

# 情報デザイン応用演習I 11. ポートフォリオ作成 II

# 目次

- 初めに
- ポートフォリオサイト
- Github Pagesとは?
  - Git
  - Github
  - Github Pages
- HP制作 ワークフロー
  - ワークフロー
  - 環境
  - やってみよう

# 初めに

## 前回の内容

- Git
- Github
- Github Pages

をメインに行ないました。

## 今後の予定

前々回決定したように

ポートフォリオ作成

を最終課題として、それに対して今回入れて5回課題に取り組んでいこうと思います。

企画から公開までを全てこなす課題となります。

# ポートフォリオサイト

## 前提条件

- Github Pagesとして公開すること
- レスポンシブ対応であること

# HP制作 ワークフロー

## AI Tool紹介

- [Magic Todo](#)

を利用して、ポートフォリオサイトに必要なステップを洗い出してみましょう。



## この後のページ

ほぼ前回のおさらい

## ワークフロー

通常以下のような流れとなります。

1. ヒアリング
2. 企画立案・サイト設計
3. コンテンツ制作
4. Webサイトデザイン
5. コーディング
6. 校正・品質チェック
7. 確認・チューニング
8. サイト公開

## 環境

通常では

1. 開発・テスト環境
2. ステージング環境
3. プロダクション(本番)環境

と使い分けます。

1. 個人のPC
2. 関係者がアクセスできるサーバ
3. 不特定多数がアクセスできるサーバ

ですが、今回2番目は使いません。

2のところで、テストなどを行います。

## ヒアリング

クライアントがいる場合、要望などを聞きます。

また、この場合、現状サイトや競合サイトの分析も行います。

今回は、ポートフォリオサイトとしてどんなものがあるか、分析調査してみましょう。

## 企画立案・サイト設計

分析調査を踏まえたうえで、ユーザに効果的な

- Webサイトの企画
- サイトマップ

を制作します。

## コンテンツ制作

コンテンツに必要な

- コピー
- 文章
- 画像

などを制作します。

今回は、ポートフォリオサイトのため、これまでに制作し、掲載したいコンテンツをWebで掲載できるように準備しましょう。

例えば、psd, ai, movファイルでは掲載できません。PDF,PNG,mp4形式に変換しましょう。

## IFRAME

HTMLではIFRAMEというタグがあります。

Webページ内に別のWebページや画像、動画などのコンテンツを読み込んで表示することができます。

ってことは、前半にやったJavaScriptの作品もこれで公開できるはずです。

- [iframeとは？基本的な仕組みと使い方・使用例を解説](#)

## Webサイトデザイン

情報を整理し、ユーザビリティを考慮したWebページをデザインします。

一から制作するのは大変ですので、XDのテンプレートを利用してもOKです。レスポンシブ対応なことにも留意しましょう。

- [Free XD responsive portfolio template](#)

こんなの使うと作りやすいかもです。自分でも探してみましょう。「XD ポートフォリオ テンプレート」等

- トップページ
- 各作品ページ
- 自己紹介ページ

があれば、最低限足りると思います。

後述する**モバイルファースト**も意識しましょう。



## コーディング

最近では**モバイルファースト**でレスポンシブ対応することが多いようです。(Googleさんの意向も大きいようですが)

- **モバイルファーストデザインとはスマホ版から作ればOK？手順やCSS解説つき**

```
@media screen and (min-width:768px){  
  /*タブレット用として画面幅を768pxまでに設定*/  
}  
@media screen and (min-width:1024px){  
  /*PC用として画面幅を1024pxまでに設定*/  
}
```

で追記していきましょう。最悪、モバイルとPC版でOKとします。

- **参考：レスポンシブ対応のレイアウトを実装する最新テクニックを解説、モバイルファーストとデスクトップファーストの現状**

## コーディングの方法

- 全体的な構造(ヘッダー・フッター・サイドバー・コンテンツ)等を実装する
- 各エリアを個別に実装する。
- レスポンシブ化する。

## ハンバーガーメニュー

レスポンス対応時にほとんどは、レイアウト変更で対応できるかと思いますが、それに対応できないのがハンバーガーメニューとなります。

この辺参考にしてみましょう。

- **【コピペで簡単】ハンバーガーメニューをCSSだけで作る方法**

## 校正・品質チェック

様々な環境で表示エラーや動作の不具合がないかチェックしましょう。

- ResponsivelyApp

もうまく使っていきましょう。

## 確認・チューニング

クライアントにチェックをうけ、直していきます。

今回は、なくてもいいですかね。

## サイト公開

Githubで公開できるようにして終了です。

今回はデザインを終えた状態で、

1. 新たにGithubで「portfolio」というリポジトリを作成
2. VSCでクローン
3. index.htmlを作成し、公開されることを確認
4. 制作に取り組む

とすると無駄がないかと思います。

## 就活で利用する場合

ソースを見れば、採用担当者はHTMLやJavaScriptの技術がどのくらいするか判断できることを意味します。

```
<!--  
    HTMLのコメントアウト  
-->  
// JavaScriptの1行のコメントアウト  
/*  
    JavaScriptの複数行のコメントアウト  
*/
```

等も気にすると、評価する人にアピールもできますね。

## ワークフロー(再掲)

1. ヒアリング
2. 企画立案・サイト設計
3. コンテンツ制作
4. Webサイトデザイン
5. コーディング
6. 校正・品質チェック
7. 確認・チューニング
8. サイト公開



## ガントチャート

ガントチャートとは、プロジェクト管理や生産管理などあらゆる管理工程に使う表です。

残り5回ありますが、ステップがいくつもあるので、いつくらいまでにどう終わるかスケジュールを立ててみましょう。

NotionのTimelineという便利な機能があるのでそれを使ってみましょう。

- [Notionでガントチャートを瞬殺で作る。【プロジェクト管理】](#)

## やってみよう

それでは、まず1-4あたりをやっていきましょう。

今日を入れて、残り5回となります。

トラブったらどんどん質問してください。