

# Technical-16<sup>th</sup> サイト構築-4

---

## テンプレート作成

テーブルくむかCSSを使うか

文章構造のキープ

ファイル名のルール

使用フォントのルール

フレームのルール

ロゴ規定など

## テストサーバーでのユーザビリティ・テスト

チーム内でのテスト

第三者によるチェック

テスト結果をフィードバック・修正

再テスト

クライアントチェック

## テンプレート作成

---

パーツデザインでの確認を終えたら、次はそのデザインを実際にhtmlにして機能性や見た目のバランスなどをチェックする。この際には、各パターンのページ1ページずつの作業にとどめる。実際のhtmlを量産する段階では、ここで作られたhtmlファイルをテンプレートとして作業していくことになる。実際にhtmlにするにあたって注意する点を以下に確認する。

### テーブルで組むかCSSを使うか

W3CはCSSでのレイアウトを推奨しており、ブラウザも対応を推し進めているが、想定ユーザーに対して適していない場合は、テーブルによるレイアウトの採用も考えなければならない。

### 文章構造のキープ

読み上げソフトなどに対して、CSSを使えば構造の確保は比較的容易に実現できるが、テーブルレイアウトを行うときにもできるかぎり文章構造は正しくしておきたい。この点について留意するならば、htmlは上から読まれていくという特徴をふまえ、以下の順で作業を進めると比較的効率がよい。

- 1 ページ内に置くコンテンツを書き出す
- 2 上から読んで矛盾の無いように並べ直す
- 3-a CSSを使ってレイアウトする
- 3-b テーブルを利用したレイアウト

ひとつのテーブルは、それがすべて読み込まれなければ表示されない点に注意。

### ファイル名のルール

この時点でまだファイル名のルールが確立していない場合は、決定しておく。

- ex) top\_title  
nav\_button\_home  
profile\_title

## 使用フォントのルール

ボタンなど、画像に使う文字がある場合は、使用フォントを統一する。(カテゴリーごとに大きく変える方針の場合はこの限りではない)

## フレームのルール

フレームを利用する場合は、ウインドウネームや、どの場合にどのターゲットを使うかなどを決めておくこと。

ex) 「サイズ表は別画面を出す」等

## ロゴ規定など

企業サイトを作る場合、ロゴの扱いには注意すること。柄の上に置かない、色の規定、文字周りの間隔など、詳細に決められている場合がある。

## テストサーバーでのユーザビリティ・テスト

---

各パターンのhtmlが完成したら、テストサーバーにアップして実際に使い勝手をテストする。ユーザビリティについて得た知識をもとに、客観的にチェックする必要がある。

### チーム内でのテスト

アップしたものは、まず制作に携わったチーム内で正しく動作するかどうかのチェックを行う。この際ユーザビリティについても気づいた点があれば、メモしておくこと。

### 第三者によるチェック

第三者にアップしたものを操作してもらおう。この際、サイトの目的とする問題を用意し、その問題をクリアする様子を観察するとよい。

ex) 「砂糖を使っていないライ麦の食パンを探して注文してください。」

被験者の自由な感想もきいておくこと。

少なくとも2~3人の被験者によるテストを行うのが望ましい。

### テスト結果をフィードバック・修正

テストによって得られた結果をフィードバック、修正を行う。この時点であまりに大きな変更が必要になった場合は、再度ページプランニングに戻って検討しなおさなければならない。

### 再テスト

前回とは違う第三者による再テストを行う。必要であればフィードバック 修正。

### クライアントチェック

クライアントのチェックを依頼する。