Webページデザインのワークフロー

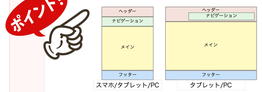
01レイアウトパターンを決める

基本レイアウトパターン

Webページの骨組みとなる基本のレイアウトパターンを、サイトの特徴に合わせて決定します。
レスポンシブWebデザインでは、スマートフォン・タブレット・PCの画面に最適なレイアウトパターンを決めましょう。

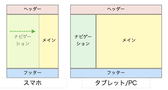
- **上部メニュー型 / 下部メニュー型**

色や画像などでイメージを伝えるのが中心のサイトに向いている。
上部メニューは、左から順に見つけやすく、自由に選びやすい。
下部メニューは、左から順に見つけやすく、順番に選びやすい。



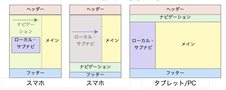
- **左サイドメニュー型 / 右サイドメニュー型**

文章で情報を伝えるのが中心のサイトに向いている。
左サイドメニューは、上から順に見つけやすく、自由に選びやすい。
右サイドメニューは、上から順に見つけやすく、順番に選びやすい。



- **逆L字メニュー型**

ページ数の多い大規模なサイトや、ブログのように更新頻度が高いサイトに向いている。
左サイドをメイン、右サイドをサブとすることが多い
さらに上・下メニューを追加した進化系も考えられる



02コンテンツとライティング

コンテンツ

「中身」「内容」「目次」という意味があり、インターネットやテレビ、紙などの媒体を通して伝えられる情報内容のことを指します。
文章や写真、映像、図、デザインなどを組み合わせて作られた「**情報の内容そのもの**」のことです。
Webページのデザインをするために、まずコンテンツが明確にならないとデザインできません。

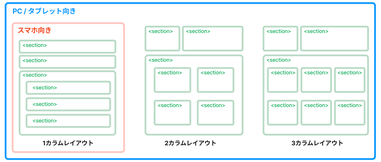
ライティング

各ブロック内のコンテンツを、それぞれ文章で表現する作業のことです。
また、文章を効果的に伝えるために必要となる、写真や動画・イラスト・グラフなどもこのタイミングで考え用意します。

「メイン」で使用する文章には、「**見出し**」となる要約文（タイトル）のほか、長い文章を短くまとめた「**箇条書き（リスト）**」など、**文章構造**をうまく活用し、コンテンツを伝わりやすくする工夫も必要になります。

コンテンツのレイアウト

一般的なレイアウトの種類に、**1カラム・2カラム・3カラム**のレイアウトがあります。
基本レイアウトパターンも、スマートフォンやパソコンによってレイアウトの種類が変わります。



03 ワイヤーフレームの作成

ワイヤーフレームを作成する

レイアウトパターンにコンテンツを配置していく作業です。
特に「メイン」のブロックにコンテンツをレイアウトしていく際は、コンテンツの**優先順位**、文章や画像などの**ボリューム**、**視線誘導**などに**注意**しながらレイアウトします。

サイズ感に注意

ー ここまでは手描きでOK ー

ワイヤーフレーム作成の注意点
- 基本レイアウトパターンで使う四角形には線を追加表示
- ブロック内の文章部分で使う四角形は、表示文字サイズより10px程度大きめで作成
- 四角形は線だらけにならない様に、背景色を変えて使い分ける

ワイヤーフレーム 第一段階

1. コンテンツの優先順位を考える
2. 文章のボリュームや画像のサイズを考え、配置場所を割り当てる

ワイヤーフレーム 第二段階

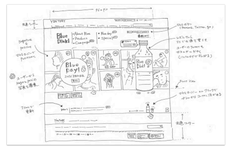
3. Adobe Photoshop / Illustrator / XD、またはFigmaで、幅1280px（スマホは幅320px）のアートボード（Figmaはフレーム）を作成
4. アートボードを選択し、レイアウトグリッドを設定
（例：カラムは12、マージンは138px、ガターは20px）
5. 1画面の高さ（750〜800px程度）の一覧性を意識しながら、四角形ツールを使って基本レイアウトパターンの5つのブロックを配置
6. 各ブロック内に表示するコンテンツが複数ある場合、四角形ツールを使って配置
7. コンテンツ内の文章を、四角形ツールを使って配置し、その上にテキストツールで文章を入力
（標準文字サイズは16pxで、文章はダミーでいい）
8. コンテンツ内の画像部分も、四角形ツールを使ってサイズや周囲の余白を調整しながら配置

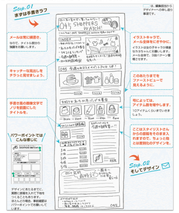
リアルに丁寧に

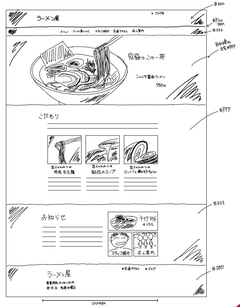
04 ラフデザインとデザインカンプの作成

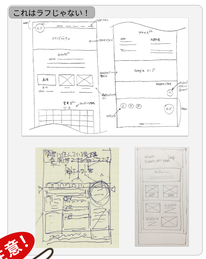
ラフデザイン

ワイヤーフレームを元に、**より具体的で詳細なデザインを手書き**で作成します。
ラフデザインは、そのままデザイナーへの指示や、クライアントにデザインイメージを伝える時にも使用するので、**具体的に・詳細に・丁寧に作成**します。









グラフィック化

デザインカンプ

ラフデザインを忠実に再現したデザインの見本となるデザインカンプです。
Adobe Photoshop / Illustrator /XD、またはFigmaを使って作成します。

デザインカンプはモックアップとも呼ばれ、XDやFigmaで作成した場合はプロトタイプと呼ばれます。
プロトタイプのページを複数作成して、プロトタイピングという作業を行うことで、操作性のチェックをすることもあります。

1. アートボード（フレーム）のサイズは、デバイスの最小画面サイズで作成（PCは1280×800px、スマホは320×568px）
2. レイアウトグリッドを設定
（例：カラムは12、マージンは138px、ガターは20px）
3. 文章を入力する（文字サイズ16px）
4. 画像を挿入
5. ワイヤーフレームとラフデザインを元にレイアウト
6. 文章や画像のサイズ調整
7. 色や装飾を追加し完成

カンプ作成の注意点
- 標準文字サイズは16〜18px、行の高さは180%程度
- 本文は太字にしない
- コンテンツの上下間は60px以上は離す
- 勝手に余白ができないように、あえて余白を作る
- 整列を守り、サイズ感（バランス）を整える...

05 コーディング前の準備

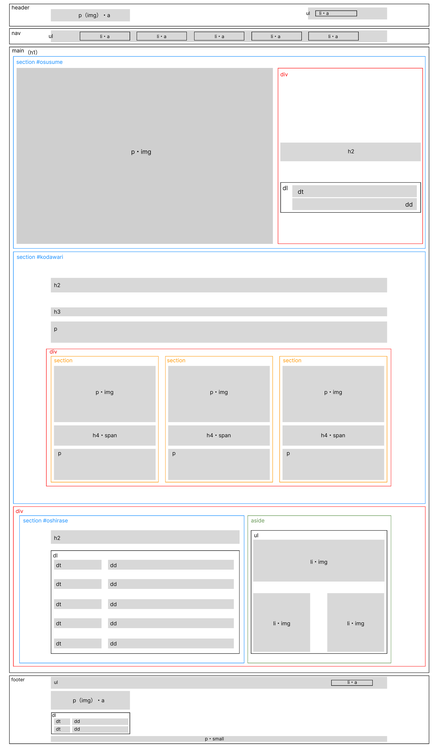
コーディング基本

HTML / CSSでWebページを作成する作業をコーディングと呼びます。
作業の流れは、テキストファイルをHTML文書ファイルへ書き換え、CSSファイルを作成しデザインを加えて完成といった流れになります。
ワイヤーフレームを作成する段階で、作成するページの文章を作成しておくことで、作業を効率良く進めることができます。

ボックスの構造化

HTML / CSS のための設計書、ボックスの構造化。
ワイヤーフレームとカンプをもとに、ページの構造や文章に適切なタグを関連付けた図を作成します。

divやid名・クラス名などの整理にも役立ち、マージンやパディングなどの使い方、タグの親子関係も理解しやすくなる。



ページに表示する文章の作成

ワイヤーフレームと一緒に作業してもいい

ページを作成する際は、必ずコンテンツが先に決定しています。
コンテンツが決まっていれば、文章を作成することができます。
文章が作成できれば、ワイヤーフレーム（デザイン）の精度とクォリティが上がります。
ワイヤーフレームの精度が上がれば、HTMLの文書構造が作りやすくなります。
デザインのクォリティが上がれば、CSSでの再現性が上がります。

『Webデザイン』は、伝えたい内容を伝わりやすくするために『デザイン』します。

06 コーディングの流れ

コーディングワークフロー

最も多いコーディング時のミスがタグの閉じ忘れです。ミスにどの時点で気付けるかによって作業時間に影響してきます。
このような無駄・非効率を無くすためにも、無意識に作業ができるまでは次の「流れ」に沿ってコーディングを進めることをお勧めします。

コーディングの流れを完全に覚え身に付けることで、自分がよくするミスの癖を知ることになり、作業スピードも効率も上げることができます。

1. **HTMLファイルの作成**
2. **viewportメタタグを記述**
<meta name="viewport" content="width=device-width,initial-scale=1">
viewportメタタグはレスポンシブデザインに必要な記述
デバイスごとの画面サイズに応じて最適な表示に変換してくれる
3. **<title>のタイトル文を入力し、ファイルを保存**
4. **文章を全て段落<p>で入力**
画像<img>を表示する部分はaltテキストを入力しておく
5. **段落<p>文章の内容に合わせて適切なタグに変換**
見出し→リスト→画像→テーブル→フォーム→リンク
6. **ページの全体の文書構造をまとめる**
<header> , <nav> , <main> , <footer>
7. **セクショニングコンテンツで文章をまとめる**
<article> , <aside> , <section>
8. **レイアウト用のボックス図を作成し、必要な箇所を でまとめる**
（**id名の入力**→articleが複数存在する時は必須）
9. **リセットCSSを参照する**
10. **CSSファイルを作成・保存し、ワイヤーフレームやボックスの構造化を元にレイアウトのみ完成させる**
主にwidth・margin・display（flex/grid系）・float・clear・overflowなど
11. **各ボックスのスタイリングとレイアウトの微調整を行う**
（headerから順番に各ボックス内のデザインを完成させる）