

目次

・ スキン概要

はじめに	01p
スキン制作キットでできること	01p
このスキン制作キットに含まれているもの	01p
まずはページの構造を理解しよう	02p
スキンの種類	02p
スキンの構造	03p
スキン構成ファイルについて	04p
段組コンテナについて	05p
ユニットについて	06p～10p

・ スキンカスタマイズ

スキンのカスタマイズにあたり・・・	11p
作業環境の構築	11p
MAMPの設定方法	12p
スキン構成ファイル詳細	13p
コーディングについて	14p
「スキン」 カスタマイズ	14p
「index.tp」 ソースI詳細	15p～16p
「グローバルメニュー」 カスタマイズ	17p
「段組コンテナ」 カスタマイズ	18p～19p
「ユニット」 カスタマイズ	20p～21p

はじめに

この「bingo!CMS スキン制作キット」は「bingo!CMS」のスキンを制作するために必要なマニュアルや参考ドキュメント、サンプルファイルをまとめたものです。

このキットを使用するには、Adobe Fireworks CS4 以上、Adobe Dreamweaver CS4 以上（スニペットを利用する場合）のソフトが必要となります。

このキットは、HTML と CSS の基礎知識がある人、上記ソフトの基本的な操作を行える人が対象となります。

※作成されたスキンは「bingo!CMS」のサポート対象外となります。

※制作に関する方法についての問い合わせには応じられません。

※Dreamweaver、Fireworks は、Adobe Systems Incorporated（アドビシステムズ社）の米国ならびに他の国における商標または登録商標です。

MacOS および Macintosh は、米国および他の国々で登録された AppleInc. の商標です。

スキン制作キットでできること

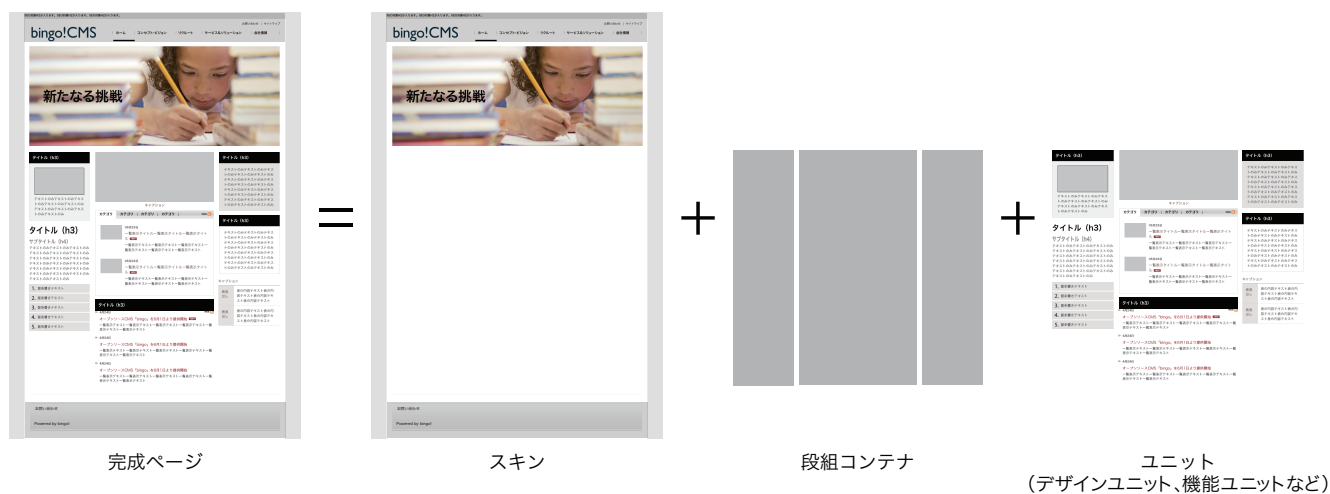
1. 「bingo!CMS」のスキンを理解しオリジナルデザインのスキンを作成することができます。
2. 段組コンテナやユニットをカスタマイズすることができます。

このスキン制作キットに含まれているもの

- スキン制作マニュアル（manual）
- カスタマイズ用サンプルスキンデザインファイル（sample-design）
トップページ用デザインと下層ページ用デザインの Adobe Fireworks（CS4 以上）のデザインファイル（png）です。
あらかじめサンプルデザインやスライス（書き出し名称設定済み）が配置してあります。
- カスタマイズ用サンプルスキンファイル（sample-skin）
カスタマイズ用スキンデザインファイルをもとにしたスキンです。
- タグリファレンス（tag-reference）
「bingo!CMS」専用タグの解説ドキュメントです。
- コンテンツサイズガイド（contents-size）
段組コンテナ内にユニットを配置した際のサイズ一覧表です。
- スニペット（snippet）
Adobe Dreamweaver（CS4 以上）用のスニペットとなります。
スニペットは Dreamweaver の機能の一つで「bingo!CMS」の専用タグを簡単に挿入することができます。
テンプレートソースを一から作成する場合に使用すると便利です。

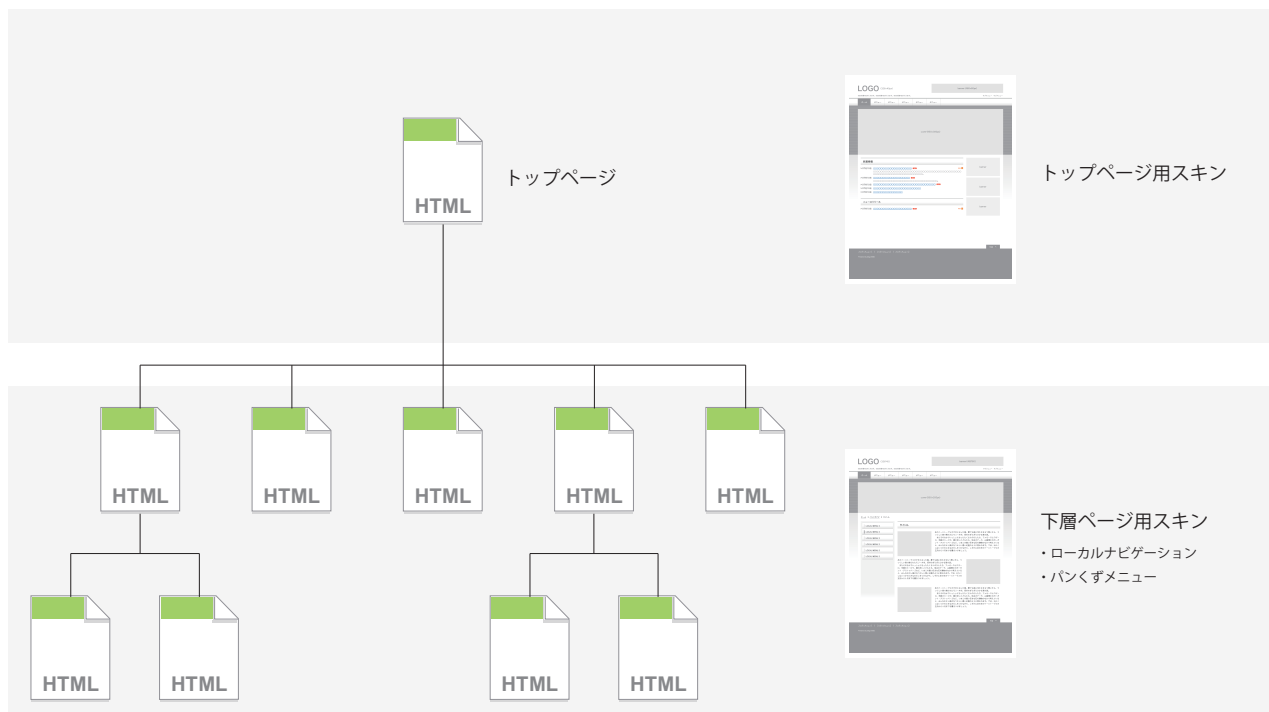
まずはページの構造を理解しよう

「bingo!CMS」では「スキン」+「段組コンテナ」+「ユニット（デザインユニット、機能ユニット）」という構造でページを形成しております。



スキンの種類

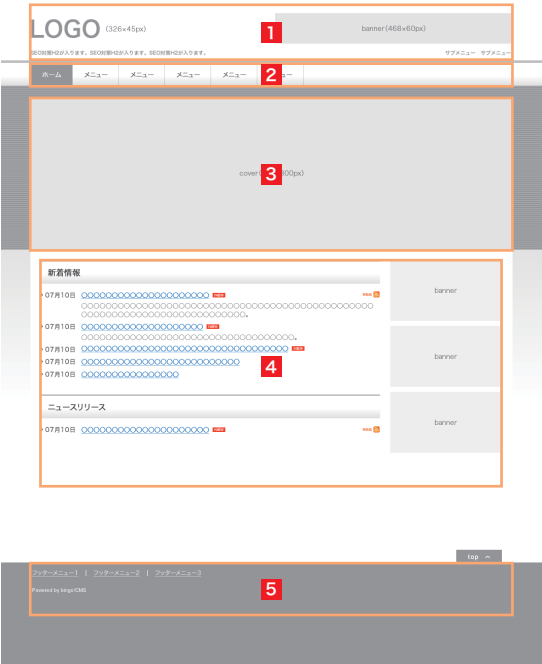
スキンは、カバー画像が大きくデザインされているトップページ用スキンと、ローカルナビゲーションやパンくずメニューが組み込まれた下層ページ用のスキンがあり、ほとんどのサイトでは二つのスキンを使用します。



スキンの構造

トップページ用スキンは大きく「ヘッダー」「グローバルナビゲーション」「カバーエリア」「ユーザーエリア」「フッター」という5つの構成となっています。下層ページ用スキンにはトップページ用スキン構成に「サイドバーエリア」が追加されます。レイアウトによっては、トップページ用スキンにも「サイドバーエリア」がある場合があります。

トップページ用スキンデザイン例（カスタマイズ用スキンデザイン）



- 1 ヘッダー**
ロゴ、SEO テキスト、バナー、サブナビゲーションは管理画面より登録いたします。各パーツは初期設定ファイルにより有無の設定が可能です。
- 2 グローバルメニュー**
管理画面より登録されたメニューが自動的に表示されます。表示しているページがどのメニューに属しているかがわかるよう自動的にアクティブ状態となります。
- 3 カバーエリア**
管理画面よりカバー画像の登録、カバー画像の高さの調整、カバー画像上に文字も流し込むことが可能です。
- 4 ユーザーエリア**
管理画面よりコンテンツ（コンテナ、ユニット）の配置を行うので、コーディングの必要はありません。
- 5 フッター**
フッターナビゲーション、コピーライトは管理画面より登録いたします。各パーツは初期設定ファイルにより有無の設定が可能です。

下層ページ用スキンデザイン例（カスタマイズ用スキンデザイン）



- 1 ヘッダー**
ロゴ、SEO テキスト、バナー、サブナビゲーションは管理画面より登録いたします。各パーツは初期設定ファイルにより有無の設定が可能です。
- 2 グローバルメニュー**
管理画面より登録されたメニューが自動的に表示されます。表示しているページがどのメニューに属しているかがわかるよう自動的にアクティブ状態となります。
- 3 カバーエリア**
管理画面よりカバー画像の登録、カバー画像の高さの調整、カバー画像上に文字も流し込むことが可能です。
- 4 バンクズメニュー**
自動的に生成されます。管理画面より表示の有無を設定可能です。
- 5 サイドバーエリア（一般的には下層ページ用スキンのみ）**
管理画面より登録されたローカルメニューが自動的に表示されます。また、ユニットを登録することができます。
- 6 ユーザーエリア**
管理画面よりコンテンツ（コンテナ、ユニット）の配置を行うので、コーディングの必要はありません。
- 7 フッター**
フッターナビゲーション、コピーライトは管理画面より登録いたします。各パーツは初期設定ファイルにより有無の設定が可能です。

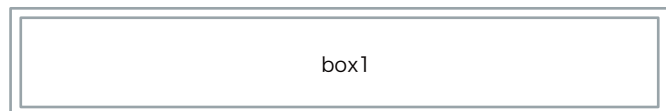
例) myskin-top スキンの場合



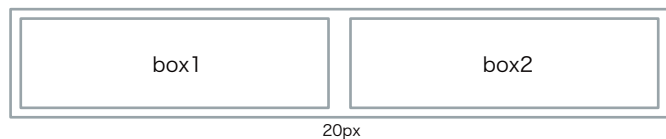
段組コンテナについて

段組コンテナ外枠のボックス幅は、どんなサイズของผู้ใช้エリアにも対応できるように、すべて 100% という設計になっています。その中に、カラムがネストされています。カラムの構造と css ファイル名のリストとなります。

コンテナユニット 1 列 (100%) cunit_1.css



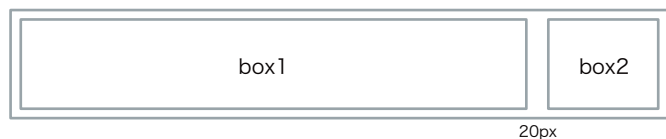
コンテナユニット 2 列 (50 : 50) cunit1_2.css



コンテナユニット 2 列 (23 : 77) cunit4_2.css



コンテナユニット 2 列 (77 : 23) cunit5_2.css



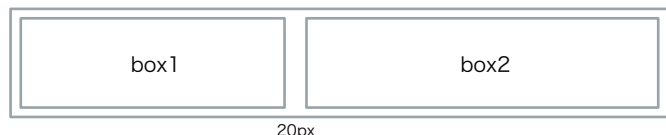
コンテナユニット 2 列 (33 : 67) cunit2_2.css



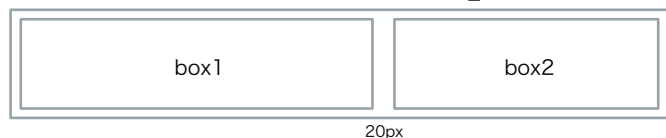
コンテナユニット 2 列 (67 : 33) cunit3_2.css



コンテナユニット 2 列 (38 : 62) cunit6_2.css



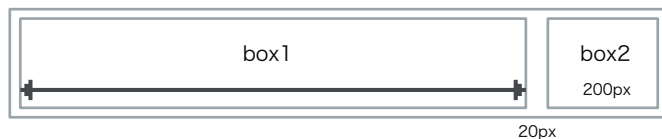
コンテナユニット 2 列 (62 : 38) cunit7_2.css



2 段組コンテナ (固定 | 可変) cunit8_2.css



2 段組コンテナ (可変 | 固定) cunit9_2.css



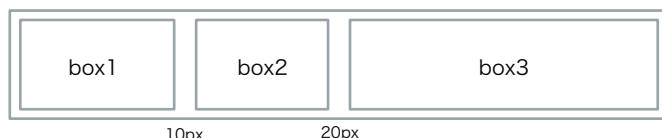
コンテナユニット 3 列 (33 : 34 : 33) cunit1_3.css



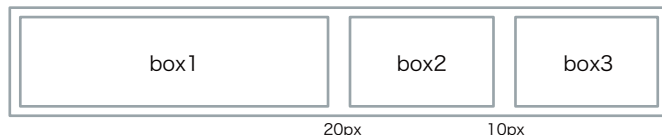
コンテナユニット 3 列 (25 : 50 : 25) cunit2_3.css



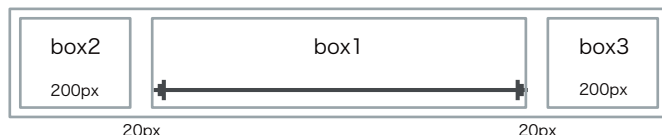
コンテナユニット 3 列 (25 : 25 : 50) cunit3_3.css



コンテナユニット 3 列 (50 : 25 : 25) cunit4_3.css



コンテナユニット 3 列 (固定 : 可変 : 固定) cunit5_3.css



コンテナユニット 4 列 (25 : 25 : 25 : 25) cunit_4.css



ユニットについて

「bingo!CMS」のユニットは段組コンテナのボックスサイズに応じて適切なバランスで表示されるようになっており、移動先のコンテナのサイズに応じて、表示サイズや配置が自動的に最適化されます。ユニット外枠のボックス幅は、どんなサイズのコンテナにも対応できるように、基本的には100%という設計になっています。

各コンテナに配置した際のユニットの画像サイズは、別紙のコンテンツサイズガイド contents-size-guide.pdf を参照してください。

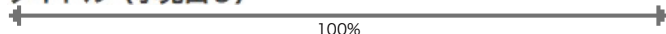
タイトル（見出し） bTtl.css

タイトル（見出し）

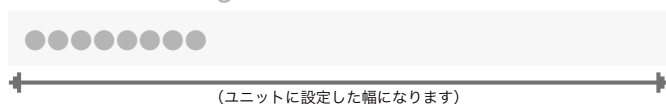


サブタイトル（小見出し） bSubTtl.css

タイトル（小見出し）



画像タイトル blmgttl.css

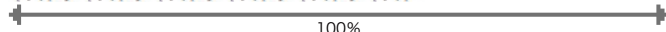


バックグラウンド付きタイトル bTtlBg.css



テキスト bTxt.css

テキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
 テキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
 テキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
 テキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
 テキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
 テキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
 テキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
 テキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト



画像のみ blmg.css



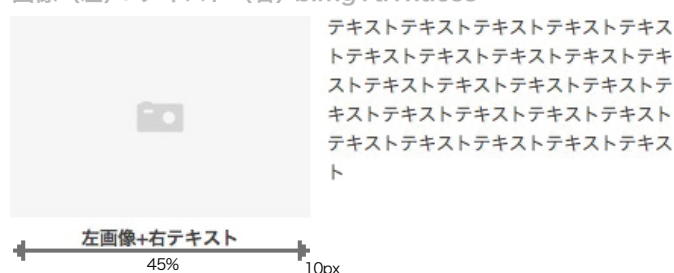
画像のみ（オリジナルサイズ表示） blmgOrg.css



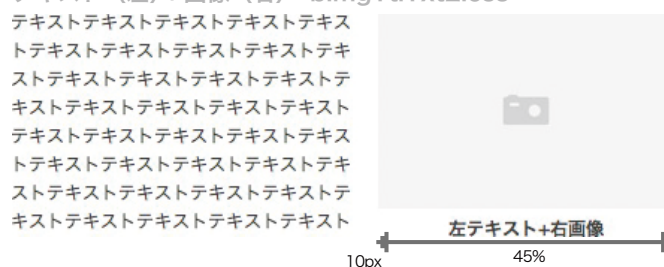
画像ボタン blmgBtn.css



画像（左）+テキスト（右） blmgTtlTxt.css



テキスト（左）+画像（右） blmgTtlTxt2.css



テキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
テキストテキストテキストテキストテキストテキストテキストテキストテキストテキスト
テキストテキストテキストテキストテキストテキスト

[illegible]

40%

20px

[illegible]

100%

[illegible]

50%

50%

0px 50%

> [詳細はこちら](#)

50% 10px

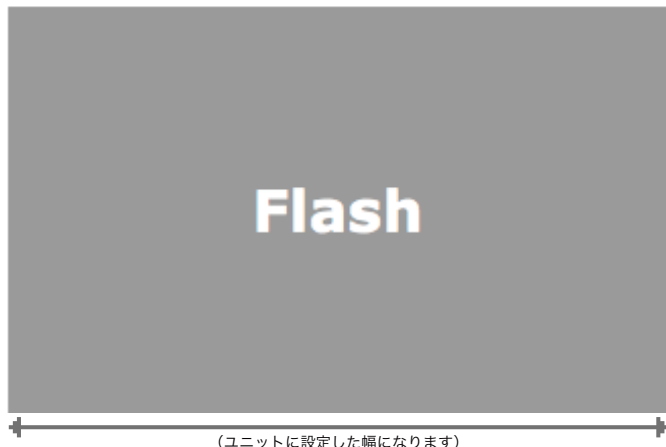
20px

50% 10px

10px

FLASH アニメーション bFlash.css

FLASHタイトル



FLASH プラグインダウンロード案内表示 bFlaDI.css



マップ bMap.css



サイトマップ bSiteMap.css

> [グローバルメニュー1](#)

- > [ローカルメニュー1](#)
- > [ローカルメニュー2](#)

> [グローバルメニュー2](#)

- > [ローカルメニュー3](#)
- > [ローカルメニュー4](#)

> [グローバルメニュー3](#)

- > [ローカルメニュー5](#)
- > [ローカルメニュー6](#)



RSS リーダー bRssReader.css

> 2011/02/10

[bingo!CMSスキンをオリジナルカラーに変えてみよう☆ 第2回](#)

第2回 下層ページ編はじめてです。こんにちは。去年シフトテック株式会社に入社した、新人の伊藤です。あと4ヶ月程で入社から一年立ちます。早いものです。先輩方が教えてくださるので、日々日進月歩をしています。早く皆さんの役にたてる様に頑張っています。さて、前回のブログではトップページのスキンの色の変更の方法について紹介しましたが、(第1回 トップページ編)今日は下層ページのスキンの色を変更していきます。使用するスキンは前回と同様のスキンの下層ページ「business1-sec」です..

> 2011/02/03

[bingo!CMSスキンをオリジナルカラーに変えてみよう☆ 第1回](#)

第1回 トップページ編お久しぶりです！山本（え）です。以前の記事で、企画のチームに異動になった山本（え）ですが、このたびデザインチームに帰ってきました！今後はまた、デザイナー向けTipsブログに度々登場いたしますので、Twitterともども、どうぞよろしくお願いします。(Twitterはこちら)さて、今日は、本日公開いたしましたVer.1.4対応の新スキン、「business01」を使用して、bingo!CMSスキンをオリジナルカラーに変更するにはどうしたらいいか、その方法に..



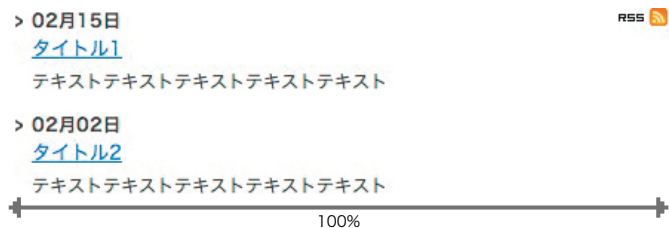
お問い合わせフォーム bContact.css

名前	<input type="text"/>
フリガナ	<input type="text"/>
会社名・団体名	<input type="text"/>
部署	<input type="text"/>
役職	<input type="text"/>
郵便番号	<input type="text"/>
都道府県	<input type="text" value="--選択してください--"/>
市・区・郡	<input type="text"/>
町名・番地	<input type="text"/>
電話番号	<input type="text"/>
メールアドレス	<input type="text"/>
メールアドレス確認	<input type="text"/>
お問い合わせ内容	<input type="text"/>
画像認証	<div>6B1UR</div> <input type="text"/> ※画像の文字を入力してください

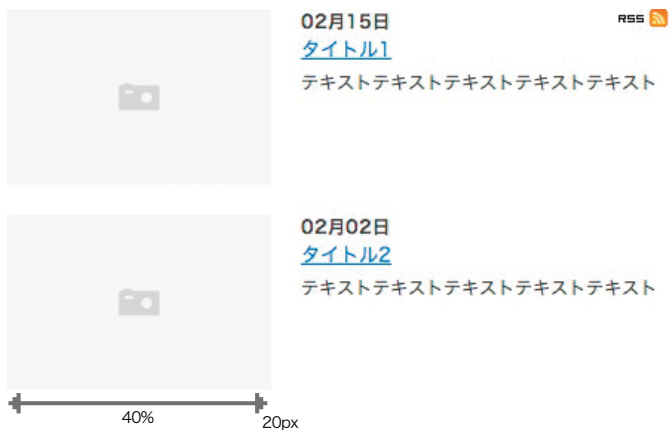
確認画面へ

Dimension line indicating the width of the contact form section is 100%.

連載記事リスト bCatListc.css



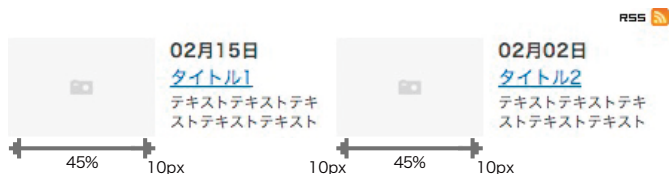
連載記事リスト（画像タイトルテキスト） bCatList2.css



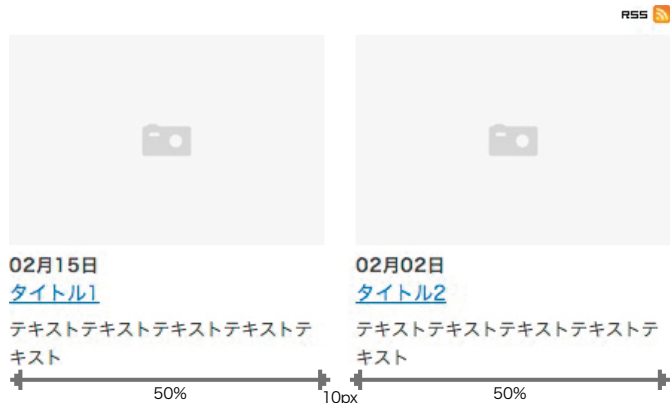
連載記事リスト縦（画像タイトルテキスト） bVCatList2.css



連載記事リスト（画像タイトルテキスト ×2） bCatList3.css



連載記事リスト縦（画像タイトルテキスト×2） bVCatList3.css



連載記事リスト縦（タブ表示） bCatList4.css



スキンのカスタマイズにあたり・・・

この資料では、スキン制作キットに含まれるサンプルスキン「myskin-top」「myskin-sec」をもとにすることを前提に、カスタマイズ方法を説明してあります。ここでの説明は方法の 1 部であり、HTML・CSS 知識の豊富な方ならばコーディング次第で、もっと自由で色々なデザインが可能です。

またスキン制作キットに含まれる、bingo!CMS 専用タグスニペット集 (bingo!CMS.mxp) などを使用して、1 から完全なオリジナルスキン作成にチャレンジしてみてください。

作業環境の構築

はじめに作業環境を構築します。

サーバーに「bingo!CMS」をインストールし、スキンを管理画面からインストールできる状態にします。

必須ではありませんが、bingo!CMS のスキンは構造上、ローカル環境ではデザインの確認がとりにくい為おすすめできません。

下記の方法で作業環境を構築していただくことで、すばやくかつ正確な作業が行えます。

1. スキンフォルダ名の変更

「myskin-top」または「myskin-sec」のフォルダ名を変更します。

スキンの用途がわかりやすい名前を半角英数でつけてください。

2. スキンの初期設定ファイルの変更

スキン内の config.inc （初期設定ファイル）を開きます。

デザインを確認しながら、各項目の初期設定値を記述します。

※記述方法の詳細は、別紙のタグリファレンス (tagreference.pdf) の 31p～38p を参照ください。

3. スキンのインストール

初期設定の変更が完了しましたら、スキンフォルダー一式を「zip」形式に圧縮して管理画面よりインストールします。

インストールされたスキンは bingo!CMS パッケージ内の「/usr-data/template/」に保存されます。

スキンのフォルダ名には「myskin-top_8」の用に番号が振られます。

そのため、同じ名前のスキンをインストールしても上書されることはありません。

※ 圧縮の際の注意点

- ・圧縮形式は「zip」になります。
- ・圧縮されたフォルダ内に通常無い不可視ファイル等があると正常にインストールできません。
- ・圧縮前と圧縮後のフォルダ名が違っていると正常にインストールできません。

4. Dreamweaver の設定

Dreamweaver を起動し、「サイトの管理」でスキンの保存先「/usr-data/template/ スキン名 /」をローカルルートフォルダに設定してサイトを作成します。

また、サーバーへの FTP 接続可能でしたら、リモート情報からサーバー情報登録して Dreamweaver 上でアップロードできるようにします。これで作業環境が整いました。

Mac をお使いのかたは・・・

Mac をお使いの方は「MAMP」というアプリケーションを使用することにより、お使いの PC に「bingo!CMS」インストールができ、ローカル環境でスキンのコーディングが可能になります。

MAMP については 12p 「MAMP の設定」をご参照ください。

MAMP の設定方法（インストールと MySQL データベースの準備）

1.MAMP のダウンロードとインストール

MAMP のホームページ（右図）

（<http://www.mamp.info/en/index.php>）にアクセスし、無料ダウンロード版（約 162MB）をダウンロードします。ダウンロードしたファイルを解凍し、ディスクイメージをマウントして、MAMP フォルダをアプリケーションフォルダにコピーします。

2. MAMP の起動

コピーした MAMP フォルダの中の「mamp.app」をダブルクリックして起動させます。

※Mac のシステム環境設定 / 共有・Web 共有は「切」にしてください。

右図のようなウィンドウが現れ、Apache と MySQL サーバが起動し、状態の表示が緑色になります。

ウィンドウの「環境設定」をクリックします。

3. ポートの設定

Apache のポートを「80」

MySQL のポートを「8889」に設定します。

4.PHP バージョンの設定

PHP5.2.13 にチェックを入れます。

5.Apache の設定

Document Root を指定します。参照ボタンをクリックし、インストール先の bingo!CMS パッケージフォルダを選択します。

6. データベースの準備

MAMP の Apache と MySQL サーバが起動すると、ブラウザが自動起動されて MAMP のスタートページ画面が表示されます。

MAMP スタートページ画面のグローバルメニューから「phpMyAdmin」をクリックし、phpMyAdmin ページを開きます。

7. 新規データベースの作成

「新規データベースを作成する」の欄にデータベース名（ここでは bingo_db）を入力し、その右の照合順序のポップアップから「utf8_unicode_ci」を選択します。入力した値を確認したら「作成」ボタンをクリックします。テーブルは bingo!CMS のインストーラが作成するので、準備はここまです。



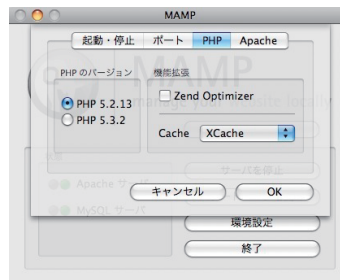
MAMP ダウンロードページ



2. MAMP の設定



3. ポートの設定



4. PHP バージョンの設定



5. Apache の設定



7. 新規データベースの作成

スキン構成ファイル詳細

【 index.tpl 】

スキンレイアウトの実態となる、HTML ファイルです。

【 config.inc 】

スキンの初期設定ファイルです。

※詳しい記述方法は、別紙のタグリファレンス (tagreference.pdf) の 31p～38p を参照ください。

【 layout.gif 】

スキンのレイアウトを表す画像です。

管理画面のスキニー覧等にサムネイルとして使用されます。

【 thum.jpg 】

スキンのイメージを表す画像です。

管理画面のスキニー覧等にサムネイルとして使用されます。

【 common-img / 以下ファイル 】

スキンで使用している画像ファイルです。

【 imported.css 】

スキン内の CSS を統括するファイルです。

index.tpl にはこの CSS ファイルが読み込まれます。

【 common.css 】

スキンで共通して使用する内容が記述してある CSS ファイルです。

【 color.css 】

カラー・イメージに関する内容が記述してある CSS ファイルです。

【 coverarea.css 】

ヘッダーエリア・カバーエリアのレイアウトに関する内容が記述してある CSS ファイルです。

あえて「layout.css」とファイルが分けてある理由としては、このファイルのみ公開表示側だけでなく、管理画面での

「ヘッダー編集」「カバー編集」の編集画面にも読み込まれます。なので「coverarea.css」に記述した内容は公開表示側だけでなく、編集画面上での、ヘッダー幅・ロゴ・SEO 文章・サブナビゲーション等にも影響します。

また、下部に記述してある「FONT (フォント設定)」の CSS セレクタの値は、編集画面での「スタイル設定」に関連しています。

【 layout.css 】

スキン全体のレイアウトに関する内容が記述してある CSS ファイルです。

【 containers/ 以下 CSS ファイル 】

コンテナに関する内容が記述してある CSS ファイルです。

1 コンテナにつき 1CSS ファイルがあります。ファイルの詳細は、4p～5p をご参照ください。

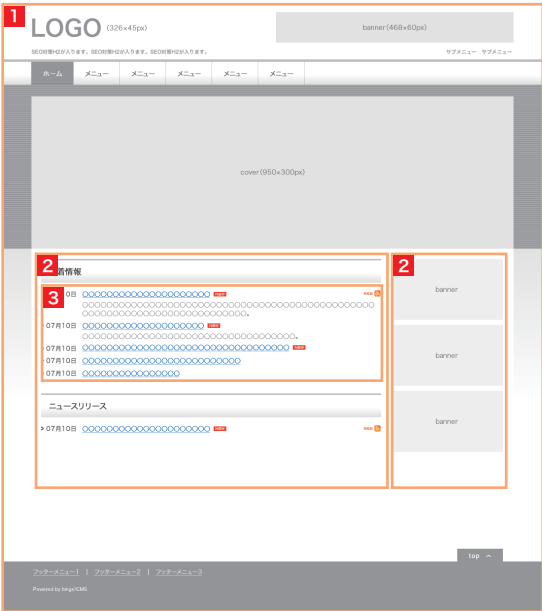
【 units/ 以下 CSS ファイル 】

ユニットに関する内容が記述してある CSS ファイルです。

1 ユニットにつき 1CSS ファイルがあります。ファイルの詳細は、4p・6p～10p をご参照ください。

コーディングについて

コーディングは、大きくの3つの行程に分けられ、「スキン」→「段組コンテナ」→「ユニット」の順にカスタマイズします。



- 1 スキン**
スキン全体のレイアウト構造や、またグローバルメニューのテキストリンクから画像ボタンへのカスタマイズ例をご紹介します。
詳細は 14p～17p をご参照ください。
- 2 段組コンテナ**
段組コンテナの構造やカスタマイズ例をご紹介します。
詳細は 18p～19p をご参照ください。
- 3 ユニット**
ユニットの構造やカスタマイズ例をご紹介します。
詳細は 20p～21p をご参照ください。

「スキン」 カスタマイズ

「index.tpl」の構造

始めに「スキン」の構成を説明します。
スキンは主に以下の5つのエリアで構成されてます。

- 1. ヘッダーエリア
- 2. カバーエリア
- 3. グローバルメニューエリア
- 4. コンテンツエリア (ユーザーエリア・サイドバーエリア)
- 5. フッターエリア

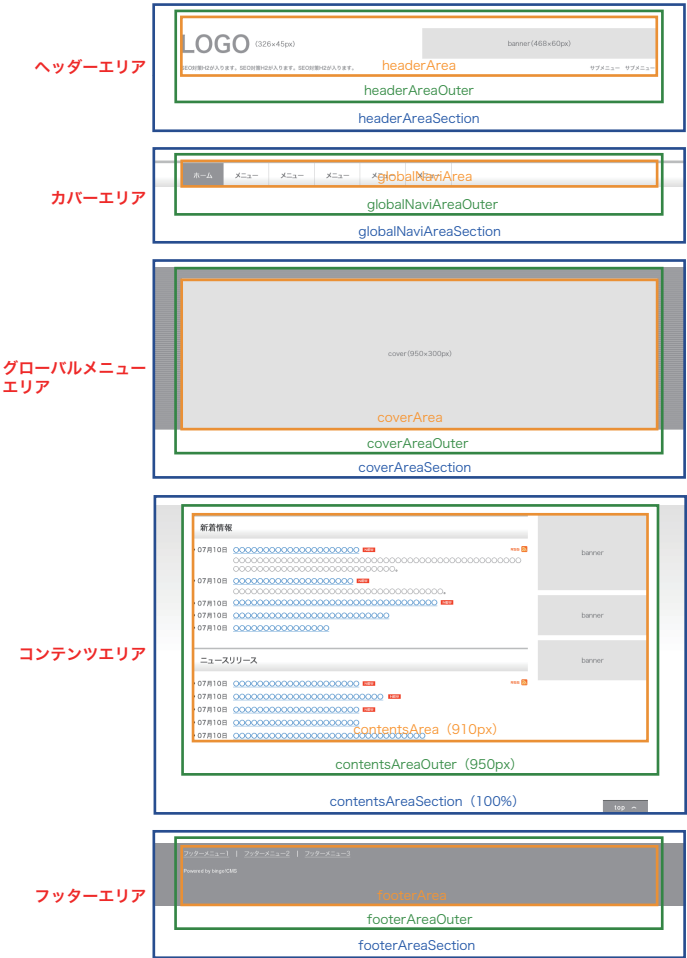
それらが積み重なった構造になっており、「myskin-sec」デザインの場合、わかりやすく分解すると右図のようになっています。
また、各エリアのタグは3層構造 (入れ子) になっていて、それぞれ決まった役割のCSSセレクタが記述してあります。
下記例のようにコンテンツエリア以外の他エリアも同様の構造になっています。
例えば「グローバルメニュー」と、「カバーエリア」の位置を変更したいと思ったら、それぞれのエリアの「●●●●AreaSection」を入れ替えるだけで簡単に変更できます。

例：コンテンツエリアの場合

contentsAreaSection
外側エリアの幅 (100%;) と、グラデーションの背景が設定されています。

contentsAreaOuter
内側エリアの幅 (950px;) と、白色の背景が設定されています。

contentsArea
さらに内側 (ユーザー編集エリア) の幅 (910px;) が設定されています。



「index.tpl」ソース詳細

※予約セレクトタについて

bingo!CMS スキンでは、システム上あらかじめ使用しているスタイルシートの ID があります。

予約セレクトタは以下「headerArea」「coverArea」「footerArea」「user-area」「usersidearea」の 5 つがあり、いずれも管理画面内での動作で必要になりますので、スキン作成時には「index.tpl」に記述する必要があります。

また下記「index.tpl」内で使用されているタグ及び、予約セレクトタの詳細は、別紙のタグリファレンス (tagreference.pdf) の 29p～30p を参照ください。

```
<!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">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta name="keywords" content="{ $METAKEY }" />
<meta name="description" content="{ $METADESC }" />
<include file=$JS>
<link href="{ $CSS }" rel="stylesheet" type="text/css" media="all" />
{ $UNITSCSS }
<title>{ $PAGE_TITLETAG[escape]</title>
{ $ANALYTICS }
</head>
<body>
<div id="wrapper">
  <!-- ページコンテナー -->
  <div id="container">

    <!-- ヘッダーエリア -->
    <div id="headerAreaSection">
      <div id="headerAreaOuter">
        <div id="headerArea" style="width: { $header.HEADERHDWIDTH }px; height: { $header.HEADERHDHEIGHT }px; background-color: #
{ $header.tpl_head_color|default:'FFFFFF' }">
          <div id="headerArealmage" style="overflow: hidden; position: relative; width: { $header.HEADERHDWIDTH }px; height:
{ $header.HEADERHDHEIGHT }px;">
            { if $header.HDLOGO }
              <h1 id="HdLogo" style="{ $header.HDLOGO_STYLE }"><a id="hlogo_anc" href="{ $m_menu.1.url }" title="{ $SITENAME }" style="width:
{ $header.HDLOGO_W }px; height: { $header.HDLOGO_H }px; background-image: url({ $header.HDLOGO });">{ $SITENAME }</a></h1>
            { /if }
            { if $header.HEADER_SEO }
              <h2 class="{ $header.HEADER_SEO_CLASS_STYLE }" style="{ $header.HEADER_SEO_CSS }">{ $header.HEADER_SEO }</h2>
            { /if }
            { if $header.HDIMG }
              <img id="HdImg" border="0" alt="" { $header.HDIMG } />
            { /if }
            { if $header.HDBANNER }
              { $header.HDBANNER }
            { /if }
            { if $header.HEADNAVI }
              <div id="subNaviArea" style="{ $header.navistyle }">
                { $header.HEADNAVI }
              </div>
            { /if }
          </div>
        </div>
      </div>
    </div>
    <!-- / ヘッダーエリア -->

    <!-- グローバルメニューエリア -->
    <div id="globalNaviAreaSection">
      <div id="globalNaviAreaOuter">
        { if $m_menu }
          <div id="globalNaviArea" class="color01">
            <ul>
              { foreach from=$m_menu item="m_menu" }
                <li class="color02 color03 color04 { $m_menu.active }"><a href="{ $m_menu.url }" title="{ $m_menu.sidx_title }"><span>
{ $m_menu.sidx_title }</span></a></li>
              { /foreach }
            </ul>
          </div>
        { /if }
      </div>
    </div>
    <!-- / グローバルメニューエリア -->
```

ヘッダーエリア

グローバルメニューエリア

カバールエリア

```

{if $header.coversetting)
<!-- カバーエリア /-->
<div id="coverAreaSection">
  <div id="coverAreaOuter">
    <div id="coverArea">
      <div id="coverAreaInner">
        <div id="coverAreaImage" style="width: ($header.HEADERBGWIDTH)px; height: ($header.HEADERBGHEIGHT)px;"><img
id="BgImag" alt="" {$header.BGIMG} /></div>
        <h3 class="{ $header.HEADERBG_TTL_CLASS_STYLE}" style="{ $header.HEADERBG_TTL_CSS}">
{$header.HEADERBG_TTL}</h3>
      </div>
    </div>
  </div>
</div>
<!--/ カバーエリア -->
{/if}

```

コンテンツエリア

```
<!-- コンテンツエリア /-->
<div id="contentsAreaSection">
  <div id="contentsAreaOuter">
    <div id="contentsArea">
      {if $PAGENA VI}
        <!-- パンくずリスト /-->
        <div id="breadcrumbsList">
          <ul>
            {foreach from=$PAGENA VI item="pnavi"}
              <li>
                {$pnavi}
              </li>
            {/foreach}
          </ul>
        </div>
        <!-- /パンくずリスト -->
      {/if}
      <div id="contentsAreaInnr">
        <!-- ユーザーエリア /-->
        <div id="mainArea" class="mainBg">
          <div id="user-area">
            {$items}
          </div>
        </div>
        <!-- ユーザーエリア -->
        <!-- サイドバーエリア /-->
        <div id="sidebarArea">
          <div id="sidebarAreaInnr">
            <div id="usersidearea">
              {$SIDE}
            </div>
          </div>
        </div>
        <!-- /サイドバーエリア -->
        <span class="brClr"><br class="clr" /></span>
      </div>
    </div>
  </div>
  <div id="pPageTop">
    <p><a href="#wrapper" title="ページトップ">ページトップ </a></p>
  </div>
</div>
<!-- /コンテンツエリア -->
```

ユーザーエリア

サイドバーエリア

フッターエリア

```
<!-- フッターエリア /-->
<div id="footerAreaSection">
  <div id="footerAreaOuter">
    <div id="footerArea">
      <!-- フッターナビゲーションエリア /-->
      <div id="footerNaviArea">
        {footer.FOOTNAVI}
      </div>
      <!-- フッターナビゲーションエリア -->
      <!-- コピーライト /-->
      <p id="pCopyright">{footer.CAP}</p>
      <!-- コピーライト -->
    </div>
  </div>
</div>
<!--/ フッターエリア -->

</div>
<!--/ ページコンテナ -->
</div>
</body>
</html>
```

「グローバルメニュー」 カスタマイズ

bingo!CMS デザインスキンのグローバルメニューはテキスト文字となっています。
ここでは、グローバルメニューの画像化のカスタマイズ例を紹介します。

グローバルメニューの画像化

グローバルメニューをテキスト文字から画像に変更します。
変更前と変更後のイメージが右図になります。
カスタマイズすることで、メニューごとに個別の画像を割り当てることができます。

変更前（テキスト文字メニュー）



変更後（画像メニュー）



固有 ID 番号の取得

グローバルメニューを画像化するにあたり必要なのが、メニューを構成している `` タグに個々のクラス（CSS）を記述する必要があります。
右の変更前 HTML ソースでは `` から `` まだが 1 メニューにあたりますが、システムで自動生成されたソースは基本的に共通のタグが生成されるため個々の判定をとることができません。

変更前 HTML ソース

```
<div id="globalNaviArea" class="color01">
  <ul>
    <li class="color02 color03 color04 active"><a href="#"><span> ホーム </span></a></li>
    <li class="color02 color03 color04 "><a href="#"><span> メニュー </span></a></li>
    <li class="color02 color03 color04 "><a href="#"><span> メニュー </span></a></li>
    <li class="color02 color03 color04 "><a href="#"><span> メニュー </span></a></li>
    <li class="color02 color03 color04 "><a href="#"><span> メニュー </span></a></li>
  </ul>
</div>
```

※ソースを一部簡略しています。

変更後 index.tpl ソース

```
<div id="globalNaviArea" class="color01">
  <ul>
    {foreach from=$m_menu item="m_menu"}
      <li class="color02 color03 color04 {$m_menu.active} gNavi{$m_menu.sidx_code}">
        <a href="{$m_menu.url}"><span>{$m_menu.sidx_title}</span></a></li>
    {/foreach}
  </ul>
</div>
```

※ソースを一部簡略しています。

変更後 HTML ソース

```
<div id="globalNaviArea" class="color01">
  <ul>
    <li class="color02 color03 color04 active gNavi1"><a href="#"><span> ホーム </span></a></li>
    <li class="color02 color03 color04 gNavi2"><a href="#"><span> メニュー </span></a></li>
    <li class="color02 color03 color04 gNavi3"><a href="#"><span> メニュー </span></a></li>
    <li class="color02 color03 color04 gNavi4"><a href="#"><span> メニュー </span></a></li>
    <li class="color02 color03 color04 gNavi5"><a href="#"><span> メニュー </span></a></li>
  </ul>
</div>
```

※ソースを一部簡略しています。

あとは固有に生成された「gNavi1～gNavi5」のクラスを使い、それぞれ個別に幅（width）や画像を割り当て、メニューを作成します。

また {`$m_menu.sidx_code`} について詳しくは、別紙のタグリファレンス（tagreference.pdf）の 28p を参照ください。

「段組コンテナ」 カスタマイズ

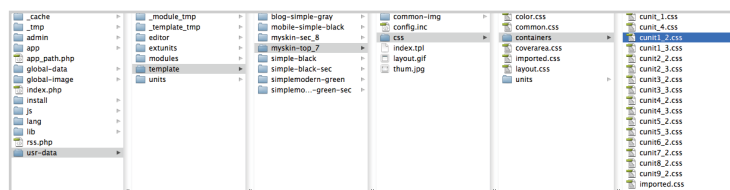
bingo!CMS は、あらかじめページのレイアウトを構成する「段組コンテナ」が数種類用意されています。ここでは、段組コンテナのファイル構造とカスタマイズ例を紹介します。

段組コンテナの構造

コンテナは「bingo!CMS」パッケージに含まれる tpl・CSS と、スキンに含まれる CSS のファイルで構成されています。

例として、「コンテナユニット 2 列 (50 : 50)」の段組コンテナのファイルディレクトリ構成及び、使用されている tpl・CSS ファイルの内容は以下のようになっています。

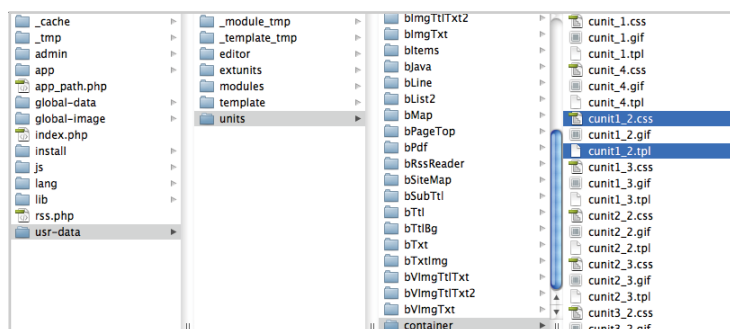
スキン側ディレクトリ



cunit1_2.css

```
.col2-1 .box1 .containerInner {
    margin: 0 10px 0 0;
}
.col2-1 .box2 .containerInner {
    margin: 0 0 0 10px;
}
```

bingo!CMS パッケージ側ディレクトリ



cunit1_2.css

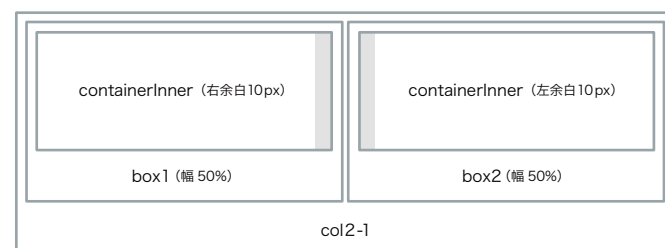
```
.col2-1 .box1 {
    float: left;
    width: 50%;
}
.col2-1 .box2 {
    float: right;
    margin-left: -1px;
    width: 50%;
}
```

cunit1_2.tpl

```
<div class="col2-1">
  <div class="box1">
    <div class="containerInner">{$data.0|default:"&nbsp;"}</div>
  </div>
  <div class="box2">
    <div class="containerInner">{$data.1|default:"&nbsp;"}</div>
  </div>
  <br class="clr" />
</div>
```

上記のファイル内容を図で表すと右図のようになります。

コンテナユニット 2 列 (50 : 50)



「段組コンテナ」カスタマイズ

カスタマイズ内容は、スキン内で完結できるよう、基本的にはスキン側の CSS ファイルに記述します。

bingo!CMS パッケージ側の CSS を変更してしまうと、カスタマイズしたスキン以外の段組コンテナにも影響がでてしまいますので、ご注意ください。

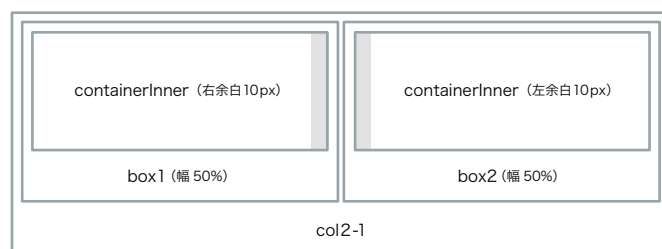
例として、「コンテナユニット 2 列(50:50)」をカスタマイズします。

変更前の内容は右のようになっています。

左右それぞれの幅 (width) や間の余白 (margin) などを変更します。

cunit1_2.css 変更前

```
.col2-1 .box1 .containerInner {  
  margin: 0 10px 0 0;  
}  
.col2-1 .box2 .containerInner {  
  margin: 0 0 0 10px;  
}
```



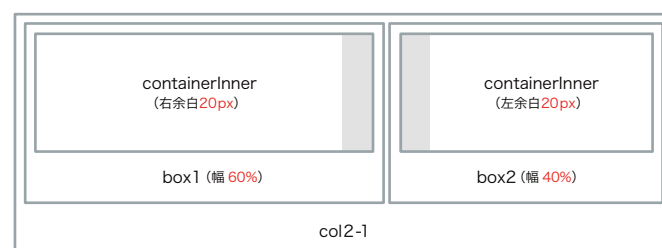
変更後の内容が右のようになっています。

幅 (width) を上書くように「左右 50%」から「左 60%」「右 40%」にと記述し、その間の余白 (margin) を 10px から 20px に変更しました。

注意点として幅 (width) を変更する際は、左右の合計値が 100% を超えないように値を記述してください。また、値を「%」ではなく「px」で指定することも可能ですが、同じように全体の幅を超える値を記述すると、レイアウトが崩れますのでご注意ください。

cunit1_2.css 変更後

```
.col2-1 .box1 {  
  width: 60% !important;  
}  
.col2-1 .box2 {  
  width: 40% !important;  
}  
.col2-1 .box1 .containerInner {  
  margin: 0 20px 0 0;  
}  
.col2-1 .box2 .containerInner {  
  margin: 0 0 0 20px;  
}
```



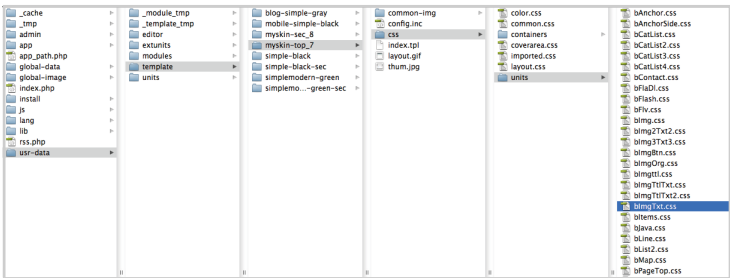
「ユニット」のカスタマイズ

bingo!CMS は、あらかじめページのコンテンツを構成する「ユニット」が数種類用意されています。
ここでは、ユニットのファイル構造とカスタマイズ例を紹介します。

ユニットの構造

ユニットは「bingo!CMS」パッケージに含まれる tpl・CSS と、スキンに含まれる CSS のファイルで構成されています。
例として、「画像（左）＋テキスト（右）」ユニットのファイルディレクトリ構成及び、使用されている tpl・CSS ファイルの内容は以下のようになっています。

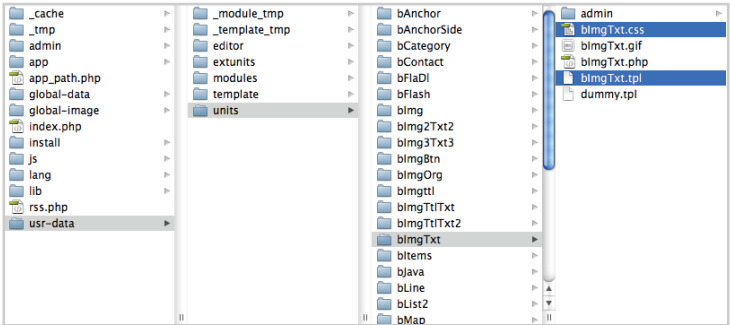
スキン側ディレクトリ



blmgTxt.css

```
/* 共通プロパティ -----*/
.blmgTxt { width: 100%; }
.blmgTxt .blmgTxtLeft { margin: 0 10px 0 0; width: 45%; }
.blmgTxt .blmgTxtLeft p.imgCap {
    margin: 0;
    padding: 5px 0;
    font-weight: bold;
    text-align: center;
    line-height: 1.4em;
}
/* コンテナ別プロパティ -----*/
/*col1 (1 カラム : 100%) */
.col1 .blmgTxt .blmgTxtLeft { width: 33%; }
```

bingo!CMS パッケージ側ディレクトリ



blmgTxt.css

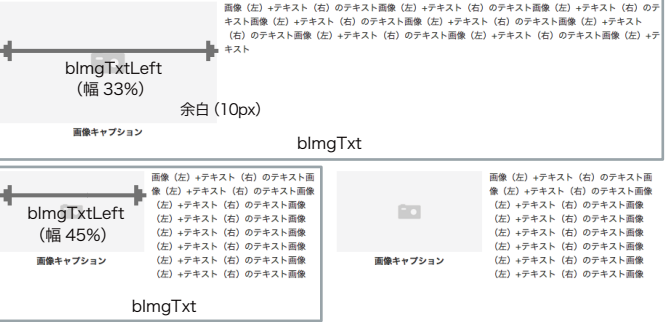
```
.blmgTxt .blmgTxtLeft { float: left; text-align: center; }
.blmgTxt .blmgTxtLeft img { width: 100%; }
.blmgTxt .blmgTxtLeft p.imgCap { }
.blmgTxt .blmgTxtRight { }
.blmgTxt .blmgTxtRight p { margin: 0; }
```

blmgTxt.tpl

```
<div class="blmgTxt">
  <div class="blmgTxtLeft">
    {$data.a1}<span></span>{$data.a2}
    (if {$data.bigtx_cap}<p class="imgCap color11">
    {$data.bigtx_cap}</p></if>
  </div>
  <div class="blmgTxtRight">
    <p>{$data.bigtx_txt}</p>
  </div>
  <span class="brClr"><br class="clr" /></span>
</div>
```

上記のファイル内容を図で表すと右図のようになります。

画像（左）＋テキスト（右）ユニット



「ユニット」のカスタマイズ

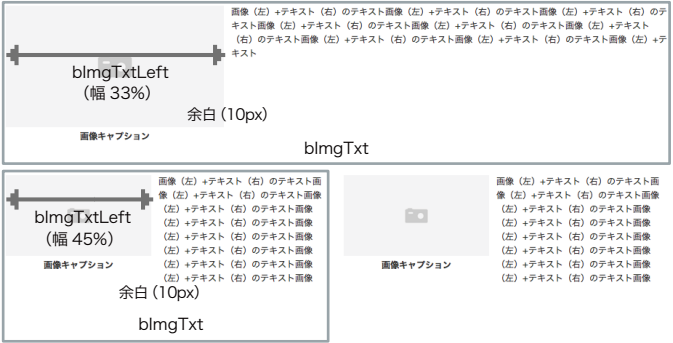
カスタマイズ内容は、スキン内で完結できるよう、基本的にはスキン側の CSS ファイルに記述します。
bingo!CMS パッケージ側の CSS を変更してしまうと、カスタマイズしたスキン以外のユニットにも影響がでてしまいますので、ご注意ください。

例として、「画像（左）＋テキスト（右）」をカスタマイズします。
変更前の内容は右のようになっています。
左の画像幅（width）や間の余白（margin）を変更します。
このユニットでは、1 列コンテナに配置された場合、は画像幅が 33%、それ以外のコンテナに配置されたときは 45% になるような記述がしてあります。
段組コンテナの幅に合わせてユニットのバランスを調整すれば、より完成度の高いサイトになります。

画像（左）＋テキスト（右）ユニット変更前

```
/* 共通プロパティ -----*/
.blmgTxt { width: 100%; }
.blmgTxt .blmgTxtLeft { margin: 0 10px 0 0; width: 45%; }
.blmgTxt .blmgTxtLeft p.imgCap {
  margin: 0;
  padding: 5px 0;
  font-weight: bold;
  text-align: center;
  line-height: 1.4em;
}
/* コンテナ別プロパティ -----*/
/*col1 (1 カラム : 100%) */
.col1 .blmgTxt .blmgTxtLeft { width: 33%; }
```

-
- 以下略
-



変更後の内容が右のようになっています。
1 列コンテナに場合は、画像幅が 25%、余白を 10px から 20px に変更しました。また、それ以外のコンテナで配置された場合は画像幅が 60% になるよう変更しました。

注意点として、画像幅の値を「%」ではなく「px」で指定することも可能ですが、段組コンテナの幅を超える値を記述すると、画像幅より狭い段組コンテナに配置した際に、レイアウトが崩れますのでご注意ください。

画像（左）＋テキスト（右）ユニット変更後

```
/* 共通プロパティ -----*/
.blmgTxt { width: 100%; }
.blmgTxt .blmgTxtLeft { margin: 0 10px 0 0; width: 60%; }
.blmgTxt .blmgTxtLeft p.imgCap {
  margin: 0;
  padding: 5px 0;
  font-weight: bold;
  text-align: center;
  line-height: 1.4em;
}
/* コンテナ別プロパティ -----*/
/*col1 (1 カラム : 100%) */
.col1 .blmgTxt .blmgTxtLeft { margin: 0 20px 0 0; width: 25%; }
```

-
- 以下略
-

