

# マークアップの基本

HTML/XHTML文書では「<要素名>内容</要素名>」のように、内容を「タグ」と呼ばれる目印(マークアップ記号)で囲みます。このことを「タグ付け」または「マークアップ」といいます。

## ▶▶ HTML/XHTMLの構文

### ④ 要素 (Elements)

「開始タグ」と「終了タグ」で囲む範囲全体を「要素」と呼びます。

たとえば、段落を表す「p要素」は、開始タグ<p>と終了タグ</p>で囲んだ範囲です。



「<」、「>」、「/」(スラッシュ)および要素名は、すべて半角の小文字で記述

### ④ 属性 (Attribute)

あるテキストにリンクを設定する場合には、次のように<要素名 属性名="属性値">の形式で、開始タグ内に記述します。

```
<a href="http://www.adobe.co.jp">Adobe</a>
```

## ▶▶ HTML文書の基本構造

HTML/XHTML文書は、基本的にhtml要素の中にhead要素とbody要素で構成されています。

### ④ html要素

全体を挟むように使用し、その文書がHTML/XHTMLであることを示します。

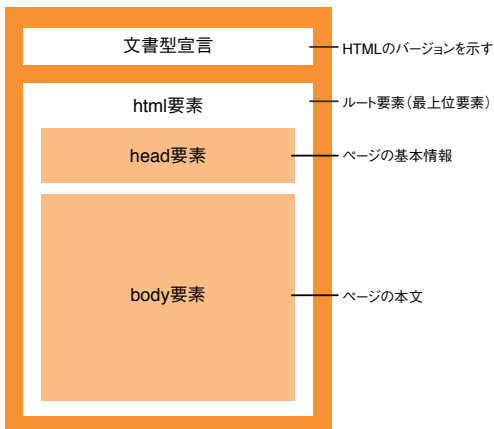
### ④ head要素

その文書のタイトル (title要素) やメタデータ (meta要素) などの基本情報を指定します。

head要素内に記述された内容は基本的にブラウザ画面には表示されません。

### ④ body要素

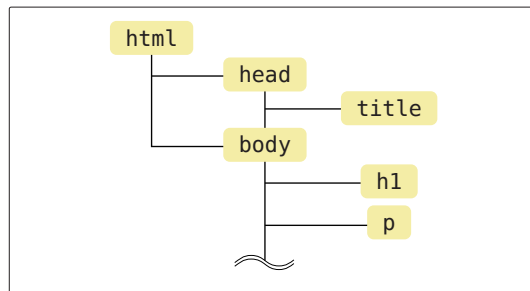
その文書をブラウザで表示したときに、見出しを指定するh1要素や段落を指定するp要素など、実際に表示される情報を記述します。



HTML

## 🔴 ツリー構造

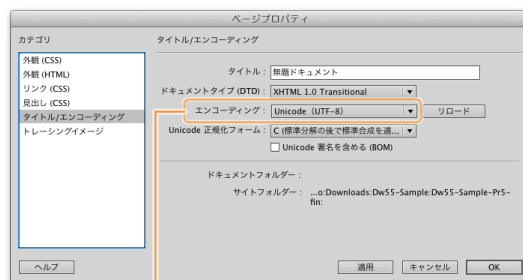
HTML 文書では、html要素を出発点（ルート）として、それぞれの要素の内部にさらに別の要素が含まれるという構造になっています。外側の要素を「親要素」、その内側に含まれる要素を「子要素」と呼びます。その構造は「ツリー構造」と呼ばれ、右図のように表現できます。



## ▶▶ 文字コード

HTML 文書では適切に文字コードを扱う必要があります。Dreamweaverでは、文字コードを確認・変更するには、次の手順で行います。

1. [修正]メニューの [ページプロパティ]をクリックして [ページプロパティ]ダイアログボックスを表示します。
2. [タイトル/エンコーディング]カテゴリの [エンコーディング]に現在の文字コードが表示されています。



```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

文書型宣言にHTML5を選択すると、文字コードは次のように記述される

```
<meta charset="UTF-8">
```

## 🔴 主な文字コード

Dreamweaverのデフォルトの文字コードは「UTF-8」です。併用するプログラムなどの兼ね合いで必要に応じて文字コードを変更してください。

| 文字コード名      | 表記          | 備考  |
|-------------|-------------|---|
| UTF-8       | UTF-8       | ひとつの文字コードで多言語を処理することが可能<br>UTF-16もあるが、UTF-8を利用するのが一般的 |
| シフトJIS      | Shift_JIS   | MS-DOSの時代から広く使われている文字コード<br>古いPC環境では一般的な日本語文字コード      |
| EUC         | EUC-JP      | UNIXの日本語環境で使われる文字コード                                  |
| ISO 2022-JP | ISO-2022-JP | 電子メールなどで使われる(俗に「JISコード」)                              |



XHTMLには次のような記述のルールがあります。HTMLでも基本的にこれに準じて記述します。

- 要素名と属性名は半角スペースで区切る
- 属性名と属性値は「=」でつなぐ
- 属性値は「"」（引用符）で囲む
- 属性名は必ず小文字で記述する
- 属性値は、状況に応じて大文字と小文字を使いわけのほか、日本語を使うことができる
- 1つの要素に複数の属性を設定するときは、半角スペースで区切る

## ▶▶ ドキュメントタイプ (文書型宣言/DOCTYPE 宣言)

### 📌 ドキュメントタイプとは

HTML/XHTML 文書では、その文書がどのバージョンの HTML/XHTML で作成されているのかを明示するための「ドキュメントタイプ」(Document Type Definition/DTD: 文書型宣言)を、html 要素の前に記述します。

文書型宣言は、公開識別子とシステム識別子で構成されています。「-/W3C//DTD HTML 4.01 Transitional//EN」が公開識別子、「"http://www.w3.org/TR/html4/loose.dtd"」がシステム識別子です。

### HTML 4.01 Transitional

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

### XHTML 1.0 Transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

### XHTML 1.0 Strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

### HTML5

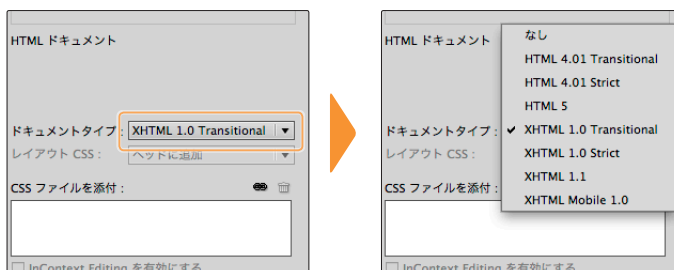
```
<!DOCTYPE html>
```

### 📌 主なドキュメント宣言

|                        |  |
|------------------------|--|
| HTML 4.01 Transitional | HTML 4.01 の仕様に沿ってコードを記述するという宣言。文書構造としての厳密さがやや低い形式。                           |
| HTML 4.01 Strict       | HTML 4.01 の仕様に沿ってコードを記述するという宣言。文書構造としての厳密さが高い形式。                             |
| XHTML 1.0 Transitional | HTML4.01 を XML 1.0 で再定義した、XHTML 1.0 の仕様に沿ってコードを記述するという宣言。文書構造としての厳密さがやや低い形式。 |
| XHTML 1.0 Strict       | HTML 4.01 を XML 1.0 で再定義した、XHTML 1.0 の仕様に沿ってコードを記述するという宣言。文書構造としての厳密さが高い形式。  |
| XHTML 1.1              | XHTML 1.0 をモジュール化した(小さな単位に分け、組み合わせて使えるようになった)仕様に沿って記述するという宣言。                |
| XHTML Mobile 1.0       | モバイル業界団体 OMA が定義した「XHTML Mobile Profile」仕様に沿って記述するという宣言。                     |
| HTML5                  | HTML5 の仕様に沿ってコードを記述するという宣言。ただし、HTML5 仕様はまだ策定途中                               |

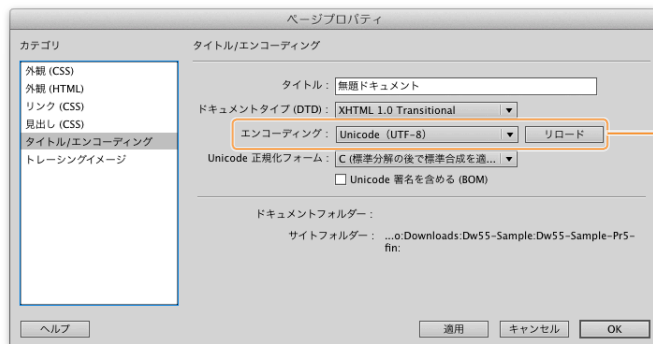
### 📌 Dreamweaver での文書型宣言の設定

Dreamweaver では、新規ドキュメントを作成する際に、[新規ドキュメント] ダイアログボックス内で [ドキュメントタイプ] (文書型宣言) を設定します。



## ④ドキュメントタイプの確認

[ページプロパティ]ダイアログボックス ([修正]メニューの [ページプロパティ]をクリックして表示)の [タイトル/エンコーディング]カテゴリの [ドキュメントタイプ (DTD)]でドキュメントタイプ (文書型宣言)を確認・変更することができます。文字コードの変更 ([エンコーディング])も、このダイアログボックスで行います。



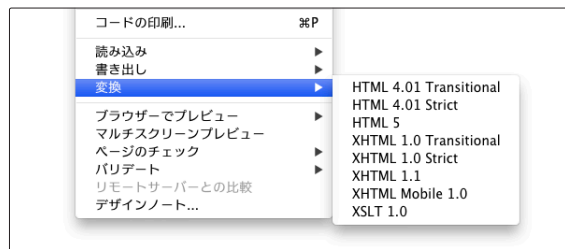
文字コードの  
確認・変更

## ④文書型宣言の変更

Dreamweaverでは、[ファイル]メニューの [変換]のサブメニューから変更したい文書型宣言を選択し、文書型宣言を変更することができます。

「HTML 4.01 Transitional」から「XHTML 1.0 Transitional」に変更すると、<br>は<br />のように変更されます。

これを利用して、HTMLとXHTMLの記述の揺れを正す (どちらかに統一する)ことができます。



[ファイル]メニューの [変換]のサブメニューから文書型宣言を選択して、文書型宣言を変更できる

新規に作成するドキュメントのデフォルトの文書型宣言を変更するには、環境設定の [新規ドキュメント]カテゴリの [初期設定ドキュメントタイプ (DTD)]を修正します。



XHTML文書では、文書型宣言の前に「<?xml version="1.0" encoding="UTF-8"?>」のようにXML宣言を記述することがあります。

XHTML文書でXML宣言を行うと、Internet Explorer 6での表示が、Internet Explorer 5/5.5 互換の表示になってしまいます。これによって、基本フォントサイズやボックスサイズの計測方法が異なるため、さまざまな調整が必要になってしまいます。

現在、多くの現場ではXML宣言を入れないことが一般的です (文字コードが UTF-8か UTF-16の場合)。

## ▶▶ ブロックレベル要素

見出しや段落など、文書を構成する基本要素は、ブロックレベル要素としてタグ付けします。ブロックレベル要素は、「ブロック」(通常改行を伴う表示上のまとまり)としてブラウザー上で扱われます。

### 🔗 ブロックレベル要素とは

段落を表す p 要素や、大見出しを表す h1 要素などはブロックレベル要素です。

ブラウザー上では、ブロックレベル要素ごとに改行され、前後に空白が取られて表示されるのが一般的です。

### 🔗 主なブロックレベル要素

<body> ~ </body> の範囲内にある内容はすべて、ブロックレベル要素内にマークアップする必要があります。

DOCTYPE が Strict の場合、使用できるブロックレベル要素は右表の 13 種類です。

上記の要素のほかに、文書内の追加した箇所を示す ins 要素、削除した箇所を示す del 要素は、ブロックレベル要素のほか、インライン要素にもなる特殊な要素です。

そのほかには、JavaScript 関連の script 要素、nonscript 要素があります。

|      |               |              |
|------|---------------|--------------|
| テキスト | h1 ~ h6 要素    | 見出し          |
|      | p 要素          | 段落           |
|      | div 要素        | 任意の範囲、グループ化  |
|      | blockquote 要素 | 引用           |
|      | address 要素    | 連絡先          |
|      | pre 要素        | 整形済みテキスト     |
| リスト  | ul 要素         | リスト(箇条書き)    |
|      | ol 要素         | 番号リスト        |
|      | dl 要素         | 定義リスト        |
| その他  | table 要素      | 表            |
|      | form 要素       | フォーム         |
|      | fieldset 要素   | フォーム内容のグループ化 |
|      | hr 要素         | 横罫線          |

主なブロックレベル要素

## ▶▶ インライン要素

### 🔗 インライン要素とは

インライン要素は、ブロックレベル要素内で特別な役割や機能をもたせる要素と考えるとよいでしょう。

### 🔗 空要素とは (Empty Elements)

img 要素や br 要素、hr 要素のように、内容をもたない要素を「空要素」(からようそ)と呼びます。

XHTML 文書では終了タグを用いず、<br /> のようにスペースと「/」(スラッシュ)を利用して空要素(タグ)を表現します。

|      |                           |                |
|------|---------------------------|----------------|
| 強調   | em 要素                     | 強調             |
|      | strong 要素                 | より強い強調         |
| その他  | a 要素                      | リンク箇所          |
|      | br 要素                     | 改行(ブロックレベル要素内) |
|      | img 要素                    | 画像             |
|      | span 要素                   | 汎用的に使用         |
|      | object 要素                 | オブジェクトの埋め込み    |
| フォーム | input                     | 入力部品           |
|      | textarea                  | テキストエリア        |
|      | select/option<br>optgroup | プルダウンメニュー      |
|      | button                    | ボタン            |
|      | label                     | ラベル            |

主なインライン要素

## Dreamweaverでのマークアップ

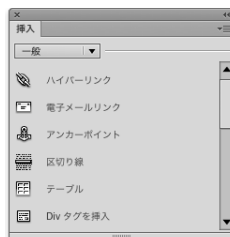
Dreamweaverには、さまざまなマークアップ方法があります。

作業の内容や流れに応じて、適切なマークアップ方法を選択できるようにしましょう。

## ④ [挿入] パネル

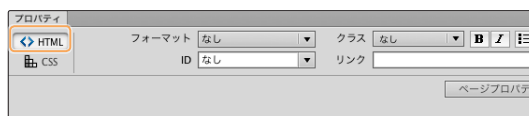
[挿入] パネルには、ブロックレベル要素やインライン要素がアイコンとして用意されています。

[テキスト] カテゴリの見出しや段落、リストなどをマークアップするとき、デザインビューとコードビューでは挙動が異なります（コードビューでは、対象となる文字列を正確に選択しておく必要があります）。



## ④ [プロパティ] パネル

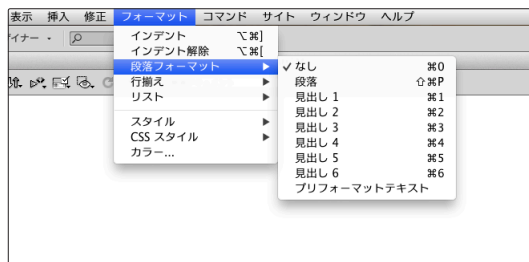
見出しやリスト、リンクなどを設定したり、要素を設定後、id/classなどの属性を設定するのに向いています。



[プロパティ] パネルでマークアップを行うときは [HTML] モードを使う

## ④ メニュー

マークアップ作業の多くは、メニューから設定できます。よく使う操作はキーボードショートカットを覚えましょう。



設定されているキーボードショートカットはメニュー項目内に表示される

## ④ キーボードショートカット

すべての操作にキーボードショートカットが用意されているわけではありません。必要に応じてカスタマイズしましょう。

## ④ クイックタグ編集

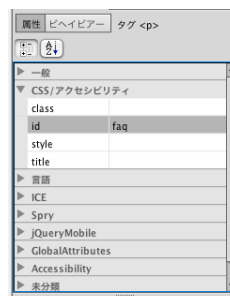
デザインビュー上で、編集箇所のソースコード編集を行う際に重宝します。文字通り「クイック」に編集できるので、ぜひ身につけたい機能です。



クイックタグ編集は、デザインビューで視覚的に作業を行いながら、ソースコードを編集できる便利機能

## ④ [タグインスペクター] パネル

要素の属性を設定／編集するのに便利なパネルです。



要素の属性を設定／編集するには [タグインスペクター] パネルが便利

## ④ コードヒント

コードビューで編集中に表示され、ソースコード入力をアシストしてくれます。

## ブロックレベル要素のマークアップ

それぞれの要素をマークアップするには、[挿入]パネル、[プロパティ]パネル、メニュー、キーボードショートカットで行う方法があります。タグによって方法が異なります。

## ▶▶ 段落のマークアップ

文章のひとかたまりは、p要素 (paragraph) でマークアップします。

段落内で強制的に改行を行うには、br要素を利用します (br要素を挿入するには `shift` + `return` / `Shift` + `Enter` を押します)。

```
<p>迷ったときにはp要素でマークアップしましょう。</p>
```

## ④ [プロパティ]パネルから

対象となる段落内にカーソルをおき、[プロパティ]パネルの [フォーマット] のプルダウンメニューから「段落」をクリックします。



## ④ メニューから

対象となる段落内にカーソルをおき、[フォーマット]メニューの [段落フォーマット] → [段落] をクリックします。

## ④ キーボードショートカットから

`⌘` + `shift` + `P` (`Ctrl` + `Shift` + `P`) を利用します。

## ④ キーボード

デザインビューで `return` (`Enter`) キーを押すと新しいp要素が作成されます。

`shift` + `return` (`Shift` + `Enter`) を押すと、br要素 (段落内改行) が挿入されます。

テキストエディターで空白行をあけておき、文字列をコピー＆ペーストでDreamweaverのデザインビューに持ち込むと、それぞれのブロックはp要素でマークアップされます (空白行がない場合には、br要素が入ります)。

- p要素 (段落) になっていれば、対象となる文字列をすべて選択せず、段落内にカーソルをおくだけで見出し、リストなどに変更できる
- Wordから、Dreamweaverにテキストをコピー＆ペーストで持ち込む際、見出しは見出し、箇条書きは箇条書きとしてマークアップされる

これらを利用すると、マークアップ作業の効率化につながります。

## ▶▶ 見出しのマークアップ

見出し要素 (heading) には、h1 から h6 の 6 つのレベルがあります。

h1 要素は「大見出し」、h2 要素は「中見出し」、h3 要素は「小見出し」として使い、h4～h6 は必要に応じてそれ以下の見出しに利用します。

### 🕒 [プロパティ] パネルから

対象となる段落内をクリックし、[プロパティ] パネルの [フォーマット] のプルダウンメニューから [見出し 1] ([見出し 2] [見出し 3] ……) を選択します。

```
<h1>CSS Niteについて</h1>
```

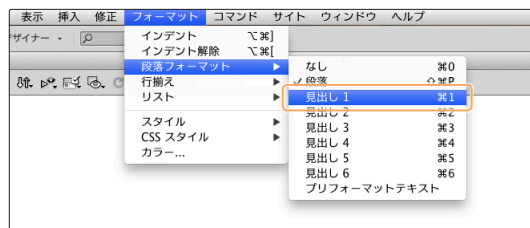
見出し要素を設定すると、多くのブラウザでは、文字サイズが大きくなり、太字で表示される



段落が設定されている場合には、対象となる文字列をすべて選択せず、段落内にカーソルをおくだけで適用することができる

### 🕒 メニューから

対象となる段落内をクリックして、[フォーマット] メニューの [段落フォーマット] → [見出し 1] (または [見出し 2] [見出し 3] …… [見出し 6]) をクリックします。



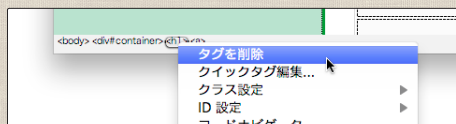
### 🕒 キーボードショートカットから

見出し 1 は **⌘ + 1** (**Ctrl + 1**)、見出し 2 は **⌘ + 2** (**Ctrl + 2**) を利用します。

見出し 3 以降、同様に、見出し 6 まで用意されています。

| マークアップ | Mac          | Windows         |
|--------|--------------|-----------------|
| 見出し 1  | <b>⌘ + 1</b> | <b>Ctrl + 1</b> |
| 見出し 2  | <b>⌘ + 2</b> | <b>Ctrl + 2</b> |
| 見出し 3  | <b>⌘ + 3</b> | <b>Ctrl + 3</b> |
| 見出し 4  | <b>⌘ + 4</b> | <b>Ctrl + 4</b> |
| 見出し 5  | <b>⌘ + 5</b> | <b>Ctrl + 5</b> |
| 見出し 6  | <b>⌘ + 6</b> | <b>Ctrl + 6</b> |

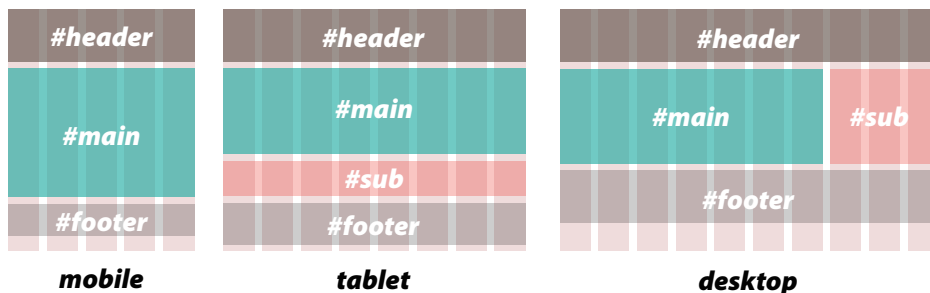
マークアップした要素を削除したい場合には、ドキュメントウィンドウ下部のタグセレクターで右クリックして、[タグを削除] をクリックします。



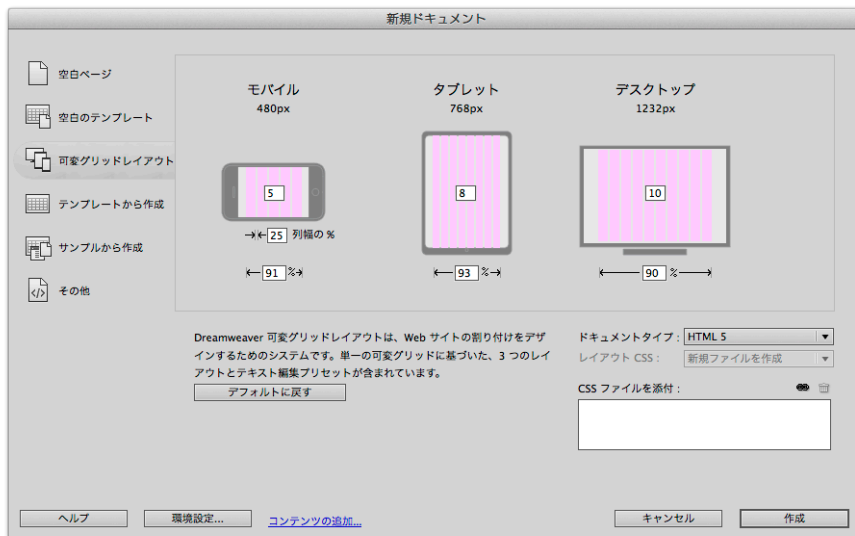


# 可変グリッドレイアウトによるレスポンスWebデザイン

「モバイル」(iPhoneなどのスマートフォン)、「タブレット」(iPadなどのタブレット端末)、「デスクトップ」(パソコン)の3つのデバイスごとにレイアウトを切り分ける「レスポンスWebデザイン」を実装します。



同じHTMLを使い、デバイスの幅(ウィンドウサイズ)に応じて、CSSで異なるレイアウトを与えます。この作業を効率化するために、Dreamweaver CS6から搭載された「可変グリッドレイアウト」を使います。



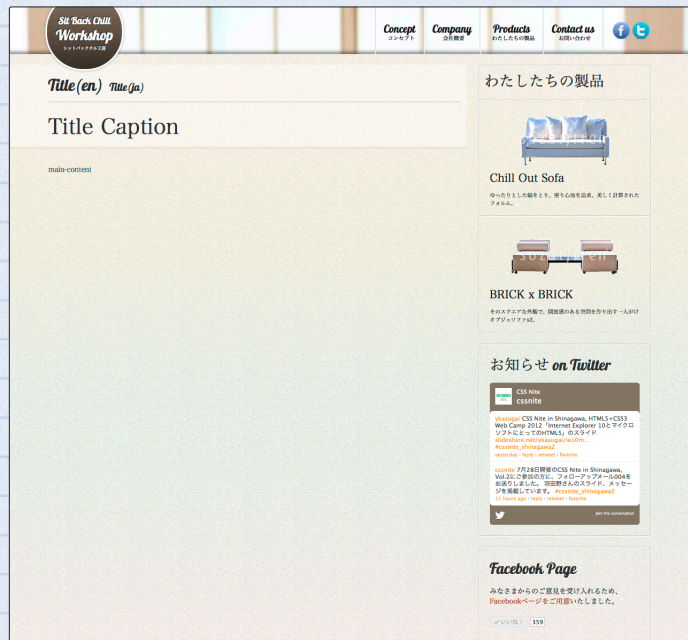
サイトとしてページを作る前に、全ページの共通部品を入れたページを作成し、後工程でこれをDreamweaverテンプレートに変換していきます。

## サンプル

「モバイル」（iPhoneなどのスマートフォン）、「タブレット」（iPadなどのタブレット端末）、「デスクトップ」（パソコン）の3つのデバイスごとの完成見本です。

次の点でレイアウトが異なります。

- ヘッダー部分のグローバルナビゲーション
- ヘッダー部分のソーシャルメディアアイコン
- サイドバーの位置、表示/非表示
- フッターの地図、ナビゲーションの表示/非表示



## デスクトップ

2段組のレイアウトです。  
HTML内のコンテンツがすべて表示されている状態です。



## タブレット

デスクトップレイアウトのサイドバー領域がコンテンツの下（フッターの上）に位置し、全体が1段組になっています。  
また、フッターのナビゲーションが非表示になっています。

## スマートフォン

デスクトップレイアウトのサイドバー領域は非表示になっています。  
フッターの地図、ナビゲーションも非表示になっています。

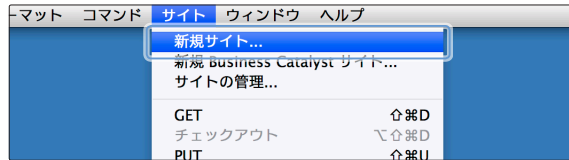
## STEP 1 サイト定義とクローク

Dreamweaverでサイト制作を行うには、公開するサイトと同じ構造の「ローカルサイト」を作成し、完成後にサーバーに転送(アップロード)するという流れです。

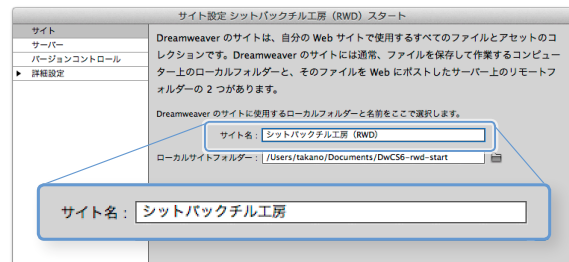
アップロードする必要のないファイルは、特定のフォルダーにとりまとめ、クローク設定を行います。

1 サンプルファイル「DwCS6-rwd-start」をダウンロードして、「書類」フォルダー(「マイドキュメント」フォルダー)におきます。

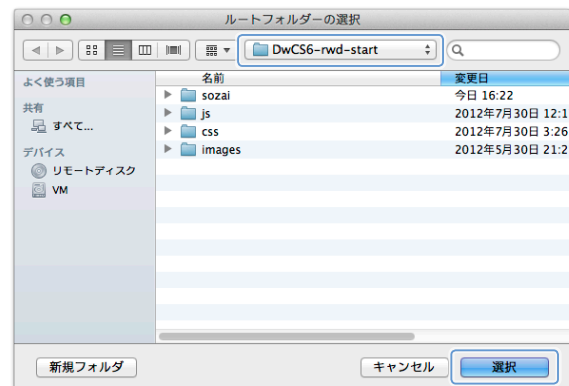
2 [サイト]メニューの[新規サイト]をクリックします。



3 [サイト設定]ダイアログボックスが開いたら、[サイト名]に「シットバックチル工房」を入力します。

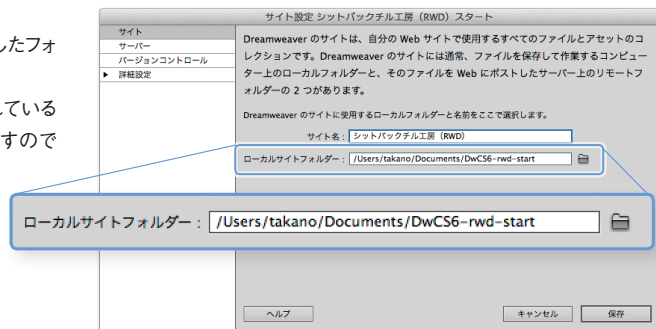


4 [ローカルサイトフォルダー]の📁をクリックして、「書類」フォルダー(「マイドキュメント」フォルダー)内の「DwCS6-rwd-start」フォルダーを指定します。

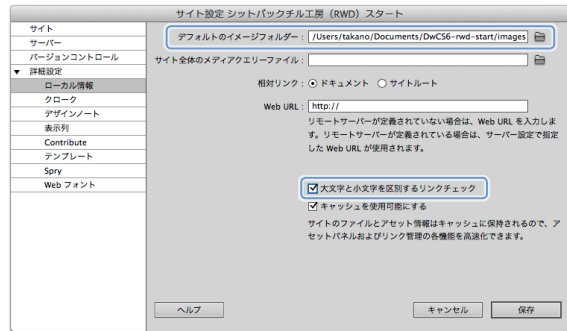


[ローカルサイトフォルダー]に、指定したフォルダーへのパスが表示されます。

パス(フォルダー名)に日本語が含まれていると、トラブルの原因になることがありますので注意してください。

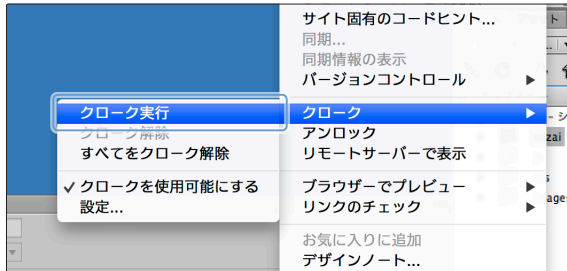


5 [詳細設定]を展開して、[ローカル情報]カテゴリを開き、[デフォルトのイメージフォルダー]にローカルサイト内の「images」フォルダーを指定します。  
また、[大文字と小文字を区別するリンクチェック]にチェックをつけます。

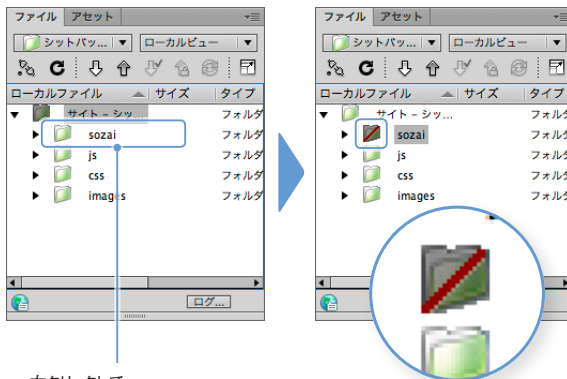


6 [保存]ボタンをクリックして、ダイアログボックスを閉じると、[ファイル]パネル内に、指定したローカルフォルダーが表示されます。

7 「sozai」フォルダーを右クリックして表示されるメニューから[クローク]→[クローク実行]をクリックします。



「sozai」フォルダーに赤い斜線が付きます。



右クリックして、  
[クローク]→[クローク実行]をクリック

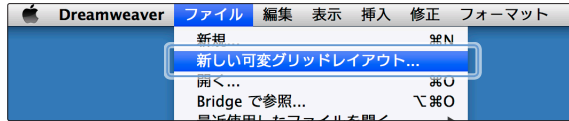
## 手を動かしながらチュートリアルを進める方へ

- 文字入力が必要なものは、すべて「sozai」フォルダー内に用意されています。
- HTMLやCSSの作業時に●行目のように指示がある場合、ソースフォーマットの適用が行われていることを前提としています([ソースフォーマットの適用]の手順は、基本的に省略しています)。

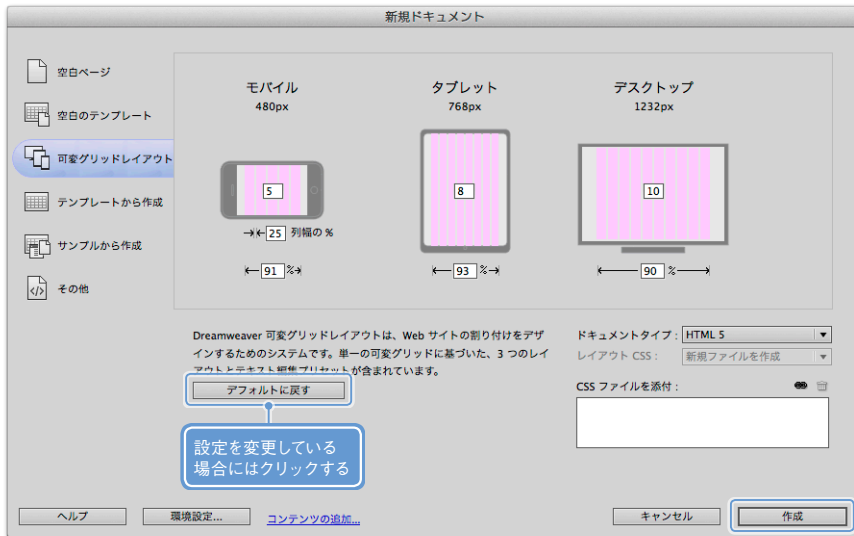
## STEP 2 可変グリッドレイアウトを使ったドキュメントの準備

レスポンシブWebデザイン対応のサイト制作を進めるために、Dreamweaver CS6から用意された「可変グリッドレイアウト」を使ってみます。[ファイル]メニューの[新規]をクリックして、[新規ドキュメント]ダイアログボックスで[可変グリッドレイアウト]をクリックして進めることもできます。

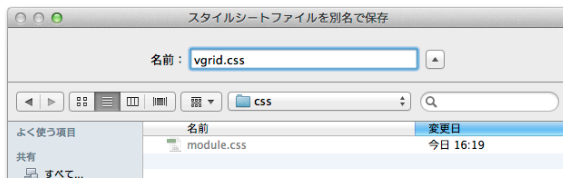
- 1 [ファイル]メニューの[新しい可変グリッドレイアウト]をクリックします。





- 2 [新規ドキュメント]ダイアログボックスが開きます。デフォルトの設定のまま、[作成]ボタンをクリックします。

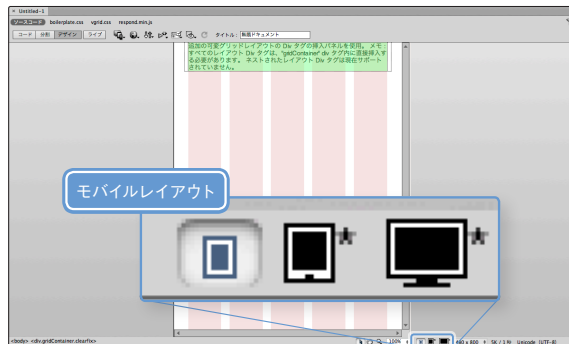


- 3 [スタイルシートファイルを別名で保存] ダイアログボックスが開くので、サイト内の「css」フォルダー内に『vgrid.css』というファイル名で保存します(ファイル名は『vgrid.css』でなくてもOKです)。

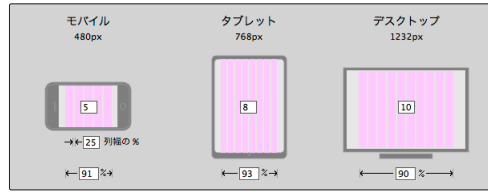


新しく作成したドキュメントが開きます。ドキュメントウインドウ下部の  ([モバイルのサイズ (480×800)] ボタン) が押された状態になっていて、薄いピンクのグリッドが5本表示されます。薄いピンクのグリッドは、Dreamweaverでの編集時のみ表示されます。

 ([モバイルのサイズ (480×800)] ボタン) が押された状態をモバイルレイアウトと呼びます。

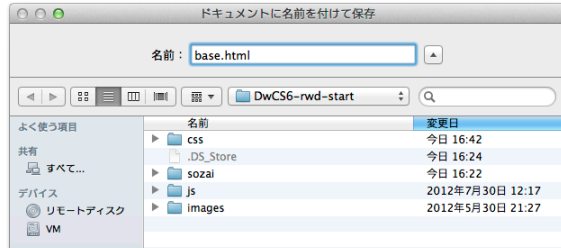


■ [タブレットのサイズ(768×1024)] ボタンをクリックすると8本 (タブレットレイアウト)、  
 ■ [デスクトップのサイズ(1000w)] ボタンをクリックすると10本のグリッドが表示されます (デスクトップレイアウト)。

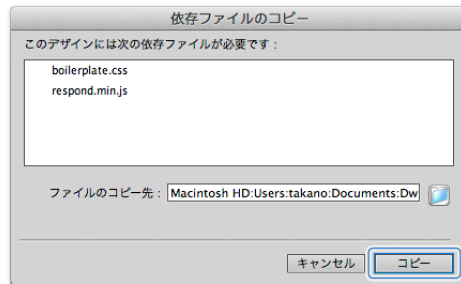


グリッドの数は、[新規ドキュメント]ダイアログボックスで決定している

4 [ファイル]メニューの[保存]をクリックして、ドキュメントをサイトルートに『base.html』のファイル名で保存します。

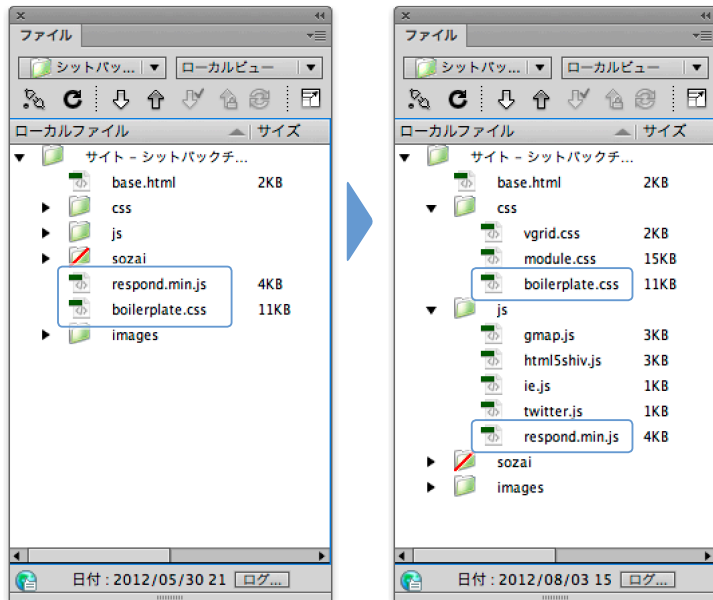


5 [依存ファイルのコピー] ダイアログボックスが開くので、[コピー] ボタンをクリックします。



6 [ファイル] パネルを確認すると、『boilerplate.css』と『respond.min.js』がローカルサイトのルートに保存されます。  
 『boilerplate.css』を「css」フォルダーに、『respond.min.js』を「js」フォルダーに移動します。

移動の際、[ファイルの更新]ダイアログボックスが表示されるので、[更新] ボタンをクリックします。



『boilerplate.css』は、リセットCSSの一種で基本要素に対する最低限のCSSを設定しています。  
 『respond.min.js』は、Internet Explorer 8以下の環境でメディアクエリーを有効にするためのJava Script ファイルです。