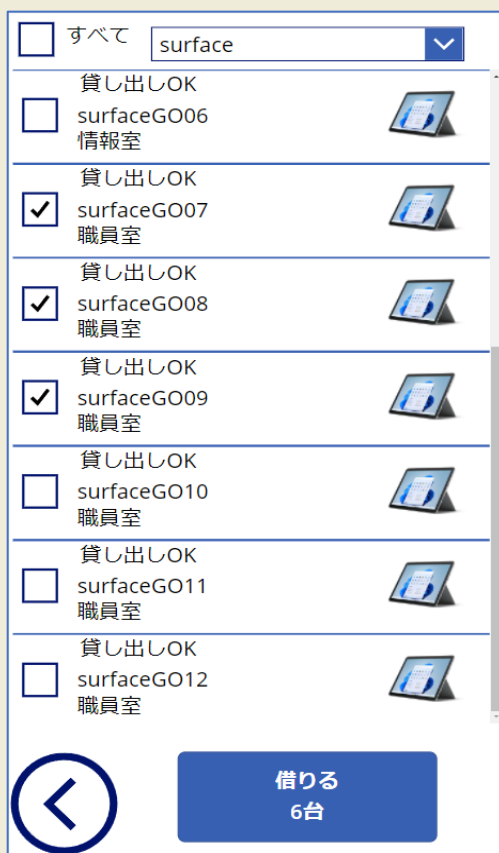
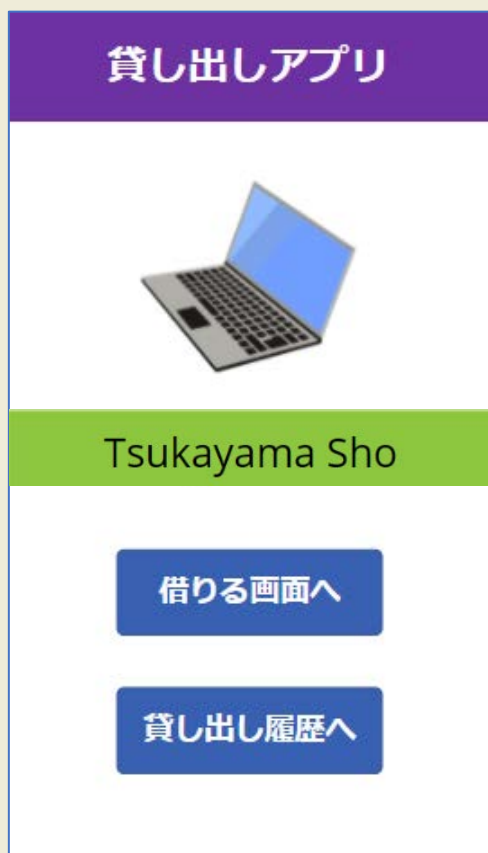


# 貸し出しアプリを作ろう

Microsoft Education Day 2022

Power Apps | 貸し出しアプリ

校務で使えるアプリを  
ローコードで作ってみませんか？



- **PowerApps** は短時間でローコードのアプリを作成できるツールです
- 校内の機器をまとめて管理し、貸し出し状況や貸し出し履歴がすぐに分かるアプリを作成します



MIEE Powerapps ワークショップ企画資料2022

2024年2月改訂【PowerApps入門】貸し出しアプリ手順書Ver3



# 貸し出しアプリ 目次

アプリの構成 P3

アプリの準備 P4~P6

## ① スタート画面 P8~9

- [①-1 画面を増やす](#)
- [①-2 ラベルを追加する](#)
- [①-3 ボタンを追加する](#)

## ② 貸し出し画面 P11~22

- [②-1 SharePointと接続する](#)
- [②-2 貸し出し機器検索ボックスを作成する](#)
- [②-3 貸し出し機器ギャラリーを作成する](#)
- [②-4 ギャラリー内にラベルを追加する](#)
- [②-5 ギャラリー内に枠を追加する](#)
- [②-6 ギャラリー内に画像を追加する](#)
- [②-7 チェックボタンを追加する](#)
- [②-8 すべて チェックボタンを追加する](#)
- [②-9 貸し出し機器検索ボックスに関数を追加する](#)
- [②-10 戻るボタンを追加する](#)
- [②-11 借りるボタンを追加する](#)

## ③ 貸し出し履歴画面 P24~26

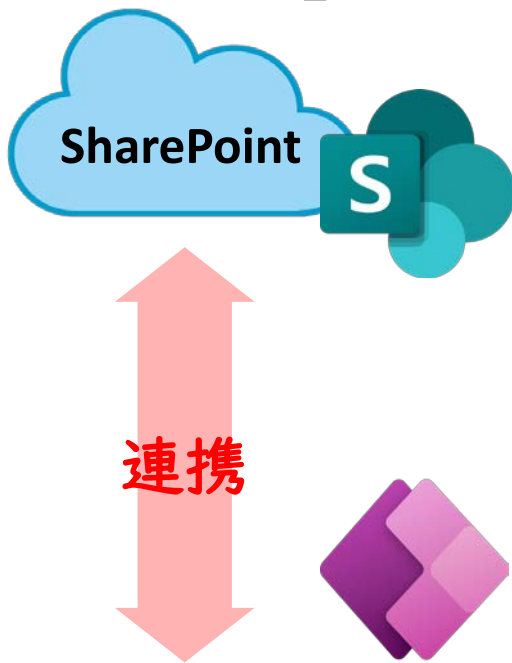
- [③-1 ギャラリーを作成する](#)
- [③-2 ギャラリー内にラベルを追加する](#)
- [③-3 返却ボタンを追加する](#)
- [③-4 体裁を整える](#)

## PowerApps 補足説明

- [◇PowerApps操作画面について](#) P7
- [◇画面の確認・アプリ保存について](#) P10
- [◇各関数について](#) [P15](#)・[P19](#)・[P20](#)・[P23](#)
- [◇アプリ動作確認](#) P27
- [◇アプリに乘せる画像URLの調べ方](#) P28
- [◇PowerAppsアプリをチーム名で共有する方法](#) P29
- [◇PowerAppsアプリをTeams上で利用する方法](#) P30

# 貸し出しアプリ 構成

## 【PowerApps 画面遷移図】



### ①スタート画面



### ②貸し出し画面

<input type="checkbox"/> すべて	surface	
<input type="checkbox"/>	貸し出しOK surfaceGO06 情報室	
<input checked="" type="checkbox"/>	貸し出しOK surfaceGO07 職員室	
<input checked="" type="checkbox"/>	貸し出しOK surfaceGO08 職員室	
<input checked="" type="checkbox"/>	貸し出しOK surfaceGO09 職員室	
<input type="checkbox"/>	貸し出しOK surfaceGO10 職員室	
<input type="checkbox"/>	貸し出しOK surfaceGO11 職員室	
<input type="checkbox"/>	貸し出しOK surfaceGO12 職員室	

◀ 借りる  
6台

- ・ 借りられる機器を確認
- ・ 機器を選択しチェック

### ③貸し出し履歴画面

2022年1月23日	AkiyamaKaori	借用中	surfaceGO01 surfaceGO02	返却
2022年1月23日	AkiyamaKaori	借用中	iPad03 iPad04 iPad05	返却
2022年1月14日	AkiyamaKaori	返却済	surfaceGO06	
2022年1月14日	Tsukayama Sho	返却済	surfaceGO01 surfaceGO02 surfaceGO03 surfaceGO04 surfaceGO05	
2022年1月13日	Tsukayama Sho	返却済	surfaceGO01 surfaceGO02 surfaceGO03	
2022年1月12日	AkiyamaKaori	返却済	surfaceGO10 surfaceGO11	

◀

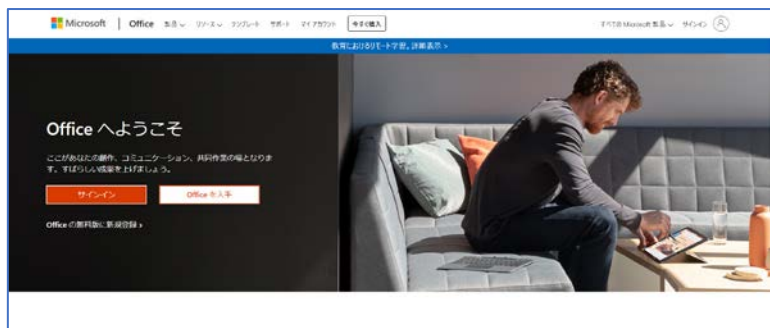
- ・ 借りた機器を確認
- ・ 機器の返却

\*貸し出しデータは、指定のSharePointList内に蓄積される

# 貸し出しアプリ 準備

## 準備 1. Office365にサインイン

- <https://www.office.com/> よりMSアカウントでサインイン



- アプリを作成する前に、アプリで利用するデータソースを用意する

\*データソースExcelファイルは、本手順書と共に「授業・校務素材ポータル」よりダウンロード可

## 準備 2. アプリに接続するデータを用意

### 手順 1

Excelシートをダウンロードする

### Sheet1 「貸し出し機器」シート

id機器名	id機器の種類	id画像	id場所
surfaceG001	surface	https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/情報室	
surfaceG002	surface	https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/情報室	
surfaceG003	surface	https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/情報室	
surfaceG004	surface	https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/情報室	
surfaceG005	surface	https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/情報室	
surfaceG006	surface	https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/情報室	
surfaceG007	surface	https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/職員室	
surfaceG008	surface	https://img-prod-cms-rt-microsoft-com.akamaized.net/cms/api/職員室	

- 本手順書と共にサイトにアップされている「貸し出しアプリに使う関数&データソースリスト」Excelファイルをダウンロードする

### Sheet2 「借用者」シート

sp機器名	sp名前	spメールアドレス	sp借用日	sp貸し出し状態
surfaceG002 surfaceG003	MIEE太郎	miee@edu.jp	2022/01/05	借用中
surfaceG006 surfaceG007	MIEE花子	miee@edu.jp	2022/01/04	返却済
surfaceG001 surfaceG003	MIEE次郎	miee@edu.jp	2022/01/03	借用中

- \*学校で利用する際にはSheet1「貸し出し機器」シートの機器名や画像を校内に導入されている機器に編集して利用する (id画像のURLはP28参照)

### Sheet3 「アプリに使う関数」シート

関数名	プロパティ	機体
ScreenTransition	パーツの名称	機体
OnSelect	対象の画面に移動	Navigator貸し出し機体.SCREENTRANSITION
OnDropDown	ドロップダウンで選ぶ項目	Navigator貸し出し機体.SCREENTRANSITION
OnChange	項目を変更すると起動する関数	Clearレンタル機体List
OnClick	ボタンをクリックすると起動する関数	Filter貸し出し機体情報一覧機体
OnCheck	チェックボックスの状態が変化する	ThiItem機体情報
OnCheckbox	チェックボックスの状態が変化する	Collectレンタル機体List.ThiItem機体

- 編集後は上書き保存をしてExcelを閉じる
- Sheet3「アプリに使う関数」(予備としての関数の一覧表)

# 貸し出しアプリ 準備

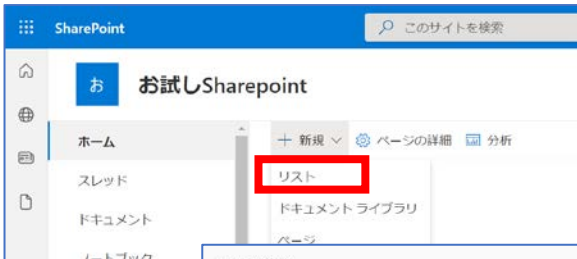
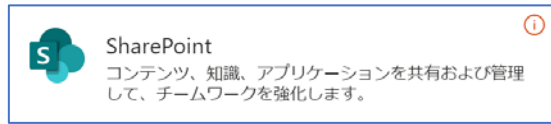
## 手順2

## SharePointにExcelをインポートする

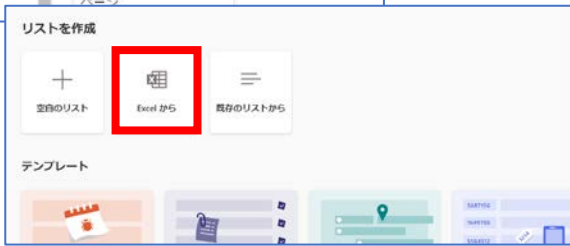


### 【SharePointを開く】

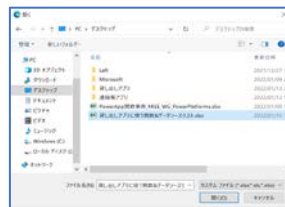
- Office365ホーム画面→左下「すべてのアプリ」よりSharePointを開く



- SharePointホーム画面→アプリのデータを共有したいチームを選択する  
(新たにサイトを作成でも可)



- 「新規」→「リスト」→「Excelから」を選択→「ファイルをアップロード」よりExcelをアップロードする



- 「カスタマイズ」画面で確認
- 基本は「一行テキスト」、文字数が多いものは「複数行テキスト」に設定する  
(貸し出し機器のid画像と借用者のsp機器名は複数行テキストにする)



- 「借用者」と「貸し出し機器」の2回分をアップロードする  
(ホーム→新規→リストより)



データソースにそれぞれの名前を付ける

- SharePointにリストが作成できたら準備2は完了！



# 貸し出しアプリ 準備

## 準備3. Power Appsを開く



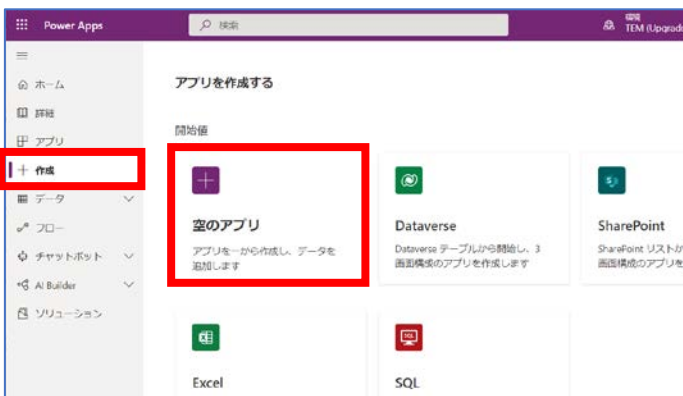
- Office365ホーム画面へ戻る  
左側のアイコンにPowerAppsがあれば選択する



- なければ左下  
「すべてのアプリ」より  
Power Appsを開く

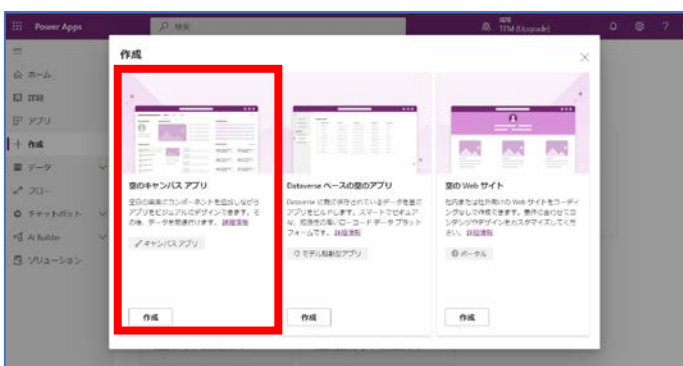


## 準備4. 新しいアプリを作成する



### 【新規アプリを作成】

- Power Appsホーム
- 「+作成」→「空のアプリ」を選択する



- 「空のキャンパスアプリ」の  
「作成」を選択する



### 【アプリ名を入力】

- 作成するアプリの名前を入力
- 今回の貸し出しアプリの場合は、  
「形式」を「電話」にチェック  
(縦長の画面)  
→ 「作成」をクリックする

## 画面操作の基本的な説明

- ・ **キャンバス** : アプリの部品(オブジェクト)を配置
- ・ **プロパティリスト** : Onselect/Text/Visible などの機能の設定
- ・ **数式バー** : 関数を入力する数式バー
- ・ **アプリ作成メニュー** : 画面構成がわかるツリービューやSharePointなどのデータに接続、図などの挿入ができる
- ・ **画面や部品** (ボタン・ラベル) : アプリを構成するオブジェクト一覧
- ・ **部品(オブジェクト)のプロパティ** : 高さや幅、色などの設定

Power Apps | キャンバス

← 戻る | ファイル ホーム 挿入 ビュー アクション 設定 貸し出しアプリ練習 | 🔍 | ↶ ↷ ↻ ▶ 🔍 ?

□ 新しい画面 ▾ |  ラベル ▾ |  テキスト ▾ |  入力 ▾ |  メディア ▾ | AI Builder ▾ | Mixed Reality ▾ ▾

OnSelect = fx /\*貸し出し画面に移動する\*/  
Navigate(貸し出し画面, Fade)

ツリービュー  
画面 コンポーネント

検索

- App
- スタート画面
- Button2
- Button1
- Label2
- Label1
- 貸し出し画面
- 貸し出し履歴画面

貸し出しアプリ

apps57

借りる画面へ

貸し出し履歴へ

Button1

プロパティ 詳細設定 アイデア

テキスト 借りる画面へ

表示モード 編集 ▾

表示 オン

位置 X 151 Y 617

サイズ 幅 338 高さ 124

パディング 上 5 下 5 左 5 右 5

色 A

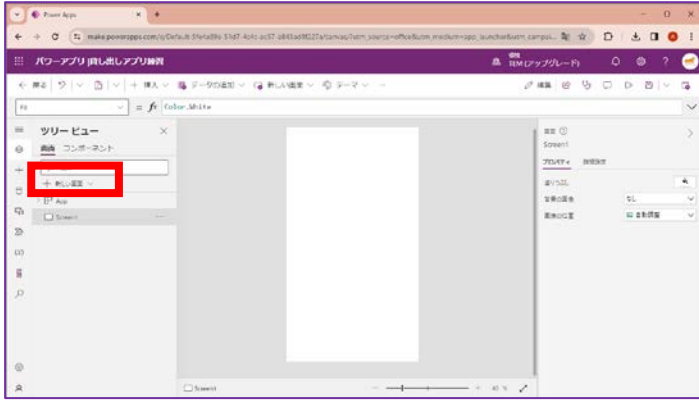
罫線 2

境界半径 10

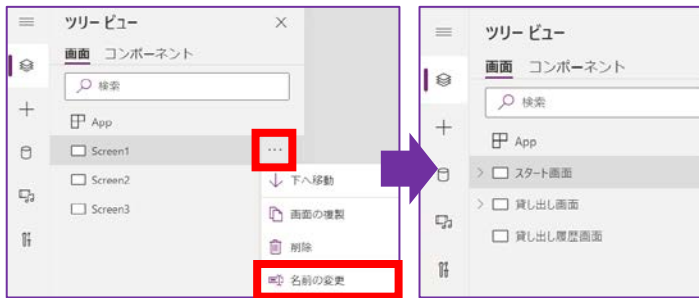
フォント Open Sans ▾

# ①スタート画面

## ①-1. 画面を増やす



- Power Apps新規作成画面
- 「新しい画面」→「空」より画面を3つ増やす



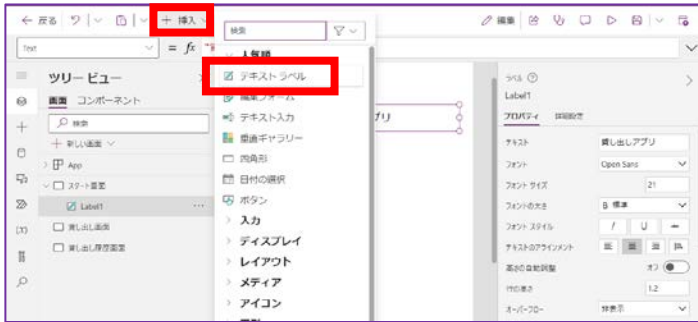
- 「Screen1」三点リーダーより「名前の変更」→各画面の名前に変更する

### 【今回作成する画面】

- スタート画面
- 貸し出し画面
- 貸し出し履歴画面

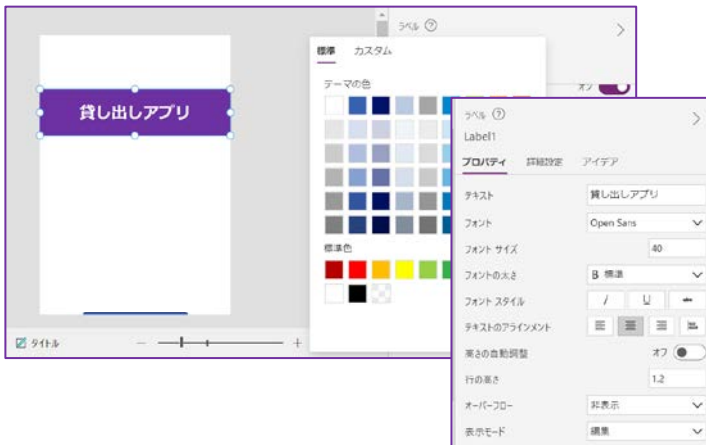
\*直接「Screen」をダブルクリックしても名前の変更は可能

## ①-2. ラベルを追加する



### 【ラベル（タイトル）を追加】

- 「スタート画面」を選択
- 「挿入」→「人気順」よりテキストラベルを追加する
- 「貸し出しアプリ」等と文字を入力する



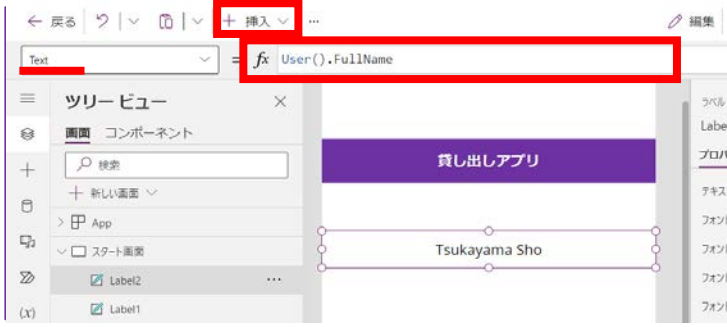
### 【パーツのプロパティを編集】

- 右側「プロパティ」より文字の大きさや色、テキストのアライメント等を調整する

以下、テキストラベルの説明ではプロパティに関するところは省略します。適宜調整してください。



# ①スタート画面

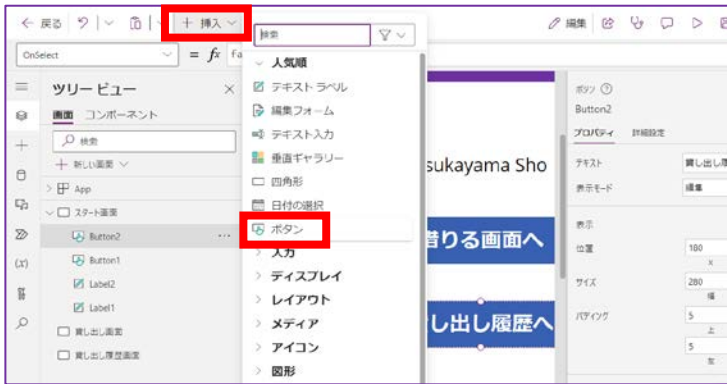


【ラベル（名前）を追加】

- ・「挿入」→「テキストラベル」を挿入しプロパティのTextに関数「User().FullName」を入力する

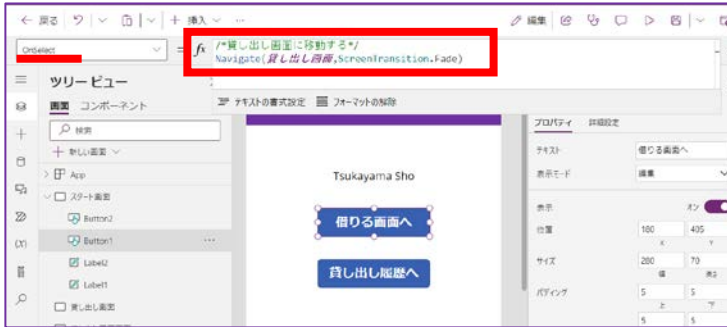
プロパティ	関数
Text	User().FullName

## ①-3. ボタンを追加する



【ボタンを挿入】

- ・「挿入」→「ボタン」よりボタンを2つ追加する  
マウスでダブルクリックして  
[借りる画面へ]ボタン  
[貸し出し履歴へ]ボタン  
と入力する。



- ・右側「プロパティ」よりボタンの大きさ等を調整する

【ボタンに関数を入力】

- ・追加したボタンを選択しプロパティのOnSelectに以下の関数を入力する

【借りる画面へ ボタン】

プロパティ	関数
Onselect	/*貸し出し画面に移動する*/ Navigate(貸し出し画面,ScreenTransition.Fade)

【貸し出し履歴へ ボタン】

プロパティ	関数
Onselect	/*貸し出し履歴画面に移動する*/ Navigate(貸し出し履歴画面,ScreenTransition.Fade)

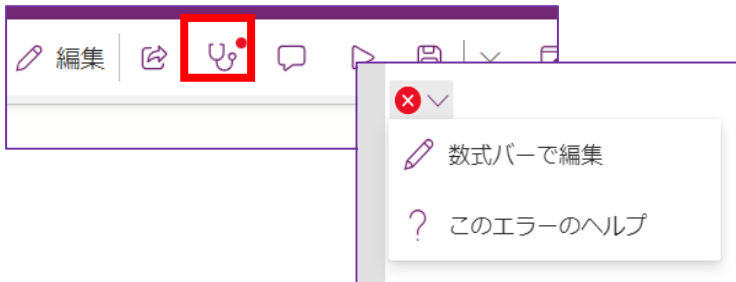
## \*プレビュー画面で確認する



【プレビューボタン】

- ・右上のプレビューボタンより動作確認ができる

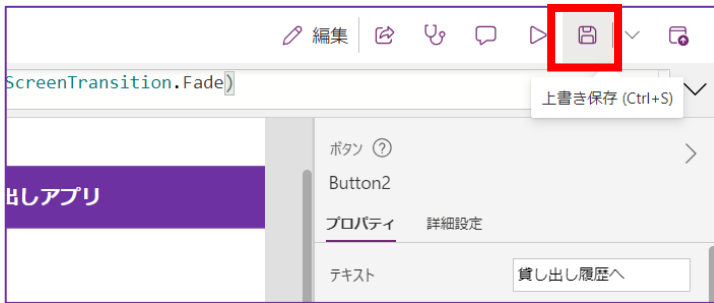
\*Windows端末の場合、「Altキー」+「Enterキー」でもプレビューを確認できる




【エラー表示】

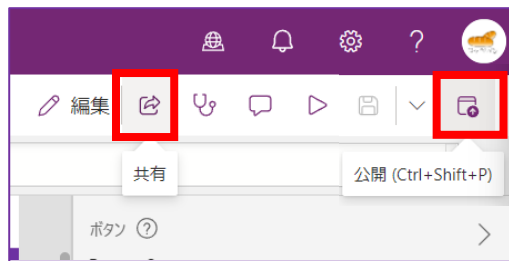
- ・赤いX = エラーが表示されたらクリックし「数式バーで編集」より数式（関数）を確認する

## \*アプリを保存する



- ・上部「

\*一度保存すると後は2分ごとに自動で保存される(初期設定時)



【保存】…クラウドに保存することで他の端末からでも利用できる

【公開】…スマホやTeamsでアプリを利用する場合は公開が必要

【共有】…アプリを利用できるユーザーや※チームを指定できる

注) アプリを共有する際、アプリ以外にデータソース (sharepoint) も共有している状態でないといけないため、データソースはアプリを利用しようとしているチーム内に作るのが良い

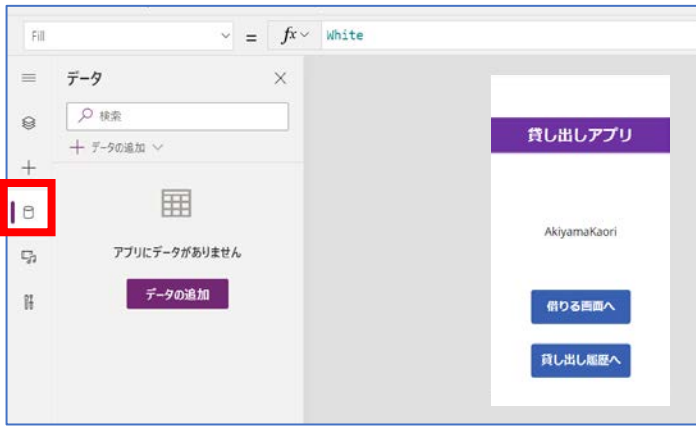
※ チームで共有する場合にはコツがある。[詳しくはポータルサイト参照](#)

# ②貸し出し画面

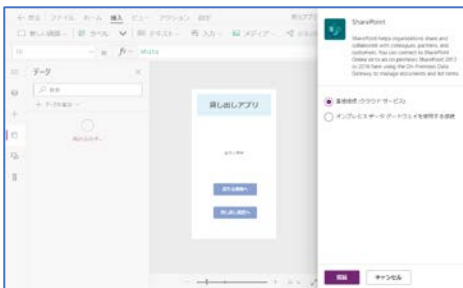
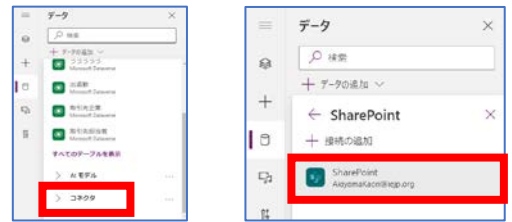
## ②-1. SharePointと接続する

Point!

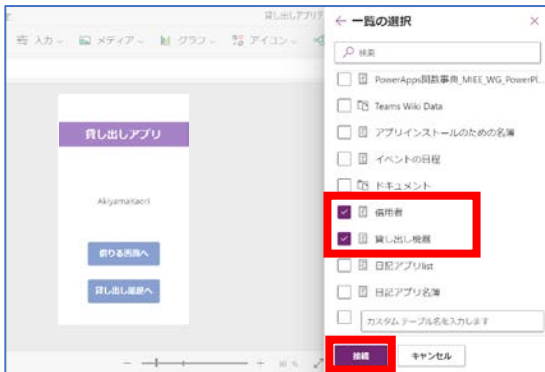
- ・ SharePointと接続することで、リストに保存してあるデータを参照したり、保存したりすることができる



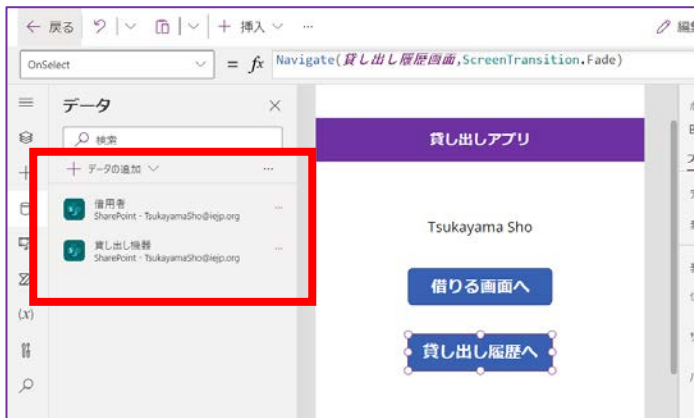
- ・ SharePointのリストデータを追加する
- ・ 「データ」 → 「+データの追加」 → 「コネクタ」 → 「SharePoint」



- \* 「SharePoint」が表示されない場合は「+接続の追加」をクリックする
- ←この画面が表示された場合は「直接接続」を選択する



- ・ 「一覧の選択」より準備2で用意した「借入者」と「貸し出し機器」にチェック → 「接続」をクリックする

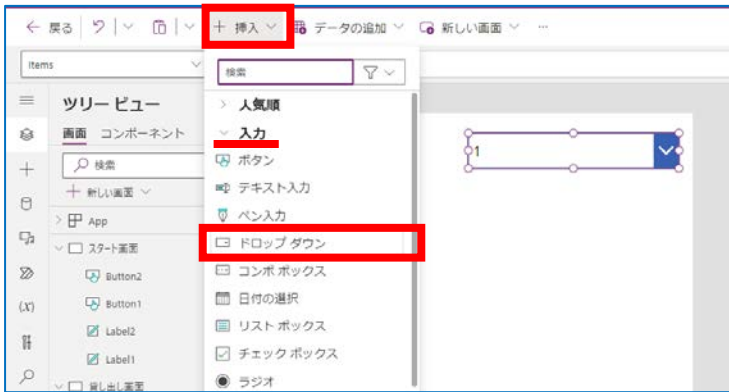


【接続を確認】

- ・ 接続できたら「データ」に「借入者」や「貸し出し機器」が表示されているか確認する

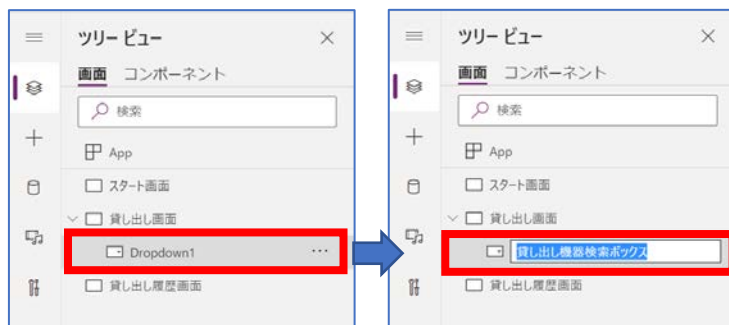
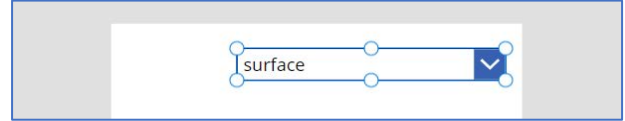
# ②貸し出し画面

## ②-2. 貸し出し機器検索ボックスを作成する



【検索ボックスを作成】

- ・ Screen 「貸し出し画面」 を選択
- ・ 「挿入」 → 「入力」 より「ドロップダウン」を選択する



【Dropdown1の名前を変更する】

- ・ ツリービュー「Dropdown1」をダブルクリックし名前を「貸し出し機器検索ボックス」に変更する



- ・ プロパティのItemsにして数式バーに以下の関数を入力する

### 【貸し出し機器検索ボックス】

関数をそのままコピーするならPDFはブラウザで閲覧が超おすすめ

プロパティ	関数
Items	<pre>/*重複を排除する*/ Distinct(貸し出し機器,id機器の種類)</pre>

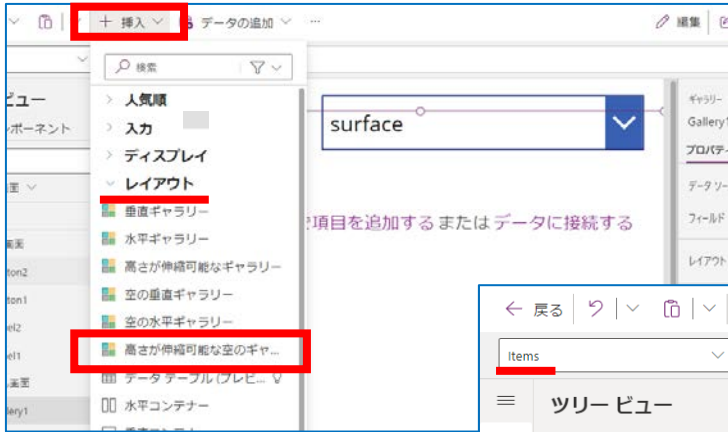
### Point!

【名前を変更する理由】 名前を変更しないと、他の関数で利用した場合にエラー表示となる

- ・ **Distinct関数**は、重複したデータを避けてドロップダウンリストにすっきりと表示させることができる

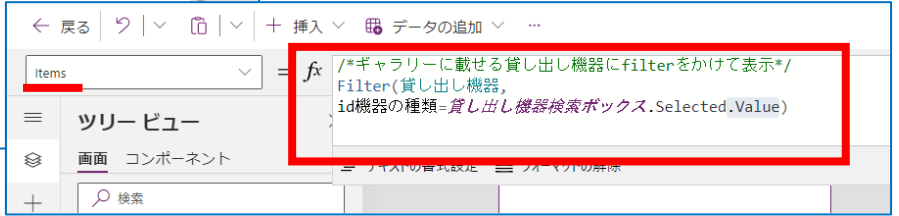
# ②貸し出し画面

## ②-3. 貸し出し機器ギャラリーを作成する



【ギャラリーを追加】

- ・「挿入」→「レイアウト」→「ギャラリー」→「高さ伸縮可能な空のギャラリー」を選択

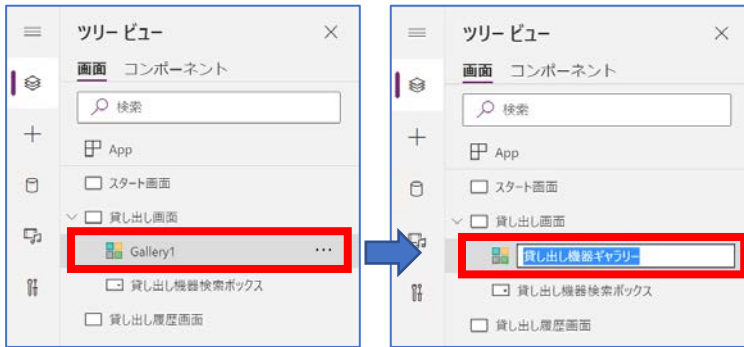


\*ギャラリーのプロパティをItemsにして

【貸し出し機器ギャラリー】 数式バーに以下の関数を入力する

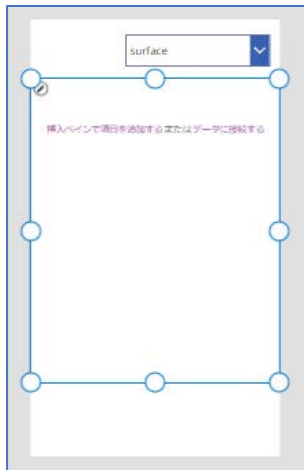
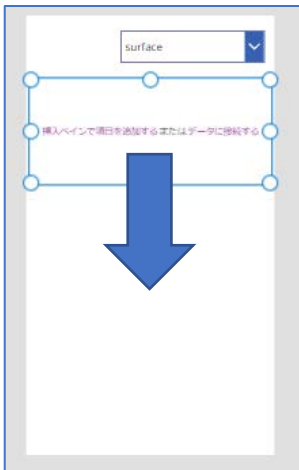
プロパティ	関数
Items	<code>/*ギャラリーに載せる貸し出し機器にfilterをかけて表示*/ Filter(貸し出し機器, id機器の種類=貸し出し機器検索ボックス.Selected.Value)</code>

\*関数をそのまま貼り付けた際に、改行によるエラーが出る可能性があるがあるので適時数式バーを修正する



【名前を変更】

- ・ギャラリーの名前を「貸し出し機器ギャラリー」に変更する



【枠を拡大】

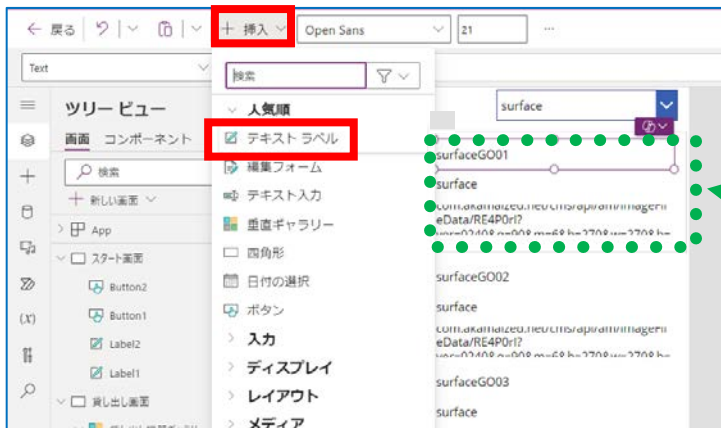
- ・ギャラリー枠を下まで拡大する

\*一番下には「戻るボタン」や「借りるボタン」を追加するためスペースを空けておく



# ②貸し出し画面

## ②-4. ギャラリー内にラベルを追加する



### 【ラベルを追加】

- ・ ギャラリーの小窓を選択した状態で「ラベル」を3つ追加する

POINT!!

必ずギャラリーの小窓を選択した状態でラベルを追加すること

- ・ それぞれのラベルに関数を入力する

### 【貸し出し状態を表示するラベル】

コピーするならPDFはブラウザで閲覧が超おすすめ

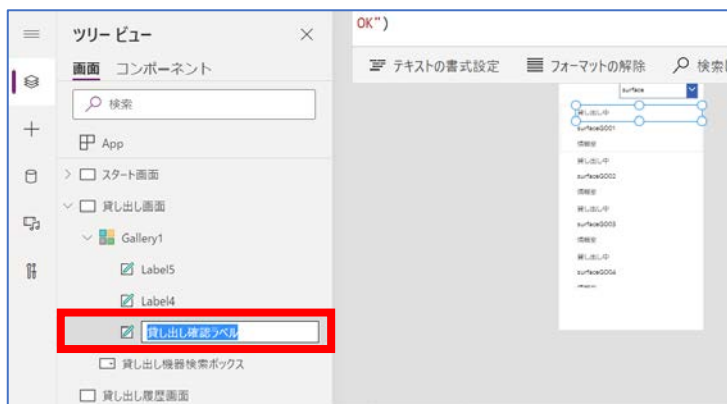
プロパティ	関数
Text ※このラベルだけ名前変更	<pre>/*貸し出し状態を表示_Concat関数は表の対象列を解除し一行テキストにする*/ If(ThisItem.id機器名 in Concat(Filter(借用者,sp貸し出し状態="借用中"),sp機器名),"貸し出し中","貸し出しOK")</pre>

### 【機器名を表示するラベル】

プロパティ	関数
Text	<pre>/*機器名を表示*/ ThisItem.id機器名</pre>

### 【保管場所を表示するラベル】

プロパティ	関数
Text	<pre>/*保管場所を表示*/ ThisItem.id場所</pre>



### 【名前を変更】

- ・ 『貸し出しOK』を表示させたラベルの名前を「貸し出し確認ラベル」に変更する

# 「貸し出し確認ラベル」に記述する 関数について

目次へ

```
If(  
  ThisItem.id機器名 in  
    Concat(  
      Filter(借用者,sp貸し出し状態="借用中"),  
      sp機器名),  
  “貸し出し中” , “貸し出しOK” )
```

## If関数

- プログラムを分岐させる
  - 「true」 =” 貸し出し中” 「false」 =” 貸し出しOK”

## in関数

- 含まれるか調べる関数
  - 1行テキスト化した文字の中に**対象の文字** (id機器名) が含まれるか調べる (含まれる=true/含まれない=false)

## Concat関数

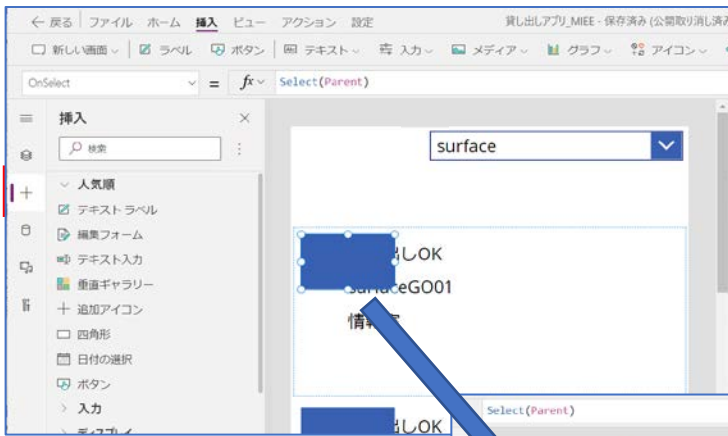
- 表を解除する関数
  - 表の” sp機器名” の**表を解除**し、1行テキストに変換 (ラベル表示可能)

## Filter関数

- フィルタリングする関数
  - 表の中の “借用中” を**集める** (ラベル表示不可)

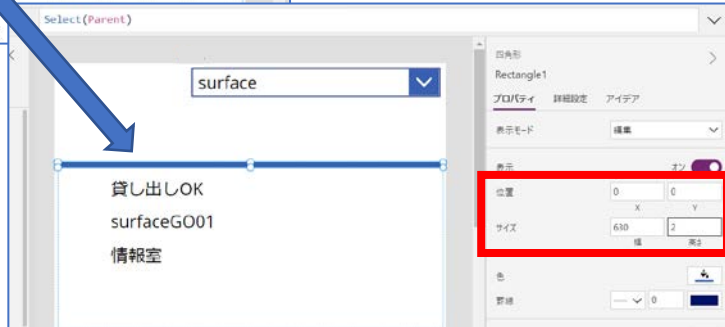
# ②貸し出し画面

## ②-5. ギャラリー内に枠を追加する

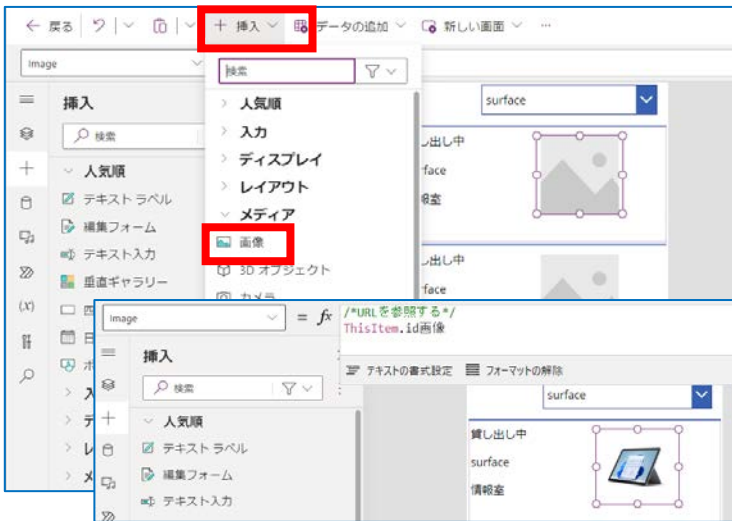


### 【枠を付ける】

- ・ ギャラリーの小窓を選択した状態で「+」→「四角形」より四角形を追加し細長くして線にする
- ・ 右側の「プロパティ」より線の太さや色を調整できる



## ②-6. ギャラリー内に画像を追加する



### 【画像を追加】

- ・ ギャラリーの小窓を選択した状態で「挿入」→「メディア」→「画像」より画像を挿入する
- ・ 位置や大きさを調整する
- ・ 画像のプロパティをImageにして数式バーに以下の関数を入力する

注) リンク切れの場合は画像は表示されない場合があります。  
オリジナル画像を使うならP28参照  
もしくは、左の「○上達へのコツ」参照

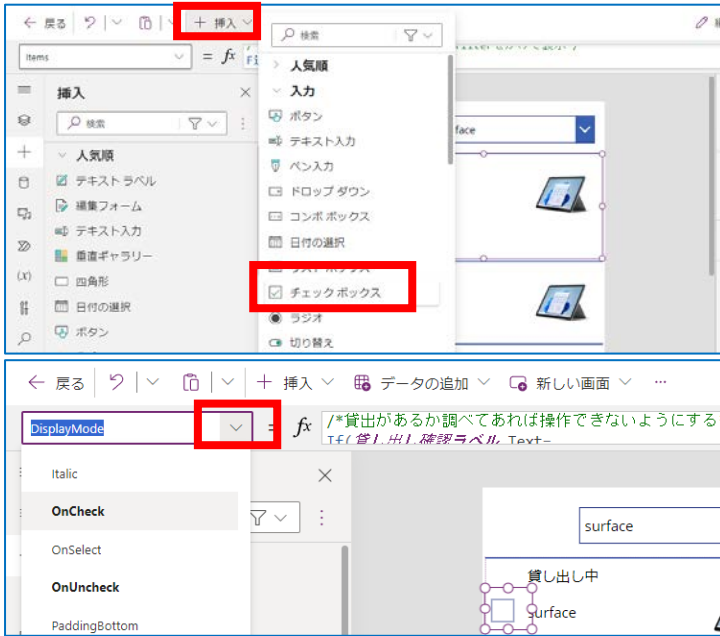
### ○上達へのコツ

Listに画像列を追加してlistから直接画像を呼び出すことも可能  
Google検索「SharePointのイメージ列でPower Appsの画像を操る」等参照

### 【画像】

プロパティ	関数
Image	/*URLを参照する*/ ThisItem.id画像


## ②-7. チェックボタンを追加する



- ・ギャラリーの小窓を選択した状態で「挿入」→「入力」→「チェックボックス」を選択し追加する

- ・チェックボックスのプロパティに以下のそれぞれの関数を入力する

**Point!!**

 をクリックしてプロパティを変更することができる

**【チェックボックス】** そのままコピペするならPDFはブラウザで閲覧が超おすすめ

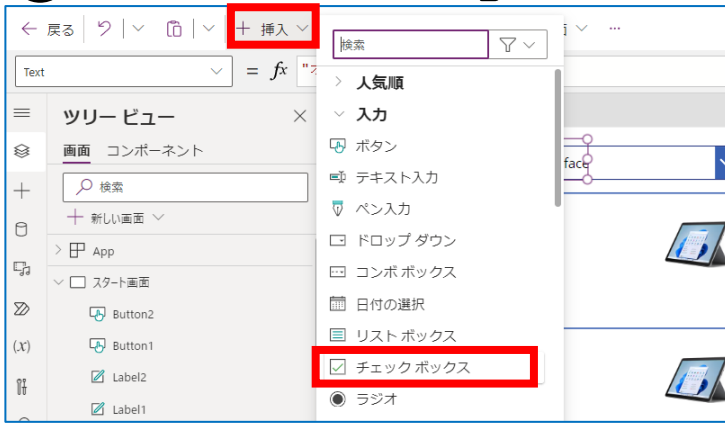
プロパティ	関数
Text	/*空欄（削除する）*/
OnCheck	/*レンタル機器Listに対象のアイテムを追加*/ Collect(レンタル機器List,ThisItem.id機器名)
OnUncheck	/*レンタル機器Listの中から対象のアイテムを削除*/ RemoveIf(レンタル機器List,Value=ThisItem.id機器名)
Default	/*レンタル機器Listに項目があればチェック済みにする*/ ThisItem.id機器名 in レンタル機器List
DisplayMode	/*貸出があるか調べてあれば操作できないようにする*/ If(貸し出し確認ラベル.Text= "貸し出しOK",DisplayMode.Edit, DisplayMode.Disabled)

### Point!

- ・ **Collect関数** を使うとアプリ内だけで保存できる「コレクション」というデータlistが自動で作成される
- ・ **Default関数** は、入力状況を自動で設定できる便利な関数  
テキスト入力や様々なパーツで利用できる

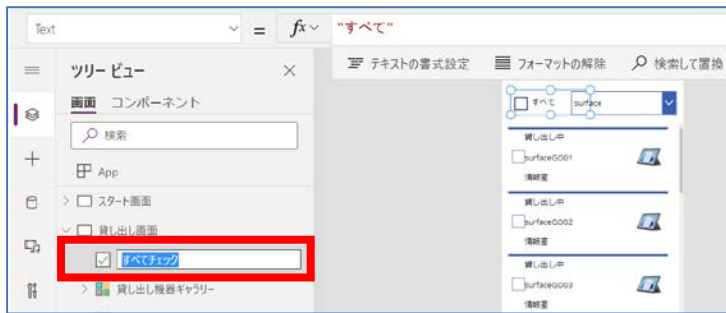
# ②貸し出し画面

## ②-8. 「すべて」チェックボックスを追加する



【「すべて」チェック枠を作成】

- ・ギャラリー外（画面上部）に「すべて」チェックボックスを追加する  
※ギャラリーの小窓は選択しない！！
- ・「挿入」→「入力」  
→「チェックボックス」を選択
- ・以下のプロパティをそれぞれ選択し関数を入力する



【名前を変更】

- ・「すべて」チェックボックスの名前を「**すべてチェック**」に変更する

【すべてチェックボックス】 コピペするならPDFはブラウザで閲覧が超おすすめ

プロパティ	関数
Text	"すべて"
OnCheck	<pre>/*レンタル機器listへ重複登録を防ぐためにいったんlistの中身を空にする*/ Clear(レンタル機器List); /*ForAll関数は貸し出し機器ギャラリーに表示しているすべての項目を対象とし、If関数で正しいと判断したものをレンタル機器listにCollectする*/ ForAll(貸し出し機器ギャラリー.AllItems, If(ThisRecord.貸し出し確認ラベル.Text="貸し出しOK",Collect(レンタル機器List, ThisRecord.id機器名)))</pre>
OnUncheck	<pre>/*保存されている項目をすべて削除*/ Clear(レンタル機器List)</pre>

### Point!

- ・ **ForAll関数**を使うと多くの作業を一度に処理することが出来る  
○表のまま処理する ○並列処理する ※詳しくは20ページ参照



```
ForAll(
```

```
    貸し出し機器ギャラリー.AllItems,
```

```
        If(
```

```
            ThisRecord.貸し出し確認ラベル.Text="貸し出しOK",
```

```
            Collect(レンタル機器List,ThisRecord.id機器名)
```

```
        )
```

```
    )
```

## ForAll関数

- 表(ギャラリー)のすべての行に対し同じプログラムを行う

## If関数

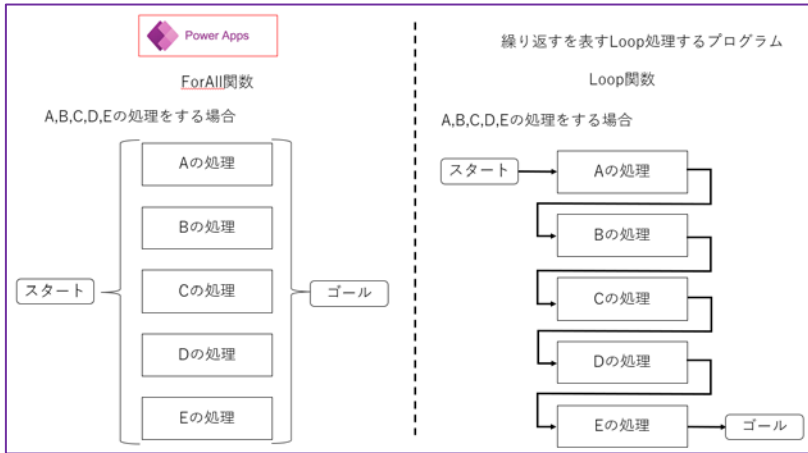
- ラベルの文字が"貸し出しOK"か調べる関数
- 一致した場合、Collect関数が起動

## Collect関数

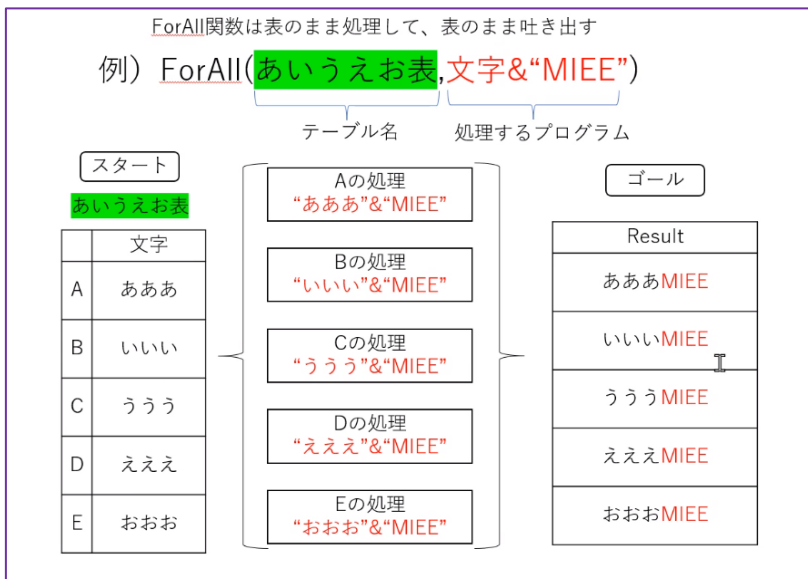
- 表にアイテムを追加する関数
- レンタル機器ListにThisRecord.id機器名を追加

## Point!

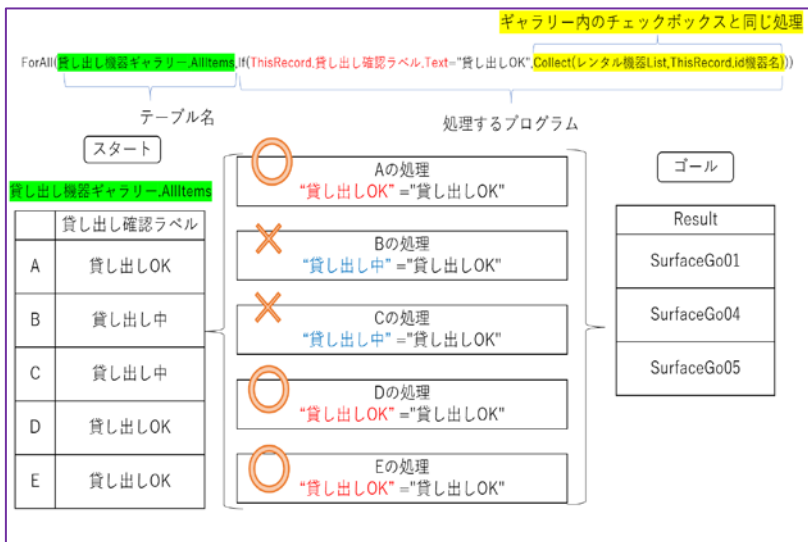
- **ForAll関数**を使用すると一度に多くの作業を処理することが出来る
- ただし、**ForAll関数は並列処理するプログラム**



**【Loop関数との違い】**  
\*Power AppsにLoop関数はないため、ForAll関数を使って多くの作業を処理していく



**【表のまま処理する】**  
\*例えば、「あいうえお」という表を処理すると、それぞれのセルに対して文字を付け足すというプログラムになる

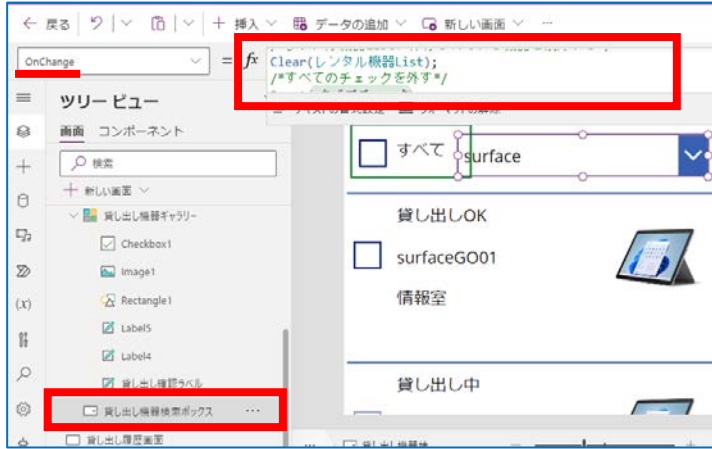


**【貸し出し中か判断】**  
\*ラベルのテキストを読み取って“貸し出しOK”なら対象の機器名を表に集めるプログラムになっている

# ②貸し出し画面

## ②-9. 貸し出し機器検索ボックスの

### OnChangeに関数を追加する



【OnChangeに関数を追加】

- ・ ツリービューの「貸し出し機器検索ボックス」を選択する
- ・ プロパティを「OnChange」に切り替える
- ・ 以下の関数を入力する

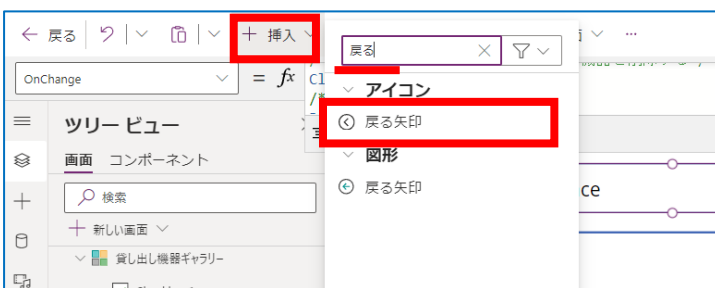
【貸し出し機器検索ボックス】 コピペするならPDFはブラウザで閲覧が超おすすめ

プロパティ	関数
OnChange	<pre>/*レンタル機器listに保存されている機器を削除する*/ Clear(レンタル機器List); /*すべてのチェックを外す*/ Reset(すべてチェック)</pre>

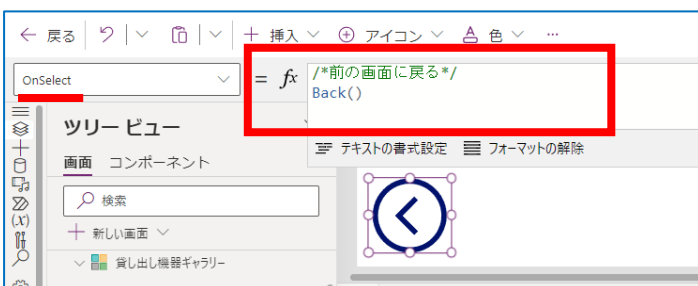
### Point!

- ・ 貸し出し機器検索ボックスを変更したときに選択している機器をリセットするために**OnChange関数**でリセットをかける

## ②-10. 戻るボタンを追加する



- ・ 「挿入」→検索バー→「戻る」を検索して画面左下に挿入する
- ・ 戻るアイコンのプロパティの数式バーに関数を入力する

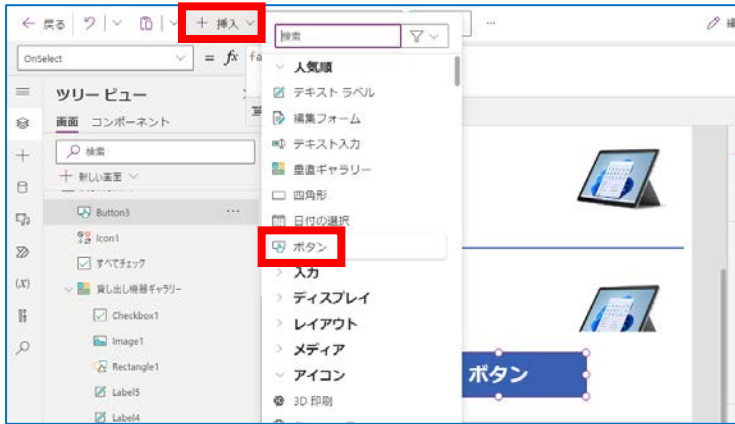


【戻るボタン】

プロパティ	関数
Onselect	<pre>/*前の画面に戻る*/ Back()</pre>

# ②貸し出し画面

## ②-11. 借りるボタンを追加する



- ・ 「挿入」 → 「ボタン」より「借りる」ボタンを作成する
- ・ ボタンをクリックしそれぞれのプロパティに以下の関数を入力する

【借りるボタン】 そのままコピペするならPDFはブラウザで閲覧が超おすすめ

プロパティ	関数
Onselect	<pre>/*借用者リストに貸し出し情報を追加する*/ Patch(借用者, Defaults(借用者), {   /*[列名]:保存させたいテキスト*/   sp借用日: Text(Today(),"[\$-ja]yyyy/mm/dd"),   sp名前: User().FullName,   spメールアドレス: User().Email,   sp貸し出し状態: "借用中",   /*Mid(で先頭の" "を削除する_Concatでlist内の機器を一行テキストにする。*/   sp機器名: Mid(Concat(Sort(レンタル機器List,Value)," " &amp; Value),2) }); /*一時保存していたリストを削除する*/ Clear(レンタル機器List)</pre>
Text	<pre>/*CountRowsでいくつリストに保存させているかを調べる*/ "借りる "&amp;CountRows(レンタル機器List)&amp;"台"</pre>

### Point!

- ・ **Patch関数**を使うことでformを使わなくてもデータソースの「借用者」にアイテムを追加することができる。

```
Patch(借用者, Defaults(借用者),
{ [列名]:[保存させたいテキスト]
  sp借用日: Text(Today(), "[$-ja]yyyy/mm/dd"),
  ~省略~
  sp機器名: Mid(
    Concat(
      Sort(レンタル機器List, Value)
      , "|" & Value)
    , 2)
})
```

## Patch関数

- Sharepointのlistにアイテムを追加したり変更したりする関数

## Text関数

- 今日を表すToday()関数を2020/02/05にテキスト変換

## Mid関数

- 左から数えて指定した文字から開始する関数
- 頭に“|”が付くためそれを削除する役目

## Concat関数

- 表を解除する関数
  - レンタル機器Listの表を解除し、間に“|”を入れて1行テキストに変換

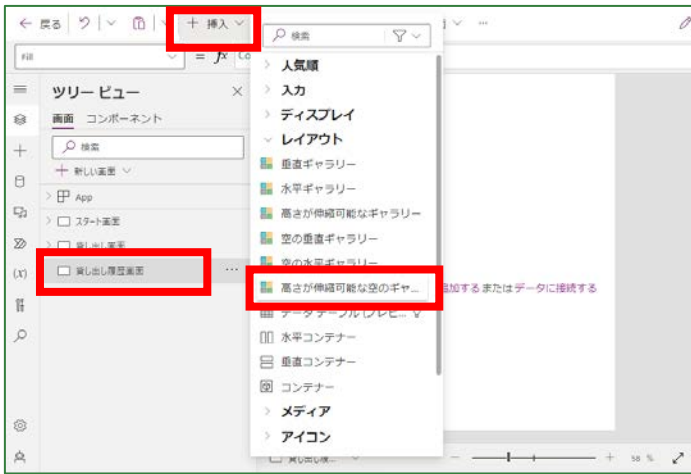
## Sort関数

- 表をソート(並び替え)する関数
- ばらばらにチェックした表を機器名でソートする



# ③貸し出し履歴画面

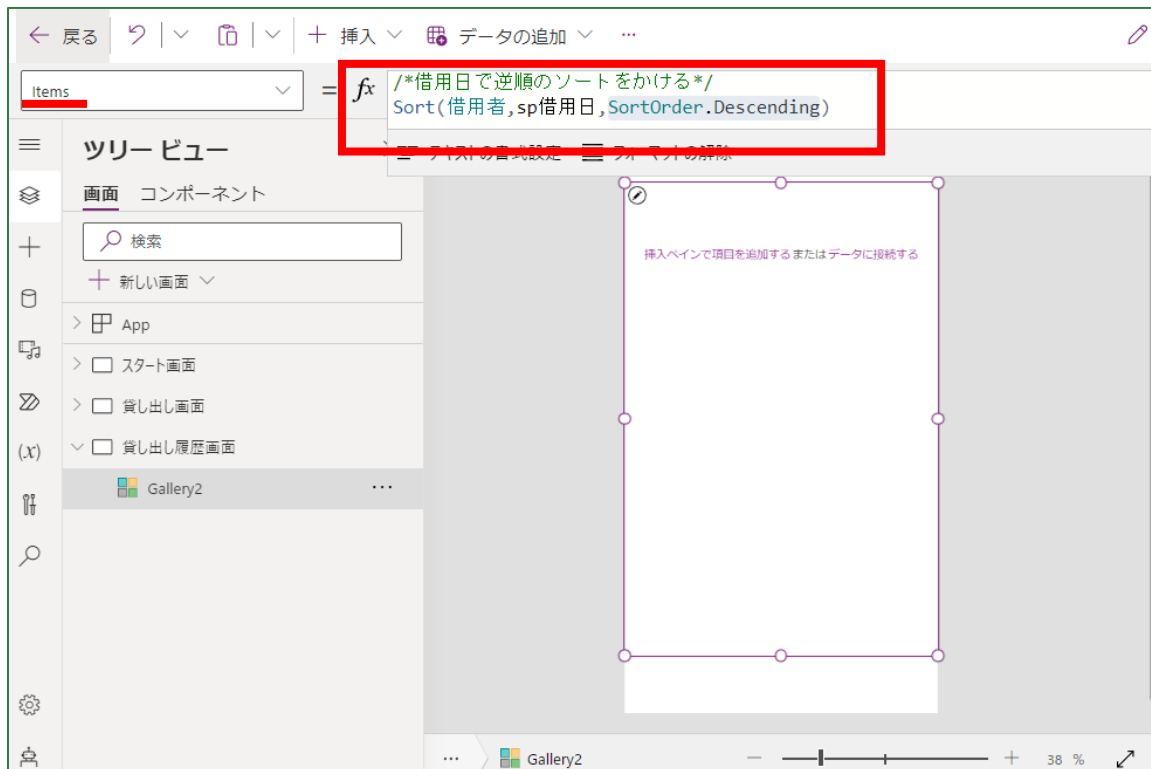
## ③-1. ギャラリーを作成する



- Screen 「貸し出し履歴画面」 を選択し  
「挿入」 → 「レイアウト」 → 「ギャラリー」 → 「高さが伸縮可能な空のギャラリー」 を選択

\*ギャラリーのプロパティのItemsに以下の関数を入力する【貸し出し履歴ギャラリー】

プロパティ	関数
Items	<code>/*借用日で逆順のソートをかける*/ Sort(借用者,sp借用日, SortOrder.Descending)</code>



- 関数入力後、ギャラリーを下の方に伸ばしておく
- 一番下には戻るボタンを後ほど追加するので隙間を作る

# ③貸し出し履歴画面

## ③-2. ギャラリー内にラベルを追加する



・ ギャラリー内の小窓を選択した状態で「ラベル」を4つ追加する

・ それぞれのラベルに関数を入力する

【ラベルの位置を調整】

・ 追加した各ラベルの位置を調整する

【借用日を表示するラベル】 **コピーするならPDFはブラウザで閲覧が超おすすめ**

プロパティ	関数
Text	ThisItem.sp借用日

【借用者を表示するラベル】

プロパティ	関数
Text	ThisItem.sp名前

【貸し出し状態を表示するラベル】

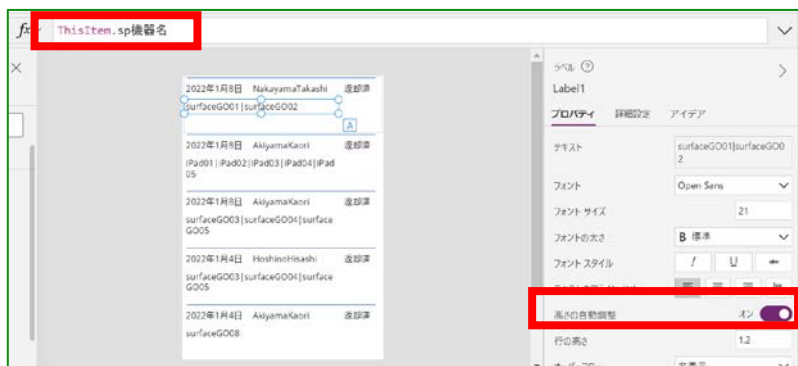
プロパティ	関数
Text	ThisItem.sp貸し出し状態

【貸し出し機器を表示するラベル】

プロパティ	関数
Text	ThisItem.sp機器名

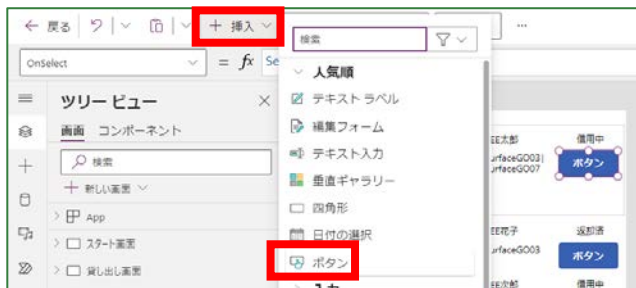
※注意

「機器名」は借し出し台数が多い場合でもすべて表示させられるように、ディスプレイ右のプロパティの「高さの自動調整」をONにする



# ③貸し出し履歴画面

## ③-3. 返却ボタンを追加する

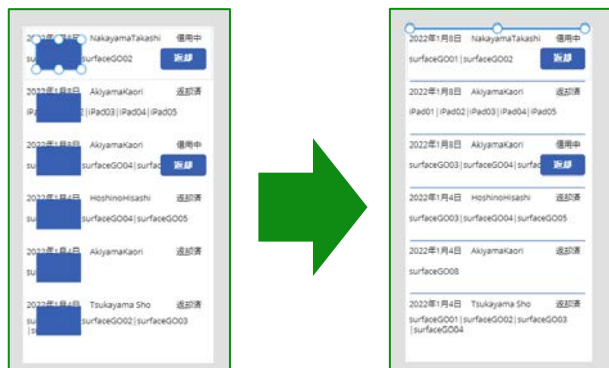
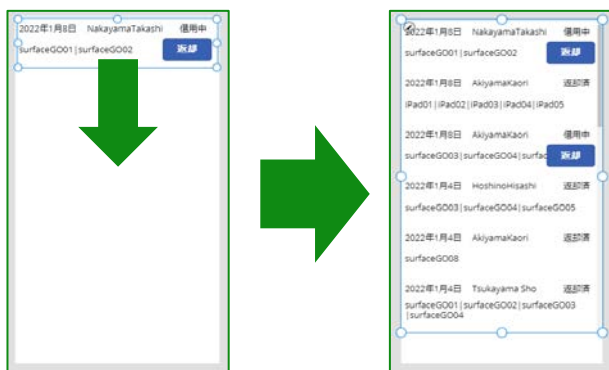


- ・ギャラリーの小窓を選択した状態で「挿入」→「ボタン」より「返却ボタン」を作成する
- ・「返却ボタン」のプロパティに以下の関数をそれぞれ入力する

### 【返却ボタン】

プロパティ	関数
Text	"返却"
OnSelect	/*listに保存されているアイテムを更新する*/ Patch(借用者,LookUp(借用者,ID=ThisItem.ID), {sp貸し出し状態:"返却済"})
DisplayMode	/*自分以外は押せないようにする*/ If(ThisItem.spメールアドレス =User().Email,DisplayMode.Edit, DisplayMode.Disabled)
Visible	/*貸し出し中ならば表示させない*/ If(ThisItem.sp貸し出し状態="借用中",true,false)

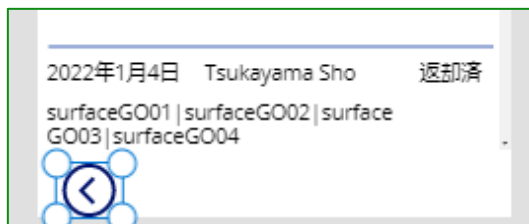
## ③-4. 体裁を整える



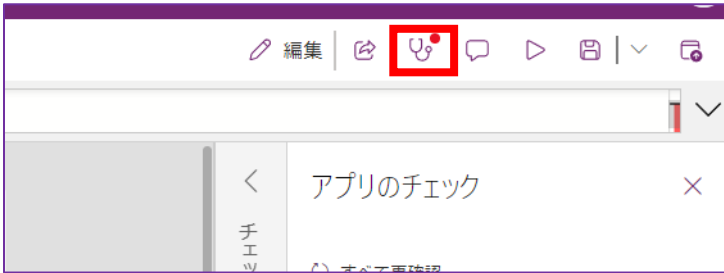
- ・「ギャラリー」の枠を下まで拡大し、他の人が借りた履歴も表示させる

- ・P16②-5 「貸し出し画面」と同じように枠に線を付ける

- ・P21②-10 「貸し出し画面」と同じように「戻るボタン」を「アイコン」より挿入し関数を入力する



## \*アプリの診断



### 【エラーを確認】

- ・診断マークに赤い丸があったらクリックしてアプリのチェックをする
- ・赤い×=エラーが表示されたらクリックし「数式バーで編集」より数式（関数）を確認する



## \*よくあるエラー

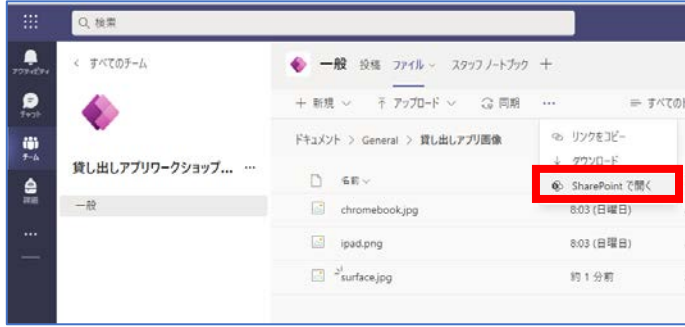
- ・間違ったプロパティに関数を入力している
- ・ギャラリー内にラベルなどのパーツが正しく配置されていない
- ・関数入力時に誤って複数のパーツを選択した状態で入力してしまった
- ・テキストを直接入力するときに「” ”」を忘れている
- ・ドロップダウンを参照するとき「.Selected.Value」ではなく「.Selected」や「.SelectedText」で参照している
- ・関数に全角半角スペースが紛れ込んでいる

\*ラベルやギャラリーなど必ず名前を記入しないといけないところに入力がない

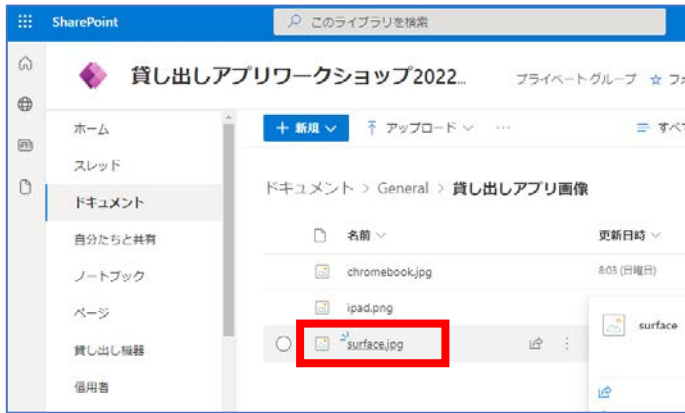
- ⇒今回の貸し出しアプリの場合、貸し出し画面の
- ドロップボックス：「貸し出し機器検索ボックス」(P12)
  - ギャラリー：「貸し出し機器ギャラリー」(P13)
  - 貸し出し確認ラベル：「貸し出し確認ラベル」(P14)
  - すべてチェックボックス：「すべてチェック」(P18)
- に名前を変更する必要あり

\*プレビューで画面を確認、テスト入力をしてみましょう。  
問題なく動けば完成です！！お疲れ様でした(^^)

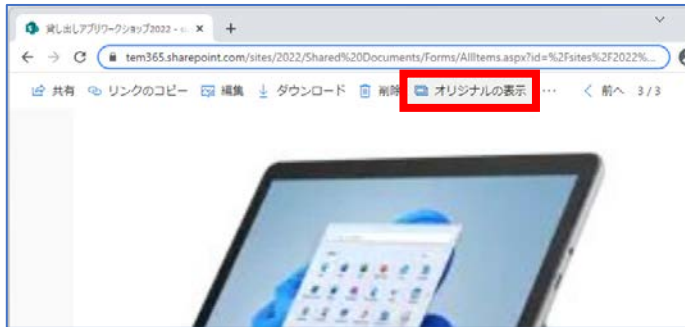
注) 画像ファイルの共有リンクのコピペではうまくいきません



- アプリに表示させたい画像を Teamsの「ファイル」内→「貸し出しアプリ用フォルダ」内に保存する

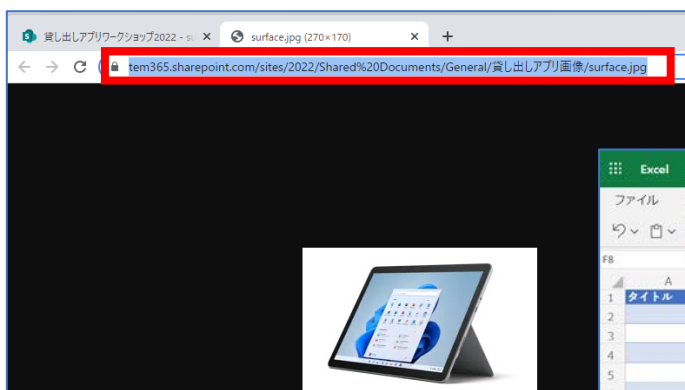


- 貸し出し機器の画像を保存しているチームのファイルから「Share Pointで開く」を選択



- SharePoint上の対象の画像をクリックして開く

- 画面上部「オリジナルの表示」をクリック、表示されたURLをコピーしExcel貸し出し機器Sheetの「id画像」の列に貼る



\*オリジナルの表示で表示されたURLは「リンクのコピー」とは違いShare Point上の場所を表すURL

\*画像の推奨拡張子：jpg png

タイトル	id機器名	id機器の種類	id画像	id場所
1	surfaceG001	surface	https://tem365.sharepoint.com/sites/2022/Shared%20Documents/...	情報室
2	surfaceG002	surface	https://tem365.sharepoint.com/sites/2022/Shared%20Docume...	情報室
3	surfaceG003	surface	https://tem365.sharepoint.com/sites/2022/Shared%20Docume...	情報室
4	surfaceG004	surface	https://tem365.sharepoint.com/sites/2022/Shared%20Docume...	情報室
5	surfaceG005	surface	https://tem365.sharepoint.com/sites/2022/Shared%20Docume...	情報室
6	surfaceG006	surface	https://tem365.sharepoint.com/sites/2022/Shared%20Docume...	情報室
7	surfaceG007	surface	https://tem365.sharepoint.com/sites/2022/Shared%20Docume...	情報室
8	surfaceG008	surface	https://tem365.sharepoint.com/sites/2022/Shared%20Docume...	職員室
9	surfaceG009	surface	https://tem365.sharepoint.com/sites/2022/Shared%20Docume...	職員室
10	surfaceG009	surface	https://tem365.sharepoint.com/sites/2022/Shared%20Docume...	職員室

\*画像データは小さいサイズのものをおすすめします。  
画像データサイズが小さければアプリがサクサク動きます！

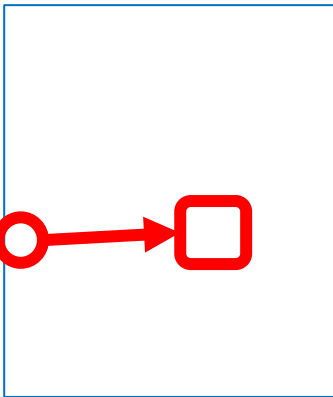




## ① クラス以外でチームを作成する

作ったアプリをチームの人にだけ共有するには、クラス以外で作成する必要があります

※“クラス”で作ったチームは、チーム名で共有することができません

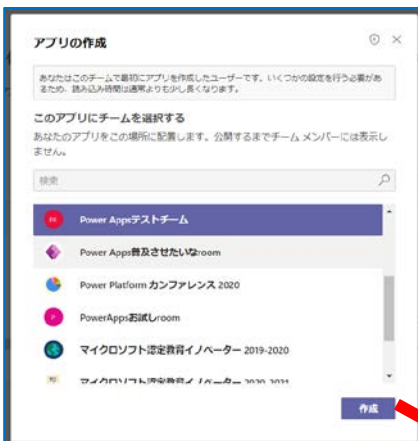


## ② TeamsからPowerAppsを開く

メニューの…(三点リーダー)から「Power Apps」を選択する

(ない場合は上のアプリ検索から検索)

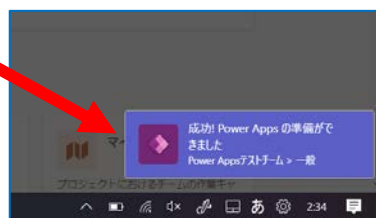
開いたら「今すぐ始める」をクリックする



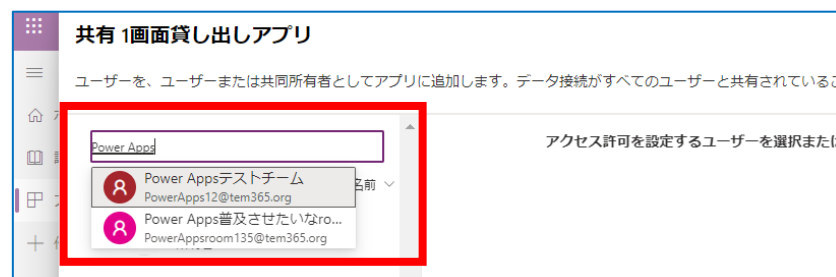
## ③ 共有したいチームを選択

共有したいチーム名を選択し、作成をクリックする

※ここにチーム名が出てこない場合は、少し時間をおいてから、もう一度お試してみる



## ④ 「成功」のアラートが出たらチーム名で共有が可能です Teamsは閉じてOK



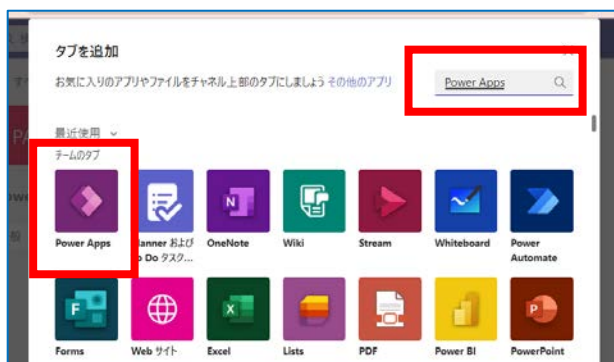
## ⑤ 共有で、チーム名を検索すると表示されるようになる

※チーム名で共有をすると、チームに所属しているメンバーがアプリを利用することができる

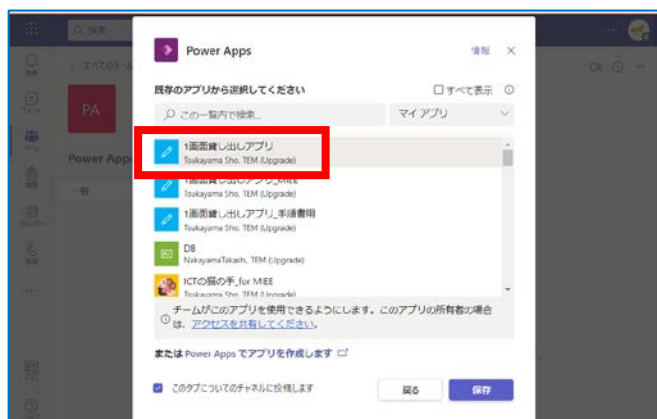
# PowerAppsアプリをTeams上で利用する方法



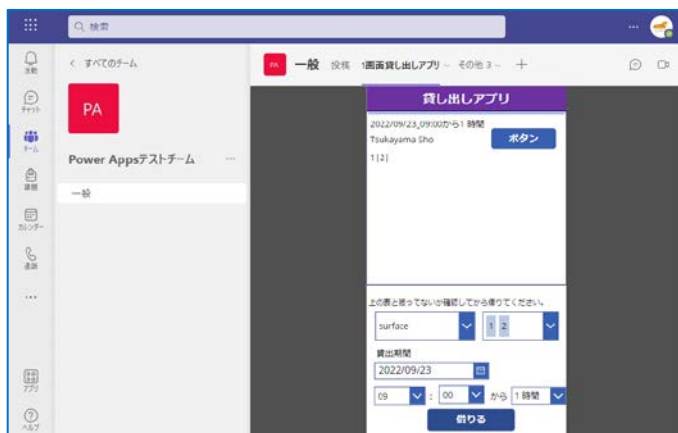
- アプリを利用したいチームの「+」タブを選択する



- 検索欄に Power Apps と検索をかけて「Power Apps」を選択する



- アプリ一覧より、チームで利用したいアプリを選択し、「保存」をクリックする



- タブにアプリが追加されて Teams上から利用できる