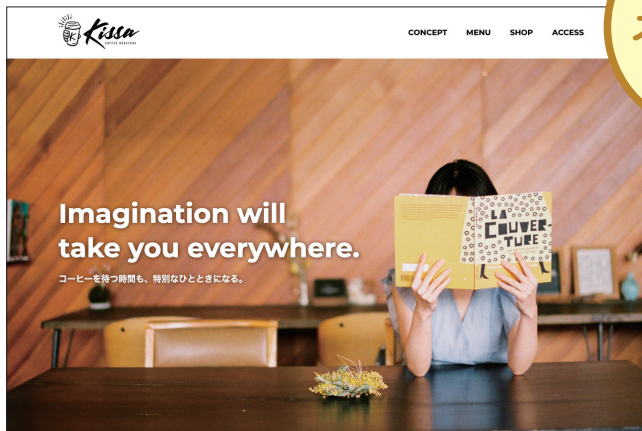


5つの レイアウト手法と レスポンスデザインが 学べる!!



フル スクリーン Chapter6

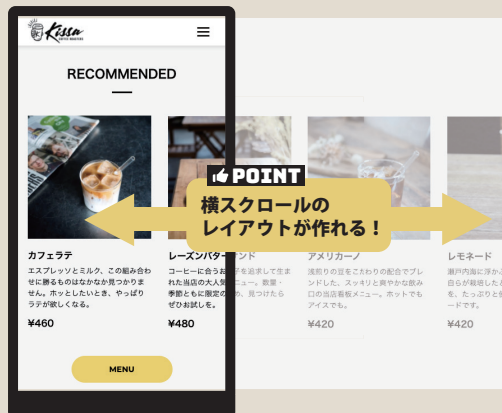
トップページでは背景に画面いっぱいの写真を配置するフルスクリーンレイアウトを作成します。イメージを重視する業種やインパクトを持たせたいウェブサイトに向いています。

フレックス ボックス Chapter7

要素を並べて配置する「フレックスボックス」の使い方を詳しく解説します。応用力の高い柔軟なレイアウト方法で、この手法を覚えるとレイアウトの自由度が格段に上がります。

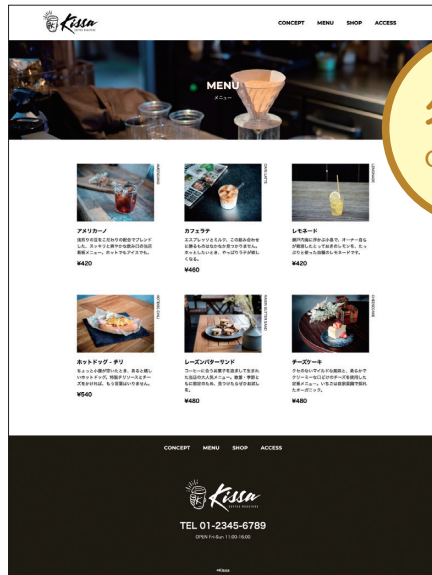
シングル カラム Chapter8

オーソドックスなシングルカラムのレイアウトです。シンプルなレイアウトも要素を左右反転して配置することでリズムが生まれます。



動きのある
ウェブデザインも
作れます!

本書は「KISSA」というオンラインショップも運営するカフェのウェブサイトを作りながら、さまざまなレイアウトパターンによるPC&モバイルサイト制作を学習します。



グリッド Chapter9

ページを格子状に区切って配置する「グリッド」を使用したレイアウトを学びます。要素を整然と並べることができ、メニューページやギャラリーページに向いています。



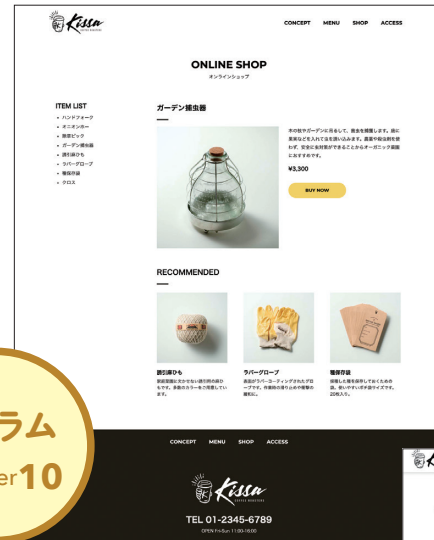
地図の 埋め込み& フォーム Chapter11

利便性の高い Google マップの埋め込み方法と、ウェブサイトに必要な問い合わせフォームの作り方を解説します。

動画や アニメーションの あるページも 作成



GIF アニメーションや YouTube 動画の埋め込み方法も解説します。

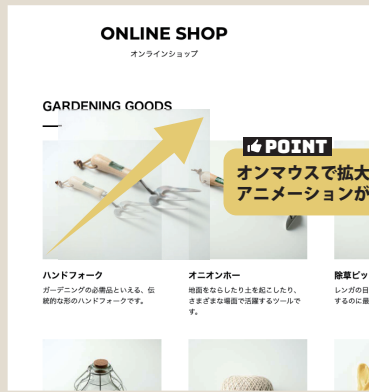


2カラム Chapter10

画面左にサイドバーを配置した 2 カラムのレイアウトを作成します。EC サイトやブログなどコンテンツが多く回遊性を高めたいウェブサイトに向いています。

レスポンス デザイン 全ページ対応

すべてのページはモバイル端末に対応したレスポンスデザインで作成します。モバイル用ウェブサイトをデザインする際のポイントも詳しく解説します。



1-2 サイト制作のフローとウェブデザイン

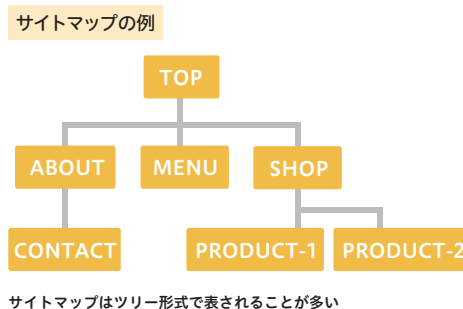
ウェブ制作者の仕事には大きく分けて「デザイン」と「コーディング」の2つがあります。この節では「デザイン」面にフォーカスし、実際にウェブサイト制作するフローや、デザイナーとコーダーのそれぞれの役割についても解説します。

1 サイトマップの作成_サイト制作のフロー①

ウェブサイトを制作する際のフローには大きく分けて3つの段階があります。

まずはじめに、ウェブサイトを制作する目的に合わせて、必要なページやコンテンツを洗い出し、ウェブサイトの大枠を決めていきます。依頼者がある場合は、依頼者からの意見や希望を聞きながら進めます。

必要なページをどのような構成で掲載するかを表した図を「サイトマップ」といい、右図のようにツリー形式で表されます。このときに、同時にディレクトリ構造やファイル名を決めておくとの作業がスムーズに進みます。



学習ポイント

ディレクトリ構造とファイル名の設定

サーバーにデータをアップロードする際には、無作為にファイルをアップロードするのではなく、一定のルールがあります。サーバー上に置かれたファイル名がそのままURLになるため、必ず半角英数字を使用するのが原則です。例えば「gihyo.com」というウェブサイトの「sample.html」というウェブページは「gihyo.com/sample.html」というURLになります。ファイル名を命名する際には、画像ギャラリーのページには「gallery.html」という名前をつけるなど、そのウェブページの内容に合ったファイル名であることが望ま

いです。また、例えば「花」「鳥」などジャンル分けされたギャラリーがいくつかあるような場合には、「gallery」というフォルダを作成し、その中に「flower.html」「bird.html」として格納するのもよいでしょう。その場合のURLは「gihyo.com/gallery/flower.html」「gihyo.com/gallery/bird.html」となります。

■ ファイルを種類別に格納

また前節で説明したとおり、ウェブサイトはHTMLファイルだけでなく、それに付随するCSS

ファイルや画像ファイル、音声ファイルなど、さまざまなファイルによって構成されます。これらのファイルも種類別に、CSSファイルは「css」、画像ファイルは「images」というフォルダにまとめておくなど、フォルダ分けをすると管理がしや

すくなります。こういったフォルダのことを「ディレクトリ」といい、このように正しくディレクトリ分けをすることもウェブ制作者の重要な仕事です。

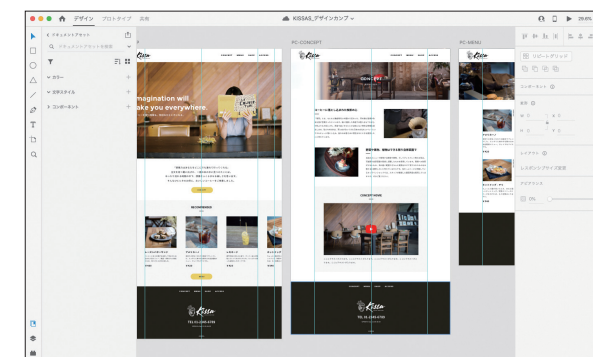
2 ワイヤフレームとキャンプの作成_サイト制作のフロー②

次に、ウェブページ内のボタンやメニューなどの大まかな配置や、どのようなレイアウトパターンを採用するかを決めます。これらの配置を記載した図面のことをワイヤフレームと呼び、丸や四角などの単純な図形を使ってざっくりとした図を作成します。一般的にこういった作業を「ディレクション」と呼び、ディレクターがいればディレクターが行いますが、ウェブデザイナーが兼務することもあります。

ディレクションによって必要な情報が集められたあとは、それらを元に、Adobe XDやSketchなどのデザインソフトや、Adobe Photoshop、Adobe Illustratorなどのグラフィックソフトを使用して、より細かいデザインを作成します。デザイナーはワイヤフレームを確認しながら、配置する画像の選定や、見出しの装飾、カラーリングなどをデザインし、ここではじめて最終的なウェブサイトの完成イメージが出来上がります。この完成イメージを「デザインキャンプ」や「モックアップ」と呼びます。



Adobe XDによるデザインキャンプ制作



1-5 ウェブデザインで知っておくべきこと

ウェブサイトで使用される文字デザインの基本的な知識や、色の指定の方法、使用する単位やレイアウトの概念など、ウェブ制作者にはさまざまな知識が求められます。ここでは、ウェブ制作者に必要な基本的な知識について確認します。

1 文字やテキストについて

まず、ウェブページを構成するもっとも基本的な要素である、文字についてです。文字は、見出しやコピー、本文、メニューなどのあらゆる場所で使用され、掲載する目的によって大きさや色を変える必要があります。例えば、キャッチコピーは大きく目立つ色で表示する、本文は読みやすいサイズで行間に

ゆとりを持たせるなど、文字が持つ役割に応じた設定をします。

文字にはサイズや色だけでなく「フォント」という概念があります。フォントは文字の見た目のことで、書体と呼ばれることもあります。



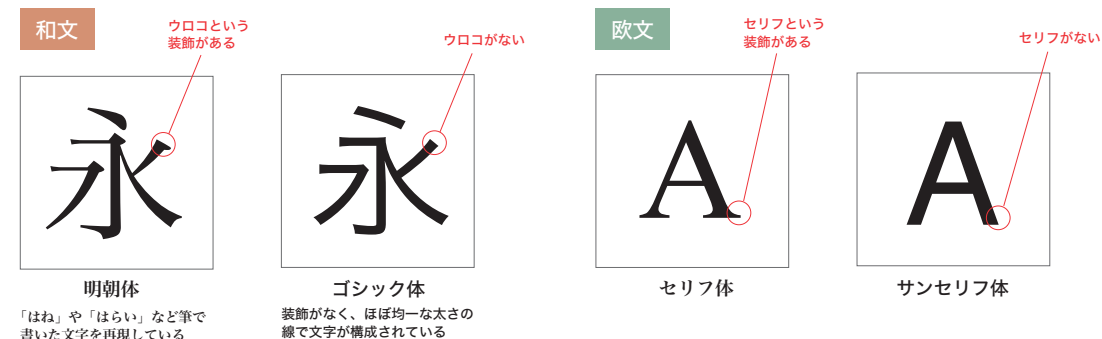
同じ「南の島へ行こう。」というキャッチコピーでも、文字の大きさや太さによってずいぶん印象が違う。コンテンツをどのように見せたいかで使い分ける必要がある。

■ 明朝体とゴシック体

和文フォントには大きく分けて「明朝体」と「ゴシック体」の2種類があります。明朝体には「ウロコ」と呼ばれる装飾があり、「はね」や「はらい」など、筆で書いた文字を再現したようなデザインとなっています。一方ゴシック体には明朝体のような装飾はなく、ほぼ均等な太さの線で構成されています。

■ セリフ体とサンセリフ体

欧文フォントでは、明朝体のように線端に装飾があるフォントを「セリフ体」、ゴシック体のようにほぼ均一の線で構成されるフォントを「サンセリフ体」と呼びます。どちらのフォントを選ぶかによってウェブサイトの印象は大きく変わりますので、伝えたいイメージに合わせて適切なフォントを選ぶようにしましょう。



■ 行間によって読みやすさが変わる

文字を掲載する際に気を配りたいのが行間の設定です。行間とは文字どおり、行と行のあいだのスペースのことで、**行間によって可読性は大きく変わります**。例えば、行間が詰まった（狭い）文章は圧迫感があり読みにくく、逆に空きすぎ（広い）てしまっても間延びした印象になります。文字のサイズや文章量によって読みやすい行間の設定は異なりますが、本文であれば、1文字のサイズの半分程度の行間に設定すると読みやすい文章になります。

行間が詰まった文章

親譲りの無鉄砲で小供の時から損ばかりしている。小学校に居る時分校の二階から飛び降りて一週間ほど腰を抜かした事がある。なぜそんな無闇をしたと聞く人があるかも知れぬ。別段深い理由でもない。新築の二階から首を出していたら、同級生の一人が冗談に、いくら威張っても、そこから飛び降りる事は出来まい。弱虫やーい。と囃したからである。小使に負ぶさって帰って来た時、おやじが大きな眼をして二階ぐらいから飛び降りて腰を抜かす奴があるかと云ったから、この次は抜かさずに飛んで見せますと答えた。（青空文庫より）

行間を適度に空けた文章

親譲りの無鉄砲で小供の時から損ばかりしている。小学校に居る時分校の二階から飛び降りて一週間ほど腰を抜かした事がある。なぜそんな無闇をしたと聞く人があるかも知れぬ。別段深い理由でもない。新築の二階から首を出していたら、同級生の一人が冗談に、いくら威張っても、そこから飛び降りる事は出来まい。弱虫やーい。と囃したからである。小使に負ぶさって帰って来た時、おやじが大きな眼をして二階ぐらいから飛び降りて腰を抜かす奴があるかと云ったから、この次は抜かさずに飛んで見せますと答えた。（青空文庫より）

2 ウェブで使用するフォントについて

フォントについて大まかに理解できたら、次はウェブで使用するフォントについて理解を深めていきましょう。

ウェブページ上で文字を**どの種類のフォント**で表示するかは、**CSSで指定**します。ページ全体でゴシック体or明朝体、セリフ体orサンセリフ体どちらのフォントで表示するかという指定や、「特定の場所だけ別のフォントで表示する」という指定も可能です。

■ コンピュータがフォントを表示するしくみ

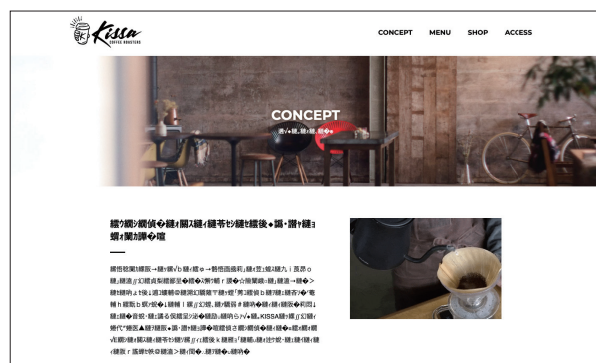
フォントにはゴシック体or明朝体という分類だけでなく、非常に多くの種類があります。和文フォントだけでも数千種類、欧文も含めると数え切れないほどの種類がありますが、そのすべてをウェブペー

ジ上で自由に使用できるかというとそうではありません。

これにはコンピュータがフォントを表示するしくみが関係しています。

4 文字コードについて理解しよう

コンピュータ上にあるデータはすべて数字に変換することができますが、文字も例外ではなく、ひとつひとつの文字にはそれぞれ数字が割り当てられています。どの文字にどの数字を割り当てるかを規定したものを「文字コード」といい、ウェブサイトを制作する際には文字コードを理解する必要があります。この文字コードには「Shift-JIS」「EUC-JP」などさまざまな種類があり、最近の日本語のウェブサイトは「UTF-8」という文字コードが主流となっています。



文字コードの指定が間違っていると文字化けして表示されることがある

■標準的なコードは「UTF-8」

日本語や英語、中国語など、世界中にはさまざまな言語がありますが、各言語にはこれまで各国独自の文字コードを使用していたため、文字化けなどの問題がありました。近年ではこれを統一しようという動きがあり、世界中の文字を一括して登録すること

を目標にしたものが「Unicode」です。UTF-8はこのUnicodeの一種なので、これから新規にウェブサイトを作成する場合はUTF-8を選択するのがよいでしょう。

5 色と色選びのきほん知識

ウェブページをデザインする際、どのような色を使うかによって印象は大きく変わります。信号を思い浮かべてもらえればわかりますが、色は人間の本能や感情と強く結びついています。

■カラー設計とサイトの使いやすさの関係

例えば企業サイトのデザインで使用するボタンの色について、こんな実験結果があります。赤や青、緑など、複数の色のボタンを同じウェブサイトを設置して、どの色が一番クリックされたかを検証したところ、色によってクリック率に明らかな違いがあっ

たそうです。業種によって違いもあるので一概にどの色がいい、ということではありませんが、この結果からもわかるように、色の設計はウェブサイトの使いやすさに大きな影響を与えます。

■アクセシビリティへの配慮も大切

また、色を選択するときに、アクセシビリティへの配慮も忘れてはいけません。特に重要なのはコントラストです。コントラストとは明度（明るさ）や彩度（鮮やかさ）の落差のことで、この落差が大きいほどコントラストは強くなります。また、色相（色の組み合わせ）によってもコントラストが生まれます。異なる色を円形に並べた色相環というチャートで、遠くにある色ほどコントラストが強い組み合わせになります。

例えばボタンなどをデザインする際、わかりやすく押しやすい＝アクセシビリティに優れたデザインにするにはコントラストを強くする必要があります。単に好みやイメージで色を選択するのではなく、こうしたさまざまな検討材料を元に、色の設計をするようにしましょう。

note

ウェブコンテンツ・アクセシビリティガイドライン (WCAG) という、ウェブのアクセシビリティに関して定められたガイドラインには、文字情報や画像と配色などによって差異を設けた背景にすることや、ボタンやナビゲーションなどのユーザーが操作する要素をはっきり際立たせること、テキストと背景のコントラストの割合は最低でも 4.5 : 1 にすべきなど、細かなルールが定められています。

明度によるコントラスト



コントラストが強い例



コントラストが弱い例



明度の落差が大きい = コントラストが強いほど読みやすい

彩度によるコントラスト



コントラストが強い例

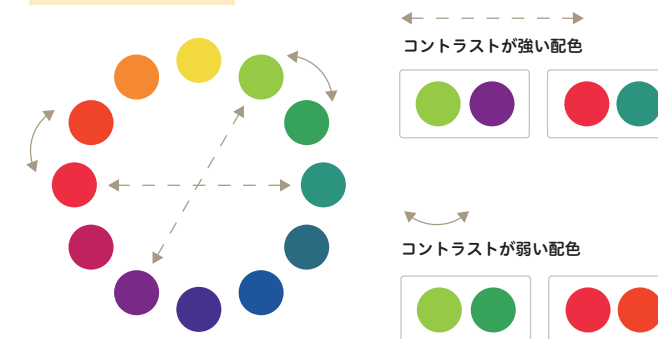


コントラストが弱い例



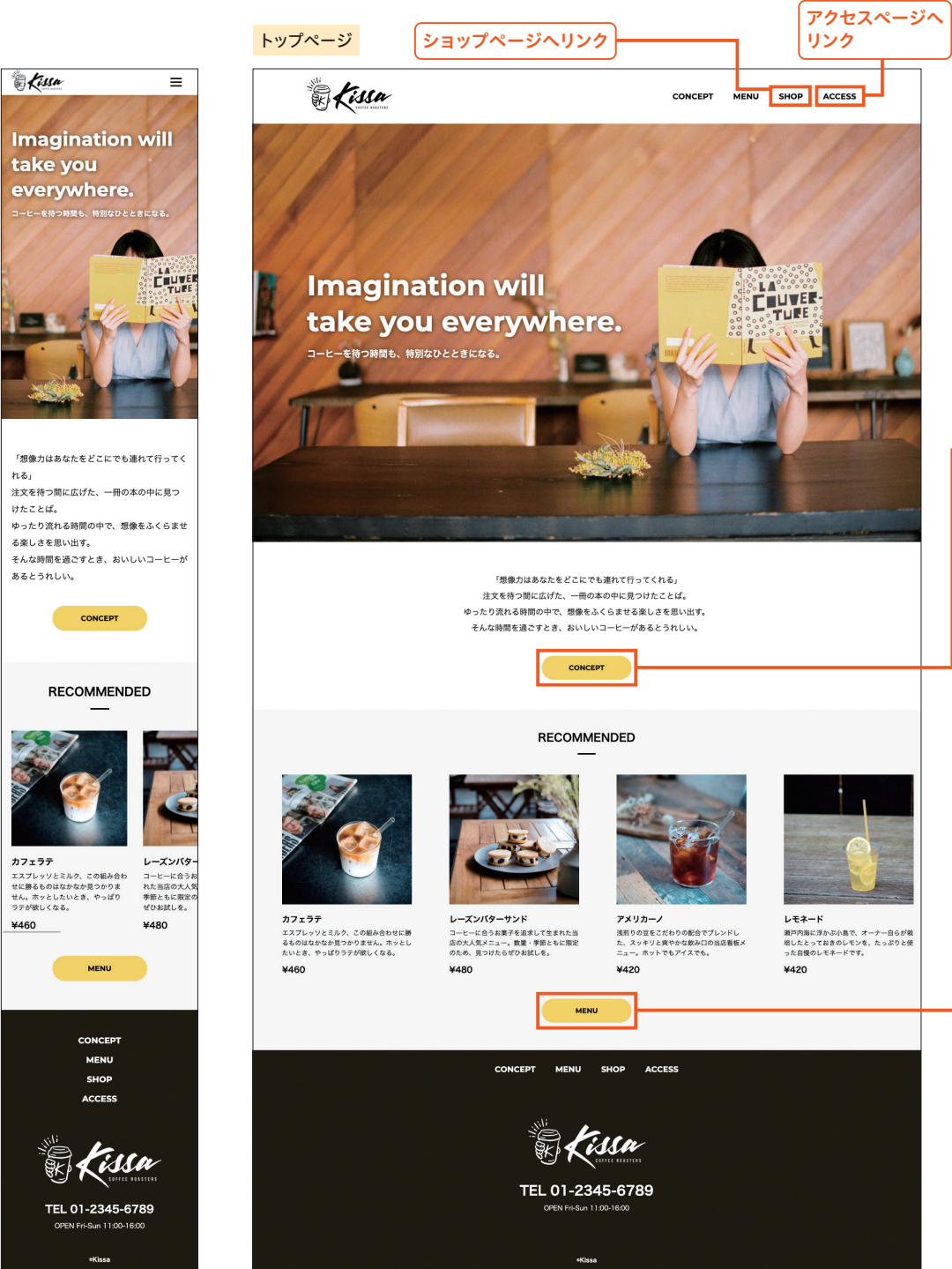
彩度の落差が大きい = コントラストが強いほど読みやすい

色相によるコントラスト



離れている色ほどコントラストが強く、近い色ほどコントラストが弱い

ウェブサイトのページ相関図



■ ページの相関を確認する

各ページのイメージとリンク関係を確認しましょう。メニューだけでなく、ロゴ画像にもリンクを張って、ウェブサイト内のページを相互につないでい

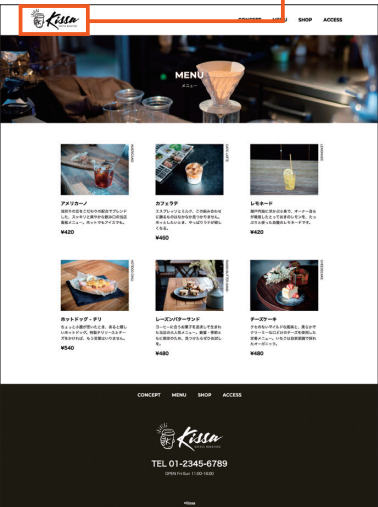
す。6章からのサイト制作ではサンプルサイトの階層だけでなく、ページ同士の関係性を意識しながら学習を進めましょう。

コンセプトページ



お店のロゴから
トップページへリンク

メニューページ



3-1 HTMLのきほん知識

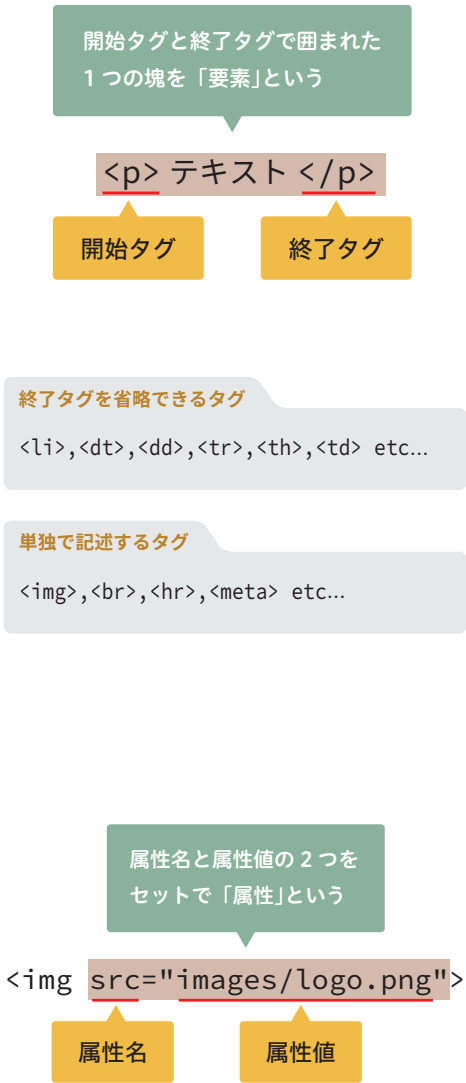
必要なソフトのインストールが完了したらいよいよコーディングの学習を開始します。まずはHTMLの基礎や文法、基本的な構造について学んでいきましょう。

1 タグのきほん的な書き方

1章で紹介（P.29）したとおり、HTMLとはテキストに目じるしをつけて文書に構造を与えるための言語です。目じるしをつけたテキストは、それぞれ役割をもった「要素」となり、それらの要素の集合体が1つのウェブページを形成しています。要素には、ヘッダー用の要素であるheader要素や、段落用のp要素などさまざまな種類があり、それらを組み合わせることでウェブページの文書構造や段落ができあがるのです。

■ HTMLの目じるし「タグ」
HTMLで使用される目じるしのことを「タグ」といいます。タグは「<」と「>」でタグ名を囲って記述し、例えばpタグは「<p>」という書き方になります。タグには開始タグと終了タグがあり、終了タグはタグ名の前に「/」（スラッシュ）をつけます。pタグの場合、終了タグは「</p>」となります。
ほとんどのタグは開始タグと終了タグが対になっており、この2つで挟まれた1つの塊が「要素」になります。ただし、タグの一部には終了タグを省略して記述できるものや、単独で記述するタグもあり、必ず対になっていないわけではありません。

■ 要素に細かな設定を与える「属性」
また、タグに「属性」を書き加えることで、要素に細かな設定をすることができます。属性の書き方は、まずタグ名の後ろに半角スペースを空け、「属性名」を入力し



ます。その後ろに「=」（イコール）を続けて入力し、「"」と「"」（ダブルクォーテーション）のあいだに「属性値」を入れます。その際に、属性名と属性値は必ずセットで記述するのがルールです。

2 HTMLの構造は2つの大きな箱に分かれている

■ html要素の中に2つの要素が含まれる
タグと属性の基本を覚えたら、次にHTML文書のおおまかな構造を理解しましょう。
HTML文書は大きく分けて「head要素」と「body要素」の2つのエリアに分かれており、その2つを包括しているのが「html要素」です。例えるならば、html要素という大きな箱の中にhead要素とbody要素という箱が入っているイメージです。このように入れ子になっている外側の要素を「親要素」といい、内側の要素を「子要素」といいます。

■ コンピュータが参照するhead要素
head要素とbody要素にはそれぞれ別の役割があります。まず、head要素はその文書に関する基本情報を記述する要素です。例えばページのタイトルや文字コード、読み込む関連ファイルの指定などです。これらはおもにウェブブラウザや検索エンジンが文書の概要を把握するために参照します。

■ ユーザーが閲覧するbody要素
一方、body要素には本文を記述します。見出しや文章、画像など、普段私たちがウェブブラウザで見ているコンテンツは、基本的にすべてbody要素に記述されています。ブラウザや検索エンジンにとってわかりやすいだけでなく、人間が理解しやすい構造で記述することが重要です。



4-1 CSSのきほん知識

CSSは、レイアウトや装飾など見た目を整えるための言語で、HTMLとともにウェブサイトをデザインするうえで欠かせない言語です。まずは基本的な文法やルールについて学びましょう。

1 文書の見た目をコントロールするCSS

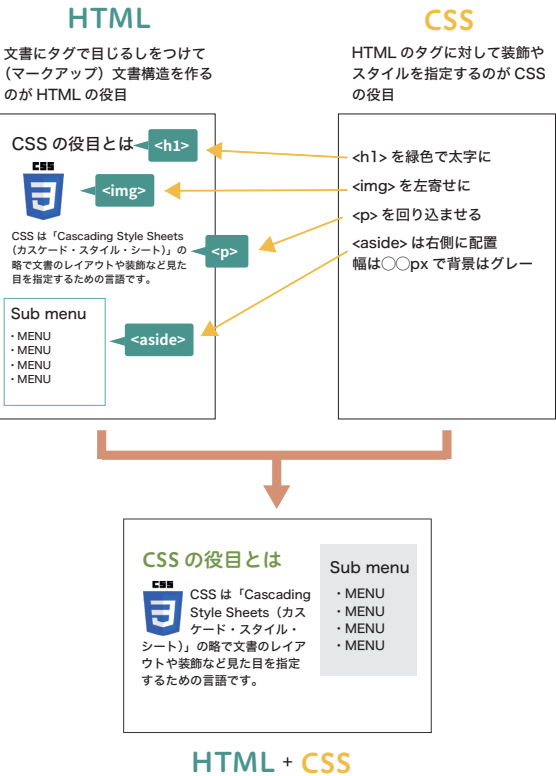
CSSは「Cascading Style Sheets(カスケード・スタイル・シート)」の略で、HTMLが文書の構造を作る言語であるのに対して、CSSはそのHTML文書のレイアウトや装飾など、見た目をコントロールするための言語です。基本的にHTMLと組み合わせて使用します。HTMLのみでも情報を伝えることはできますが、CSSを使用すれば、レイアウトを工夫したり文字の色やサイズを変更するなどして、情報をより伝えやすくすることができます。

■ CSS3から増えた表現の選択肢

HTMLと同じようにバージョンがあり、現在の最新バージョンはCSS3です。CSS3ではアニメーションや要素の変形、角丸やグラデーションなど、これまでは難しかったさまざまな表現が可能になり、ウェブデザインの新しい可能性が広がりました。

note

CSS2以前はCSS全体で機能のアップデートが行われていましたが、CSS3からは色に関連する機能は「〇〇モジュール」、背景とボーダーに関連する機能は「△△モジュール」というように、関連する機能がモジュールと呼ばれる単位で分類され、それぞれのモジュールごとで新たな機能の追加や改良が続けられています。



CSS3で追加されたプロパティの一例

border-radius	linear-gradient	box-shadow
角丸 (かどまる) CSS だけでは表現できなかった角丸を、「border-radius」という方法で可能になった。	グラデーション 「linear-gradient」という方法でグラデーション表現も可能になった。	ドロップシャドウ 「box-shadow」という方法で要素に影をつけることが可能になった。

2 CSSのきほん的な書き方

■ スタイルシートの基本文法

まずはCSSの基本的な文法を紹介します。CSSの記述はセレクト、プロパティ、値の3つで構成され、以下のように書きます。

セレクト { プロパティ : 値 ; }

「セレクト」には適用する対象(どれ)、「プロパティ」にはスタイルの種類(何を)、「値」には適用させたい結果(どうするか)を設定します。つまり

どれの { 何を : どうするか ; }

が基本的な文法になります。

■ スタイルシートの記述ルール

プロパティと値は { ~ } で囲みます。あいだにコロン(:)を記述することでセットになり、このセットのことをスタイル宣言といいます。

セミコロン (;) は、1つのセレクトに複数のスタイルを指定する際にそれぞれのスタイル宣言を区切るための記号です。

例えばHTML文書内の段落に設定した文字(どれ : p要素)の色(何を : color)を赤くしたい(どうするか : red)場合には、pタグをセレクトとして次のように記述します。

p { color : red ; }

■ 読みやすいコードの書き方

半角スペースや改行はスタイルに影響しないため、これらを使用してコードを見やすく整形しても問題ありません。本書では次のように整形して記述していきます。VS Codeの自動入力機能により、改行後のインデントやプロパティの「:」の後ろの半角スペースは自動で挿



5-1 レスポンシブデザインのきほん知識

ここからは、サイトをモバイル端末で見たときのレイアウトを作成していきます。コードの記述に入る前に、まずはレスポンスデザインの基本的な考え方について学んでおきましょう。

1 レスポンシブデザインとは

レスポンスデザインとは、端末によってフレキシブルに表示を切り替えられるウェブサイトの作り方のことを言います。スマートフォンの普及により、ウェブサイトは「PCで見るもの」から「スマホでいつでも手軽に見られるもの」に変わりました。タブレットのような大画面のモバイル端末や超小型のスマートフォンなども登場し、ウェブ制作者には、こうしたさまざまな端末での閲覧を前提としたウェブсайт設計やデザイン、コーディングが求められています。



■デザインをする際に意識したい3つのポイント
ではサイトの閲覧環境としてのPCとスマートフォンでは、どのような違いがあり、デザインにおいて、何に気をつければよいのでしょうか。大きな点として以下の3つの違いがあると考えます。

- ①画面の大きさの違い
- ②操作性の違い
- ③体験の違い

この3つのポイントをまずは理解することが、レスポンスデザインにおいて特に重要です。1つずつ解説していきます。

■①もっとも大きな違いは画面サイズ
まず何よりも大きな違いは「画面サイズ」です。説明するまでもないですが、PCの大画面で見ると、手元のスマートフォンで見るのでは、表示される情報の「量」がまったく異なります。

フッターを例に説明します。フッターは、一般的にウェブサイトが一番下に配置されますが、PCの大画面では、ユーザーがフッターにたどり着いたときにも、他のさまざまなコンテンツが画面内に表示されています。しかし、スマートフォンでは表示できる領域が限られているため、「画面の大部分をフッターが占有する」こととなります。この違いは大きいです。

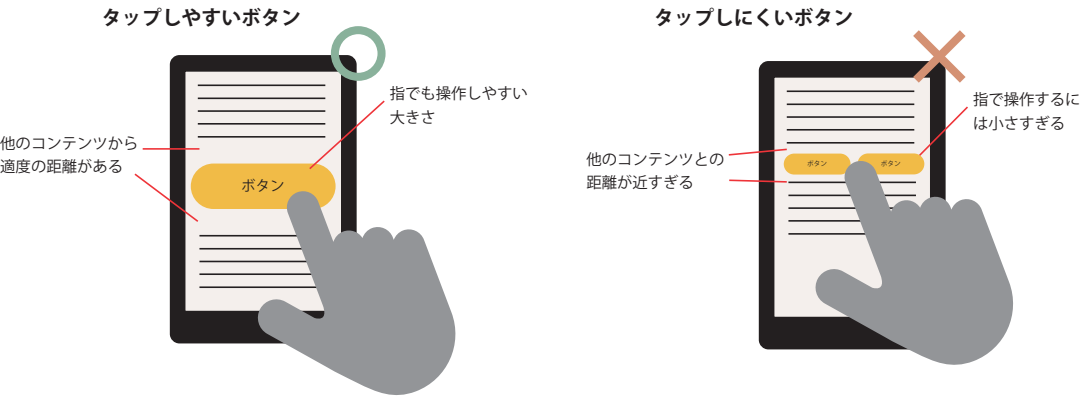
一番下までスクロールしたときにフッターがスマホの画面を占有していることを理解し、別ページへの導線や申し込みボタンなど、ユーザーに起こしてほしい行動を促す装置を置いておくことが大切になります。フッターにサイトマップのようなメニューが掲載されていることがよくありますが、これはサイト内の回遊性（P.35）を確保し、他のページを見てもらいやすくするための設計であると言えます。



■②PCとスマホでは操作性が異なることを理解する

では、小さな画面の中にいろいろと詰め込んだほうがよいのかというと、必ずしもそうではありません。特にユーザーが直接触れる場所である、ボタンなどのサイズへの配慮は重要です。マウスでのクリックと違って、タップやタッチなど指先で操作するスマートフォンでは繊細な操作は難しいものです。ボタンは押しやすい大きさにする、タップできる要素が隣接しすぎないようにするなど、指先での操作にストレスを感じさせないデザ

インを心がける必要があります。また、画面が小さく縦長なスマートフォンでは、必然的にコンテンツを縦に並べて配置することが多く、そのため「画面のスクロール量」が多くなります。ヘッダーのナビゲーションなどユーザーが頻繁に触れる要素は、常に上部に固定表示するなどして、画面をスクロールして上まで戻らなくてもすぐにタップできる位置に配置すると、親切的なサイト設計になります。

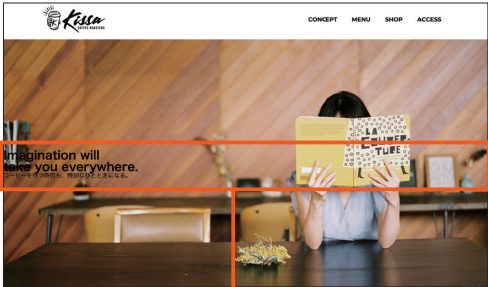


STEP 2 見出しとキャッチコピーエリアの大きさを調整する

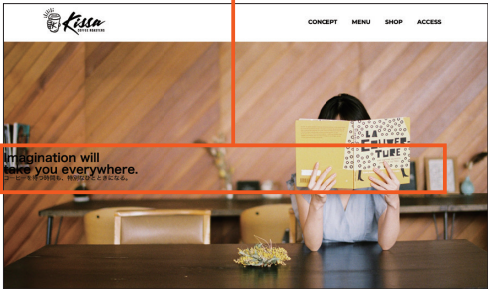
続いて、見出しとキャッチコピーエリアの大きさを調整します。2つの要素を包んでいるdiv要素につけたクラス名「first-view-text」をセレクトにしてCSSを記述していきます。まずは、親要素いっぱいに広がるよう、widthプロパティに「100%」を指定し、そのうえでmax-widthプロパティで最大幅を指定します。最大幅はヘッダーと同じく「1200px」とします。

```
.first-view-text {
  width: 100%;
  max-width: 1200px;
}
```

```
12
13 .first-view-text {
14   width: 100%;
15   max-width: 1200px;
16 }
```



ブラウザ上では変化がないが、要素の幅が1200pxに変更された



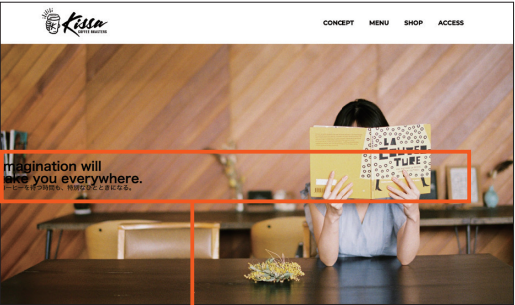
STEP 3 見出しとキャッチコピーの位置を調整する

次に、見出しとキャッチコピーの掲載位置を調整していきます。margin-left、margin-rightプロパティにそれぞれ「auto」を指定し、左右中央に配置します。

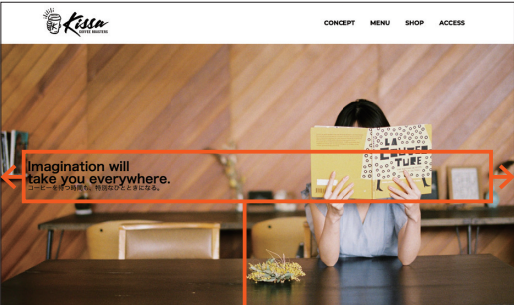
```
.first-view-text {
  width: 100%;
  max-width: 1200px;
  margin-left: auto;
  margin-right: auto;
}
```

```
13 .first-view-text {
14   width: 100%;
15   max-width: 1200px;
16   margin-left: auto;
17   margin-right: auto;
18 }
```

note
1200px 以下のブラウザ幅でのプレビュー環境では、変化がわかりにくい場合があります。



エリア全体がブラウザの左右中央に表示された

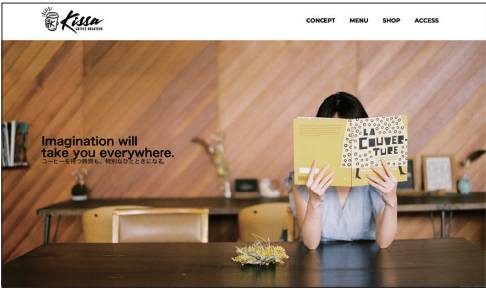


STEP 4 見出しとキャッチコピーの位置をさらに細かく調整する

paddingプロパティを使い、さらに細かく位置を調整します。やや右上に移動したいので、padding-left、padding-bottomプロパティにそれぞれ以下のように指定します。

```
.first-view-text {
  ... (略) ...
  margin-right: auto;
  padding-left: 40px;
  padding-bottom: 80px;
}
```

```
16 margin-left: auto;
17 margin-right: auto;
18 padding-left: 40px;
19 padding-bottom: 80px;
20 }
```



STEP 5 テキストの色と太さを変更する

続いて、テキストのスタイリングを進めます。まずは、テキストの色と太さを変更しましょう。colorプロパティとfont-weightプロパティでそれぞれ以下のように指定します。

```
.first-view-text {
  ... (略) ...
  padding-bottom: 80px;
  color: #ffffff;
  font-weight: bold;
}
```

```
18 padding-left: 40px;
19 padding-bottom: 80px;
20 color: #ffffff;
21 font-weight: bold;
22 }
```



テキストの色が白になり太字になりました。

7-1 フレックスボックスレイアウト制作で学ぶこと

この章では、トップページの下部におすすめ商品を掲載する「RECOMMENDED」エリアを作成します。フレックスボックスという手法を使って要素を並べる方法や、ブラウザの幅からはみ出た要素の表示のしかたについて学習していきましょう。

1 フレックスボックスレイアウト制作で学べるHTML&CSS

「RECOMMENDED」エリアでは、商品を横並びに配置し、ブラウザの幅からはみ出た分は横スクロールで表示するレイアウトを採用しています。スマートフォンのように幅の狭い端末でも多くの情報を掲載できることから、レスポンシブデザインではよく使用されるレイアウトです。横並びにはフレックスボックスという手法を使います。ブラウザ幅から要素がはみ出る場合の設定についても学習しましょう。また、各要素の説明文にはdl要素を使用し、記述リストの使用方法を学びます。



学べるHTML

- ul、li要素による商品リストの作成
- dl要素による記述リストの作成

学べるCSS

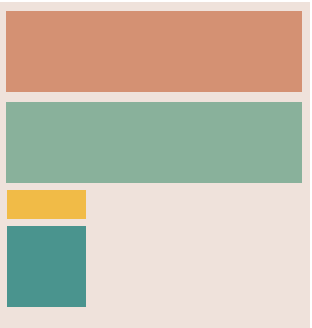
- 背景色の変更
- フレックスボックスによる要素の横並び
- 疑似要素による見出しの装飾
- はみ出た要素を横スクロールで表示する
- 疑似クラスで特定の要素を指定する

2 フレックスボックスレイアウトとは

「フレックスボックスレイアウト」とは、正確には「CSS Flexible Box Layout Module」といい、フレキシブル=柔軟性の言葉が表すとおり、ボックスを使って柔軟にレイアウトを作成する手法です。これまでの学習でも、ナビゲーションメニューを横並びにしたり、フッターの要素を縦に並べるなど、すでにいくつかのレイアウト作成で使用しています。あらためてこの章では、「フレックスボックスレイアウト」について詳しく説明します。

HTMLという言語は、各要素がボックス、つまり箱のようなイメージで、積み上げたり、箱の中に箱を入れたりすることで文書が構成されています。これらの箱は何も指定をしない場合は、単純に縦に積み上げられたり、決められたルールに沿って幅が設定されてしまいますが、箱の並び方や大きさ、並び順、並べる方向などをCSSで細かく指定して、柔軟にレイアウトできるように考えられたのがフレックスボックスレイアウトです。

何も指定をしていない HTML



フレックスボックスを使用すると...



ルールに沿って要素のサイズや配置が決まる

要素のサイズや配置を細かく指定でき、柔軟なレイアウトが可能

note

フレックスボックスが登場する前は、要素を横並びにするには「float」というプロパティを使用するケースが多かったのですが、仕様が難解できちんと理解しないと表示が崩れることもよくありました。フレックスボックスはそういった問題を解消し、より簡単に、かつ自在にレイアウトを組めることから、デザインの自由度を高めてくれる重要なプロパティといえます。

COLUMN

古いブラウザへの対応

フレックスボックスはほとんどのブラウザに対応していますが、比較的新しいプロパティのため一部の古いブラウザに対応していません。こうした古いブラウザに対しては、プロパティに「ベンダープレフィックス」という文字列を追加することで適用できる場合が多く、実務ではベンダープレフィックスを追加しておいたほうが安心です。

本書ではChromeでの表示を前提に紹介しているため、Chrome以外のベンダープレフィックスについては省略します。最新のブラウザ対応状況は、以下のリンク先からご確認ください。

<https://caniuse.com/#search=flex>

8-1 シングルカラムレイアウト制作で学ぶこと

この章では、CONCEPTページとしてお店のコンセプトや店舗紹介ムービーを掲載するページを制作します。動画コンテンツのうち1つめにはGIFアニメーションを使用して“動く画像”の掲載方法を学び、店舗紹介ムービーのエリアではYouTubeにアップした動画を埋め込む方法を紹介します。

1 シングルカラムレイアウト制作で学べるHTML&CSS

CONCEPTページでは、まず一番上のページタイトルエリアで画像をブラウザの横幅いっぱいに配置し、その上にタイトルを掲載します。そしてコンセプトエリアでは、動画とテキストの横並びの配置と、

並び順を逆転する方法を学びます。動くコンテンツの1つにはGIFアニメーションを使用します。そして最後にYouTube動画の埋め込み方法を解説します。

コンセプトページのデザインカンパ



GIFアニメーションによる動きのあるコンテンツ



YouTube 動画コンテンツ



学べるHTML

- GIFアニメーションの配置
- classを複数使う
- iframeによるYouTube動画の埋め込み

学べるCSS

- フレックスボックスによる上下中央揃え
- 横並びの左右反転
- iframe要素のサイズ指定

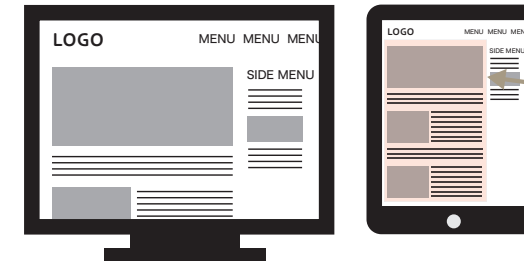
2 シングルカラムレイアウトのメリット・デメリット

ここ数年のウェブデザインの傾向として、シングルカラムのレイアウトが増えています。これはスマートフォンの普及に深く関係しており、**携帯端末の小さな画面にコンテンツを表示したときに、少しでも表示エリアを大きく確保したい**、という要望からシングルカラムが多く採用されるようになりました。同じ理由から、端末によって表示サイズを変更するレスポンシブデザイン (P.26) が導入されていく中で、**サイドバーのないレイアウトであれば文字サイズの調整などの最低限の変更だけでレスポンシブ化できる**ため、そういった制作時の利便性や合理性といった観点からも、シングルカラムのレイアウトが主流になりつつあります。

■ シングルカラムに不向きなサイトとは
しかしすべての面で優れているかというと、そうではありません。例えばECサイトなどページ数の多いウェブサイトでは、サイドバーがあったほうがサイト内での回遊性が高く使いやすいケースもあります。また、ブログなどの媒体でも、過去記事のアーカイブや人気記事の一覧、広告を表示する場所としてサイドバーは有効活用されています。このように、作成するウェブサイトの目的や、訪れる人にどのような行動をとってほしいのかによって、選択すべきレイアウトパターンは異なり、こういったレイアウトならより使いやすいのかをつねに考えながら、サイトをデザインするのがサイト制作者の仕事です。

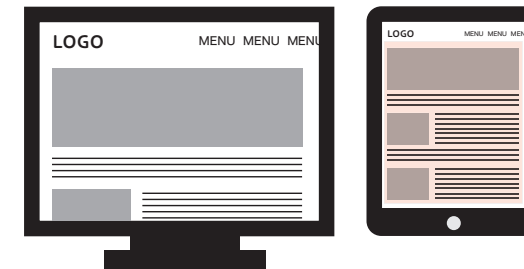
シングルカラムと2カラムページの比較

2カラムレイアウト



コンテンツエリアの幅が狭くなるので、横幅の狭い端末で見たときに、写真や文字が小さくなってしまう

シングルカラムレイアウト



端末の幅いっぱいまでコンテンツエリアとして使えるので、横幅の狭い端末でも見やすく表示できる