クォート

HTML タグでのすべての属性値は、「" (ダブルクォーテーション)」で囲まれます。  
但し、JavaScript や PHP など自動生成される場合に例外的に「' (シングルクォーテーション)」で囲む場合があります。

### インデント

HTML タグはすべて論理的な構造に沿って、タブによるインデントで記述します。

### 検証

HTML ページは「W3C Validator」を使用してマークアップが整形形式であることを検証します。

## 各 HTML の詳細仕様

デジタルキッチンでは、各 HTML タグにも詳細な記述仕様がありますが、別紙 HTML コーディングルールを参照してください。

## CSS

デジタルキッチンでは CSS を下記のルールに従って設定します。

- CSS ファイルは共通、個別に分類したファイルを複数設置します。
- CSS ファイルはルートディレクトリ直下に `css` ディレクトリを設置し、以下サイト構成と同様のディレクトリ構成でファイルを設置します。
- CSS ファイル名は下記のルールで記述します。
  - サイト全体の共通スタイル → `common.css`
  - トップページスタイル → `toppage.css`
  - 各ディレクトリ共通スタイル → `cat-xxxx.css`  
※「xxxx」はディレクトリ名、ディレクトリ名はすべて英小文字
  - 個別ページのスタイル → `in-xxxx-yyyy.css`  
※「xxxx」はディレクトリ名、「yyyy」はページファイル名  
ディレクトリ名、ファイル名はすべて英小文字
- CSS の ID・Class は下記のルールで記述します
  - ID は共通スタイルの CSS には記述しない。
  - ID 名は表示ページのコンテンツ名の英訳を使用し、英小文字で記述します。
  - Class 名はコンテンツの機能・役割名を英訳して使用し、英数小文字で記述します。
- CSS ショートハンドは共通スタイルで使用します。
- CSS の記述は、
- ベンダープレフィックスは下記の順番で記述します。
  - `-webkit-`
  - `-moz-`
  - ベンダープレフィックス無しのプロパティ

※HTML5、CSS3 実装に問題の多い ie8 以前のブラウザは別途ご相談ください。

- `font-family` は次の設定を行います。
  - 「遊ゴシック/明朝」
  - 「ヒラギノ角ゴ/明朝」
  - 「メイリオ」
  - 「MS P ゴシック/明朝」
  - 「sun-serif/serif」

※但し、Web フォントを使用する場合やサーバー上にフォントファイルを設置する場合はその旨、ご指示ください。

## JavaScript

当社では下記の仕様で JavaScript を記述します。

- 基本的に JavaScript のコード自体は外部の .js ファイルに記述されます。
- .js ファイルはルートディレクトリ直下の「js」ディレクトリに設置します。
- HTML への埋め込みは<script>タグで行い、</body>タグの直前に記述されますが、動作上の理由で<head></head>内に記述する場合があります。同様に動作上の理由で、HTML へ直接 JavaScript コードを記述する場合がありますが、直接の記述はごく特別な対応となります。
- すべてのコードは何の動作をしているのかコメントによる説明を記述します。
- JavaScript ライブラリのライブラリファイルは、軽量化されたものを使用します。
- JavaScript コードは 80 文字を目安に改行して記述します。
- 「; (セミコロン)」の直後に改行し、コードの途中の場合はタブによるインデントで構造が分かり易いように記述します。
- if、else、for、while、try のブロックは、「{ (波かっこ)」を使用し、複数行で記述します。「{ (波かっこ)」内のコードはタブによるインデントを行います。
- 基本的に文字列は「' (シングルクォーテーション)」で囲みますが、動作上の必要がある場合は、「" (ダブルクォーテーション)」で囲む場合もあります。
- var での変数宣言はグローバル変数はファイルの最初に記述し、ブロック内変数はブロックの最初にコンマ区切りで記述します。
- 変数名と関数名は省略しない完全な単語を使用し、最初の文字を小文字にした camelCase を使用します。
- 厳格な等価性比較 (===) を使用します。
- switch 文の使用は他の方法での代替が困難な場合にのみ実装します。

2016.10.11