



はじめに

「SharePoint Online」に於いて、よく耳にするのは、以下のキーワードかと推測します。

- ●サイトに係わるキーワードとして「**クラシック**とモダン エクスペリエンス」、「モダンサイト」「クラシックサイト」
- ●サイトを表示するデフォルト画面としてモダンサイトは「サイトページ」、クラシックサイトは「WikiPage」
- ●「Power PlatForm」、「Teams」、「Office365グループ」、「ハブサイト」

「働き方改革」も「クラウド環境へ管理データの一元化」「メールレス化」「ペーパレス化」の対応が加速されています。

その世の中に流れに合わせて、2年前頃より、従来の「レガシーシステム」を「Microsoft365(「SharePoint Online)」へ移行 検討された組織は、基本ポリシーは「モダンなエクスペリエンス(User Interface)」で可能な限りサイト構築を推奨し、「モダンサイト」 で「モダンページ」を採用されています。

その理由としては、マイクロソフトが発信している、以下の情報による物と推測します。

- ●ページの設定・編集は利用可能な機能・内容が少ないため操作も容易
- ●リストやライブラリのアイテム数のしきい値が「5,000件」から「20,000件」に拡大
- ●携帯端末「Padやスマートホーン」で観た画面はPC閲覧時と同じでレスポンシブ
- ●「PowerPlatForm」は可能な限りローコードで実現できるツールで「モダンUI」を採用、「Teams」も「モダンUI」を採用
- ユーザー側も「クラシックなエクスペリエンス(UI)」が数年後は使えなくなる?という不安もあり、ポリシーを決められたのかと推測します。

しかしながら、サイト展開作業ですぐに直面するのは、「設定が容易は=期待するレイアウトや機能が実現不可能」という現実です。

- ●サイトテンプレート、リストテンプレート機能が使えない
- ●ページレイアウトは期待通りのレイアウトにならない
- ●ページ編集機能を簡単にパーツ化できない

- ⇒ サイト展開に**膨大な時間(費用)**が掛かる ⇒ サイトページで使用できるパーツで可能なレイアウト編集は難しい
- ⇒ 従来のWebパーツやそれを継承した個別Webパーツは使えない
- ●従来、実現できた機能が「PowerApps」でないと実現できない ⇒ コンテンツの作成に膨大な時間(費用)が掛かる

上記の問題を解決するため、一部のユーザーは「**モダンサイト**」に対して「カスタムスクリプト」を有効にする「PowerShell」を 1行実行し、設定を変更して「クラシックサイト」の有用な機能を有効にするユーザーも多く見かけられます。 また、上記の1行の「PowerShell」をしてクラシックUI「WikiPage」を使わずにモダンUI「サイトページ」で何とかしたいという事例として、 「SharePoint Framework」での対応を検討したいユーザーにその対応手順をご紹介します。





「SharePoint Online」が「クラシックUI」から「モダンUI」が主流になってくるにあたり、どうしても「クラシックUI」で実施していた 機能が再現したいというご要望を頂くことが多くなってきました。

そこで、クラシック時代に活躍したWebパーツである「**スクリプトエディタ**」を例にして、モダンUIで利用する迄の手順を紹介します。 「クラシックUI」で一番使用されている?Webパーツ「スクリプトエディタ」を SharePoint Online 環境に反映するまでの手順は以下 の通りです。

1. スクリプトエディタの概要

- ●クラシックUIではスクリプトエディタというWebパーツが 標準機能として利用できました。
- ●このスクリプトエディタはJavaScriptを記入でき、 SharePoint Onlineの画面をスクリプトベースで カスタイマイズ可能でした。
- モダンUIになり、このスクリプトエディタ機能は廃止 されてしまっています。
- ●代替機能としては SharePoint Framework という開発手法で、基本、TypeScript で カスタマイズという手順になります。
- ●残念ながら従来のスクリプトエディタでの JavaScript によるカスタマイズよりも、SharePoint Framework によるカスタマイズ(開発)は、かなり難易度が上がって います。

	SharePoint		𝒫 このサイトを検索		0		R
参照	ページ テキストの書	式設定 挿入		◎共有 会习	770-	日 保存	
表表表	画像 ビデオと Offic オーディオ・ ビ メディア	20 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)					
カテゴリ	ע ע	パーツ		パーツについて			
■ P4 ■ ビ: ■ フィ ■ フィ ■ ブ: ■ グ: ■ 検 Web /	キュメスト セット シネス データ パルター オーム ログ ディアおよびコンテンツ 滚 環型コンテンツ パーツのアップロード 、	■Silverlight の Web パージ (ヨイメージ ビューアー コンテンジ エディター ロサイトアの体系の想給 コンテンブ エディター ロサイトアの体系の想給 コンティアの Web パーツ コンディアの Web パーツ コンディアの X5ィド ショー	σ We	スクリプト エディター 作成者に HTML スニペットまたはスクリプトの挿入を許可します。 パーツを追加します: 199チ コンテンツ >			
				通知		キャンセ	ĸ

・・・今回スクリプトエディタを利用するサイト

2. 事前に必要な環境

●SharePoint Online のサイト

PS> gulp trust-dev-cert

- SharePoint Online のアプリカタログサイト・・・ スクリプトエディタアプリを登録するサイト ※こちらはSharePointの管理者でないとサイト作成ができません。
- SharePoint Framework の開発環境 ・・・ SharePoint Framework をビルドできる環境 ※以下のURLの記載内容を参考にSharePoint Framework 開発環境の設定してください。
- $\Rightarrow https://docs.microsoft.com/ja-jp/sharepoint/dev/spfx/set-up-your-development-environment$

★SharePoint Framework の開発環境 ・・・SharePoint Framework をビルドできる環境の準備【操作手順】

① Node.js. のインストール ⇒ https://www.nodejs.org/ より [node-v12.14.0-x64.msi] をダウンロードし、 「msi」をクリック ※最新版「node-v18.12.1-x64.msi」では「 npm i」でエラーとなる ② コード エディターのインストール \Rightarrow https://code.visualstudio.com/docs/?dv=win \pounds 0 [VSCodeUserSetup-x64-1.73.1.exe] をダウンロードし、[exe]をクリック ③ 開発ツールチェーンの前提条件をインストールする 1つの行に、次の3つのツールのすべてをインストールできます。 PS > npm install qulp-cli vo @microsoft/generator-sharepoint --global ft Comporation, All rights reserved ④ Gulp のインストール フォームの PowerShell をお試しください https://aka.ms/ps PS > npm install gulp-cli --global ⑤ Yeoman のインストール PS > npm install yo --global ⑥ 最新の Web ブラウザーをインストールする ⑦開発者用自己署名証明書の信頼 「3.(4)パッケージファイルの依存関係の反映コマンド実行」完了後、 プロジェクトのルートフォルダー内で以下のコマンドを実行します。











3. スクリプトエディタのビルドから反映まで

(1) スクリプトエディタのサンプルコード入手

GitHub の SP-Framework のサンプルページよりサンプルコードを ダウンロードします。

https://github.com/pnp/sp-dev-fx-webparts すべてのサンプルが含まれているので、11/10時点で圧縮状態でも 1.54GBとかなりサイズが大きいです。 sp-dev-fx-webparts-main.zip

(2) スクリプトエディタのサンプルを取り出す

samplesフォルダから「**react-script-editor**」を取り出して作業フォルダ に持っていきます。 作業フォルダは任意の場所(以下のフォルダー)とします。 「C:¥_MicroSoft365¥WebParts¥react-script-editor」

(3) コードの読み取り専用を解除する

任意のフォルダに配置した「react-script-editor」サンプルコードの読み 取り専用を解除します。 ※この後の作業でエラーとならないようにするため。

	(💭 Why GitHub? 🗸 Team Enterp	orise Explore∨ Marketplace P	ricing \vee	Searc
📮 pnp	/ sp-dev-fx-webparts Public				
<> Cod	de 💿 Issues 45 👫 Pull requ	ests 🖓 Discussions 🕞 Actions	Projects 1 🛛 Wiki 🔅) Security 🗠 Insights	
		🏌 main 👻 🥲 6 branches 🗞 155	5 tags	Go	to file Code -
		hugoabernier Added zip file		E Clone	0
		github	Updating wrong author messag	le a https://github.com/pnp/sp-dev-f	x-webparts.
		samples	Added zip file	Use Git or checkout with SVN using the w	eb URL.
		tutorials	Web parts is written with two we	orc	
		 tutorials .gitignore 	Web parts is written with two w	orc 단 Open with GitHub Desktop	
		tutorials .gitignore CODE_OF_CONDUCT.md	Web parts is written with two we Added issue action Update CODE_OF_CONDUCT.me	d Download ZIP	
		tutorials .gitignore CODE_OF_CONDUCT.md	Web parts is written with two w Added issue action Update CODE_OF_CONDUCT.m	d Download ZIP	
∎ → PC	ン ダウンロード > sp-dev-fx	tutorials	Web parts is written with two w Added issue action Update CODE_OF_CONDUCT.m	C Dopen with GitHub Desktop	samplesの検索
→ PC	: > ダウンロード > sp-dev-fx 名前	turorials gitignore conc_of_conDuct.md -webparts-main > samples	Web parts is written with two w Added issue action Update CODE_OF_CONDUCT.m 更新日時	ef (2) Open with GitHub Desktop d Download ZIP で ひ の 2 種類	samplesの検索 サイズ
→ PC	: > ダウンロード > sp-dev-fx 名前		Web parts is written with two w Added issue action Update CODE_OF_CONDUCT.m 更新日時 2021/12/07 4:09	erec C Open with GitHub Desktop Download ZIP ・ ひ の : 種類 ファイルフォルダー	samplesの検索 サイズ
> PC	: > ダウンロード > sp-dev-fx 名前 react-remote-event react-restaurant-men		Web parts is written with two w Added issue action Update CODE_OF_CONDUCT.m 更新日時 2021/12/07 4:09 2021/12/07 4:09	co C Open with GitHub Desktop Download ZiP を を を ク ・ を ・ を ・ ・ を ・ ・ を ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	samplesの検索 サイズ
। → PC	: > ダウンロード > sp-dev-fx 名前 		Web parts is written with two wr Added issue action Update CODE_OF_CONDUCT.mr 更新日時 2021/12/07 4:09 2021/12/07 4:09 2021/12/07 4:09	COPEN with GitHub Desktop Download ZiP 20 00 20 確据 77イルフォルダー ファイルフォルダー ファイルフォルダー ファイルフォルダー	samplesの検索 サイズ
ې PC ۲ ۲	: > ダウンロード > sp-dev-fx 名前 react-remote-event-r react-restaurant-men react-rss-reader react-rxjs-event-emit		Web parts is written with two wr Added issue action Update CODE_OF_CONDUCT.mr 更新日時 2021/12/07 4:09 2021/12/07 4:09 2021/12/07 4:09 2021/12/07 4:09	CC ② Open with GitHub Desktop Download ZiP ▼ ひ の 2 催焼類 ファイル フォルダー	samplesの検索 サイズ
→ PC ス スト	: > ダウンロード > sp-dev-fx 名前 react-remote-event-r react-restaurant-men react-rss-reader react-rxjs-event-emit react-save-attachmen	turorials gitignore concernment concernment samples concernment concernment concernment concernment	Web parts is written with two wr Added issue action Update CODE_OF_CONDUCT.mr 更新日時 2021/12/07 4:09 2021/12/07 4:09 2021/12/07 4:09 2021/12/07 4:09	CODE With GitHub Desktop Download ZiP 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20	samplesの検索 サイズ







(4) パッケージファイルの依存関係の反映コマンド実行

PowerShellにて「react-script-editor」フォルダを開き、 「react-script-editor」に必要な依存関係のファイルを取得する 以下のコマンドを実行します。

※結構時間がかかります。

警告は出ますが、エラーが出なければ、この後の作業が続行可能です。

PS C:¥_MicroSoft365¥WebParts¥react-script-editor> npm i

※このコマンドが正常終了後は「2.⑦開発者用自己署名証明書の信頼 の実行が可能です。

(5) ビルドコマンドの実行

「react-script-editor」をビルドする下記コマンドを実行します。

PS C:¥_MicroSoft365¥WebParts¥react-script-editor > gulp bundle --ship

(6) ソリューションのパッケージ化コマンドの実行

「react-script-editor」をパッケージする下記コマンドを実行します。

PS C:¥_MicroSoft365¥WebParts¥react-script-editor> gulp package-solution --ship

PS C:_MicroSoft365\WebParts\react-script-editor> gulp package-solution --ship Build target: SHIP

[12:43:58] Using gulpfile C:_MicroSoft365\WebParts\react-script-editor\gulpfile.js

[12:43:58] Starting 'package-solution'...

[12:43:58] Starting gulp

[12:43:58] Starting subtask 'package-solution'...



[12:43:59]	[package-solution] Done!
[12:43:59]	[package-solution]
[12:43:59]	[package-solution] ALL DONE!
[12:43:59]	[package-solution]
[12:43:59]	Finished subtask 'package-solution' after 339 ms
[12:43:59]	Finished 'package-solution' after 342 ms
[12:43:59]	
[12:44:00]	Project pnp-script-editor version:1.0.20
12:44:00	Build tools version:3.17.17
[12:44:00]	Node version:v12.14.0
12:44:00	Total duration:3.67 s





(7) 出来上がったパッケージファイルを確認する

「react-script-editor」のフォルダに「sharepoint¥solution」というフォルダが出来上がるので、そのフォルダに入っている 「pnp-script-editor.sppkg」を取得します。



(8) アプリカタログサイトへ登録する

アプリカタログサイトにて、SharePointアプリに「pnp-script-editor.sppkg」をアップロードします。

SharePoint		T 2 1 0 ? ()
照 ファイル ライブラリ		○ 共有 ☆ フォロ- □
5	*-♪ SharePoint 用アプリ ₀	このサイト の検索 ・ 오
arePoint ストア 丘使った項目 arePoint 用アプリ	 ● 新規作成 ▲ アップロード ジ 同期 ● 共有 その他 マ すべてのアプリ お勧めのアプリ 使用できないアプリ ・・・ ファイルの検索 ク 	
377U ₹	✓ ○ タイトル 名前 アブリのパージョン 編集 製品 ID メタデータ言語 既定のメタデータ言語 更 フェイルをアップロ	見新日時 有効 有効なアプリパッケージ 身
ンテンツ	ファイルをアッフロ	コートするには、ここにトリックします





(9) アプリカタログサイトへ展開する

アップロード後に展開画面が表示されるので、展開ボタンを押します。

III SharePoint	A		
81 2HK 31799	Cher 立 ShareDoint 田文介山。	○ 其 参照 ファイル 5イブジリ	Сана Флан и сана сана сана сана сана сана сана
Sharehard 31-7 Bittlest 98	5/10/07/10/07/07/00/000075	- S	**A SharePoint 用アプリ。
SharePoint (1777) Office (1777) 77/18:8 9(1):00929	Modern Script Extor web part by mikelevenion %15%	SharePoint ストア 単近使と現日 SharePoint 用プリ の形に用プリ アプラネ サイトコンテング	*********************************
	NR HOCK		Nodern Script pnp 1.0.20.0 ↓ (1425175F- 日本語・ はい 数分前 はい はい しいえ 13-は約2ません。 Editor web script- part by editor 8FC4+ milaelovension ■ DD1497(91294)

(10) サイトへの追加

スクリプトエディタを利用したいサイトで「アプリ」クリックで表示される「追加できるアプリ」で作成した「Modern Script Editor Web part」を選択し [追加]をクリックすると、「追加しました。」と表示されると追加成功です。

	SharePoint	☆ ♪ 検索		= @ ? (%)	III SharePoint		
ଭ	モダンUI検証サイト 02.モタ	シン検証用サイト > ごみ箱	福集		🚯 SharePoint アブ	ע׳	
•	+ 新規 ~		論 サイトの利	J用状況 🖒 サイトワークフロー 🎯 サイト設定 💼 ごみ箱(O) 🎤	← モダン切機証サイトに戻る		
0 6	リスト ページ ト ドキュメントライブラリ アプリ				マイアプリ ミ フィルター 追加できるアプリ これらは、相称で許可されている5%	Q. アプリ名または発行見能に使用 ち 位く国スの基本 和しい様 ~ sarefbint ストアまたはカスタム アグリです。カスタム り	Modern Script Editor web part by mikaelsvenson
÷	サブサイト	種類	アイテム 更新	新日時	スト、ドキュメント ライブラリ、予	定表などの組み込みアプリは従来の環境にあります。	自分の組織
	 10D.水認の要求模証 11D.簡易承認機能検証 	ドキュメントライブラリ	0 2022	2/11/03 17:49 2/11/03 17:49	Modern Script Editor web part by S Devolute 7	SharePoint ストアで その他のアプリを	追加しました
	 12D.承認の要求(CUI:承認関連 12D.承認の要求(承認関連列有 	ドキュメント ライブラリ ドキュメント ライブラリ	0 2022	2/11/03 17:49	50.00 -	→	





(11) サイトへのスクリプト有効化コマンドを実行する

スクリプトエディタを利用したいサイトに対してスクリプトを有効化する必要があるため、有効化するスクリプトを実行する。

サイト設定変更コマンド

1 Connect-SPOService -Url https://contoso-admin.sharepoint.com -Credential admin@contoso.com

2 Set-SPOSite -DenyAddAndCustomizePages 0

※管理シェルが無い場合はを下記をダウンロードしてインストールしてください。 ※ SharePoint Online Management Shell https://www.microsoft.com/ja-jp/download/details.aspx?id=35588

※注意

SharePoint管理者権限が必要です。

SharePoint Online ManagementShellモジュールとSharePointClient Components SDKの間には、両方が同じコンピューターにいる場合 にモジュールの読み込みに失敗するという既知の問題があります。 この問題が発生した場合は、SharePointクライアントコンポーネントSDKをアンインストールしてください。

(12) サイトにWebパーツを追加する

サイトにて、「Modern Script Editor Web part」パーツが追加可能となりますので、追加すると下記のようにスクリプトエディタWebパーツが表示 されますので、任意のスクリプト「スペニット」を追加してください。

SharePoint ド書きとして保	ア このサイトを検索 ア スに戻す ~ 〇 変更の結果 ◎ ページの評価 Ⅲ 分析	S 0 ? 3	ジ Coサイトを検索 ※ F書きとして保存 ジ 元に戻す /> 二。 変更の破壊 ② ページの評判 国 分析	Modern Script Editor × Title to show in edit mode The Modern Script Editor web part!	× Edit HTML Code
		クイックリンク + リンクの追加 でチ 団 100.米部の要求検証 タス 団 110.都局利約報助検証 ご 110.原稿の要求検証 ご 110.原稿の要求検証 ご 110.原稿の要求検証意外的)	The Modern Script Editor web part!	Remove top/bottom padding of web part container	1 1<





サイトページの事例については、ユーザー要望に合わせ、「Edit HTML Code」プロパティへ、従来の「スクリプトエディターWebパーツ」のスペニットに記述していた 「JavaScript」が活用できるかの検証・修正が必要です。



クラシックUIでは、編集者にはコードを意識 させない専用Webパーツ化していましたが、 モダンUIでは、その機能が容易に実現できない ため、コードを意識した作業が必要になります。

おわりに

今回、追加したスクリプトエディタですが、モダンUIにて廃止になっているスクリプトエディタをどうしても利用したいというご要件を頂いた際に選択肢の一つとして対応できる可能性がありますので、ご参考になれば幸いです。

但し、モダンサイトでもクラシックUIの「WikiPage」を使えば上記の作業は不要です。

通常のサイトやコンテンツの設定画面、ドキュメントセット画面など、サイトページ・ビュー一覧、フォーム画面以外は、従来のクラシックUIで提供され画面であり、 SharePointの標準機能を最大限活用したい、ユーザー要望を最大限対応したい場合は、モダンUIには固着せず(囚われず)、従来のクラシックUIで提供 されたSharePoint標準機能を最大限活用した「**可能な限りモダンUI」**をお勧めします。





可能な限り「モダンUI」での対応は、従来の「クラシックUI」と比較し、SharePoint標準で提供される「Webパーツ」が利用できないため、想定されるユーザー要望は殆どはSharePointだけでは実現できません。

① モダンUIのリストやライブラリへは「Webパーツ」を追加不可

・ビューの表示編集は、基本「JSON」設定のみ

→ビューへのアクセス権限設定ができない。 TreeView表示機能の追加などはできない。

・フォームの表示編集や機能追加は「PowerApps」でのみ

→ヘッダー、本文、フッターの表示編集は、基本「JSON」設定のみだが、列ごとのClass指定などによる表示制御は、期待する動作設定が難しい。

→フォームへの表示制御・機能追加は、「PowerApps」での編集作業が必要。

クラシックUIは、「NewForm」「DispForm」「EditForm」ごとに編集か可能だが、モダンUI用の「PowerApps」画面は一つの画面設定で対応は 必要であり、専用コードでの記述が必要なため、専任者での対応を推奨。

② モダンUIのサイトのページに限り「Webパーツ」の追加は許可

SharePoint OnlineのJavaScript Object Model(JSOM)は利用できないため、サイトページに限り、大別して2つの対策案が考えられます。 【参考URL(内田洋行 太田さんのブログ):https://idea.tostring.jp/?p=3986】

◆ SharePoint Framework の利用【本命】

敷居が高いと思われている「SharePoint Framework」(公開されているソースコード有り)の「スクリプト エディタWebパーツ」をパッケージ化して活用 ※参考資料:SharePointを学ぼう_17.【モダンUI】サイトの編集「スクリプトエディタを利用」

◆ サイトページに標準機能の組み合わせで行う方法

クラシックUIの標準機能をサイトページへ「埋め込み Web パーツで表示」する内容であり、サイトページに固着しなくても良い場合は、SharePoint 標準の「WikiPage」の活用をお勧めします。

- ① JavaScript を埋め込んだページを作成する。-「WikiPage」(.aspx)でも代用可能?
- ② SharePoint Online のデータを利用する場合には REST API を利用する

>SharePoint Online のドキュメントライブラリにアップロードした「.aspx」ファイルからJSOMを利用 ⇒ https://idea.tostring.jp/?p=5462 >ページ上への表示に iframe が利用されるため、CSS も自分で作成しておく必要がある。

>例えば、Office UI Fabric JS \Rightarrow https://developer.microsoft.com/en-us/fabric-js

この Office UI Fabric を基に自分で「CSS」のカスタマイズを行い、ページを拡張子「.aspx」として保存して任意のドキュメントライブラリにアップロード。 ③ 埋め込み Web パーツで表示

これを SharePoint Online モダン ページの埋め込み Web パーツで呼出しは、SharePoint Online のドキュメントライブラリに保存されている ページを直接参照できる URL を指定する。

これで、ページ上に先ほど作成したページが埋め込み Web パーツによって iframe のコンテンツとして表示されます。 表示される高さなどを指定したい場合には、埋め込み Web パーツの設定に iframe タグを記述することもできます。

モダンUIサイトページの編集(カスタマイズ)