

Programming Boot Camp Learning Phase #3

Bubble Basic #2-1

2023/11/11

事前準備

- 本日は前回作成したペットの健康管理アプリに、デザインやロジックを追加していきます
- その後、さらにAPI連携やチーム開発といった応用についてもお話しします
- 本日の講義用に多少手を加えていますので、開始時点をそろえるため、こちら側で用意したアプリケーションを複製したものを利用してもらいます
- 複製したアプリケーションを配布しますので、@imahashi 宛てに、Bubbleのアカウントを作成したメールアドレスを伝えてください。
- また、動作確認用に実際にペットを5件くらい登録しておいてください。



今日やること

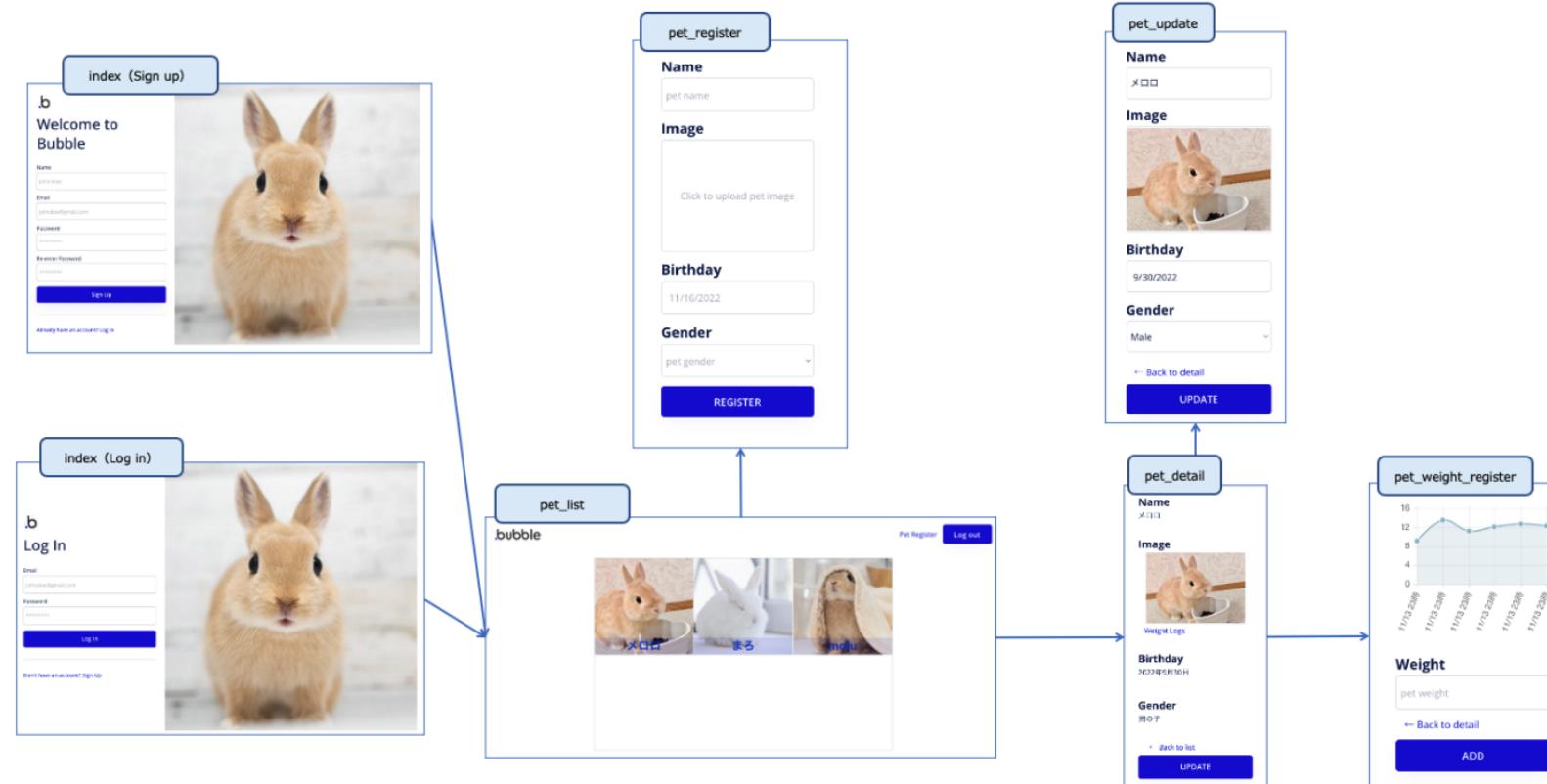
- 前回のおさらい
- デザインを作りこむ
- ロジックを作りこむ
- 外部システムと連携する
- チームで開発する

前回のおさらい

- Bubble はビジュアルプログラミングツールとうたっており、画面からポチポチと操作して、見た目や動きをプログラミングできるツールです。
- Webアプリケーション前提であり、ディスプレイサイズにあわせた対応をいれスマホやPCに対応させます。
- 前回お休みの方はこちらの資料でキャッチアップしましょう
 - <https://github.com/GuildWorks/titech-2023/tree/main/docs/Bubble1>

前回のおさらい

ペット管理アプリケーションのペットの登録、一覧、詳細、体重記録の画面を作りながら、Bubbleの基本である、Design/Workflow/Dataの使い方を学びました。



今日やること

まずは前回までのアプリケーションに対して、デザインやロジックをさらに作り込んでいきます

最終的にはこんな感じになります

トップページ

Sign up for My App

Email

Password

Confirmation

Pet Owner

Sign up

Already have an account? [Login](#)

Sign up for My App

Email

Password

Confirmation

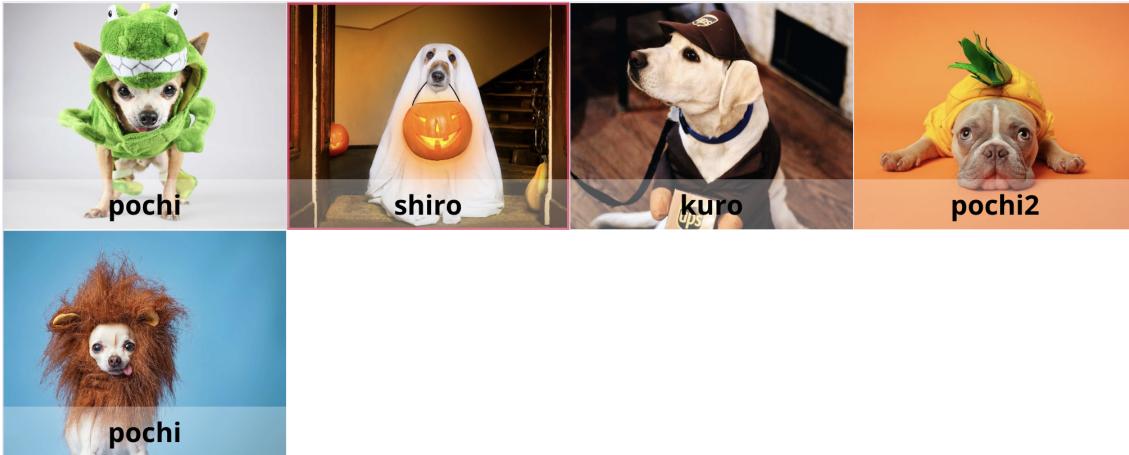
Pet Owner

Sign up

Already have an account? [Login](#)

ペットリスト

PetLog



PetLog



ペット詳細

PetLog

[← Back to list](#)



Name (Initial)

pochi(P)

Birthday

2020年8月18日

Age (as Dog/Cat)

2(24)

Gender

男の子

Latest Weight

9kg



[Pet Register](#)

[Log out](#)

PetLog

≡

[Weight Logs](#)

[← Back to list](#)

[Weight Logs](#)



Name (Initial)

pochi(P)

Birthday

2020年8月18日

Age (as Dog/Cat)

2(24)

Gender

男の子

Latest Weight

9kg

アドバイザー用ペットリスト

PetLog

[Pet Register](#)[Log out](#)

では、さっそくはじめていきましょう。

まずはデザインを作りこんでいきます

デザインの作り込みでやること

- ディスプレイサイズに合わせた画面をつくろう
 - レスポンシブウェブデザインという手法を使って、ディスプレイサイズに見た目を制御します
- Styleを使ってみよう
 - Styleを編集・追加したり、個別にスタイルをあてます

ディスプレイサイズに合わせた画面をつくるう

ディスプレイサイズに合わせた画面をつくろう

- WebアプリケーションはPCやタブレット、スマートフォンといった様々な端末で利用されます。
- 端末毎にディスプレイサイズが違うのですが、それらに対応する手法としてレスポンシブウェブデザインというデザイン手法があります
- 画面サイズに応じて、要素が伸びる／縮む、折り返す／折り返さない、表示する／表示しないといった見た目の切り替えが柔軟に行われる手法になります。
- 実現方法として、固定した配置やサイズを指定するのではなく、配置やサイズを決めるためのルールを指定します。
- Bubbleは初期設定では、固定の配置やサイズが指定されるようになっていますが、各種ルールを指定することもできるようになっています。

よく使うルール

Bubbleでレスポンシブデザインを実現するために、よく使うルールとして以下のようないがあります。

1. 親要素内の配置ルール
2. 要素のサイズ決定ルール
3. 表示の有無ルール

これらのルールを組み合わせて、レスポンシブな画面デザインを実現します。

なお、これらのルールはBubbleに限らず、Webアプリケーション全般に通じる考え方でもあります。

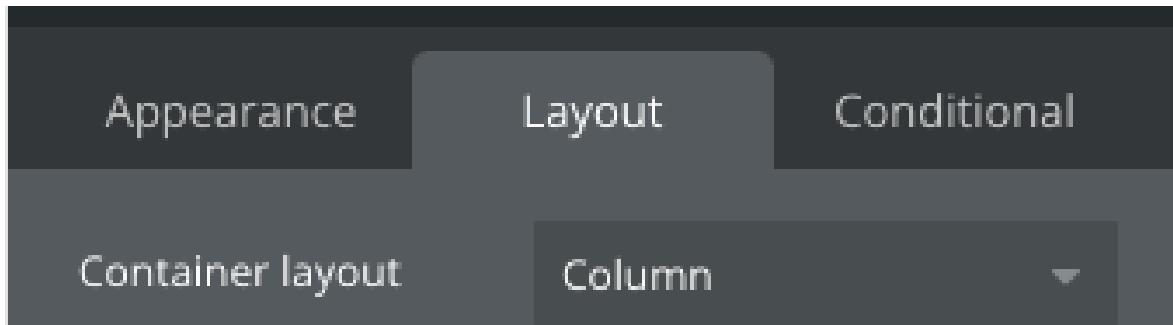
後ほど、一緒に画面に組み込んでいきますが、難しい考え方もあるので、まずは概要を説明していきます。

ルール1：親要素内の配置ルール

親要素の中でどう配置するかという、ルール指定になります。

リピーティンググループなどのグループやページ全体など個々の要素を囲むような親要素をBubbleではContainerと呼ばれています。

Containerでは中に含む子要素の配置ルールを指定できます。



子要素の配置ルールとしては以下の4つがあります。

- Fixed : 配置場所を固定で指定する
- Aligng to parent : 親要素に対する相対位置を指定する
- Row : 行方向（水平方向）に並べる
- Column : 列方向（垂直方向）に並べる

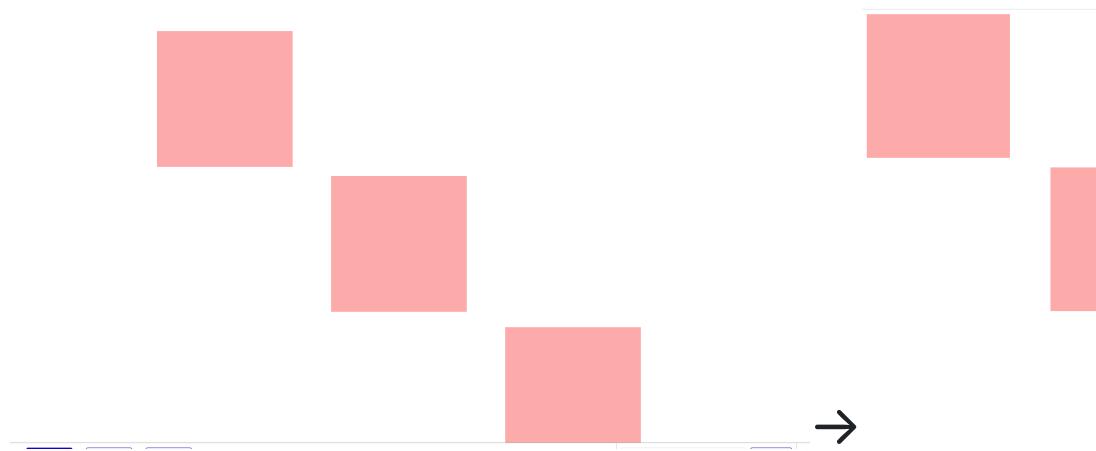
順に説明していきます。

Fixed：配置場所を固定で指定する

配置場所を固定で指定するルールです。配置場所をピクセル単位で指定します。

Bubbleで親要素を置いた場合の初期設定となります。

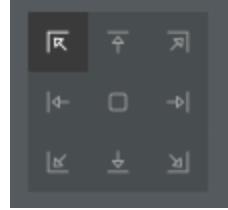
固定で指定しているため、画面幅を変更しても指定した位置から変更されません。下の例では、画面の外にはみ出てしまっています。



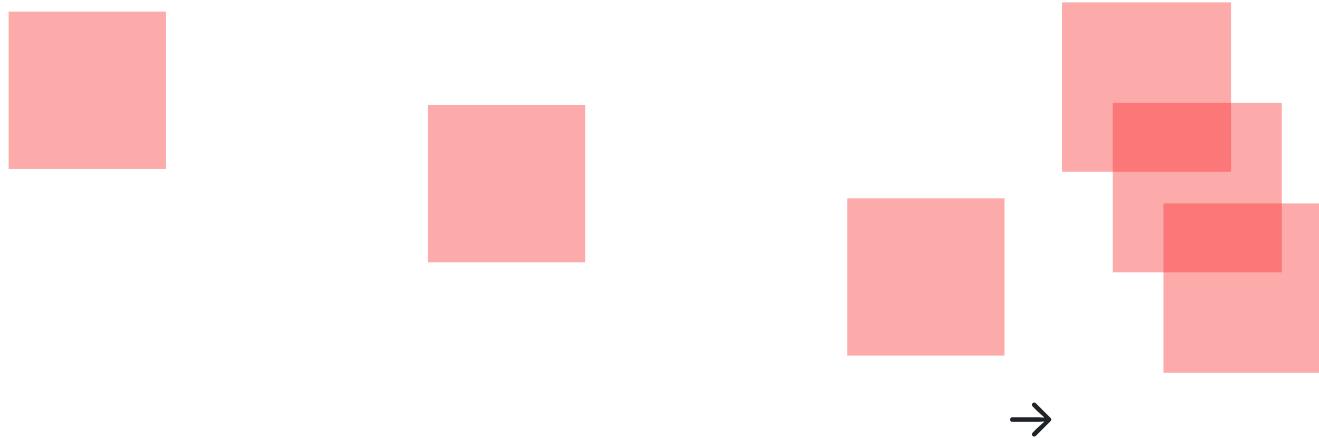
Alignng to parent : 親要素に対する相対位置を指定する

親要素に対する相対位置を指定するルールです。

Bubbleでは、9つのエリアから配置場所を指定できる。



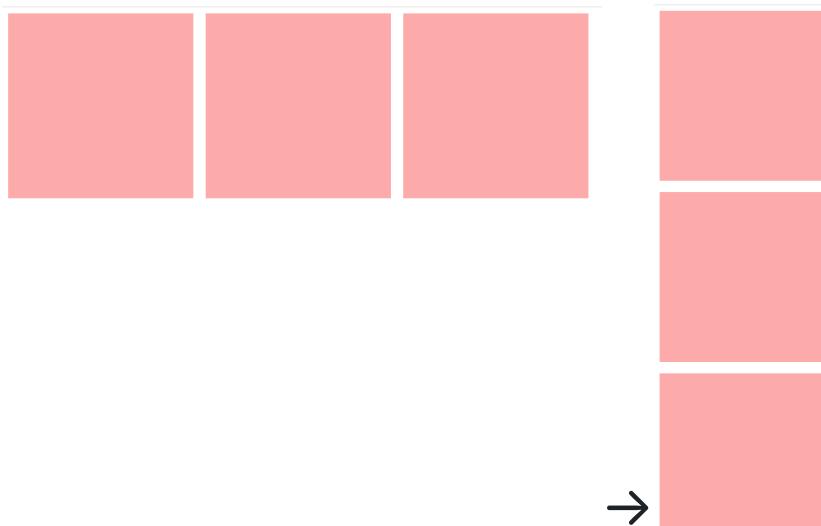
画面幅を変更したら、相対位置を維持して動く。下の例では、画面幅を狭めると、左上・中央・右下という相対位置を維持して動いている。



Row : 行方向（水平方向）に並べる

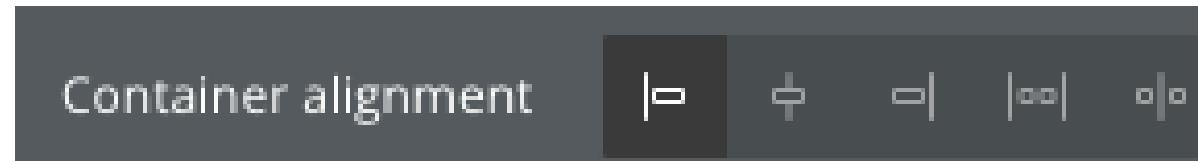
行方向（水平方向）に並べるルールになります。行は自動で折り返します。

下の例では、画面幅を狭めると、行が折り返していき、結果として縦に並んでいます。



Row : 行の中での水平方向（左右）の配置を指定できる

要素ごとに行の中での垂直方向の配置を指定できます。



(Left-aligned)



(Centered)



(Right-aligned)



(Space-around)

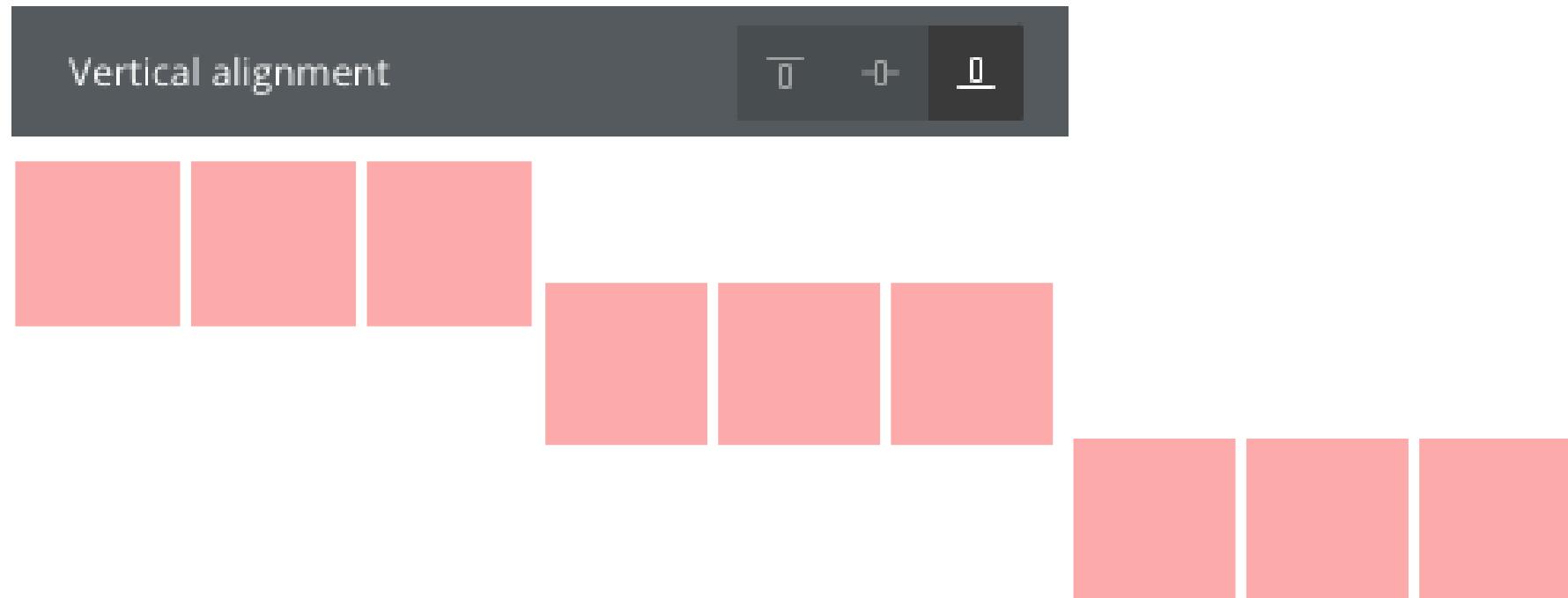


(Space-between)

Row : 行の中での垂直方向（上下）の配置を指定できる

行の中での垂直方向の配置を指定できる。

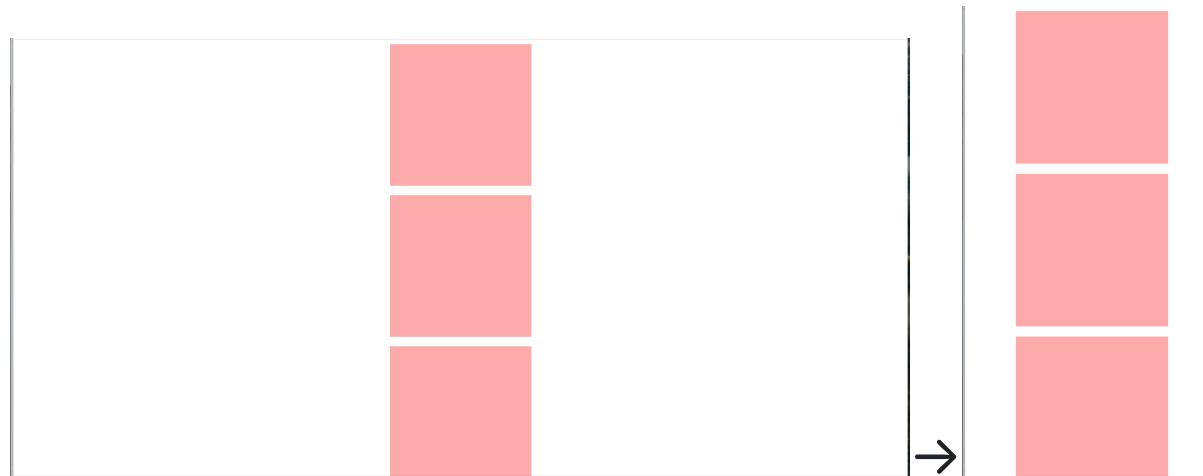
※これは親要素ではなく、子要素に対して指定するものになります。



Column：列方向（垂直方向）に並べる

列方向（垂直方向）に並べる。

下の例は左右中央寄せで縦に並べており、画面幅を小さくしても中央寄せを維持したまま縦に並んでいる。



Rowと同様に水平、垂直の配置を指定できる。（水平・垂直で指定できる内容は反対）

ルール1(おさらい)：親要素内の配置ルール

親要素の中でどう配置するかという、ルール指定になります。
子要素の配置ルールとしては以下の4つがあります。

- Fixed : 配置場所を固定で指定する
- Alignng to parent : 親要素に対する相対位置を指定する
- Row : 行方向（水平方向）に並べる
- Column : 列方向（垂直方向）に並べる

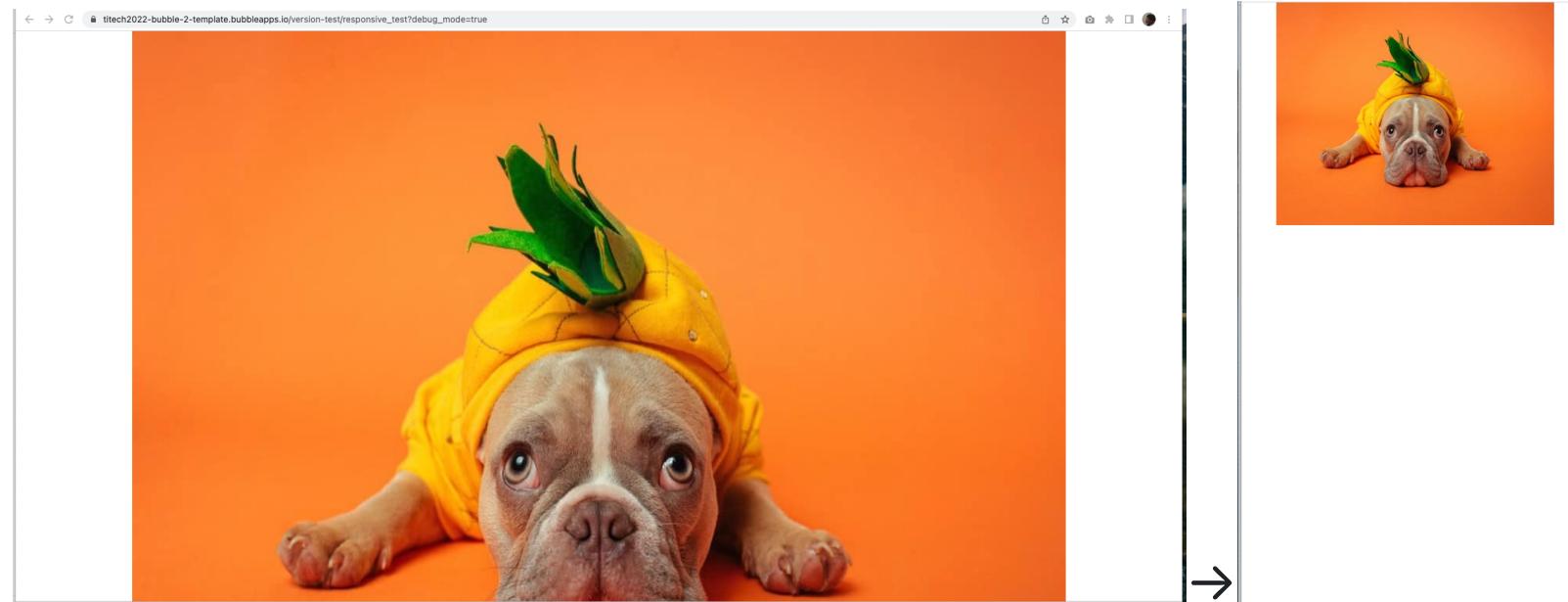
ルール2：要素のサイズ決定ルール

画面幅によって、大きくなったり小さくなったりできるように、固定サイズを指定のではなく、サイズを決めるためのルールを指定します。主に以下の二つのいずれかを利用します。

- 親要素の大きさに対するパーセンテージを指定する
- 伸び縮みした場合の最大・最小サイズを指定する
 - 最大最小は指定せずに、無制限にすることも可能

パーセント指定

下の例は、画面に対して幅が80%になるように指定しています。画面幅を小さくすると、80%の割合を維持しながら画像サイズも小さくなります。



下の例は、幅を最大800px、最小300pxに指定しています。

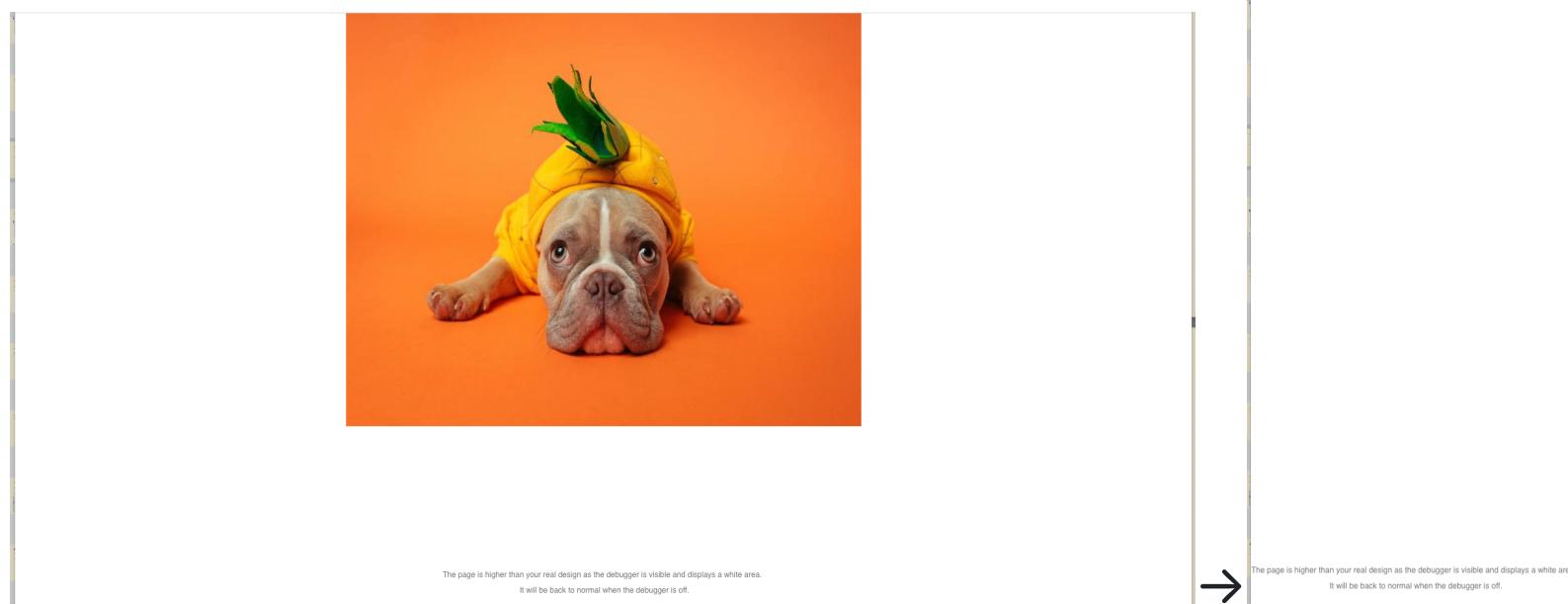
300pxから800pxの間で伸縮しますが、画面を広げきっても800px以上は大きくならないですし、逆に画面を狭めても300px以下にはなりません。



ルール3：表示の有無ルール

画面幅の下限や上限を指定して、画面内の要素の表示・非表示を切り替えることができます。

下の例は、画面幅が800px以下になった場合に画像を表示しないという指定をしています。



画面が大きい場合だけ情報を多く表示したいと言った場合によく使います。

よく使うルール(おさらい)

Bubbleでレスポンシブデザインを実現するために、よく使うルールとして以下のようなものがあります。

1. 親要素内の配置ルール
2. 要素のサイズ決定ルール
3. 表示の有無ルール

では、実際に使ってみましょう。

(再確認) 事前準備

- 本日は前回作成したペットの健康管理アプリに、デザインやロジックを追加していきます
- 本日の講義用に多少手を加えていますので、開始時点をそろえるため、こちら側で用意したアプリケーションを複製したものを利用してもらいます。
- 複製したアプリケーションを配布しますので、@imahashi 宛てに、Bubbleのアカウントを作成したメールアドレスを伝えてください。
- また、動作確認用に実際にペットを5件くらい登録しておいてください。



トップページ（ログインページ）にレスポンシブデザインを適用します

まずは、トップページ（ログインページ）に摘要されているレスポンシブデザインを確認してみましょう。

トップページでは、十分なディスプレイ幅があるときは、画像がいっぱい今まで広がっていますが、画面幅が小さくなるとサインアップやサインインのエリアしかでなくなります。

Pet Log



Name

Email

Password

Re-enter Password

Sign Up

Already have an account? [Log In](#)

Pet Log

Name

Email

Password

Re-enter Password

Sign Up

Already have an account? [Log In](#)

適用されているルール

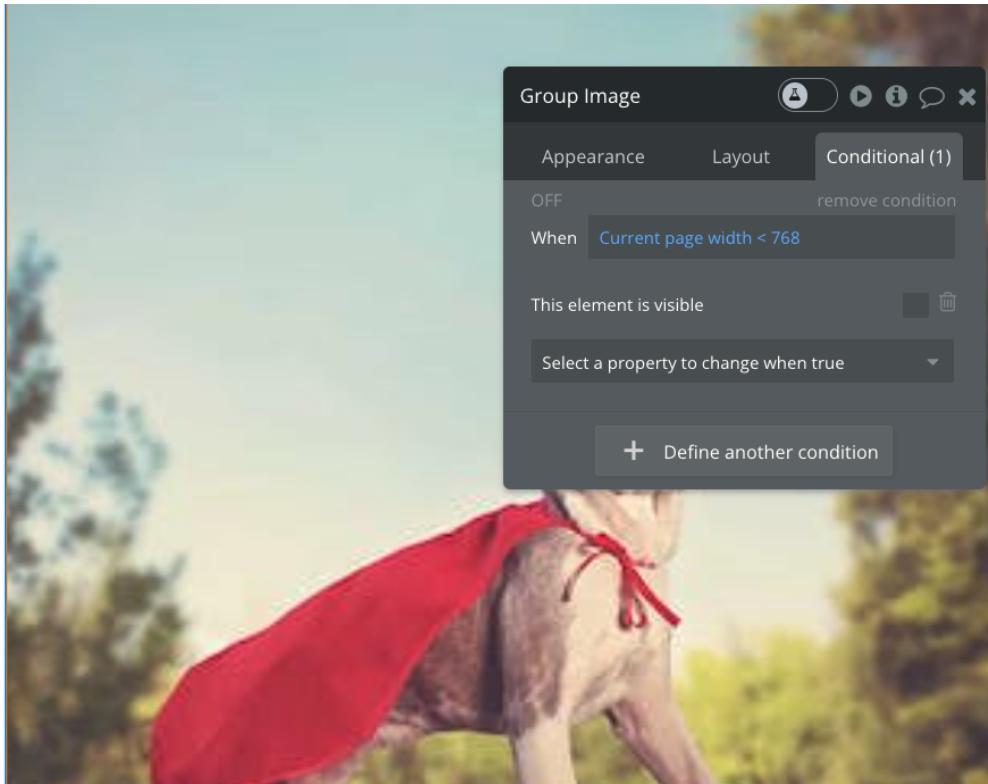
細かな設定はたくさん入っていますが、大きくは以下の二つの設定がきいています。

- 表示の有無ルール：画面幅が小さくなったとき（768pxより小さくなったとき）に、画像は出さないようにする。
- 親要素内の配置ルール：入力エリア+画像エリアは、画面全体に対して真ん中に配置する。

順に見ていきましょう。

表示の有無ルール

Group Imageという要素のConditionalというタブの中に、「768px以下の場合にはVisibleにしない」というルールが入っていますね。



この設定を無くしてしまうと、画像は消えません。

親要素内のルール

Group Sign Up/Log in 2 colという要素のLayoutタブで以下の設定がされていますね。

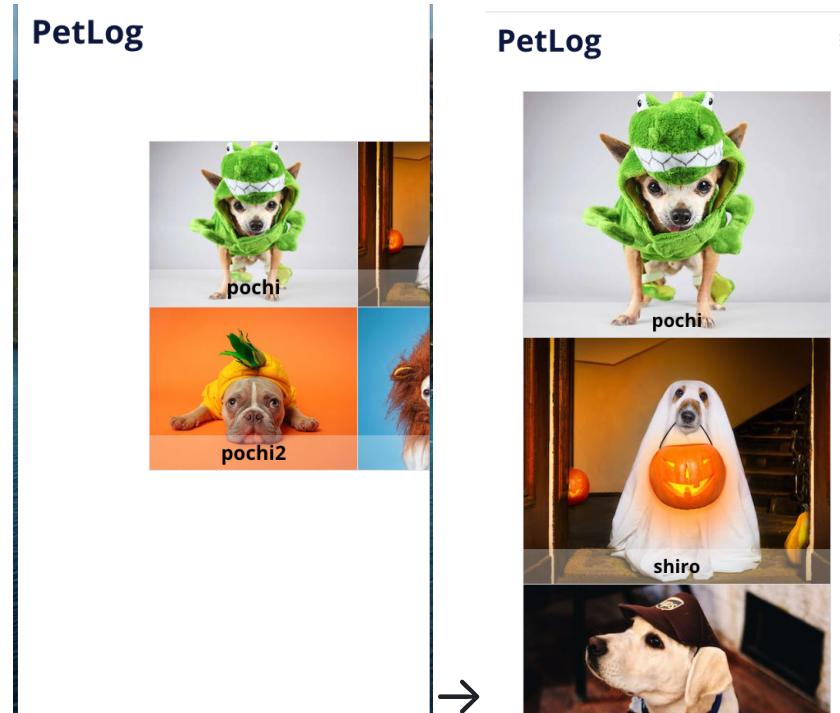
- 子ども要素は行方向にならべる
- 左右は中央寄せで配置する

この設定を変更すると、例えば左に寄ってしまうようになります。

ペット一覧ページにレスポンシブデザインを適用します

次は、ペット一覧ページにレスポンシブデザインを適用します。

対応すると、画面幅に列数が柔軟に変わり、すべてのペットが一覧で見れます。



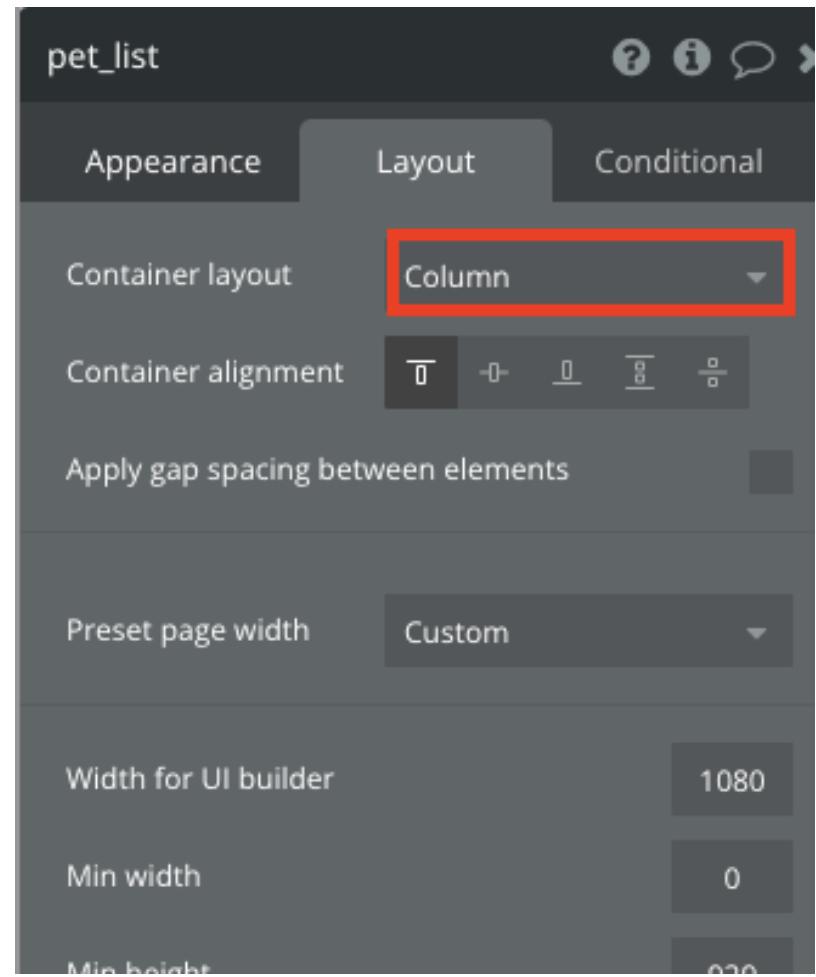
親要素内の配置ルールとして紹介した、Columnを使います。

またサイズの指定やリピーティンググループ特有のレスポンシブ対応用の設定も組み合わせていきます。

まずは、親要素となるページの
Container layout を変更しましょ
う。

今回は、 Column を指定します。

- pet_list の画面を開く
- ページの空白部分をダブルクリックして、ページ 자체の設定ウィンドウを開く
- 設定ウィンドウのLayoutタブを指
定する
- Container layout を Column に
変更する



そうすると、勝手に画面内の要素が縦に並ぶようになりましたね。

Column で設定された親要素の直下の子要素は列方向（垂直方向）に勝手に整列されるのです。この並びを基本として、整えていきます。

Current cell's Pets's Image	Current cell's Pets's Image	Current cell's Pets's Image
Current cell's Pets's Name	Current cell's Pets's Name	Current cell's Pets's Name

続いて、リピーティンググループに対して設定をいれていきます。

- リピーティンググループ `pet list` の設定ウィンドウを開き、`Layout` タブに移動する
- 以下の設定をいれる
 - Horizontal alignment: `centered`

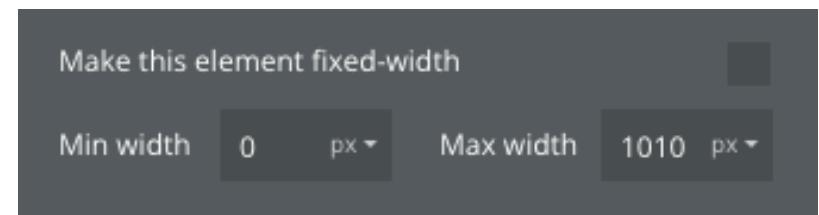


ペット一覧ページ直下の要素が列方向に並ぶようにしていますが、その際にこのリピーティンググループは左右中央寄せに整列されるようになります。

さらに、以下の設定をいれる

- Make this element fixed-width: チ
エックなし
- Min width: 指定なし
- Max width: 1010px

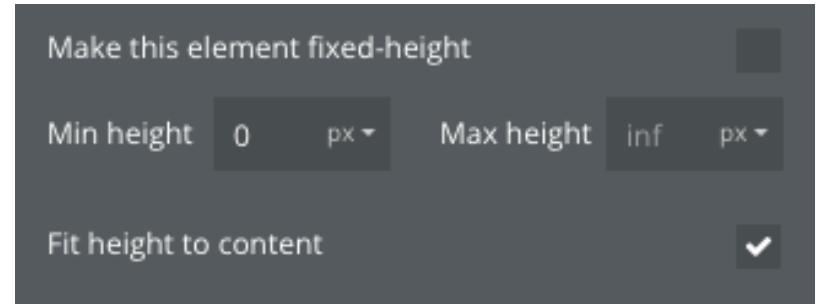
このリピーティンググループの幅は固定にせず、親要素（ここでは画面自体）の幅に応じて柔軟に伸縮します。ただし、広がりすぎると見えにくくなるので、最大幅は1010pxまでとしています。



以下の設定をいれる

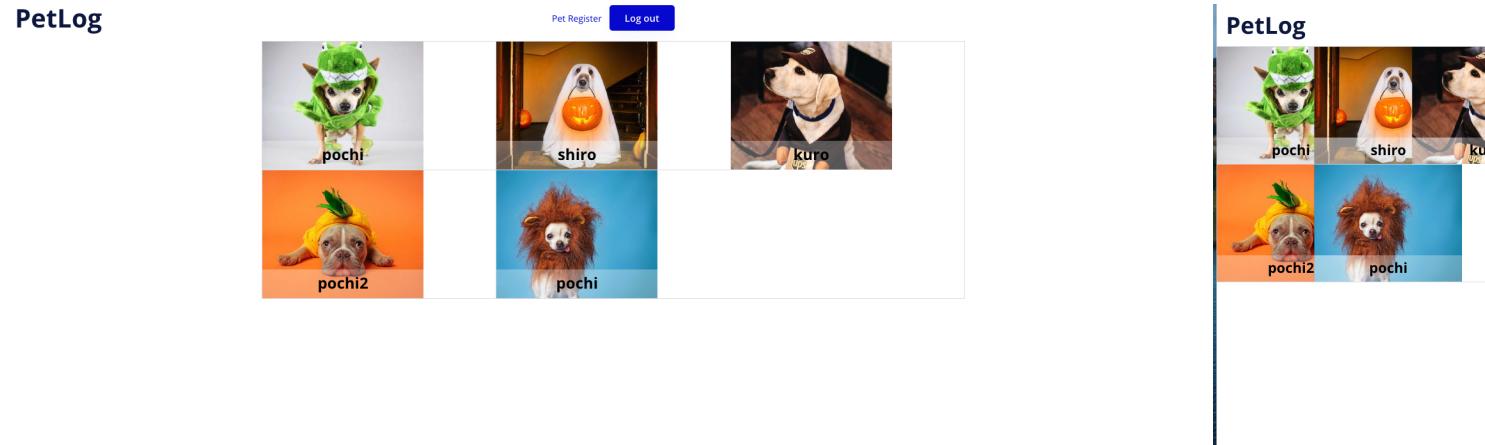
- Make this element fixed-height:
チェックなし
- Min height: 指定なし
- Max height: 指定なし(inf) ※おそらくinfinityの略
- Fit height to content: チェック

このリピーティンググループの高さは固定せず、中身に合わせてどこまでも伸縮するという設定です。



ここまでで、プレビューを表示してみましょう。

まだ不細工ですが、リピーティンググループが中央に寄っており、縦横が伸縮するようになっているはずです。



でも、セルの大きさが意図した大きさになっていたいなかったり、列数も変わらないですね。その設定をしていきましょう。

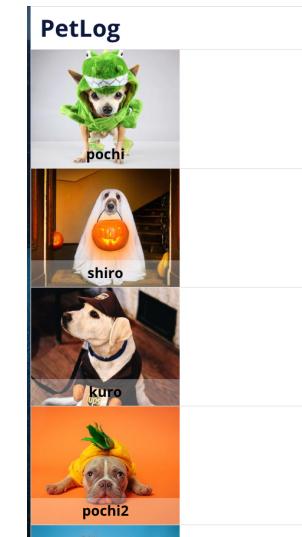
次はリピーティンググループの設定ウィンドウの Appearance タブに移ってください。
以下を設定していきます。

- Set fixed number of rows: チェックなし
- Min height of row: 200px
- Set fixed number of columns: チェックなし
- Min width of column: 200px

列数と行数は固定せず、行と列の最低幅を指定するようにしています。これによって、行数や列数が最低幅を維持しながら、表の大きさによって柔軟に切り替えられます。

では、プレビューを表示してみましょう。

まだ不細工ですが、画面の幅・表の幅に合わせて、列数・行数が変動するようになりました。



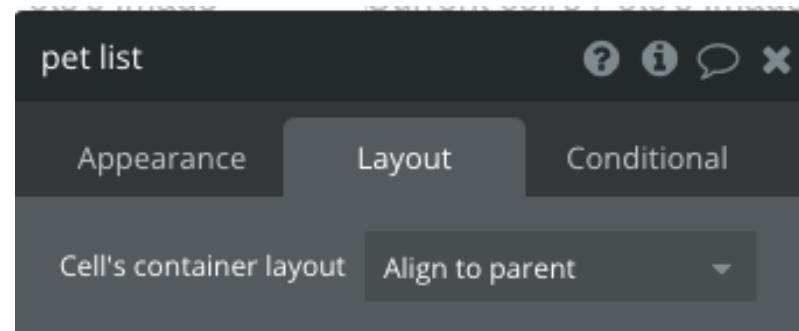
でも、セルの中で左によってしまっていたり、幅を小さくしていく際に意図しない余白ができててしまっていたりしますね。修正していきましょう。

またリピーティンググループの設定ウィンドウの Layout タブに移ってください。

以下を設定します。

- Cell's container layout: Align to parent

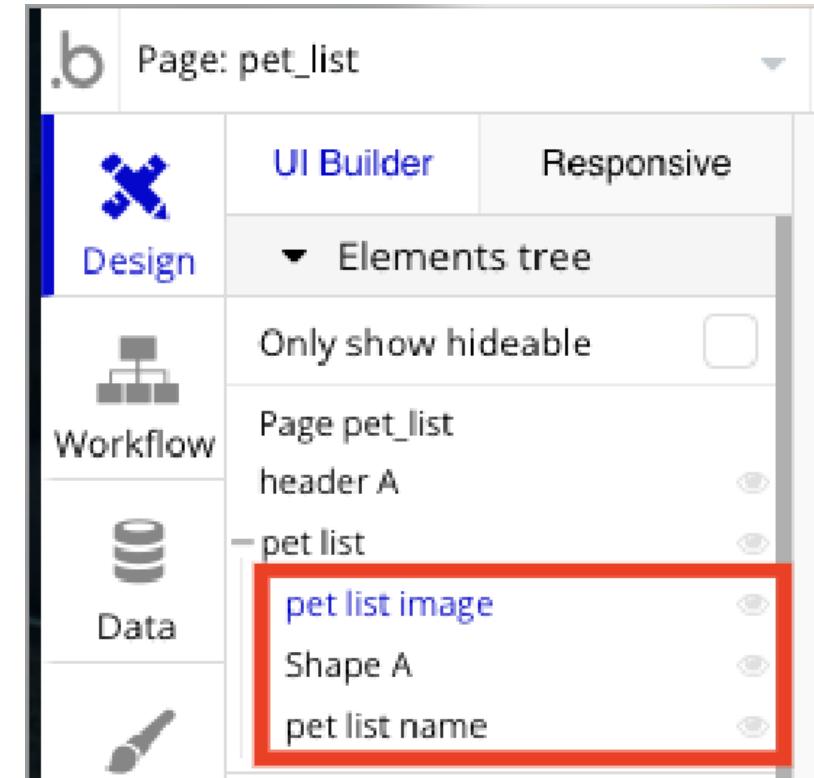
ログインページで使った設定ですね。親要素（ここではリピーティンググループの一つ一つのセル）に対して、相対位置で要素を配置できるようになります。



次にセルの中身の要素のレイアウトを
ていきます。

要素が重なっていて選びにくいので、画
面左の **Elements tree** から指定しま
す。

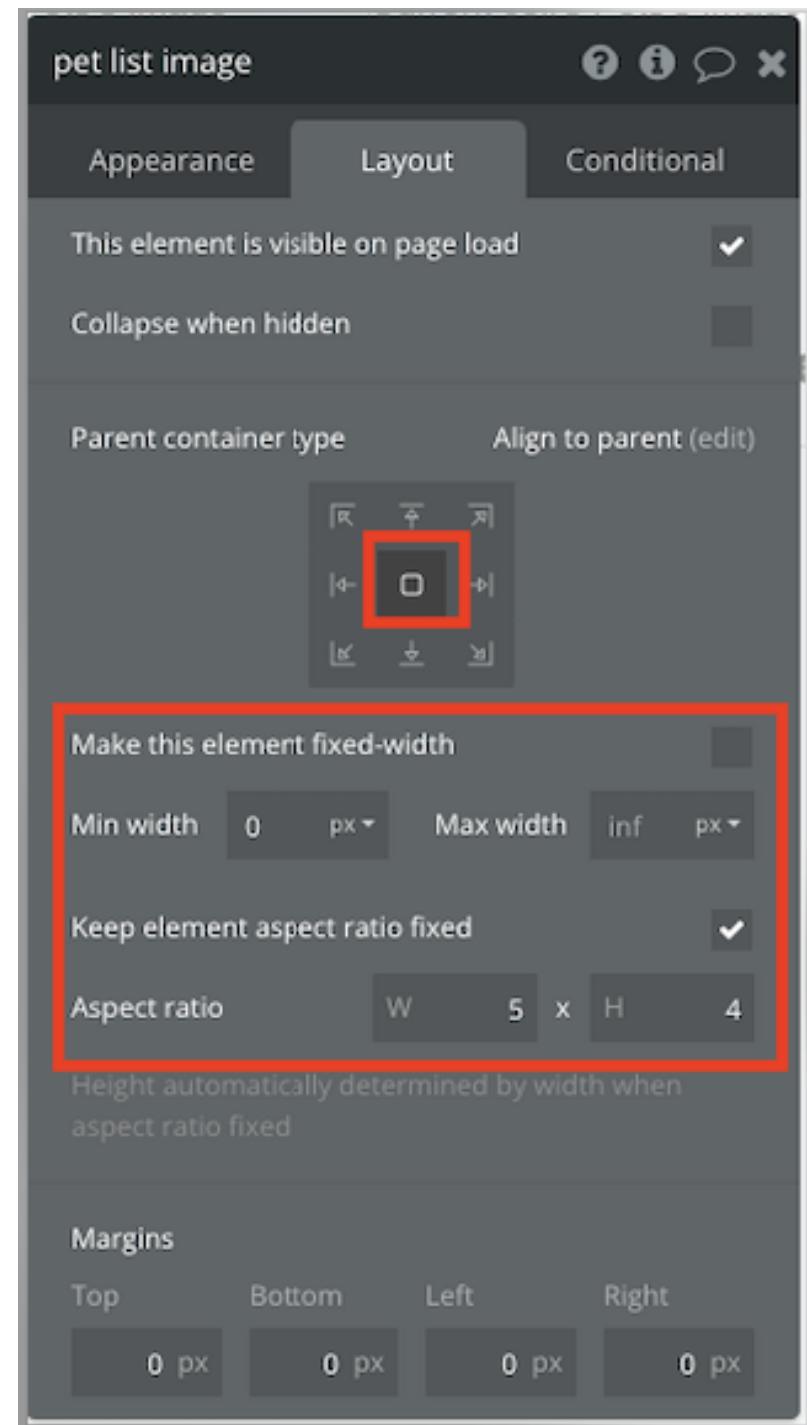
- Designメニューの **UI Builder** の
最上部に **Elements tree** という
部分があります。
- **Only show hideable** のチェック
がついていたら外してください。
- * **pet list** しか表示されていない
場合は、**+** をクリックしてツリー
を開いてください。

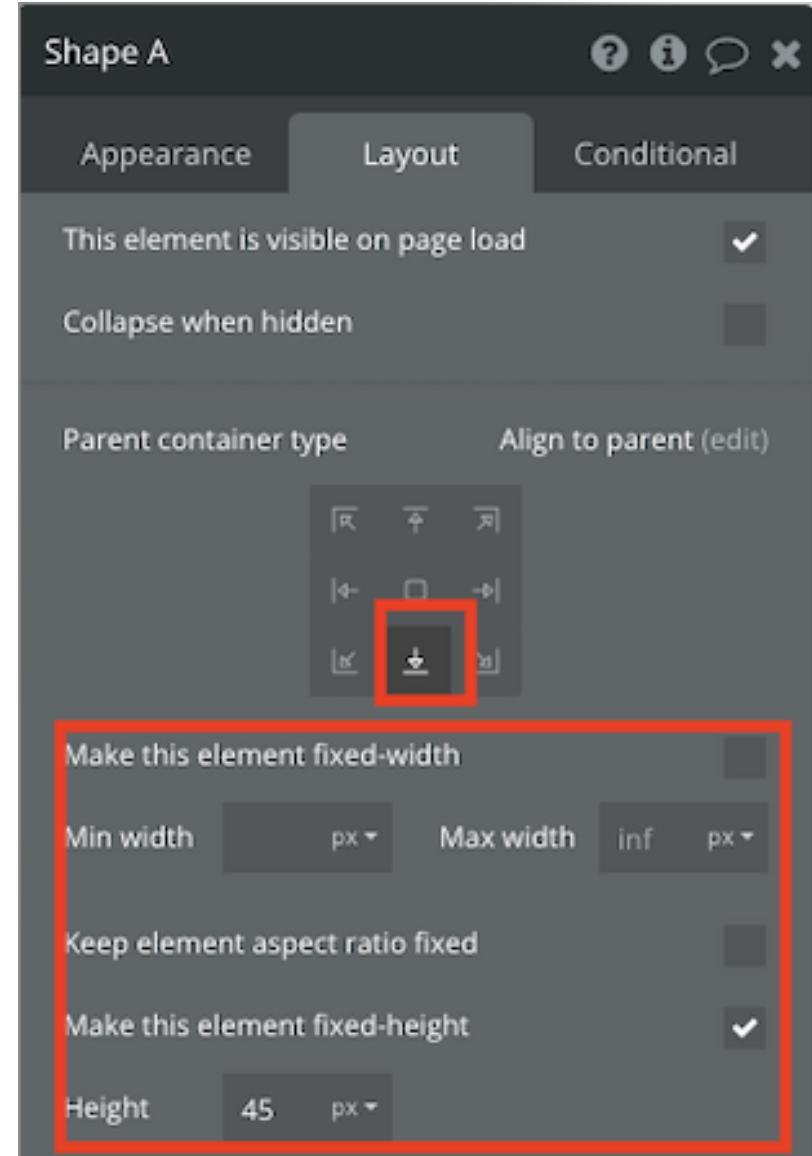


では、`pet list image` から順に設定していきます。

- `Elements tree` の `pet list image` をクリックして設定ウィンドウを開く
- `Layout` タブ内で右の画像と同様に設定してください。

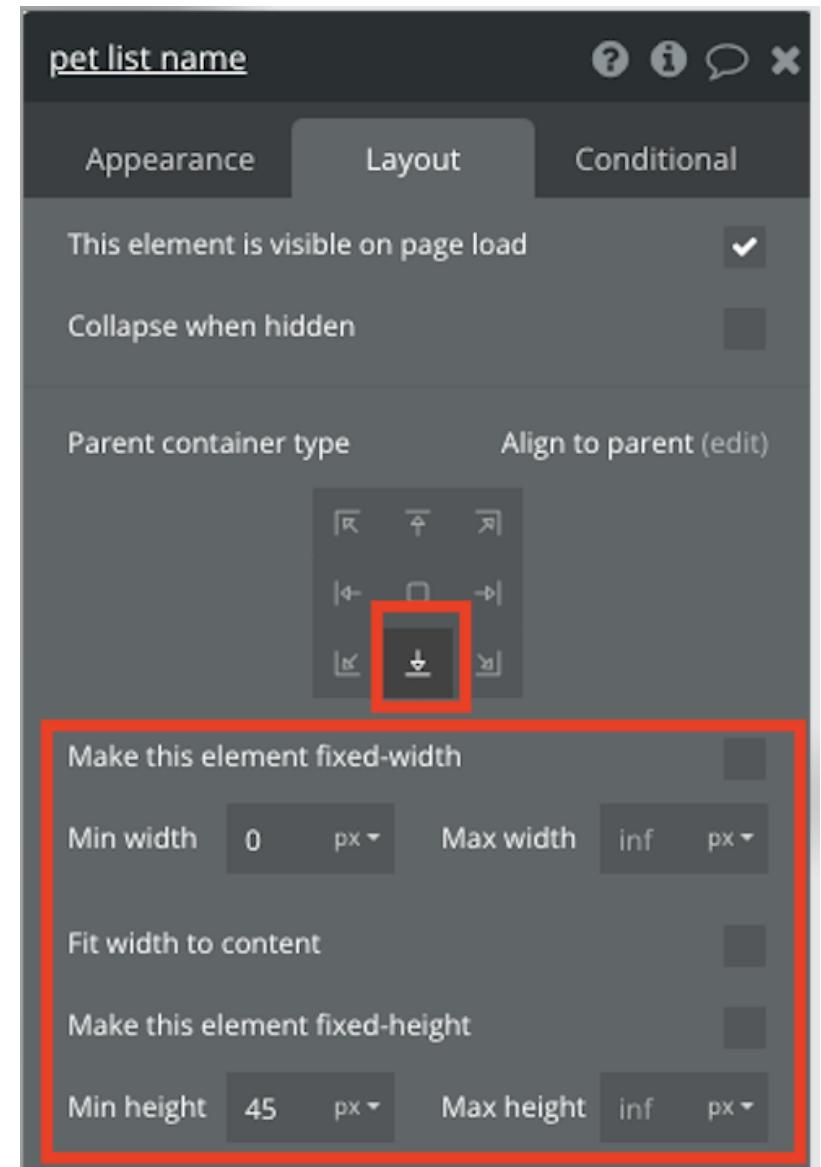
セル内の真ん中に配置し、縦横比を4:5に保ったままセル内いっぱいまで大きく表示する、という設定になります。





次は、Shape A の Layout に右の画像と同様の設定をいれます。

セル内の下部に配置し、高さ45pxを保ったまま左右いっぱいまで大きく表示する、という設定になります。



次は、`pet list name` の Layout に右の画像と同様の設定をいれます。

Shape A と同じ内容ですね。

プレビューします。

惜しいですね。あとはヘッダーの位置や余白が気になります。

PetLog

Pet Register Log out

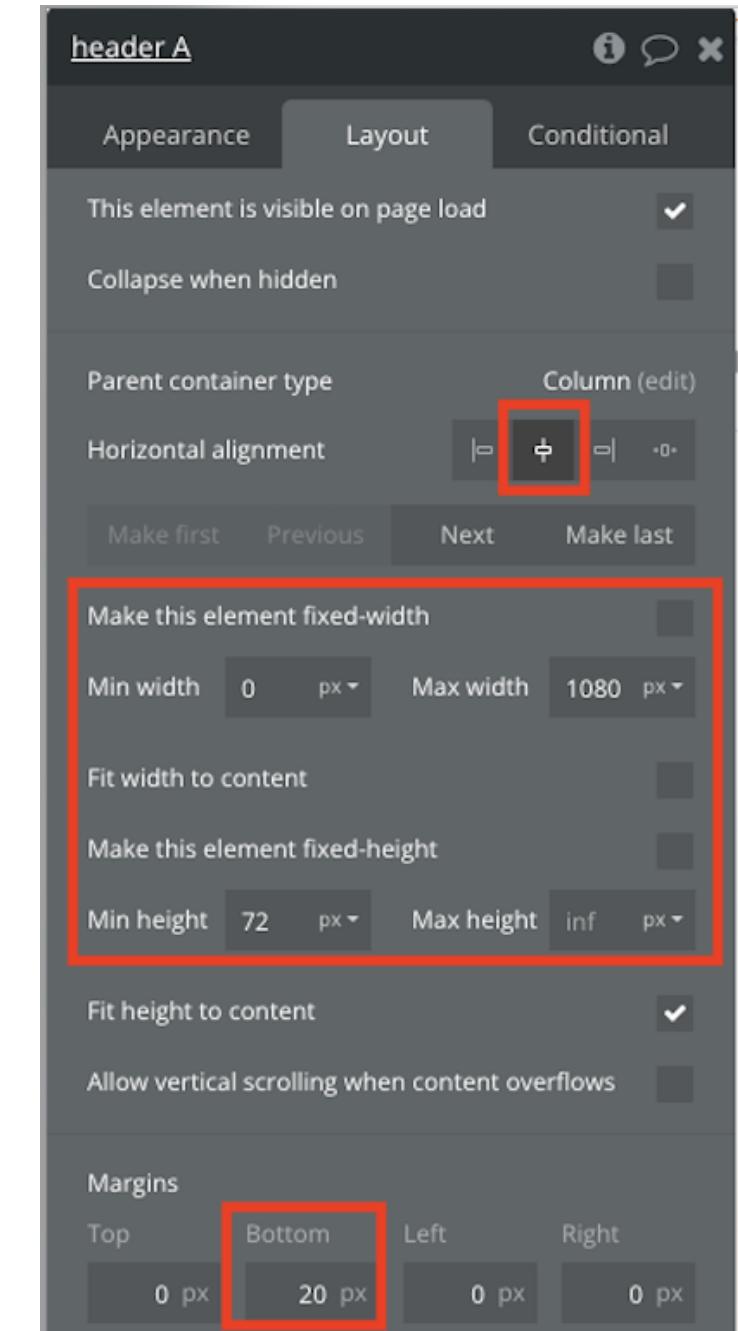
pochi shiro kuro pochi2

pochi

PetLog

pochi

shiro



ヘッダーの設定ウィンドウを開き、
Layout に右の設定を入れます。

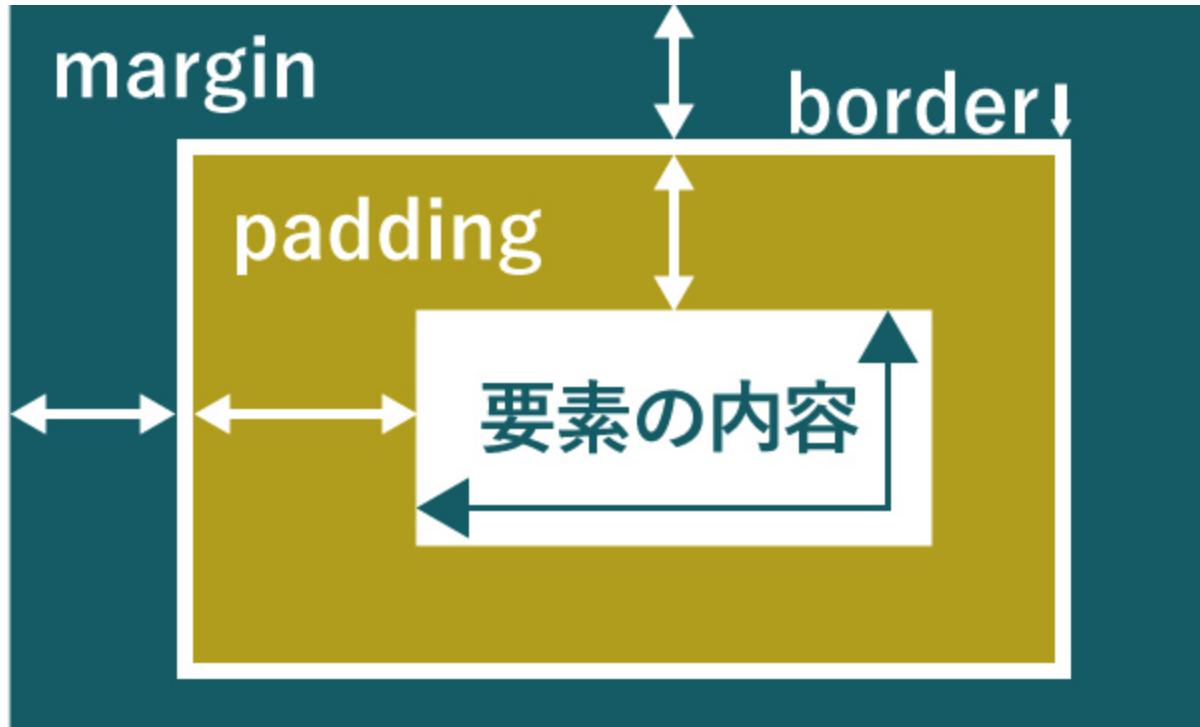
ページ内で列方向に並べられる際に、中央揃えで表示する。横幅は固定せずページ幅に応じて、最大1080pxまで伸びる。下側に20pxの余白をとる。という設定です。

MarginとPaddingについて

Marginという言葉がはじめて出てきましたので、説明します。

余白を表す言葉に、MarginとPaddingというものがあり、それぞれ以下を意味します。

- Margin : 要素の境界の外側の余白
- Padding : 要素の境界の内側の余白

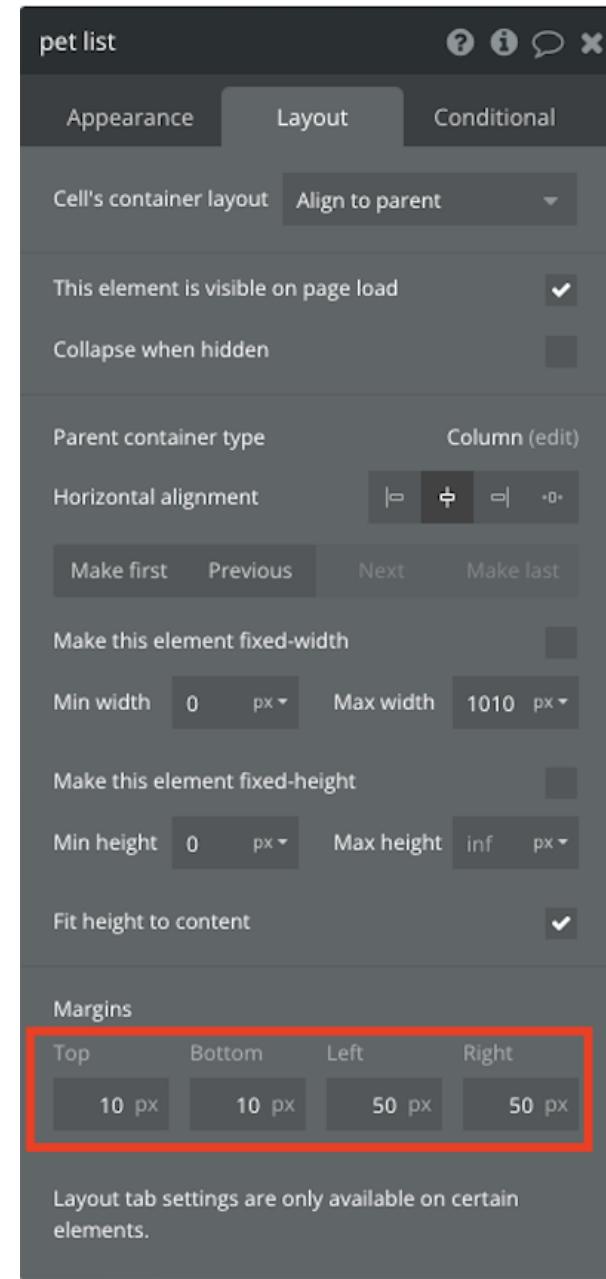


境界線の内と外とで、余白を使い分けたい時に意識してみてください。

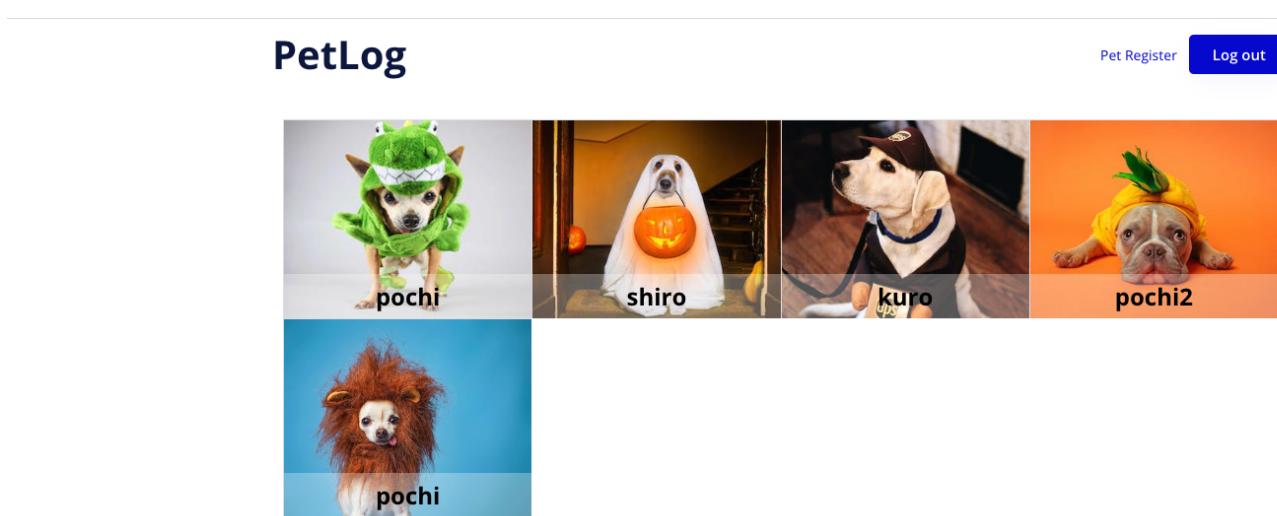
ついでに、リピーティンググループにも余白をいれておきます。

リピーティンググループの Layout タブで以下を設定してください。

上下の余白に10px、左右の余白に50pxをとるという設定になります。



では、プレビューしましょう。



やったー。

なお、ハンバーガーメニューが出てますが動作しないのは気にしないでください。

ペット詳細画面にレスポンシブデザインを適用しよう

ペット詳細画面がせっかくのPC画面の広さを活かせないようになっているので、レスポンシブ対応しましょう。

親要素内の配置ルールのRowを利用して、多段組の設定をします。
また要素の表示、非表示の制御も行います。

PetLog

Pet Register Log out

← Back to list

Image


Name
pochi

Birthday
2022年11月23日

Gender
男の子

Weight Logs

← Back to list

Image


Name
pochi

Birthday
2020年8月18日

Gender
男の子

Weight Logs

← Back to list

10
9
8
7
6
5
4
3
2
1
0

Nov 25, 2022 11:01 pm Nov 25, 2022 11:01 pm Nov 25, 2022 11:01 pm Nov 25, 2022 11:01 pm

→

グループを組み合わせて、こんな段組みをつくります。

PetLog

[← Back to list](#) [Weight Logs](#)

Image
Current Page Pet's Image

Name
Current Page Pet's Name

Birthday
Current Page Pet's
Birthday: formatted as 2022年11...

Gender

A line chart showing weight logs for a pet over seven months. The y-axis represents weight in kilograms, ranging from 0 to 90. The x-axis shows the months from January to July. The weight starts at approximately 65 kg in January, dips to about 60 kg in February, rises to a peak of around 80 kg in March and April, then gradually declines to about 40 kg by July.

Month	Weight (kg)
January	65
February	60
March	80
April	80
May	55
June	55
July	40

55

こんな感じになります。

PetLog

[← Back to list](#)



Name

pochi

Birthday

2020年8月18日

Gender

男の子



[Pet Register](#)

[Log out](#)

[Weight Logs](#)

PetLog

[← Back to list](#)

[Weight Logs](#)

Image



Name

pochi

Birthday

2020年8月18日

Gender

男の子

まずは、このようなグループをつくるっていきます。

Group (Row Container)

Group (Fixed Container)
Image

Current Page Pet's Image

Group (Fixed Container)

Name

Current Page Pet's Name

Birthday

Current Page Pet's
Birthday: formatted as 2022年11...

Gender

Group (Row Container)

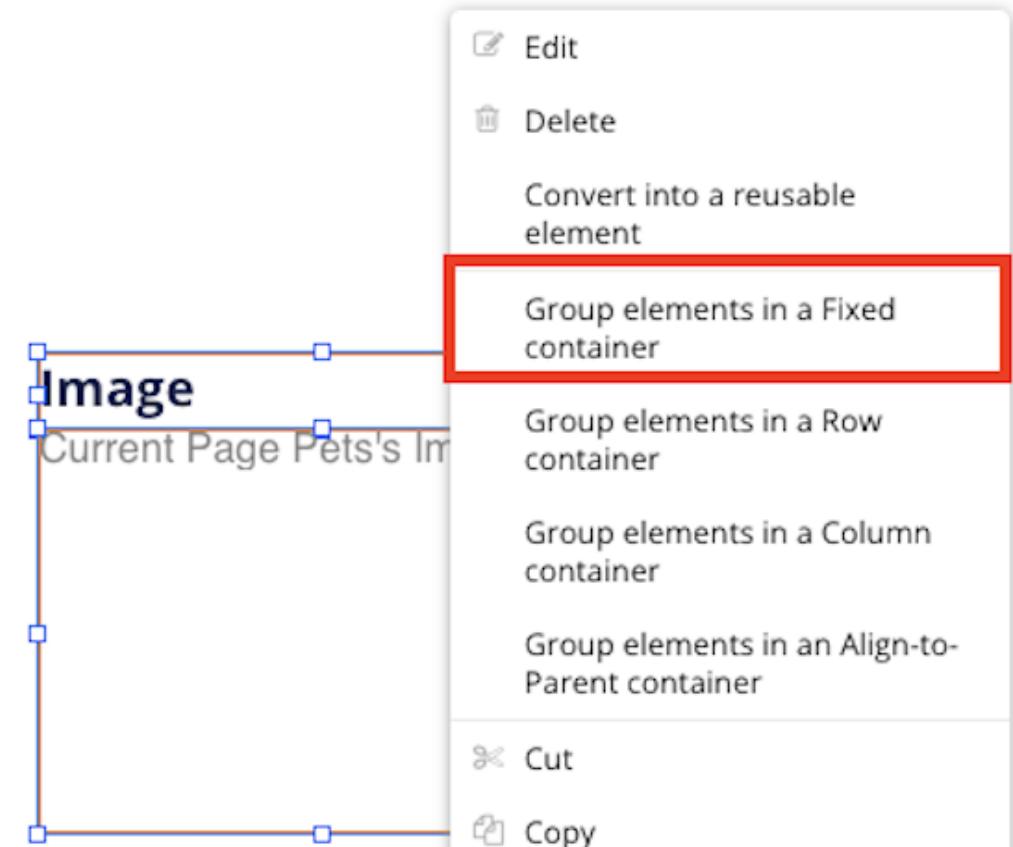
Weight Logs

← Back to list

画像エリアとテキストエリアに分けたいので、それぞれのグループを作成します。

- 画像のラベルと画像表示部分をドラッグして選択する
- 右クリック（もしくは、ダブルタップ）してメニューを出す
- `Group elements in a Fixed container` を指定する。

選択した要素が含まれるグループが作成されます。固定配置（Fixed）のグループとして作成しているので、グループ内では固定の配置になります。



同様に、名前、誕生日、性別の要素をドラッグして、Fixedのグループを作成します。

Image

Current Page Pets's Image

Name

Current Page Pets's Name

Birthday

Current Page Pets's
Birthday:formatted as 2022

Gender

- Edit
- Delete
- Convert into a reusable element
- Group elements in a Fixed container
- Group elements in a Row container
- Group elements in a Column container
- Group elements in an Align-to-Parent container
- Cut
- Copy
- Paste
- Select all
- Copy with workflows
- Paste with workflows
- Bring to front
- Send to back
- Center horizontally
- Center vertically
- Align left
- Align horizontal centers
- Align right

作成したグループは、あとでわかりやすいように名前をつけておきましょう。

- Elements tree からグループをクリックし、設定ウィンドウを出す
- 設定ウィンドウのタイトル部分をクリックすると、名称を変更する
 - Group A → Image Group
 - Group B → Text Group

The screenshot shows the UI Builder interface with the 'Elements tree' panel open. The panel has a header with tabs: 'UI Builder' (selected), 'Responsive', and 'Design' (highlighted in blue). Below the tabs is a section titled '▼ Elements tree' with a checkbox labeled 'Only show hideable'. The tree view lists several elements:

- header A
- Link ← Back to list
- Link Weight Logs
- + Group A (with a plus sign icon)
- + Group B (with a plus sign icon)

Each element has a small eye icon to its right, indicating visibility status.

また、グループと要素がピッタリ重なっていたら、あとでグループを選択したいとなった時に選択しづらいので、少しグループを大きく広げておきましょう。

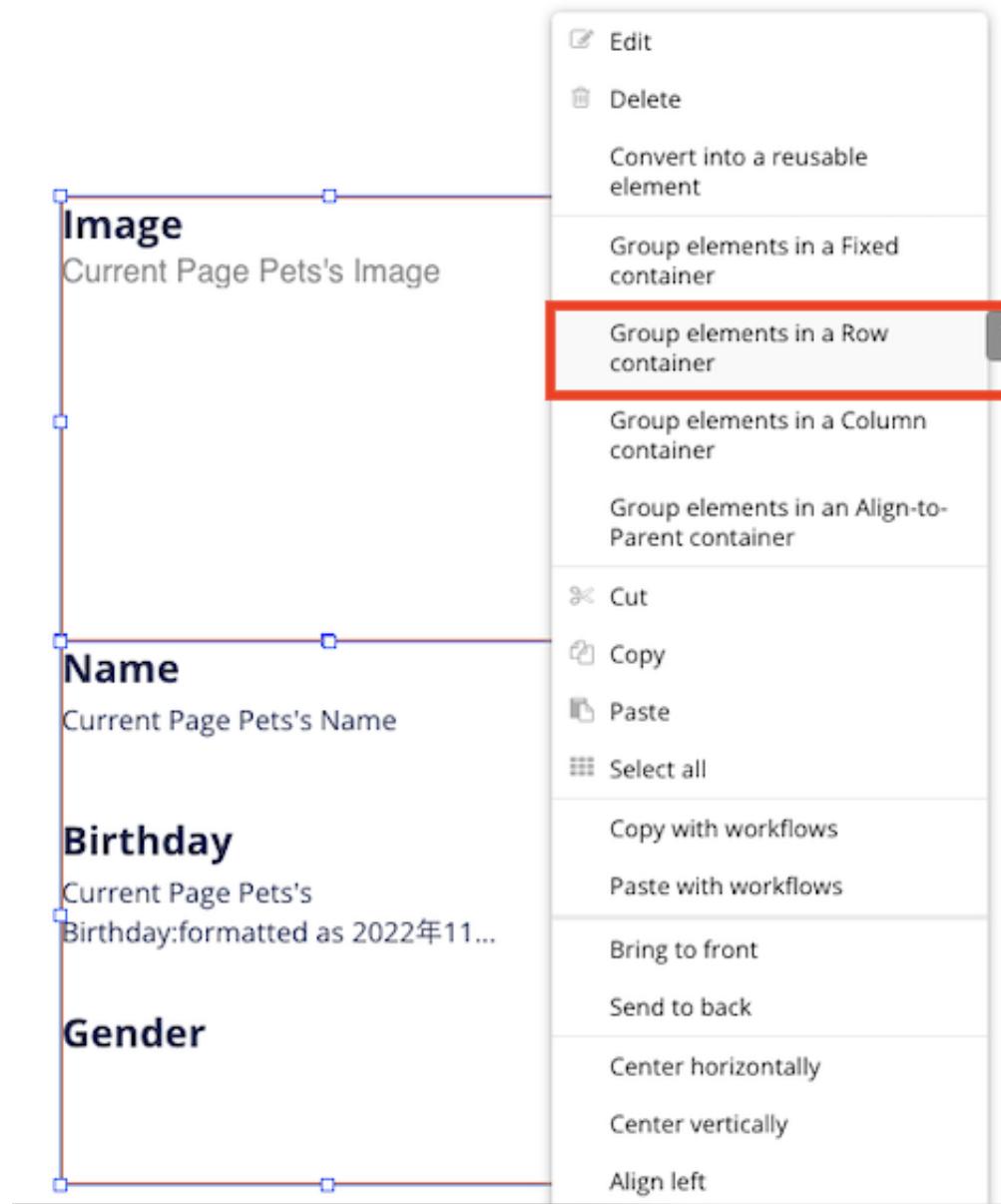
- Elements tree からグループをクリックし、設定ウィンドウを出す
- Groupの要素の端の四角をドラッグして広げる



続いて、画像エリアとテキストエリアが
横に並ぶような段組みを作りたいので、

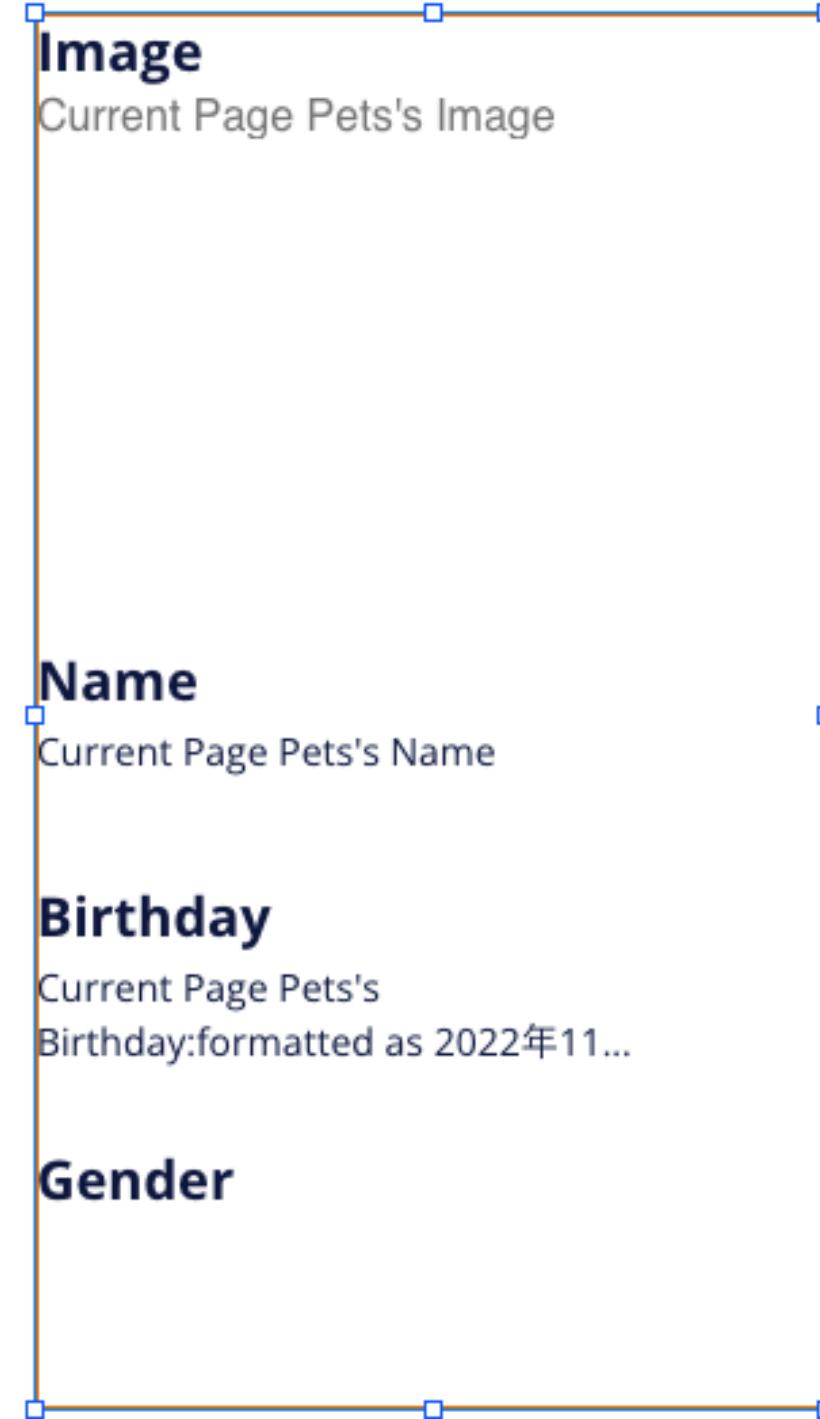
Image Group と **Text Group** を選択
して、さらにグループを作ります。

- **Image Group** を選択（少し大きく
しておいたので選択しやすいです
ね）
- **Text Group** も Shift を押しながら
選択（二つのグループが選択状態
になる）
- 右クリック（もしくは、ダブルタ
ップ）してメニューを出す
- **Group elements in a Row
container** を指定する。



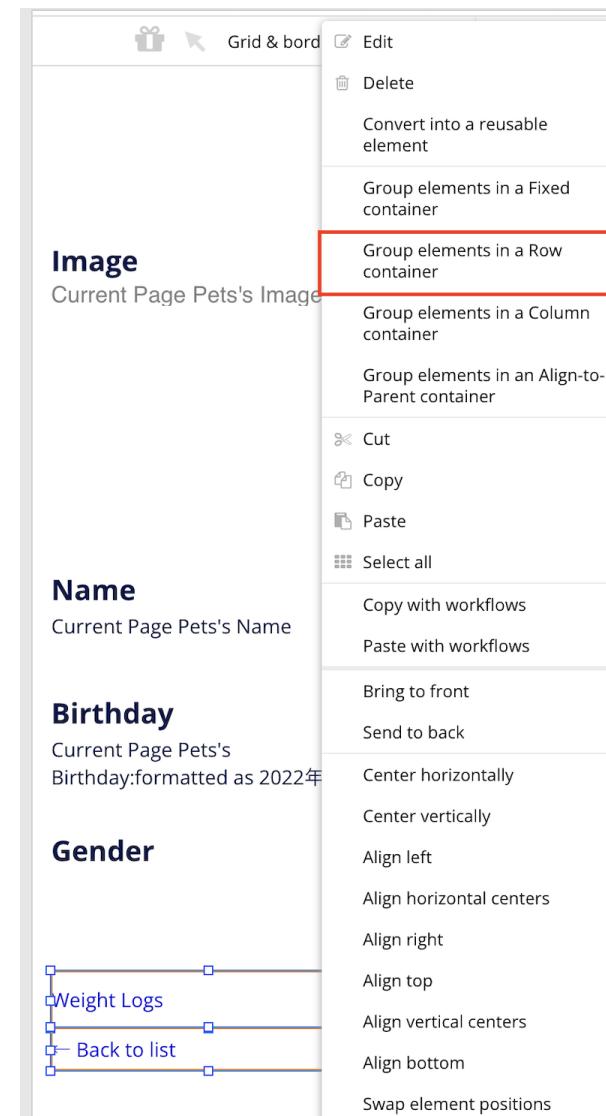
後でグループを選択しやすいように、また少し広げておきましょう。

グループに対して `Row` で指定することで、子要素を行方向（水平方向）に並べるグループになっています。
行は自動で折り返すようになっており、ここで作成した状態では、グループの幅が小さいので折り返されている状態になります。試しに幅を十分に広げると、行方向に並びます。



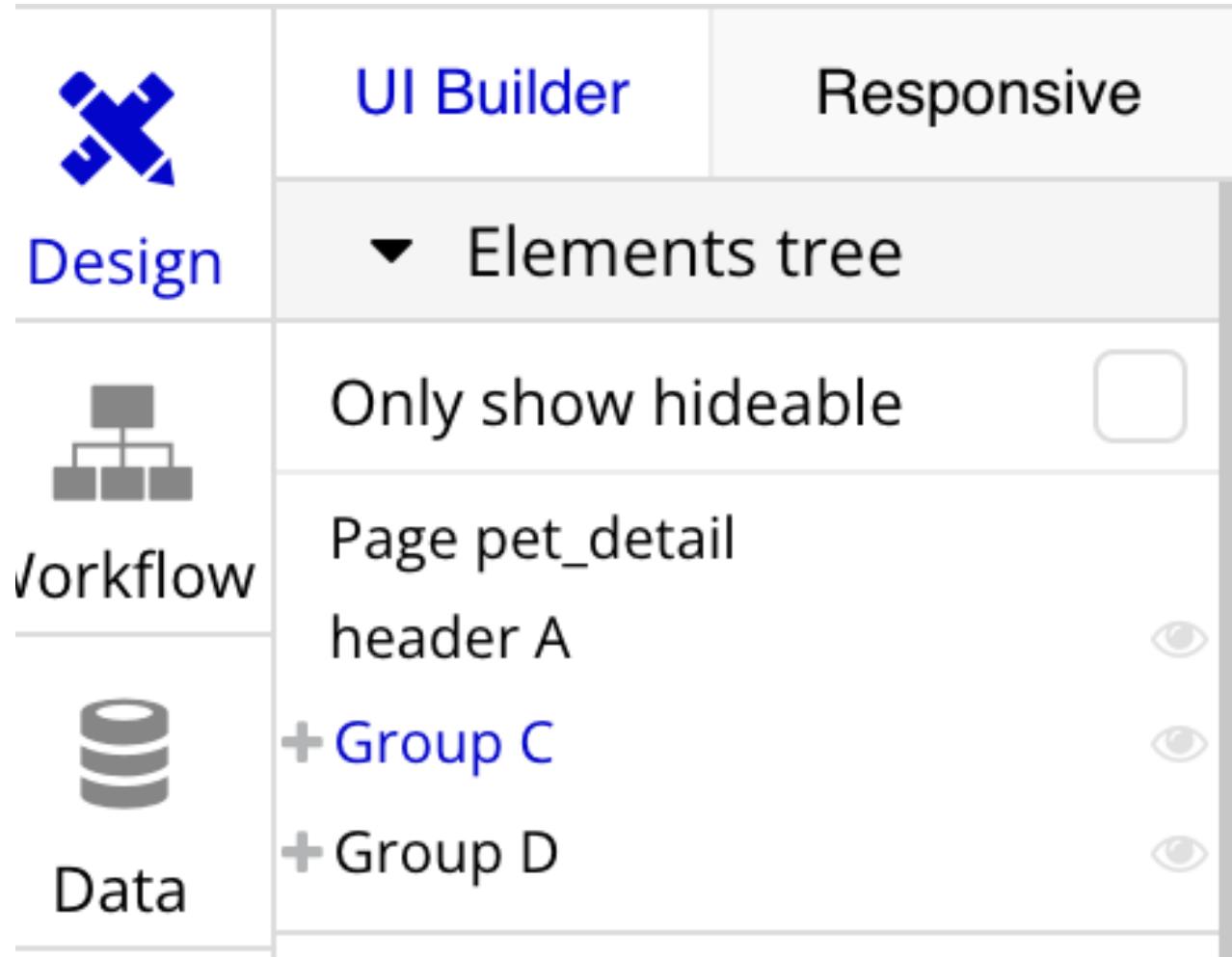
Weight Logs リンクと **Back to list** リンクも横に並ぶような段組みを作りたいので、グループを作ります。

- **Weight Logs** リンクを選択
- **Back to list** リンクをShiftを押しながら選択
- 右クリック（もしくは、ダブルタップ）してメニューを出す
- **Group elements in a Row container** を指定する。
- また少し広げておきましょう



作成したグループは、あとでわかりやすいように名前をつけておきましょう。

- Elements tree からグループをクリックし、設定ウィンドウを出す
- 設定ウィンドウのタイトル部分をクリックすると、名称を変更する
 - Group C → Contents Group
 - Group D → Navigation Group



グループをたくさんつくりましたが、うまく作れていれば、右のような状態になっているはずです。

Group (Row Container)

Group (Fixed Container)
Image

Current Page Pets's Image

Group (Fixed Container)

Name

Current Page Pets's Name

Birthday

Current Page Pets's
Birthday:formatted as 2022年11...

Gender

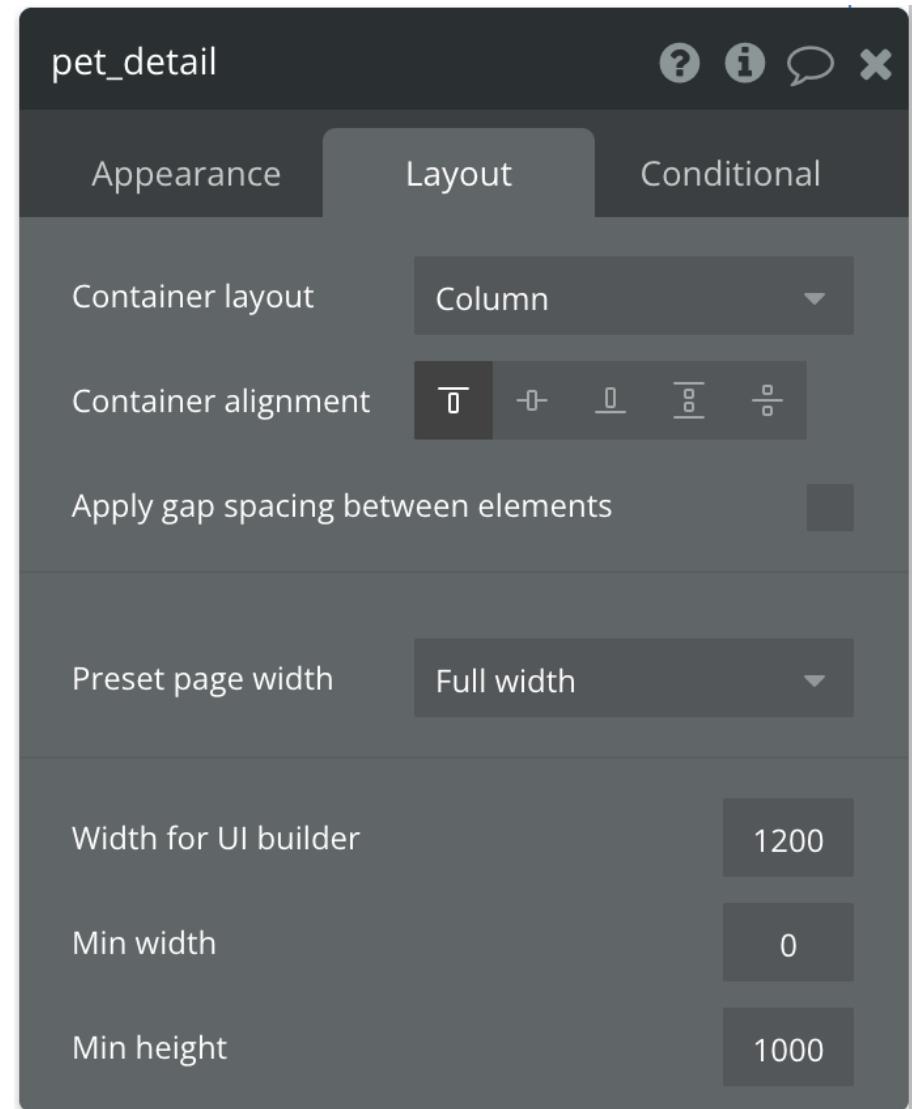
Group (Row Container)

Weight Logs

← Back to list

`pet_detail` ページ自体を `Column` に指定します。

- 画面の空いている部分をクリックして、`pet_detail` を選択する
- `Layout` タブを開く
- `Container layout` に `Column` を指定



PetLog

Image

Current Page Pets's Image

Name

Current Page Pets's Name

Birthday

Current Page Pets's
Birthday:formatted as 2022年11...

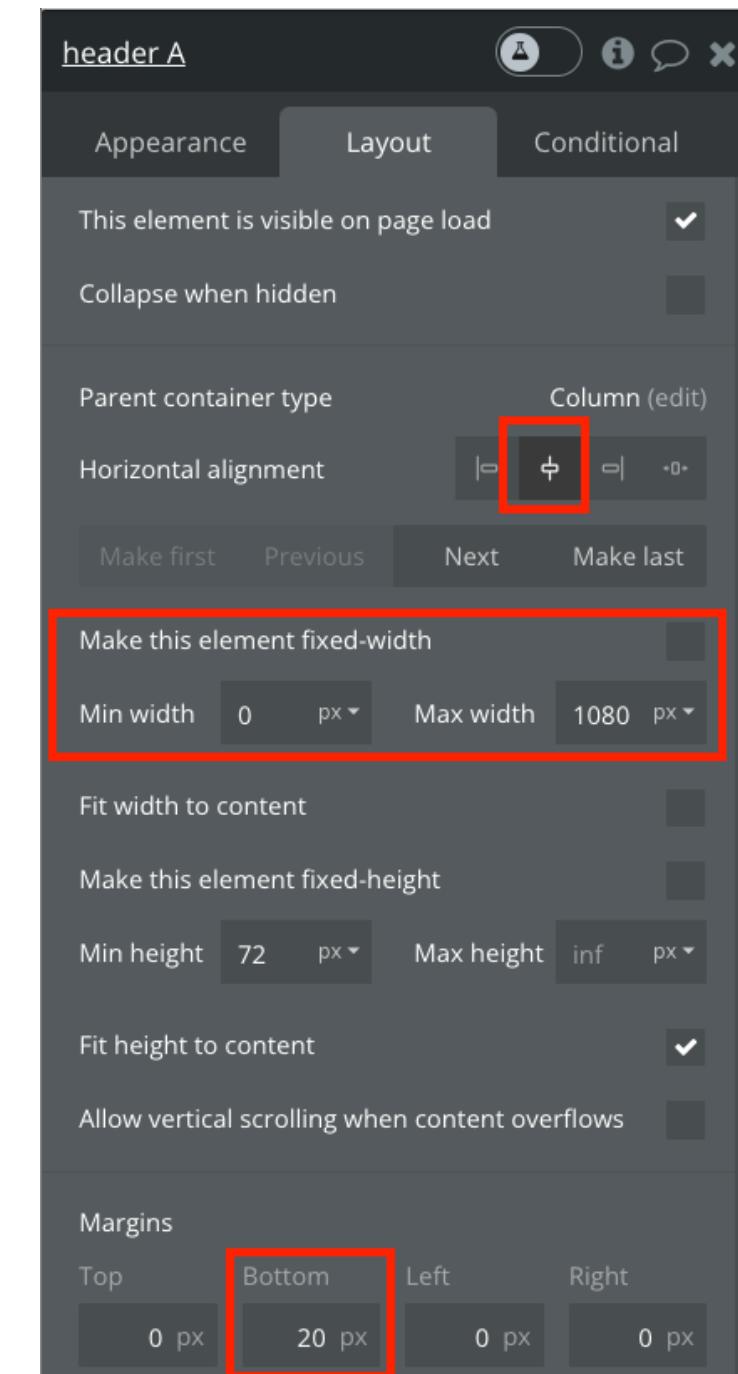
Gender

Weight Logs

← Back to list

そうすると、要素が左側によって、縦一列に並んでいるはずです。

`Column` で指定されたことにより、列方向に整列させられている状態ですね。



ヘッダー（header A）を選択して、
Layoutを右のよう指定します。

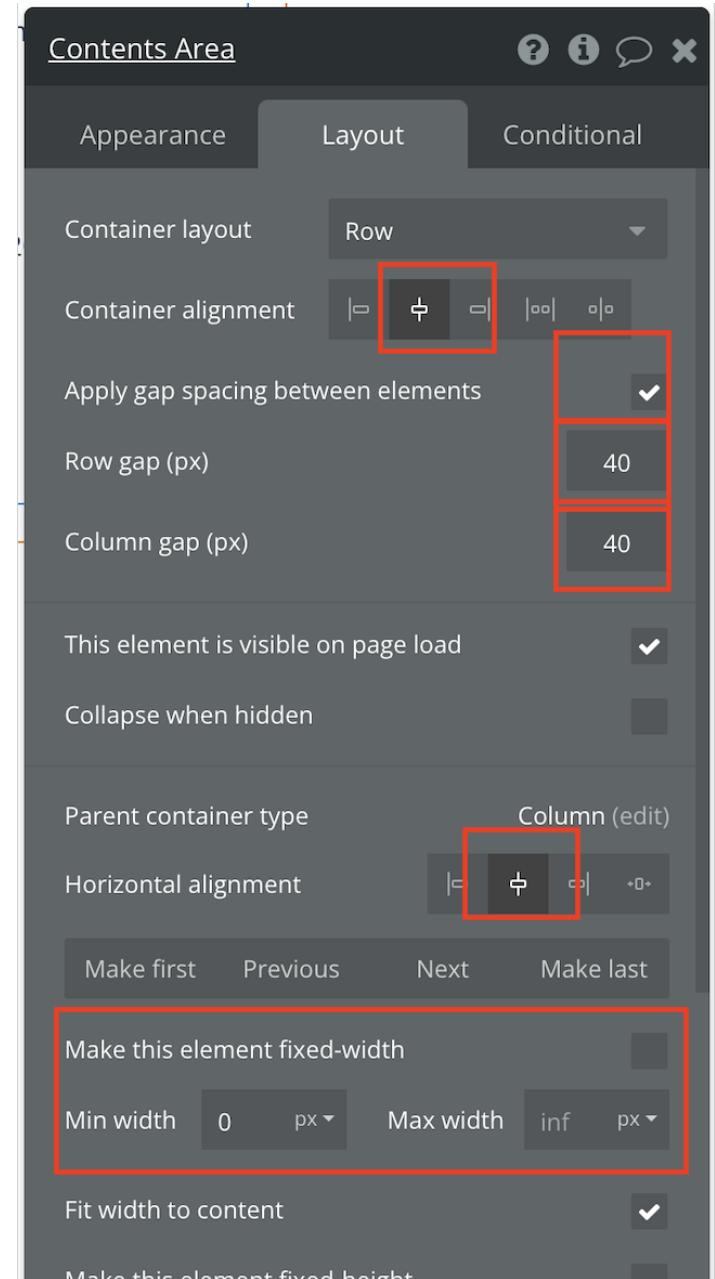
設定内容の意味は、理解できています
ね。

- 列方向に整列する際には、親要素に対して中央揃えにする
- 幅は親要素に合わせて、最大1080pxまで伸び縮みする。
- 下側に20pxの余白をとる。

Contents Group を選択して、Layout を右のように指定します。

新しい設定が出てきましたね。以下の意味になります。

- 行方向に整列させる際には、子要素（ここでは、Image Group と Text Group）は真ん中に寄せる。
- 子要素を整列させるときに、要素間の間の行間隔と列間隔は40pxあける。
- 親要素に対して、列方向に整列させる際には中央に寄せる。
- 親要素に合わせて横幅は伸縮する。



Appearance

Layout

Conditional

Parent container type

Column (edit)

Horizontal alignment



Make first

Previous

Next

Make last

Make this element fixed-width

Min width

0

px

Max width

inf

px

Fit width to content

Make this element fixed-height

Min height

0

px

Max height

inf

px

Fit height to content

シェイプのスタイル

Allow vertical scrolling when content overflows

Margins

Top

Bottom

Left

Right

20 px

20 px

20 px

20 px

Padding

edit style →

Top

Bottom

Left

Right

0 px

0 px

0 px

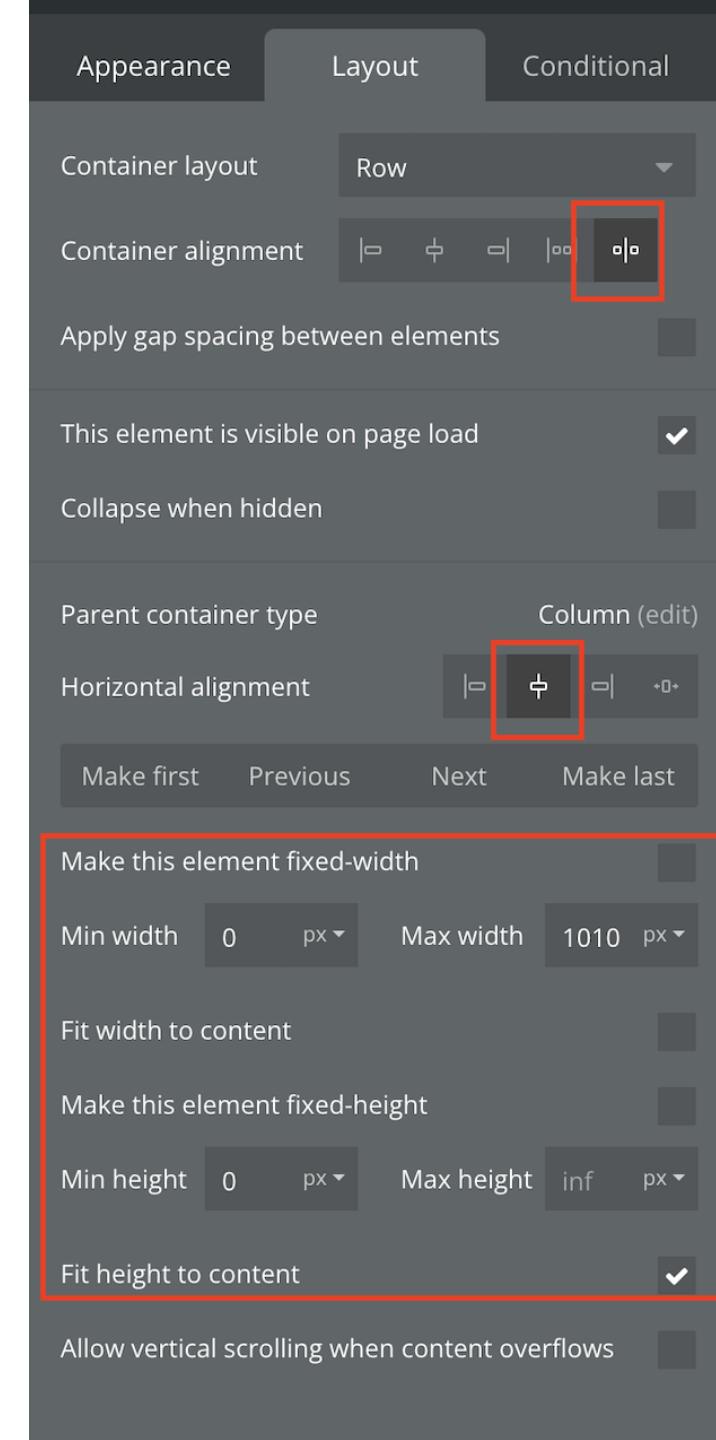
0 px

余白も指定しておきましょう。
もし、Min heightに値が入っていたら0にしておいてください。

次に **Navigation Group** を選択して、
Layoutを右のよう指定します。

以下の意味になります。

- 行方向に整列させる際には、子要素は左右に広げる。
- 親要素に対して、列方向に整列させる際には中央に寄せる。
- 横幅は親要素に合わせて伸縮する。
- 縦幅はコンテンツに合わせて伸縮する。

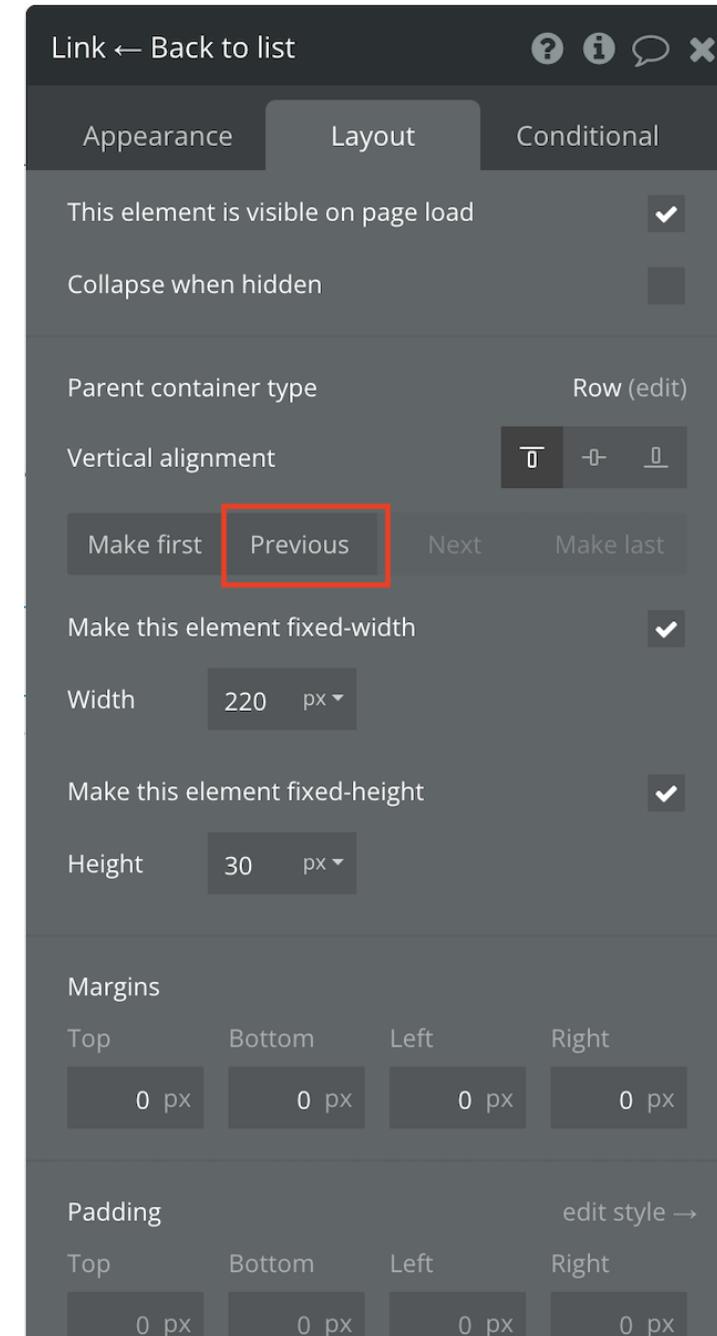


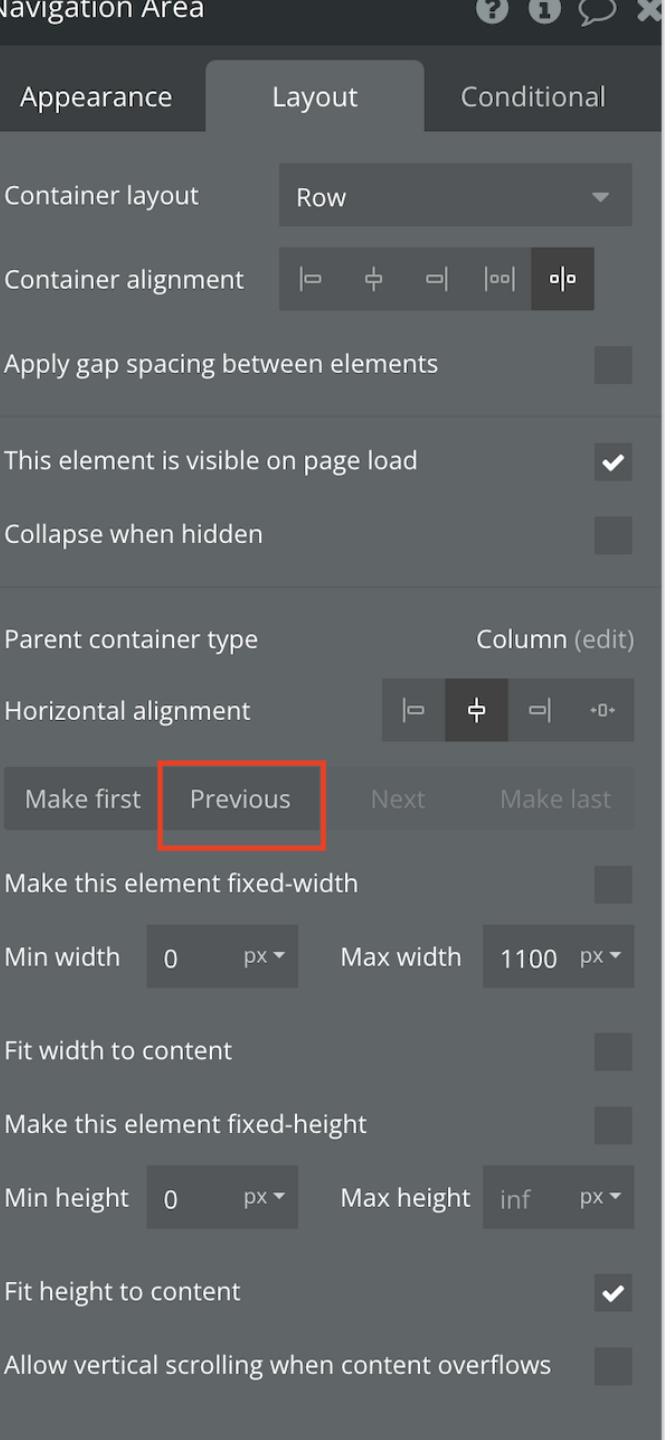
Back to list リンクと Weight

Logs リンクの位置を並べ替えましょ
う。

- Back to list リンクを選択して設定ウィンドウを開く
- Layout タブを開く
- Previous ボタンを押す

このように、Row や Column で指定された親要素内での整列順序を変更することができます。ドラッグでも変更できますが、グループの外にだしてしまったりするので、こちらの方が指定しやすいです。

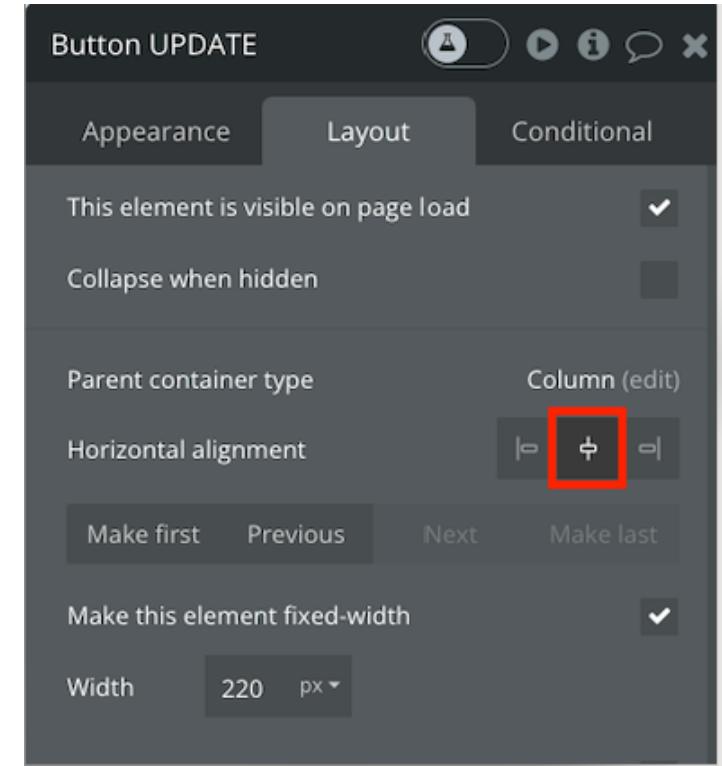




同様に Contents Group と Navigation Group の位置を並べ替えましょう。

- Navigation Group リンクを選択して設定ウィンドウを開く
- Layout タブを開く
- Previous ボタンを押す

Updateボタンを中心寄せになるように
しましょう



プレビューしましょう。

細かなところが多少気になりますが、段組みのレイアウトができました。

スマホだと縦に並びますね。

PetLog

[← Back to list](#)

**Name**

pochi

Birthday

2020年8月18日

Gender

男の子

[Pet Register](#)

[Log out](#)

[Weight Logs](#)

PetLog

[← Back to list](#)

[Weight Logs](#)

**Name**

pochi

Birthday

2020年8月18日

Gender

男の子

<Excercercises>

レイアウトの微妙な部分が気になる方は以下のようなことをやってみてください。

- `Back to list` リンクと `Weight Logs` リンクの横幅をコンテンツに合わせて伸縮するようにする。
- `Navigation Area` の左右に余白をとる。
- 画像サイズを広げたり、テキストの整列を整える

こんな感じになります。

PetLog

[← Back to list](#)

Image



Name

pochi

Birthday

2020年8月18日

Gender

男の子

[Pet Register](#)

[Log out](#)

[Weight Logs](#)

PetLog

[← Back to list](#)

[Weight Logs](#)

Image



Name

pochi

Birthday

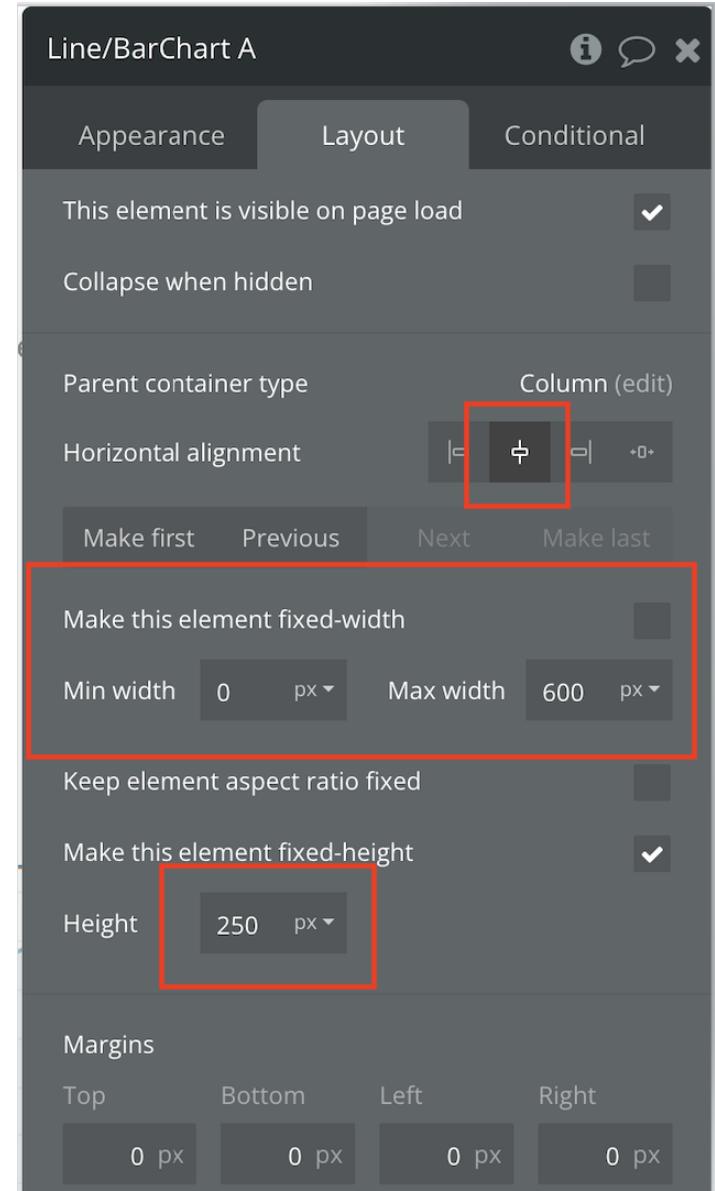
2020年8月18日

Gender

男の子

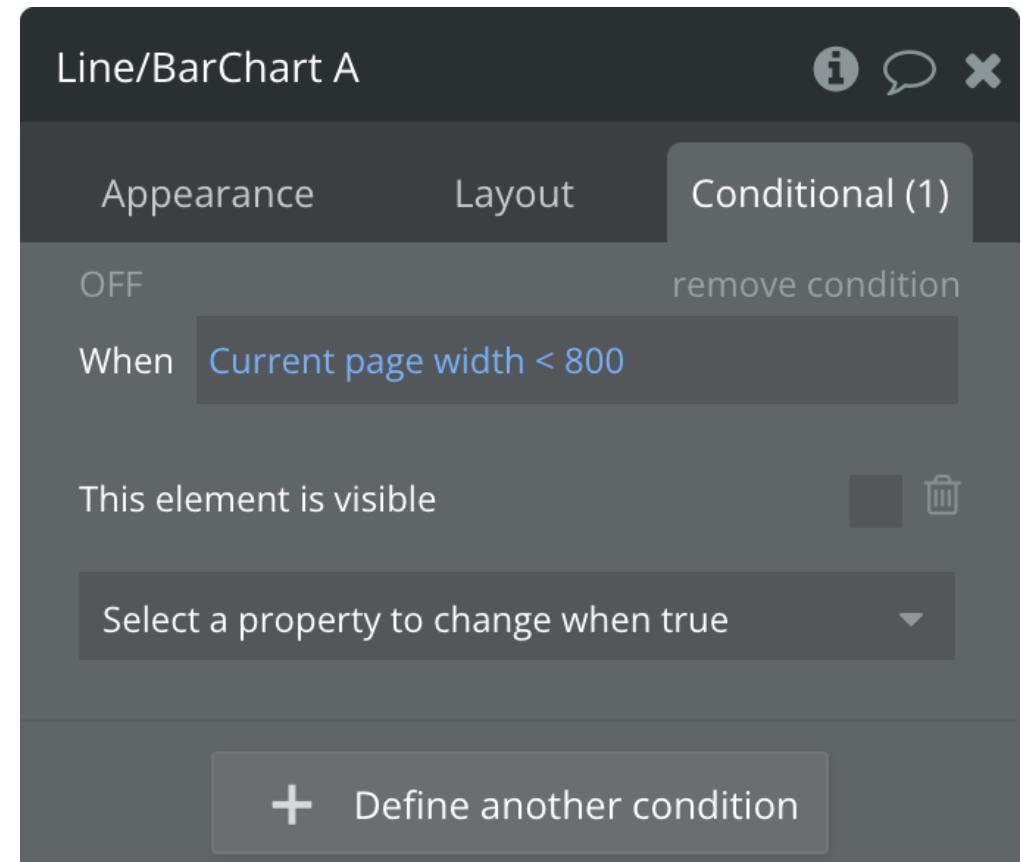
次に体重のグラフをもってき ます

- pet_weight_registerの画面で、グ
ラフをクリックしてコピーする
- pet_detailの画面に貼り付ける
- Layout で Next を押して行っ
て、一番下に配置する
- さらに Layout の他の項目を右の
図のように指定する



スマホで1画面にグラフまで出すと窮屈なので、画面幅が小さいときには表示させないようにします。

- グラフの設定ウィンドウを開いて、**Conditional** タブを開きます
- 条件を新しく作成して、**When** に **Current page width**、**<**、**800** を指定します。いまの画面サイズが800より小さければという意味ですね。
- その場合の変更内容を定義します。**This element is visible** を選択して、チェックなしの状態のままにしましょう



プレビューしましょう。

PetLog

[← Back to list](#)



Name

pochi

Birthday

2020年8月18日

Gender

男の子

[Pet Register](#)

[Log out](#)

[Weight Logs](#)

PetLog

[← Back to list](#)

[Weight Logs](#)

Image



Name

pochi

Birthday

2020年8月18日

Gender

男の子



スマホ表示では、グラフが表示されないですね。

よく使うルール(おさらい)

Bubbleでレスポンシブデザインを実現するために、よく使うルールとして以下のようなものがあります。

1. 親要素内の配置ルール
2. 要素のサイズ決定ルール
3. 表示の有無ルール

これらを組み合わせて、レスポンシブデザインを適用しました。

参考：UIデザインで考えるとよいこと

なんとなくいい感じのデザインにするというのもいいですが、ターゲットユーザーを思い浮かべながら、以下のようなことを意識的に考えてみるとよりよくなります。

1. レイアウトと構造
2. 視覚的要素
3. インタラクション
4. 一貫性と標準化

Styleを使ってみよう

Styleを使ってみよう

- これまでBubbleが標準で用意してくれていたスタイルを利用してきました
- 実際のプロダクトでは、プロダクトにあったデザインコンセプトを描き適用していきます
- ここからスタイルを変更する方法を説明します

Styleの適用方法には大きく3つあります

- 既存のスタイルを編集する
- 個別でスタイルを適用する
- 新しいスタイルを追加する

順にやっていきましょう

既存のスタイルを編集する

既存のスタイルを変更して、ボタンやリンクの色を変更したいと思います。

PetLog

Pet Register

Log out

← Back to list

Weight Logs

Image



Name

pochi

Birthday

2020年8月18日

Gender

男の子



Style variablesをつかってみよう

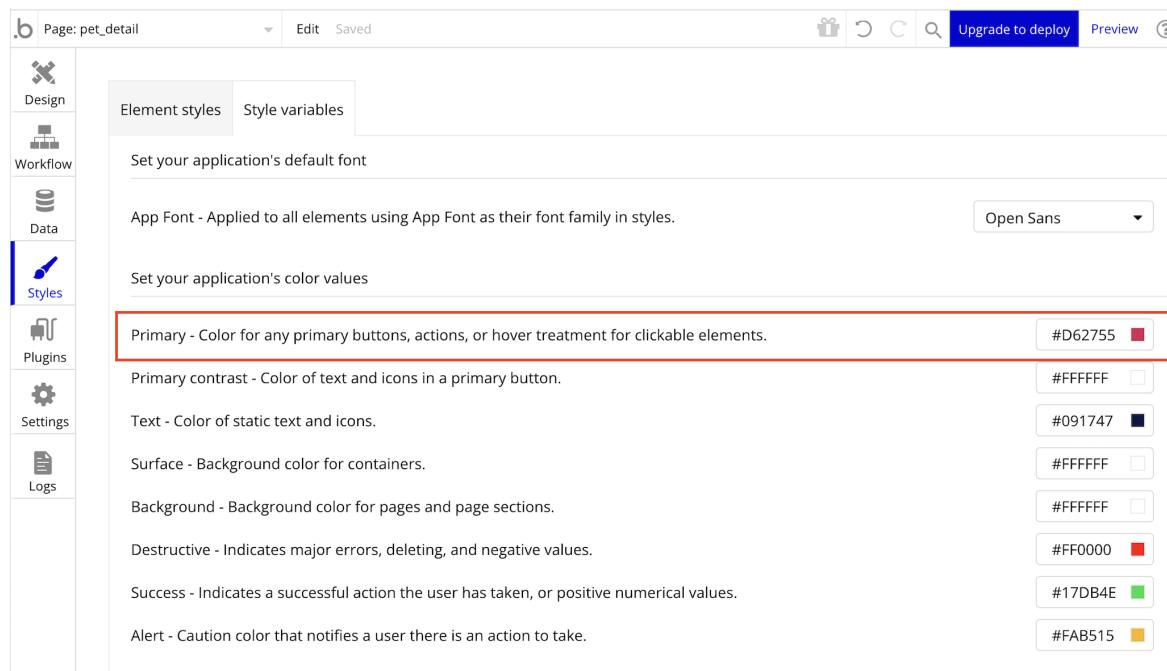
Bubbleでは、基本となる色やフォントが **Style variables** として設定されています。

左メニューの **Styles** > 画面上部のタブで **Style variables** に移動してみてください。ここで指定された色は、スタイルの作成時や編集時に利用することができます。例えば、**Primary**という色設定は、基調色を意味し、Primaryボタンなどで利用されています。

The screenshot shows the Bubble app's interface. On the left is a vertical sidebar with icons for Design, Workflow, Data, Styles (which is selected and highlighted in blue), Plugins, Settings, and Logs. The main area has a header with 'Page: pet_detail', 'Edit' and 'Saved' buttons, and a toolbar with icons for gift, refresh, search, and user. Below the toolbar is a navigation bar with 'Element styles' and 'Style variables' tabs, where 'Style variables' is active. A sub-navigation bar shows 'Element type: Button' and a 'Search by name' input field. To the right are two main sections: one for 'Button - Flat Button' and one for 'Button - Primary'. The 'Primary' section contains a 'Primary Button' style card with a trash bin and a comment icon. Below it is a button labeled 'Find all elements using this style'. To the right is a detailed configuration panel for the 'Primary' style. It includes tabs for 'Appearance', 'Layout', and 'Conditional (1)'. Under 'Appearance', there are settings for 'OFF' (disabled) and 'When' (set to 'This Button is hovered'). A specific setting for 'Background color' is highlighted with a red box and set to 'Primary (#0205D3)'. There are also dropdowns for 'Select a property to change when true' and a button to 'Define another condition'.

`Style variables` の `Primary` の設定を変更すると利用されているすべての箇所に変更が適用されています。

- 左メニューの `Styles` > 画面上部のタブで `Style variables` と選択
- `Primary`を変更する。 (私は暗い目の赤にしたいので、 `#D62755` で指定します。)



Primaryボタンのスタイルを確認すると、変更されていますね。

The screenshot shows the Figma style editor interface. At the top, there are two tabs: "Element styles" (selected) and "Style variables". Below the tabs, there are filters for "Element type" set to "Button" and a search bar. There are also "Apply theme" and "Add style" buttons.

In the left sidebar, there are three style variants listed: "Button - Flat Button", "Button - Outline Button", and "Button - Primary". The "Button - Primary" variant is currently selected, indicated by a blue border around its name.

The main area displays the "Primary Button" style. It includes a preview box containing a red button with the text "...edit me...". To the right of the preview is a conditional panel titled "Conditional (1)". The panel shows a condition "When This Button is hovered" and a setting for "Background color" which is "Primary (#D62755)". There is also a dropdown menu labeled "Select a property to change when true" and a button to "Define another condition".

画面をプレビューしてみましょう。

ログイン／ログアウトボタンなど、基調となっていたボタンやリンクの色が変っています。

PetLog

[← Back to list](#)

[Pet Register](#)

[Log out](#)

[Weight Logs](#)

Image



Name

pochi

Birthday

2020年8月18日

Gender

男の子



Style variablesの使いどころ

このように、 Style variables を編集することで、標準の基調カラーを一括で変更できます。また、新たにスタイルを作成・編集する際も自分でルールを決めて Style variables を利用しておくことで、あとから一括で変更するなどメンテナンスが楽になります。

次に、個別でスタイルを指定してみましょう

ヘッダーロゴを基調色にあわせつつ、フォントももう少しかわいらしい感じにしたいと思います。

PetLog

[← Back to list](#)

Image



Name
pochi

Birthday
2019年11月18日

Gender
女の子

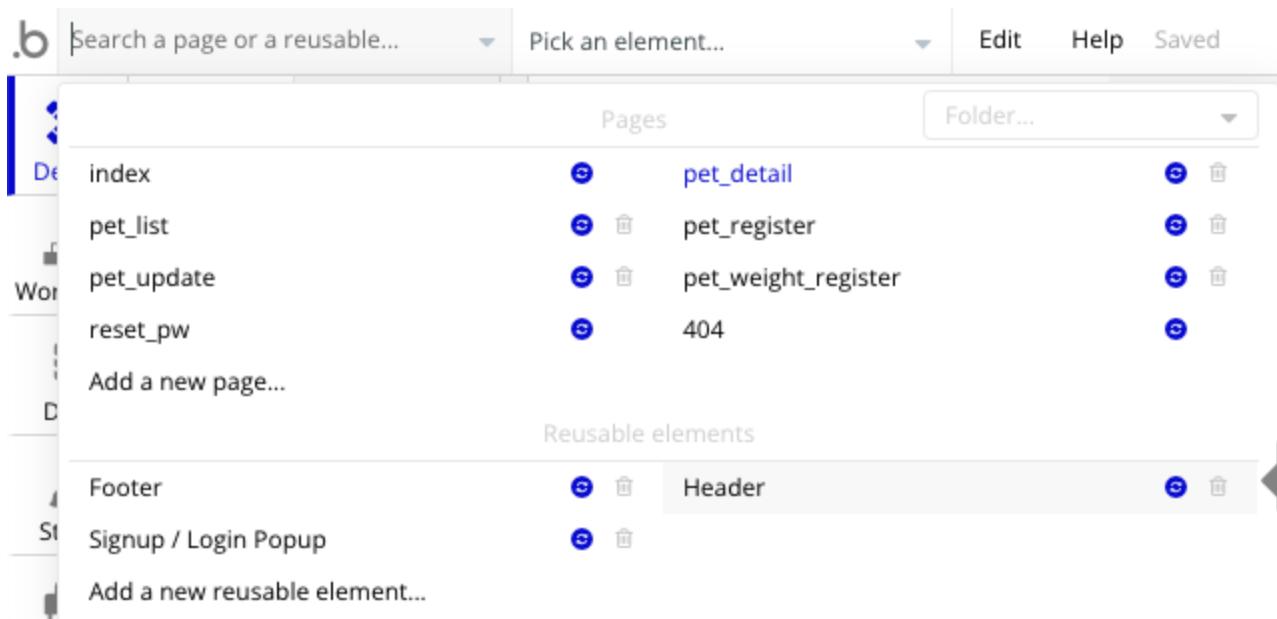
UPDATE

Weight Logs

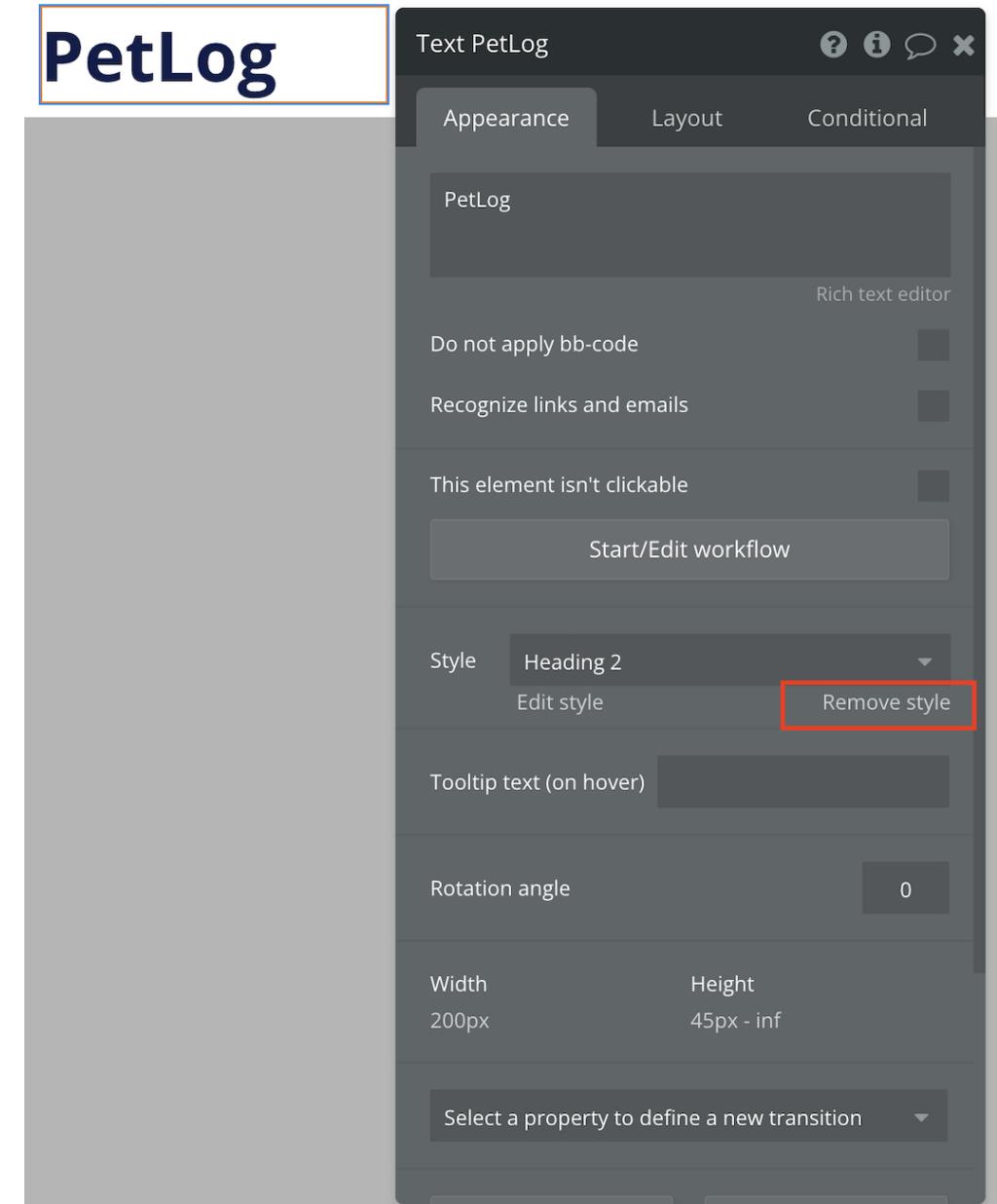


LOG OUT

b口ゴ右のメニューを開いて、headerを選びます

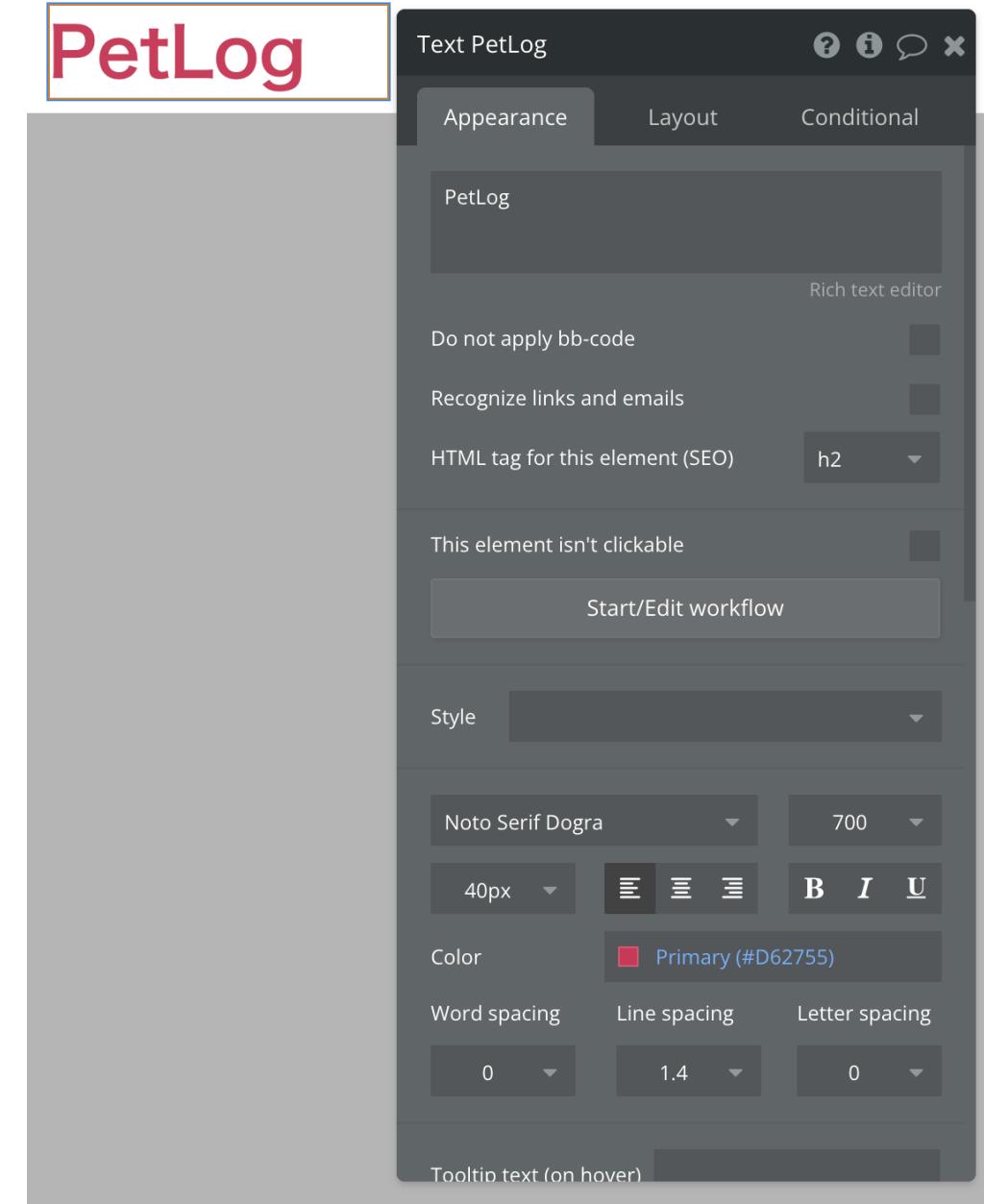


- ロゴをダブルクリックして、設定を開く
- 設定ウィンドウの **Style** の部分に移動する
- プルダウンの右下あたりの **Detach style** をクリック
 - 定義されたStyleを適用するのではなく、個別で指定できるようになります



好きなスタイルを指定しましょう

- 私はフォントカラーは、 Style variables で指定したPrimaryカラー
- フォントは Noto Sans Mono が気に入ったので、それを指定します
- フォントを変えると、ロゴが見切れたりすることがあるので、幅は適宜調整してください



プレビューしましょう

変わりましたね。

PetLog

Pet Register

Log out

← Back to list

Weight Logs

Image



Name

pochi

Birthday

2020年8月18日

Gender

男の子



ログイン画面のロゴは変わっていないと思います。
個別にスタイルを指定しているため、個別に変更が必要になります。
ここでは変更しませんが、気になる人は変更しておきましょう。

次は、新しいスタイルを追加してみましょう

ラベルが大きすぎて気になるので、ラベル用のスタイルを作っちゃおうと思います。

PetLog

Pet Register

Log out

← Back to list

Weight Logs

Image



Name

pochi

Birthday

2020年8月18日

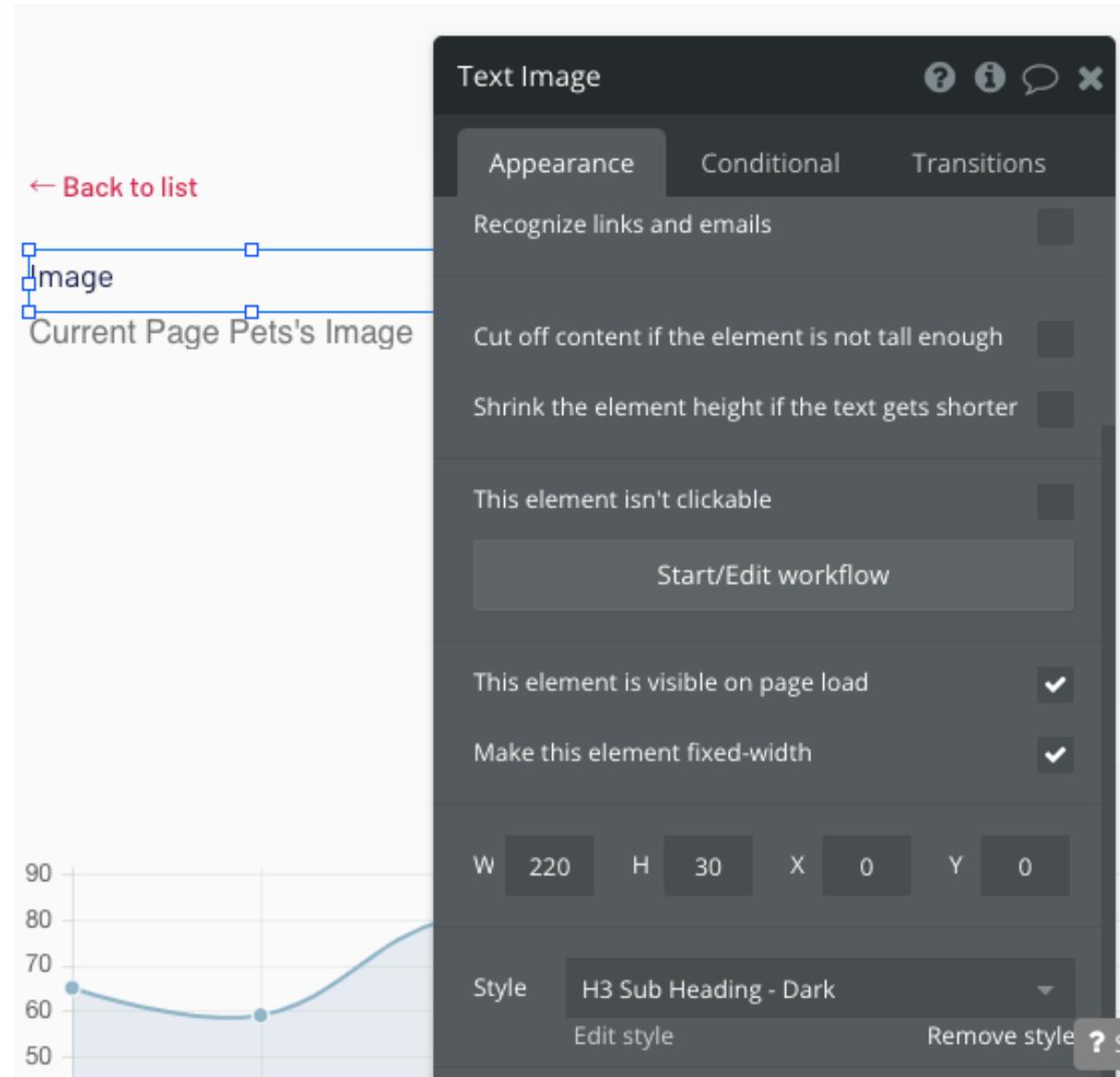
Gender

男の子



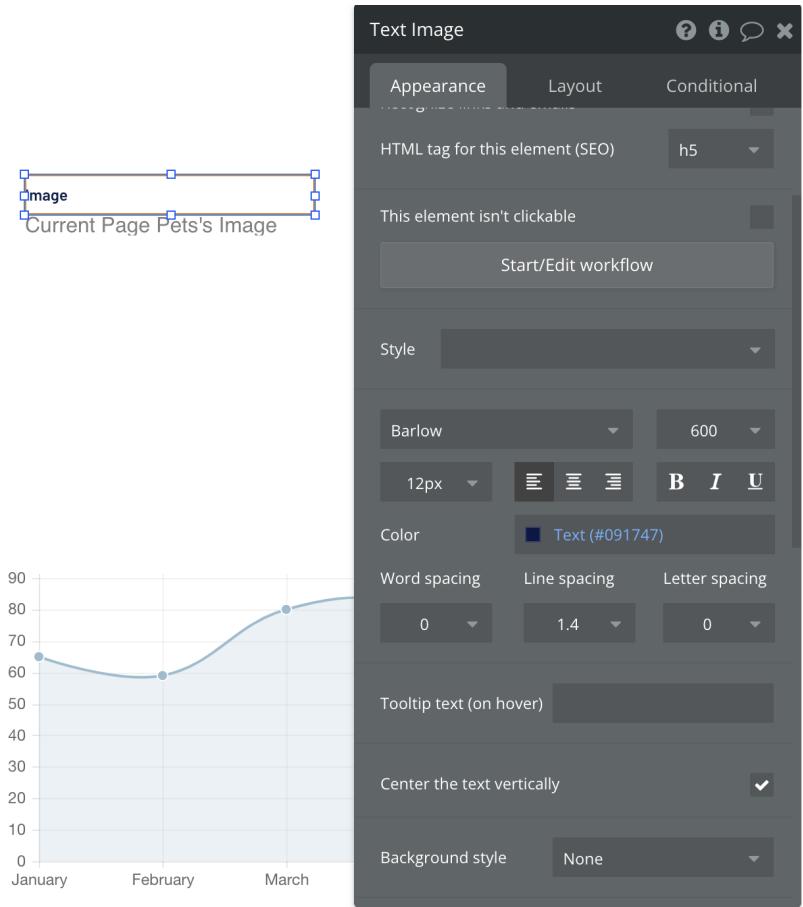
まずは個別にスタイルを指定します。

- pet_detailを開いて、Image テキストをダブルクリックして、設定を開く
- Style のプルダウン右下の Detach style をクリック



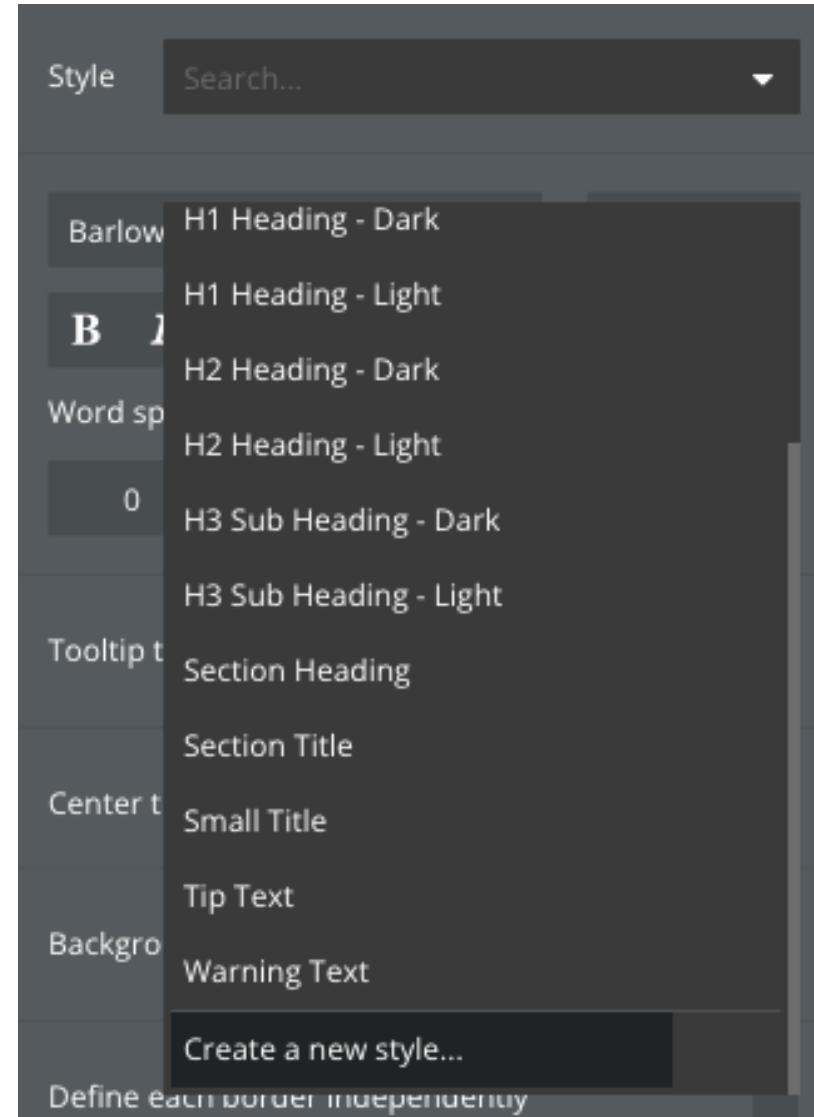
以下の設定にします

- フォントは Barlow
- フォントウェイトは 600
- フォントサイズは 14
- Center the text vertically
にチェック

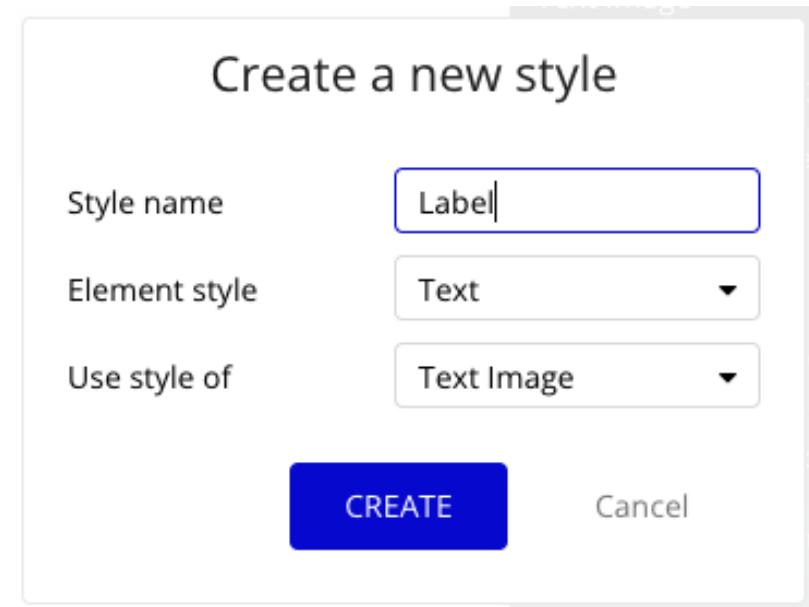


続いて、指定した個別のスタイルを共通のスタイルとして定義します

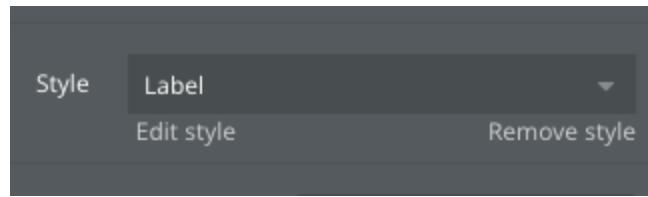
- そのまま **Image** の設定の中で、**Style** のプルダウンを開く
- 一番下の **Create a new style..** をクリック



- Style名に Label と入れる
- Element styleは Text のまま、テキストElementのスタイルであることを示しておく
- Use style ofは Text Image のまま、Text Image を元としてスタイルをつくります



スタイルにLabelが指定されるようになっているはずです。



個別に指定したスタイルを共通のスタイルとして定義するのではなく、
先にスタイルを定義する方法もありますが、個別でスタイルを
指定したものを共通のスタイルとして設定する方が
デザインビューでイメージを確認しながらつくれるのでやりやすいです。

では、定義したスタイルを他のラベルにも適用していきましょう。

- pet_detail: Name, Birthday, Gender
- pet_register: Image, Name, Birthday, Gender
- pet_weight_register: Weight

ちょっと多くて手間ですね。

最初からスタイルを分けていたら、一ヵ所のスタイルを変更するだけで済んだので、意味合いが異なる画面要素がでてきたら、スタイル定義しておくということを意識しておくとよいでしょう。

プレビューしましよう

PetLog

[Pet Register](#)[Log out](#)[← Back to list](#)[Weight Logs](#)**Image****Name**

pochi

Birthday

2020年8月18日

Gender

男の子



Styleについては以上です。

ロジックを作りこむ

ロジックを作りこむ

アプリケーションには様々なところにロジックが埋め込まれています。

- 画面操作に対するフィードバックを返す
- データを抽出、加工する
- 画面を権限によって切り替える など様々です

Bubbleでも様々なところにロジックを埋め込めるので一緒にやっていきましょう。

画面操作に対するフィードバックを返す

画面操作に対するフィードバックを返す

Bubbleでは画面要素に対してロジックを埋め込みます。

画面操作に対するフィードバックを作り混むのに使えます。

ペットリストにホバーしたら、赤枠が付くような動きをつけてみましょう

PetLog

Pet Register

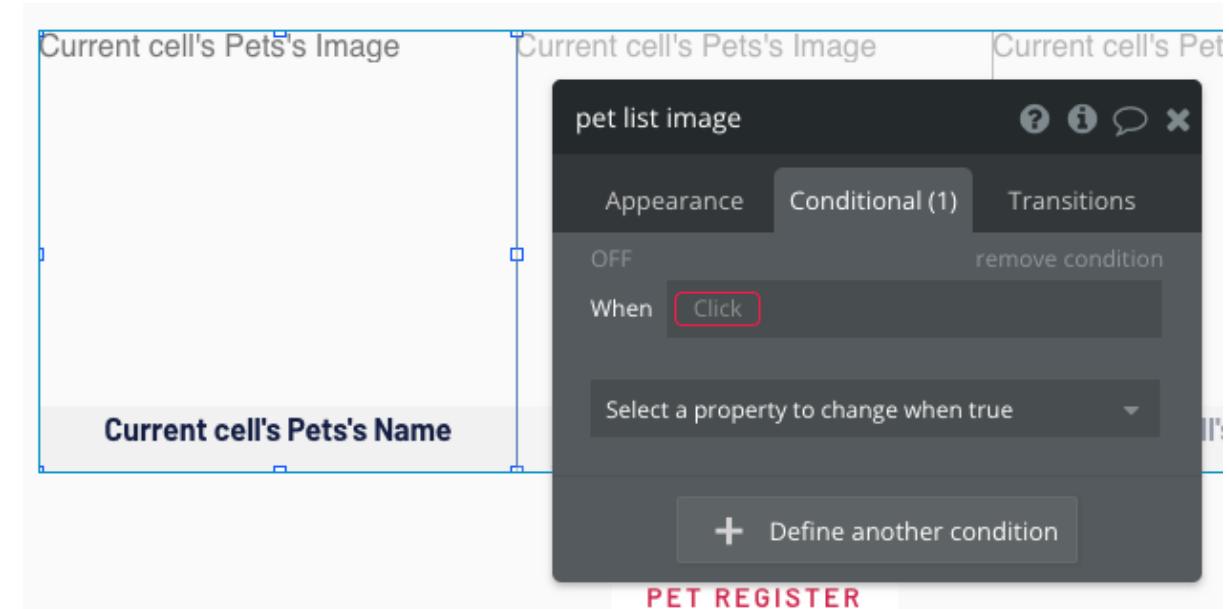
Log out



画像Elementにロジックを埋め込んでいきます

- pet_listを開く
- ペット画像の画像Elementをダブルクリックして、設定を開く
- タブから Conditional を指定する
- Define another condition ボタンをクリックする

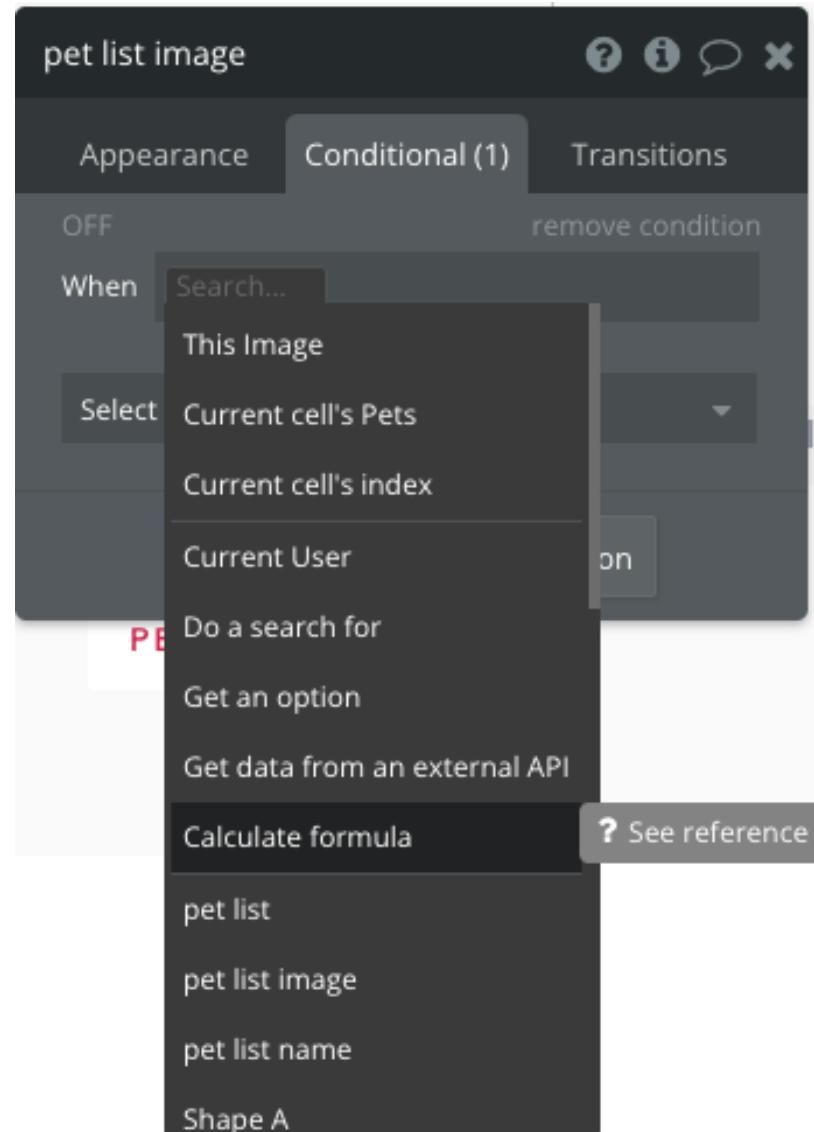
レスポンシブの表示有無のルール指定でもふれましたが、ここで条件と条件に合致した場合にプロパティをどう変更するかを定義できます。



まずは何に関する条件が指定できるか見てみましょう。

- 該当の画像Elementやその親要素や画面内の他の要素
- ログインユーザー
- 新規でのデータ検索
- 現在、日付や現在位置、ページ幅、スクロール位置など現在の状態

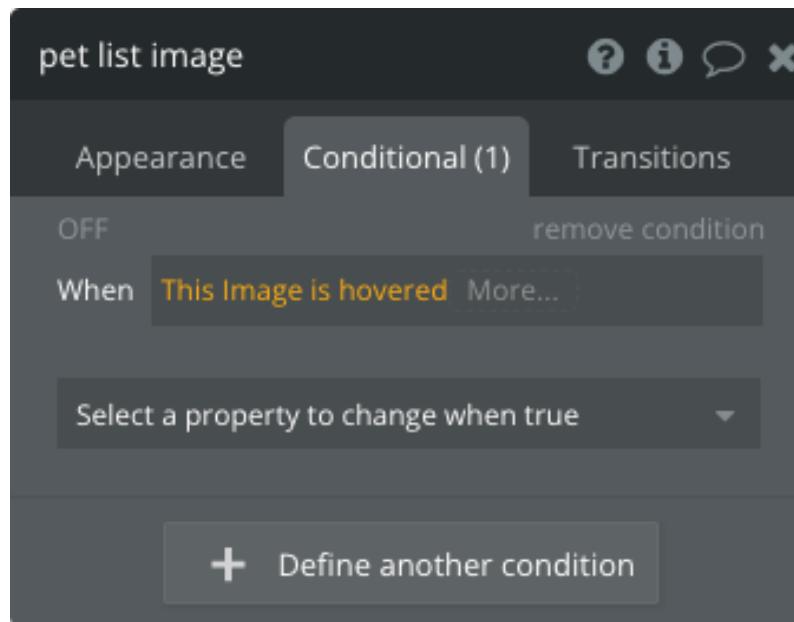
という具合に、なんかいろいろな条件が指定できそうなののがわかると思います。



今回は単純に該当の画像 `This Image` を選択しましょう。

すると、次は画像の状態がならびます。ここもいろいろありますが、今回は `is hovered` を選択してください。

これで、該当の画像がホバーされたらという条件になります。



次に条件に合致した場合に、どのプロパティをどう変更するかという指定をします。

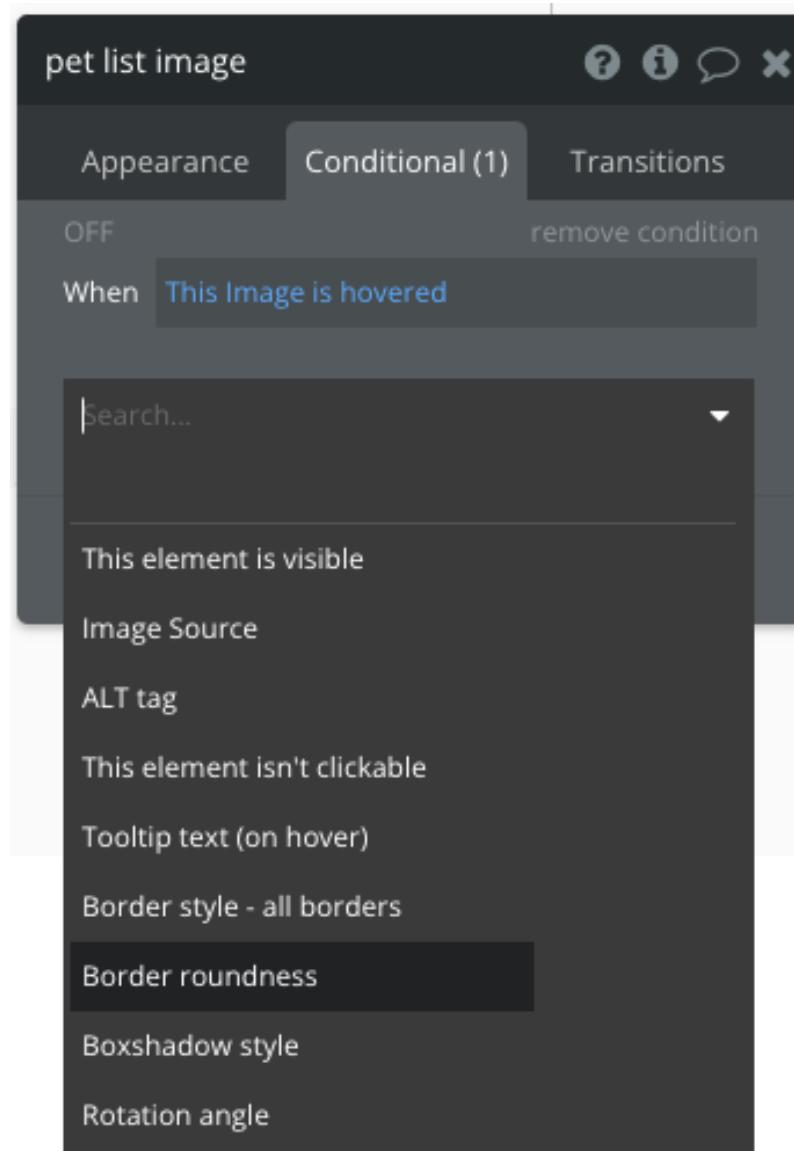
Select a property to change

when true をクリックして、中を眺めてみましょう。

- 画像のソース、alt属性
- クリックできるか、ボーダーなど

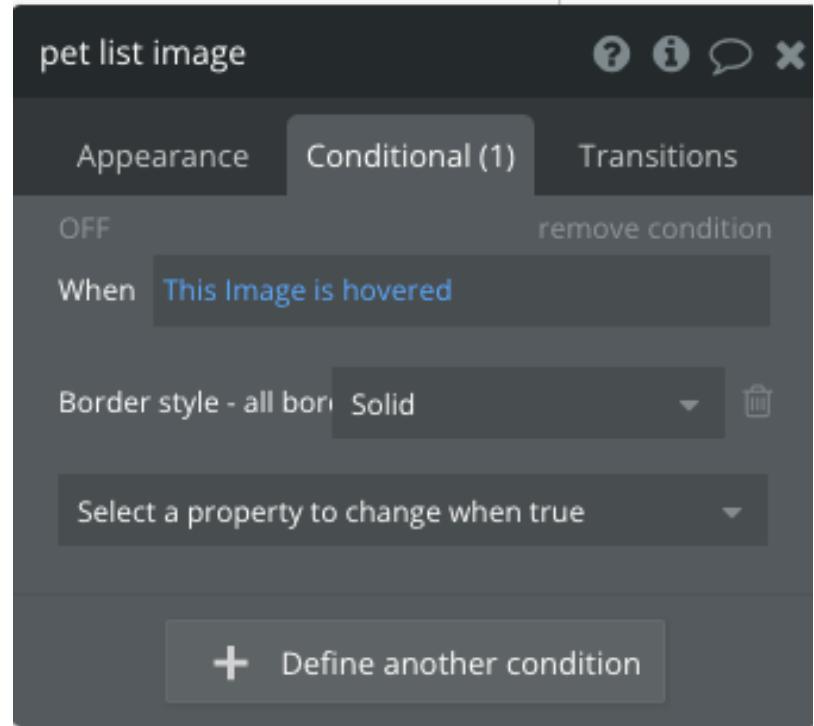
いろいろ変更出来そうだというのがわかります。

ここに並ぶものはElementの種類によって異なります。



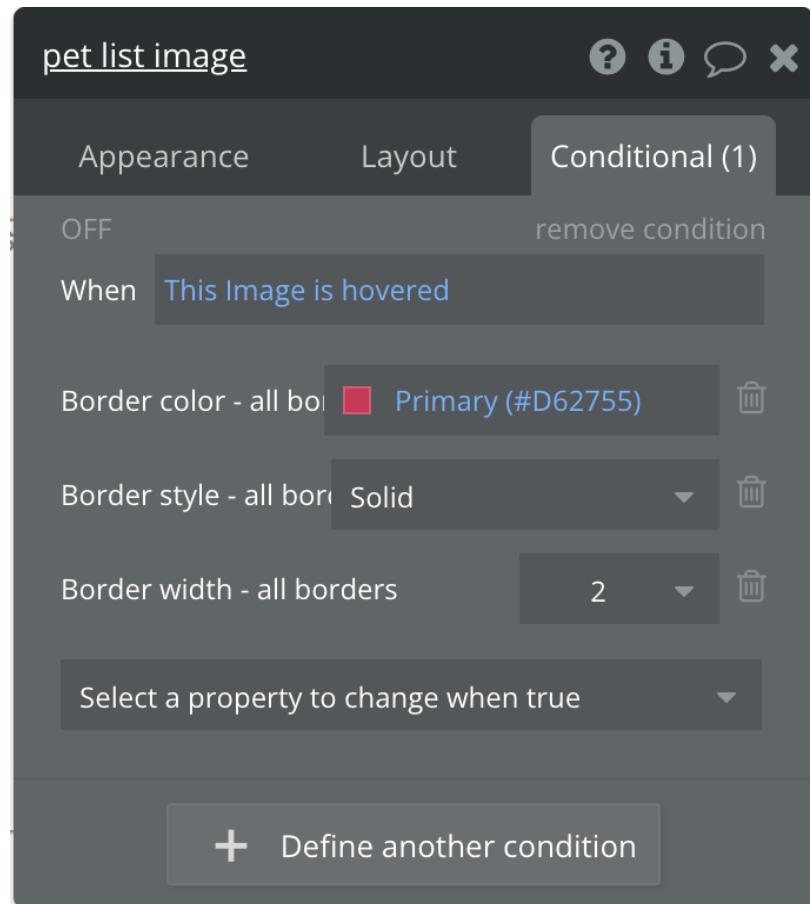
今回はホバーされたら、赤の枠線が付く
ようにします

- Border style - all borders をクリックする
- None となっている箇所を Solid に変更する
- 枠線が なし と指定されているのを 実線を表示 に変更しているという意味になります。



- Select a property to change when true を選ぶ
- Border color - all borders をクリックする
- 色を指定できるようになるので、定義してあるPrimaryを選ぶ
- 同様にして次は、 Border width - all borders を指定し、 2 を設定する

以上で設定完了です。



プレビューしましょう

ホバーしたら、赤枠がでるようになりました。

PetLog

Pet Register

Log out



こうやって、ユーザー操作に対してわかりやすいフィードバックを返したり、画面の装飾を状態によって切り替えたりといったロジックを埋め込んで、プロダクトを作りこんでいくことができます。

データを抽出、加工する

データを抽出、加工する

特定のデータだけを抽出したり、取得したデータを加工や計算して出したりすることができます。

ペットのイニシャル、年齢、最新の体重を表示するようにします。

Image



Name (Initial)

pochi(P)

Birthday

2020年8月18日

Age (as Dog/Cat)

2(24)

Gender

男の子

Latest Weight

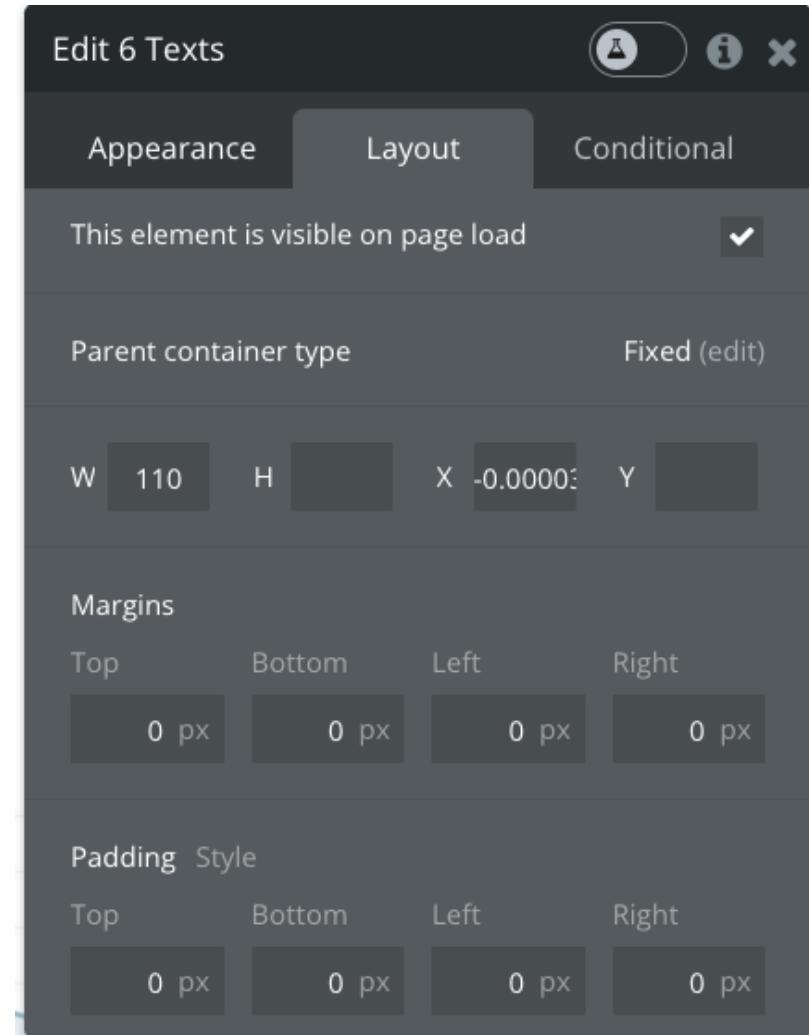
9kg



事前準備

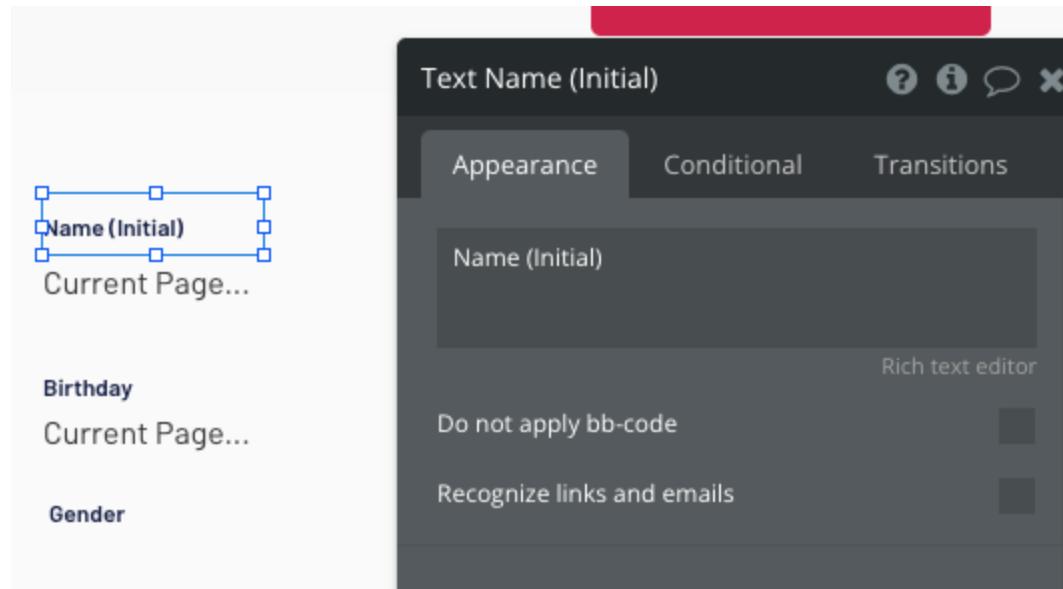
これから要素を追加していくので、
Name、Birthday、Genderの幅を縮め
ておきましょう。

- pet_detailで、Nameをダブルクリックして設定を開く
- Shiftを押しながら、NameのラベルからGenderのテキスト要素までを選ぶ
- Layout タブを洗濯し Width と表示されている部分をクリックする
- W(幅)に 110 を指定する



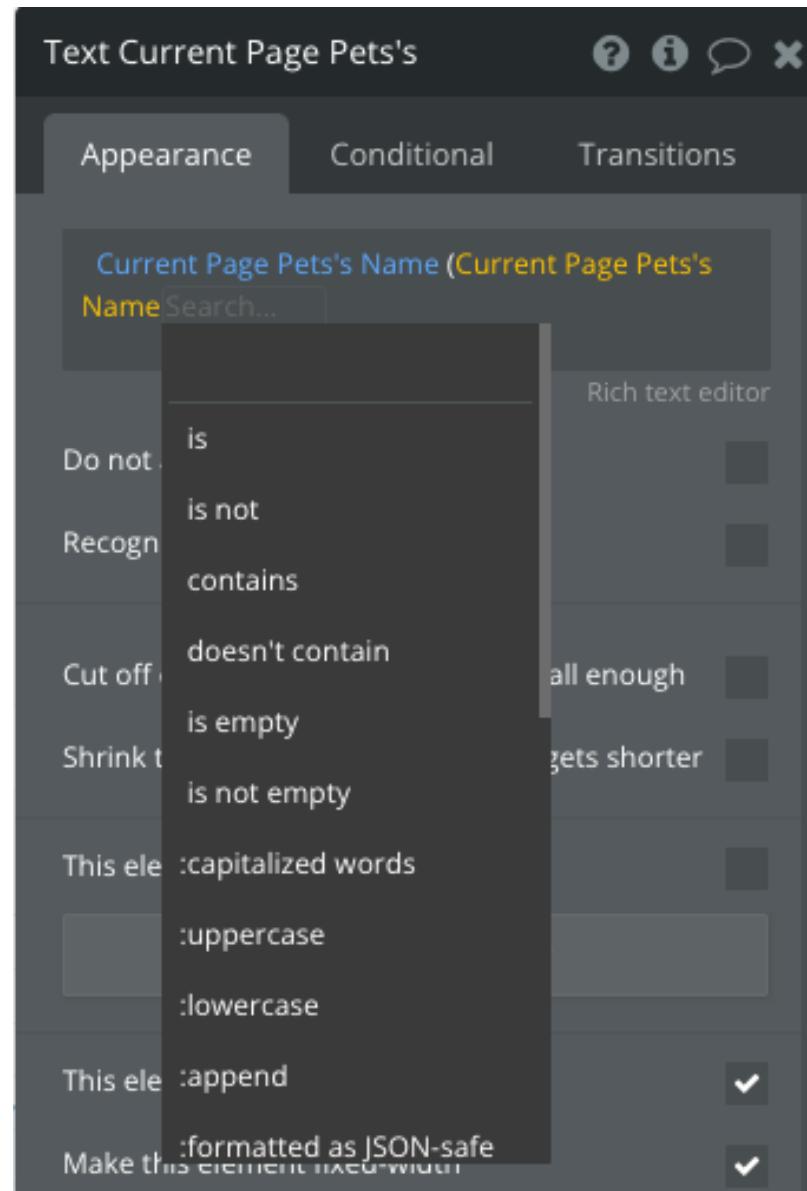
まずはイニシャルを表示します

- Nameのラベルの中身を、イニシャルを含むというのがわかるように **Name (Initial)** に変更する

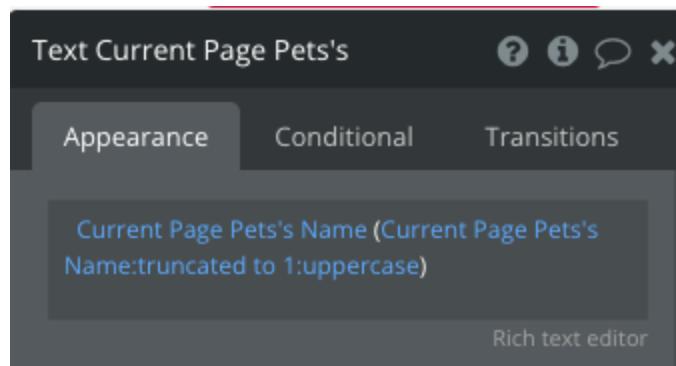


- Nameの内容を出しているテキストを選択する
- テキスト入力欄の文字がない部分をクリックして、フォーカスをあてる
- (と入力する
- Insert dynamic dataを選択する
- Current Page Pets > 's Nameを選択する
- More...と薄く出ているはずなので、クリックする

ここで様々な加工方法が選択できます。眺めて見ましょう。



- truncated to を選択する
 - これは指定された文字数までを切り取るという意味になります
- 1 と入力して、Enterキーで確定する
- また More と出るのでクリック
- :uppercase を選択する
 - これは大文字に変換するという意味になります
 - (日本語名をついている方にとっては意味がないですが)
- テキスト入力欄の文字がない部分をクリックして、) を入力する



プレビューしましょう

Image



Name (Initial)

pochi(P)

Birthday

2020年8月18日

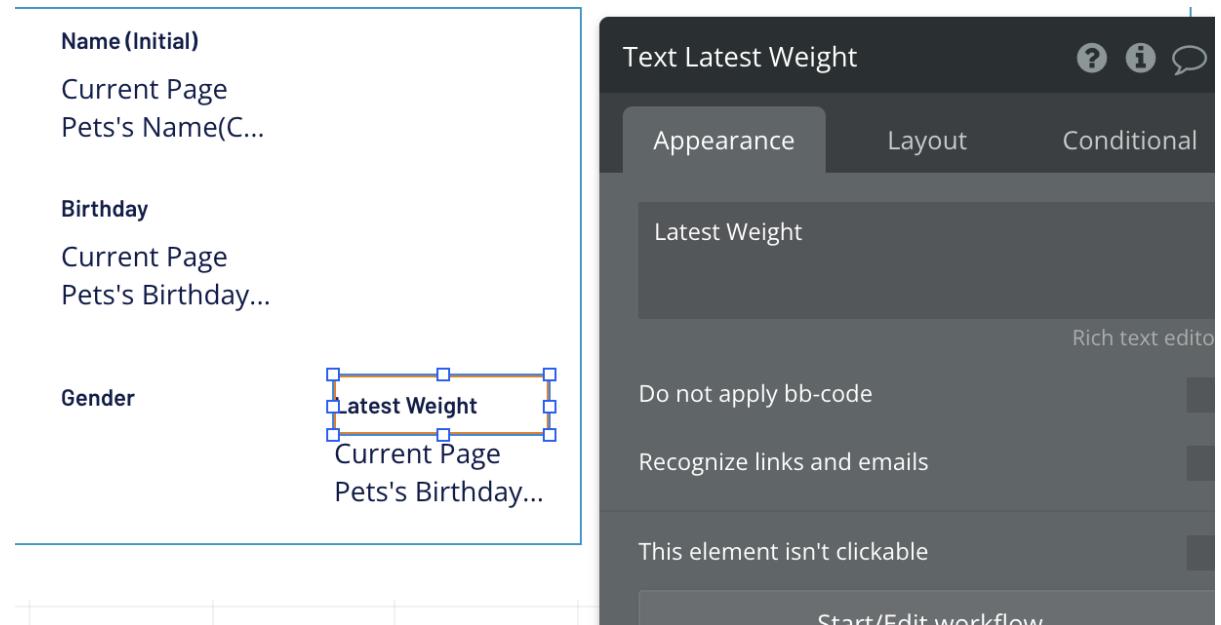
Gender

男の子

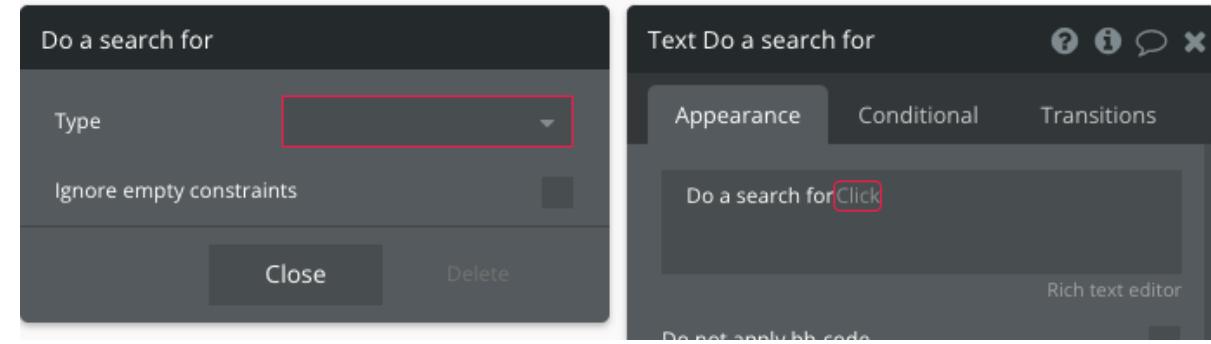


最新の体重を表示します

- Birthdayのラベルとテキストをコピー&ペーストして配置しましょう
- ラベルを **Latest Weight** に変更しましょう



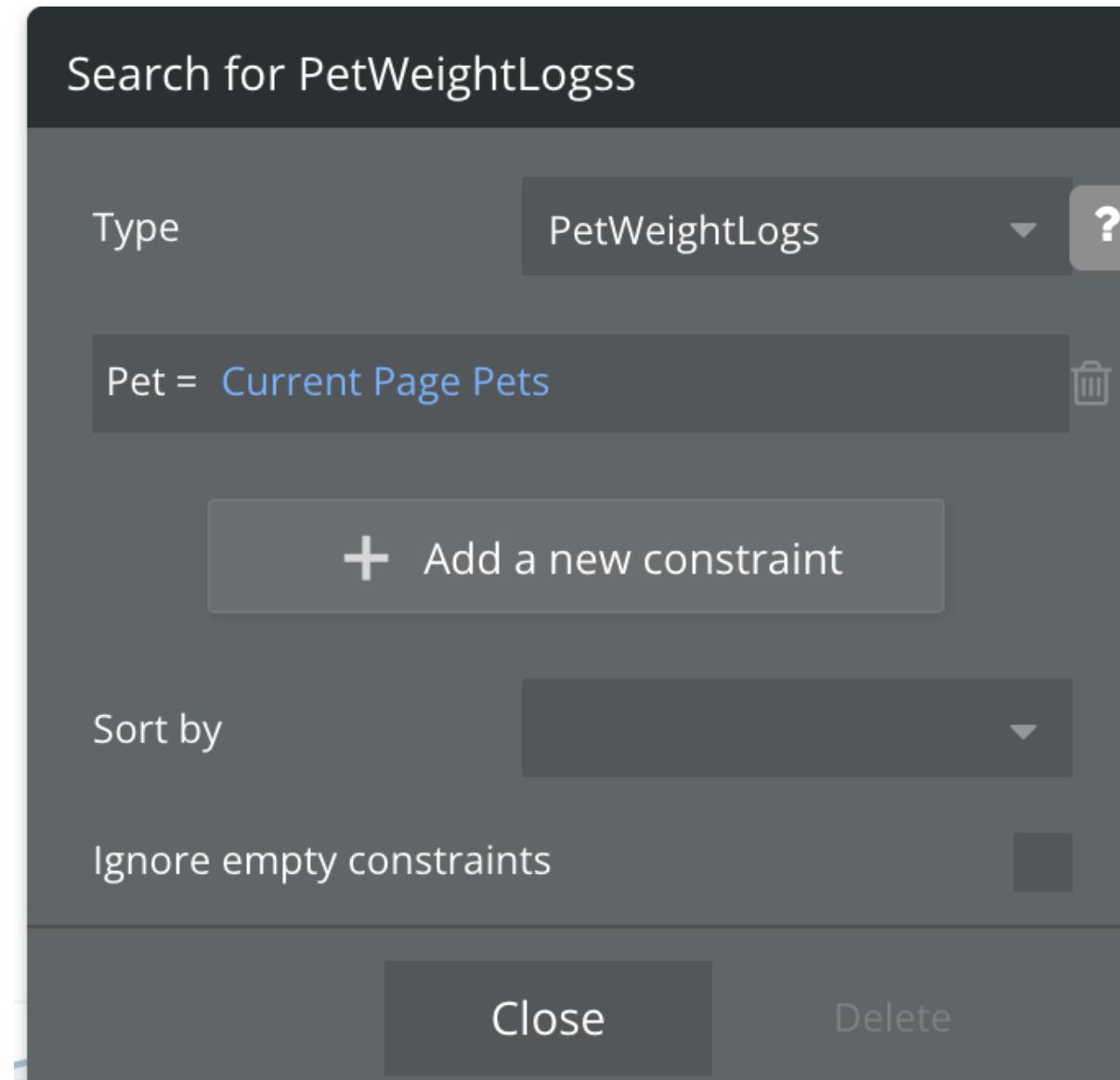
- Latest Wightの中身を出すテキストの設定を開いて、テキスト入力エリアを空にします
- フォーカスをあてて、 `insert dynamic data` をクリックします
- `Do a search for` をクリックします
 - データを検索するという意味ですね



現在ページで表示しているペットの体重を取得するという指定をします

- Type に PetWeightLogs を指定する
- Add a new constraint ボタンをクリックする
- 条件入力欄が現れるのでクリックして、Pet 、 = 、 Current Page Pets の順に指定する

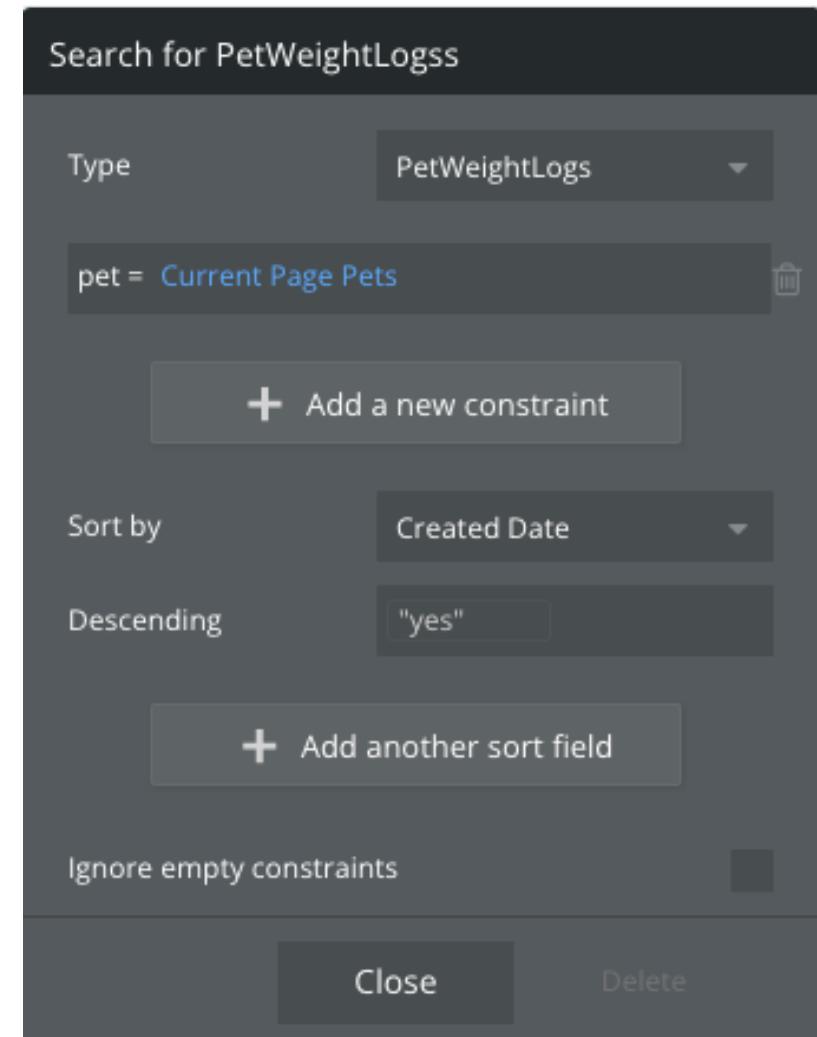
様々な条件で取得ができるので、どんな条件があるのか眺めておきましょう



作成日の降順、すなわち新しく作られた
もの順に並べるという指定をします

- Sort by に Created Date を指
定する
- Descending に yes を指定する
- Closeする

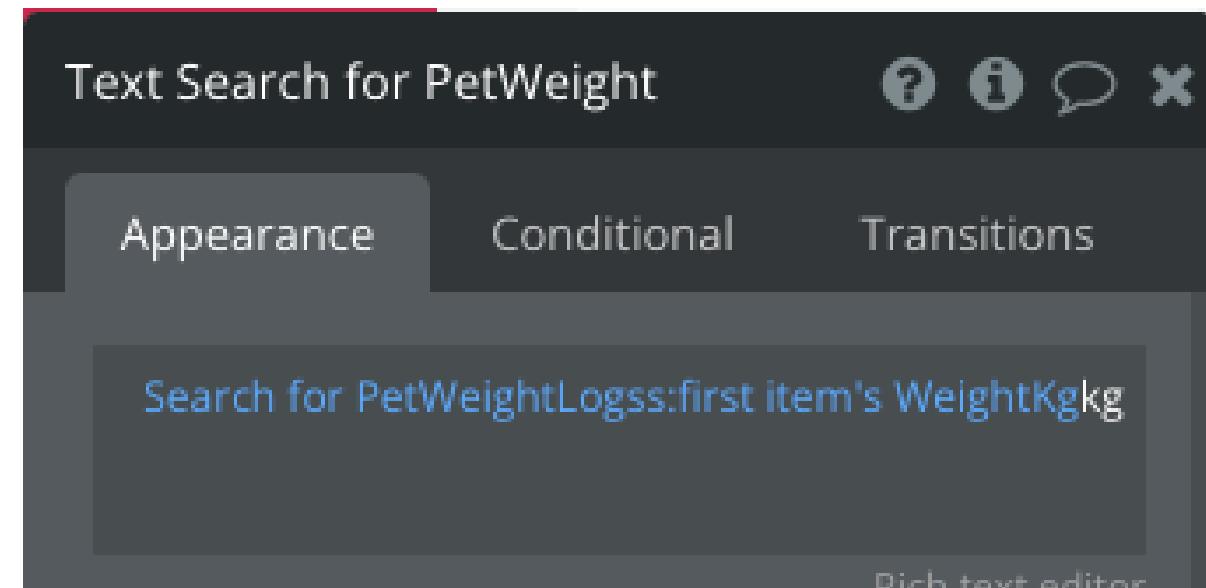
並び順の指定は忘れがちですが、重要な
ことがあります。



最新の1件の体重を表示します

- テキスト入力欄の `More` をクリックして、中身を眺めましょう
- 最初の1件目を取得したいので、`:first item` を指定する
- 続いて、`'s WeightKg` を指定する
- 空白部分をクリックして、`kg` と入力する

以上で、最新の体重の設定は完了です。
データ抽出、リスト加工の方法として覚えて起きましょう。



プレビューしましょう。

Image



Name (Initial)

pochi(P)

Birthday

2020年8月18日

Gender

男の子

Latest Weight

9kg



<Advanced>

ちょっと横道にそれで、

数値のMore、日付のMoreを眺めてみましょう。

Bubbleでは数値や日付についても、様々な加工・計算方法が提供されています。

<Advanced>

年齢を計算する

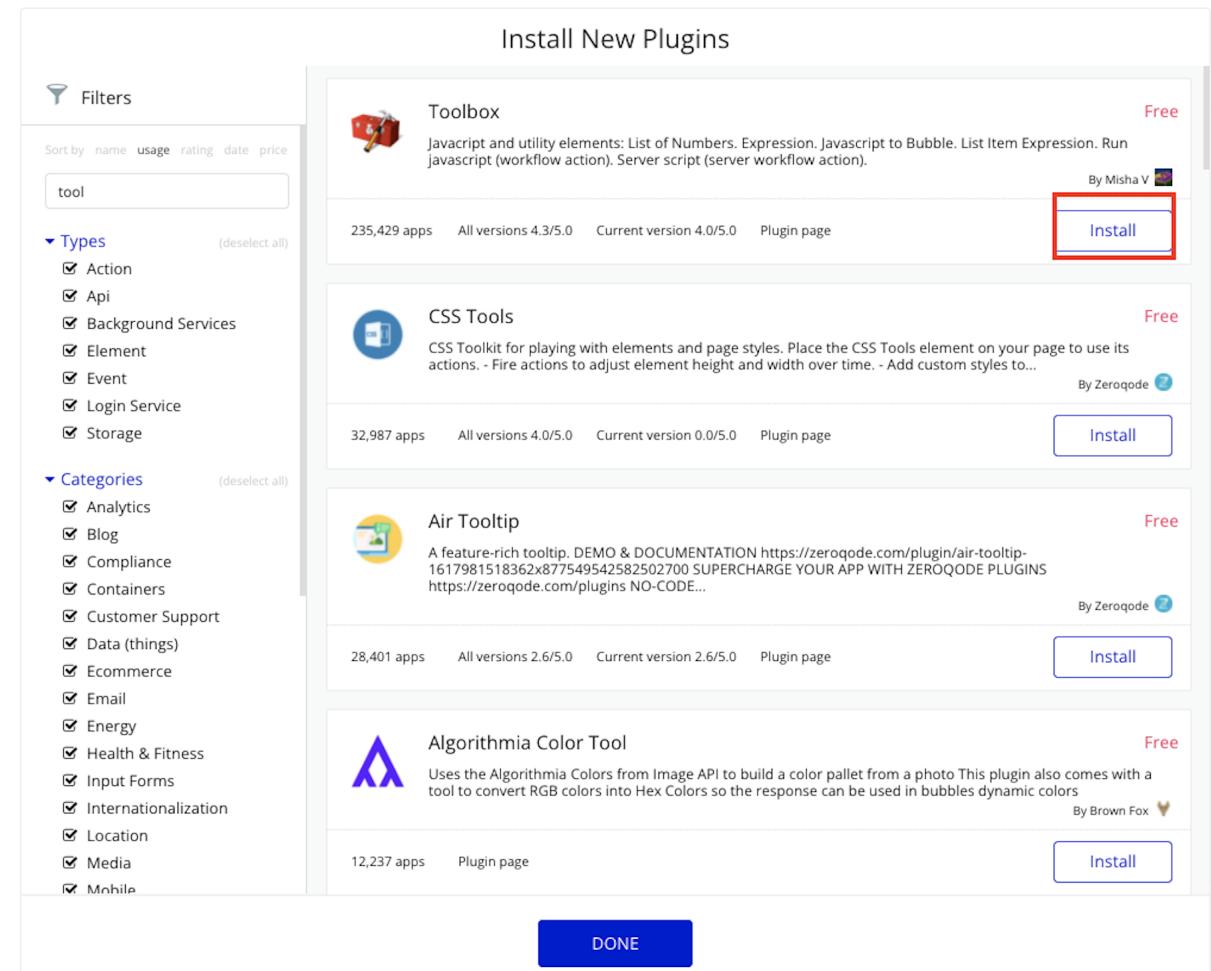
次は年齢を出します。先ほど見たとおり、数値や日付の加工・計算はできるのですが、ちょっと年齢計算は難しそうなので、直接コードを埋め込んで実現しようと思います。

Bubbleではpluginを導入することで、javascriptというプログラミング言語を使った簡易的な処理を動作させることができます。

<Advanced>

javascriptのコードを埋め込むために
は、 **Toolbox** というプラグインを利用
します。
インストールしましょう。

- 左メニューの中の **Plugins** を指定
- 検索用のテキストボックスに
tool と入力（検索には少し時間
かかる）
- 検索結果の一番上にでてくる
Toolbox の **Install** ボタンを押
す



<Advanced>

Toolboxでのコードの埋め込み方は大きく2つあります、今回は以下の2通りを紹介します

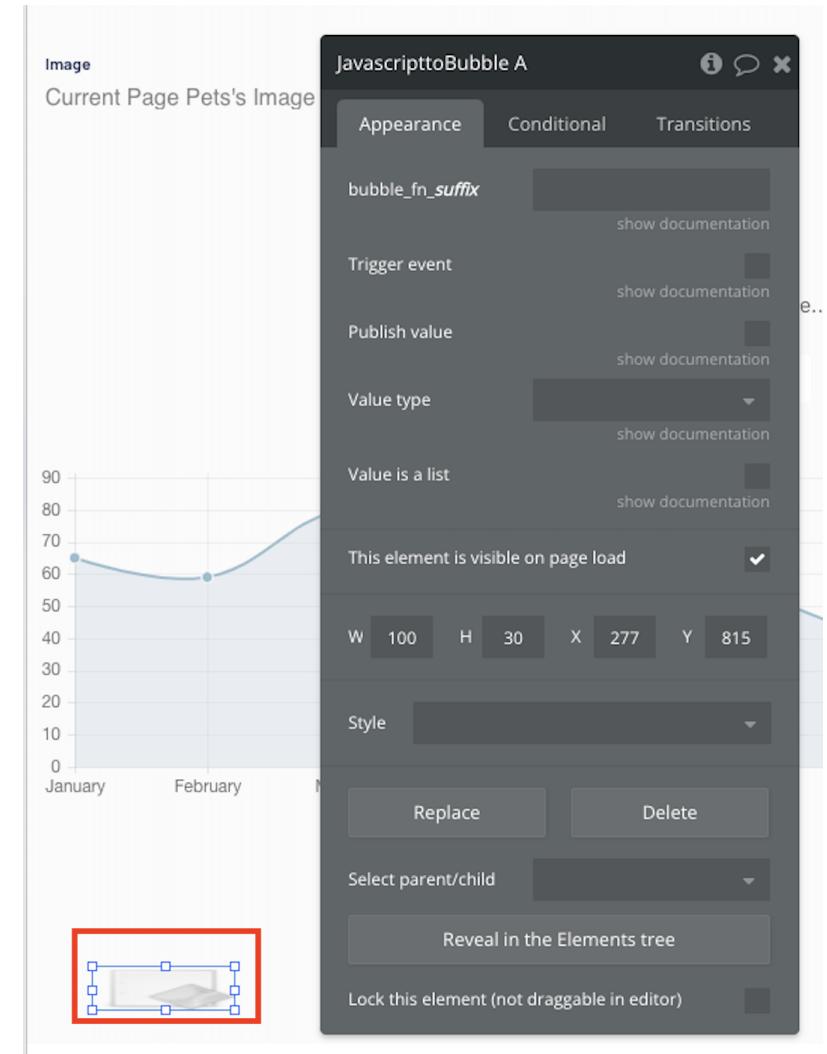
- Workflow上の `Run javascript` で実行／Design上の `Javascript to Bubble` で受取
 - 複数行にわたるような複雑な処理を使う
- Design上の `Expression` で実行兼受け取り
 - 単発で終わるような処理を使う

<Advanced>

では、年齢を計算します。

Run javascript / Javascript to Bubbleで行います。まずは、pet_detail画面に、Javascript to Bubbleを仕込んでおきます。

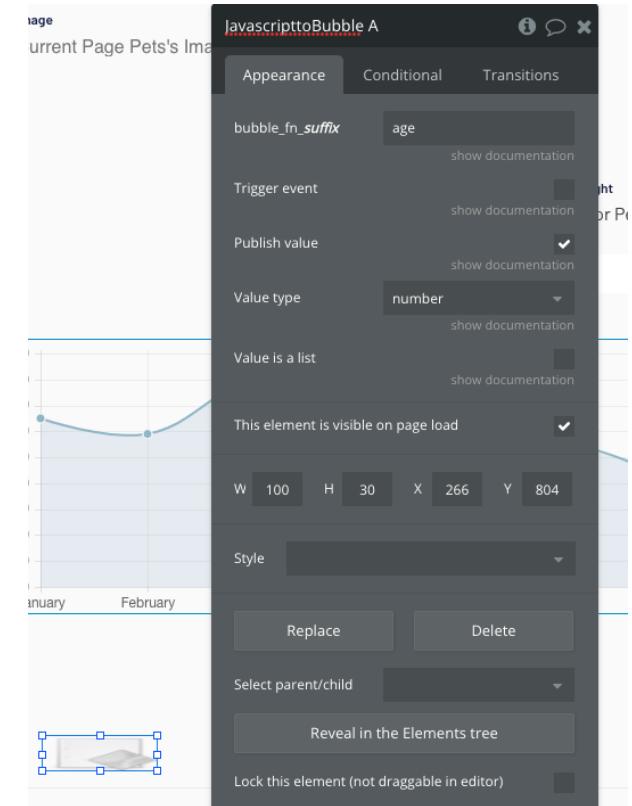
- 左メニューの Visual elements から javascript to Bubble を選択する
- 画面の最下部など邪魔にならない場所におく
- javascriptの結果を受け取るためのものなので、プレビューなど実行時には表示されません



<Advanced>

- `bubble_fn_suffix` に `age` と指定する
- `Publish value` にチェックする
- `Value type` に `number` と指定する

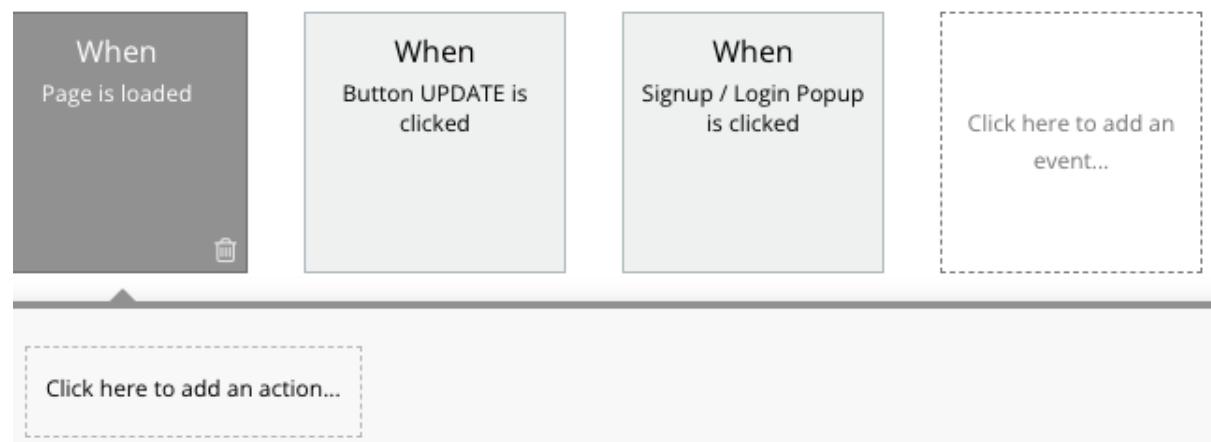
以上で、`bubble_fn_age` という関数（処理のかたまり）にjavascriptから値を渡すと、この画面要素で受け取れるようになります。



<Advanced>

次にjavascriptを実行する箇所を定義します。

- 左メニューからWorkflowを選択する
- 正方形がならんでいるが、その一番右の **Click here to add an event...** を選択する
- General > Page is loaded** を選択する



<Advanced>

- Click here to add an action... をクリックする
- Plugins > Run javascript をクリックする
- 設定が開くので、 Script の欄に次のページのコードを貼り付ける



The screenshot shows a 'Run javascript' dialog box. The 'Script' field contains the following code:

```
//生年月日
const birthday = {
  year: ,
  month: ,
  date: 
};

function getAge(birthday){

  //今日
  let today = new Date();

  //今年の誕生日
  let thisYearsBirthday = new
Date(today.getFullYear(), birthday.month-1,
birthday.date);

  //年齢
  let age = today.getFullYear() - birthday.year;

  if(today < thisYearsBirthday){
    //今年まだ誕生日が来ていない
    age--;
  }

  return age;
}

bubble_fn_age(getAge(birthday));
```

Below the script, there are buttons for 'Rich text editor' and 'show documentation'. At the bottom, there is a dropdown menu set to 'Asynchronous' with a checked checkbox, and another 'show documentation' button.

<Advanced>

```
//生年月日
const birthday = {
  year: ,
  month: ,
  date: 
};

function getAge(birthday){

  //今日
  let today = new Date();

  //今年の誕生日
  let thisYearsBirthday = new Date(today.getFullYear(), birthday.month-1, birthday.date);

  //年齢
  let age = today.getFullYear() - birthday.year;

  if(today < thisYearsBirthday){
    //今年まだ誕生日が来ていない
    age--;
  }

  return age;
}

bubble_fn_age(getAge(birthday));
```

<Advanced>

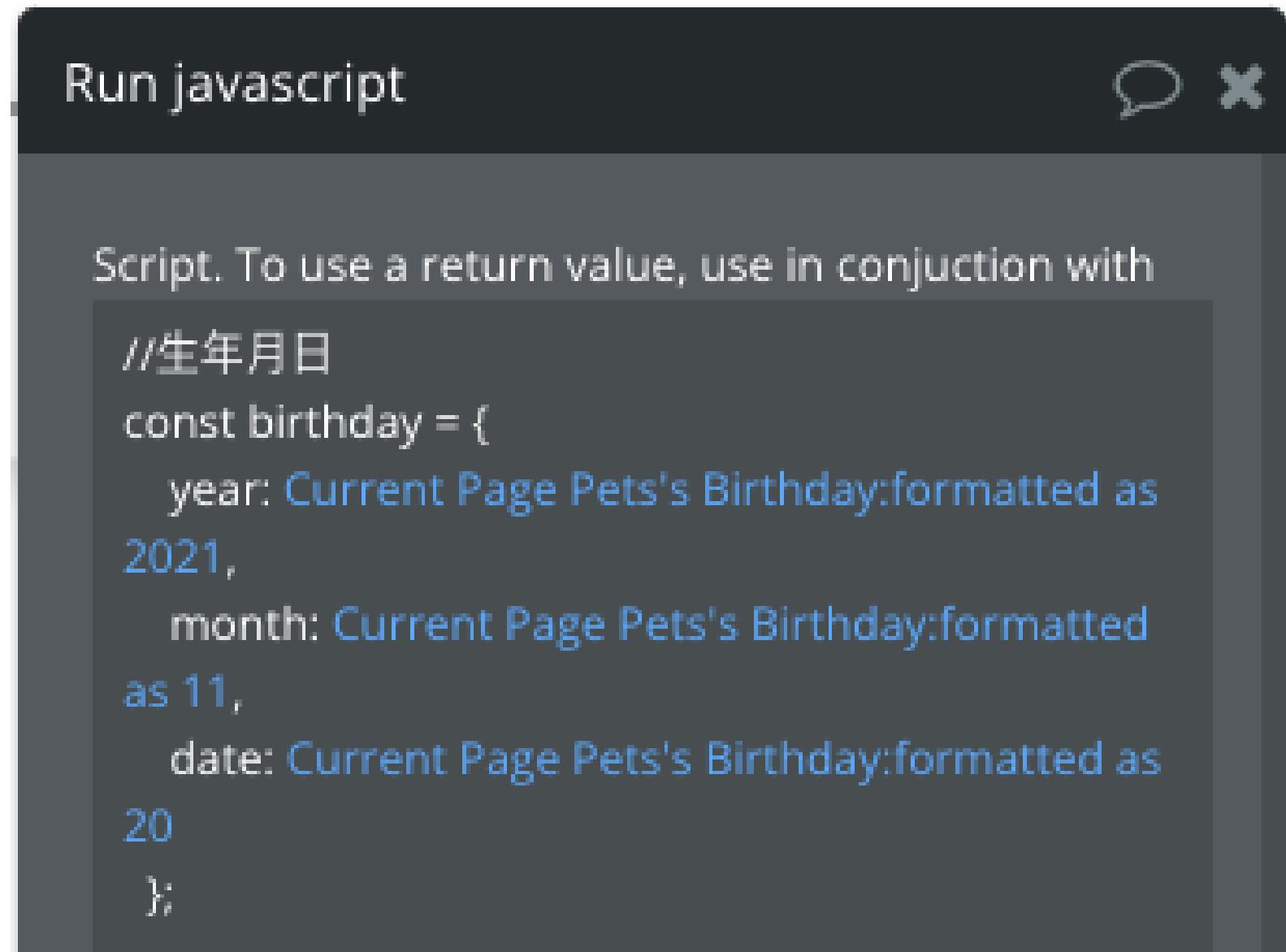
3行目から5行目の、`year:`、`month:`、`date:` の後ろに `insert dynamic data` で年月日を差し込む

- `year:` の後ろ (, の前) にカーソルを置く
 - `insert dynamic data` > `Current Page Pets` > 's Birthday
 - `More` > `formatted as 11/20/21`
 - `Format type` に `Custom` を指定する
 - `Custom format` に `yyyy` を指定する
- 同様に `month:` の後ろにも、`Custom format` を `m` にして差し込む
- 同様に `date:` の後ろにも、`Custom format` を `d` にして差し込む

※入力後イメージは次のページ

<Advanced>

入力後のイメージ



<Advanced>

表示するための画面要素を配置しましょう

- Birthdayのラベルとテキストをコピー&ペースト
- ラベルをAgeに変更
- テキストの中身を `JavascripttoBubble A > 's value` と指定する

<Advanced>

プレビューしてみましょう

Image



Name (Initial)

pochi(P)

Birthday

2020年8月18日

Age

2

Gender

男の子

Latest Weight

9kg

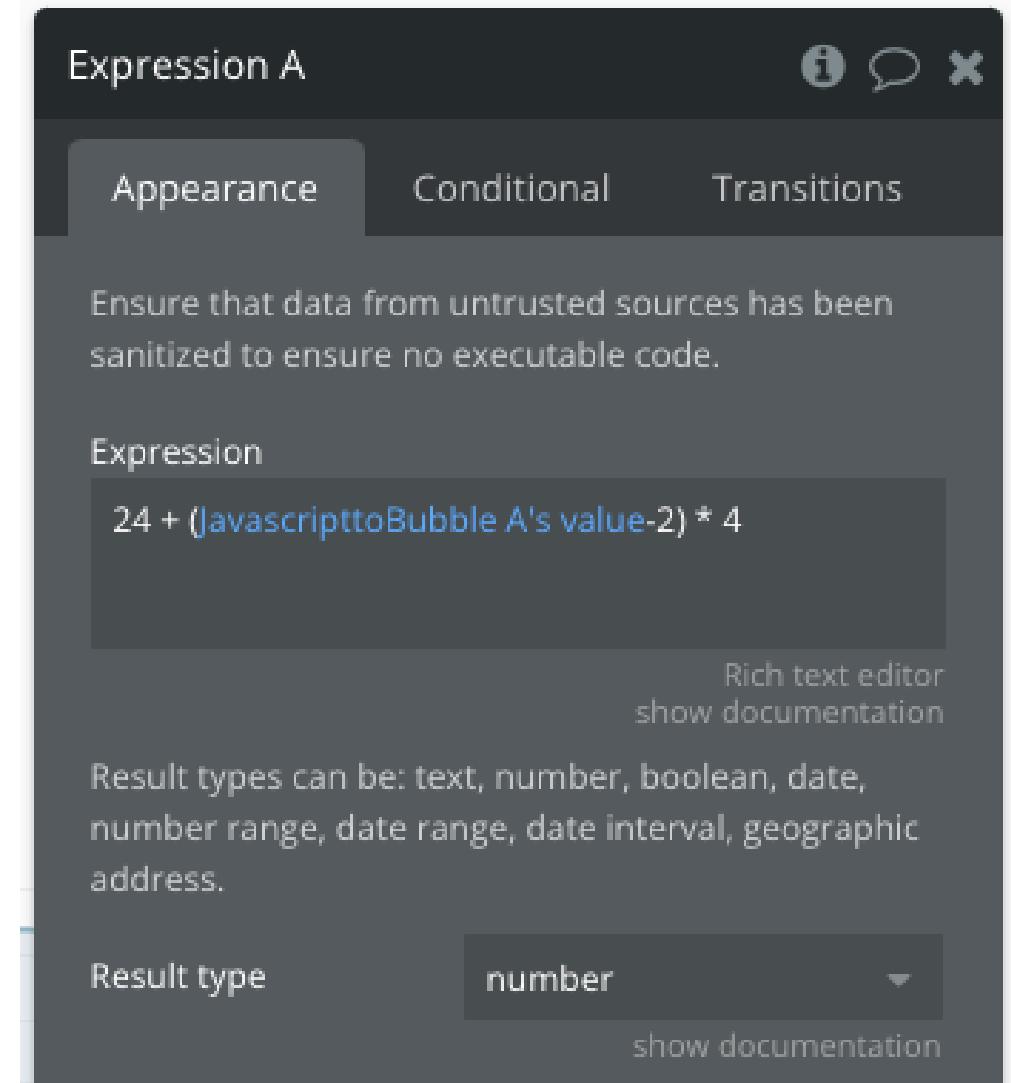


<Advanced>

次に犬猫年齢に換算したら何歳かも表示しようと思います。

Expression を用います。

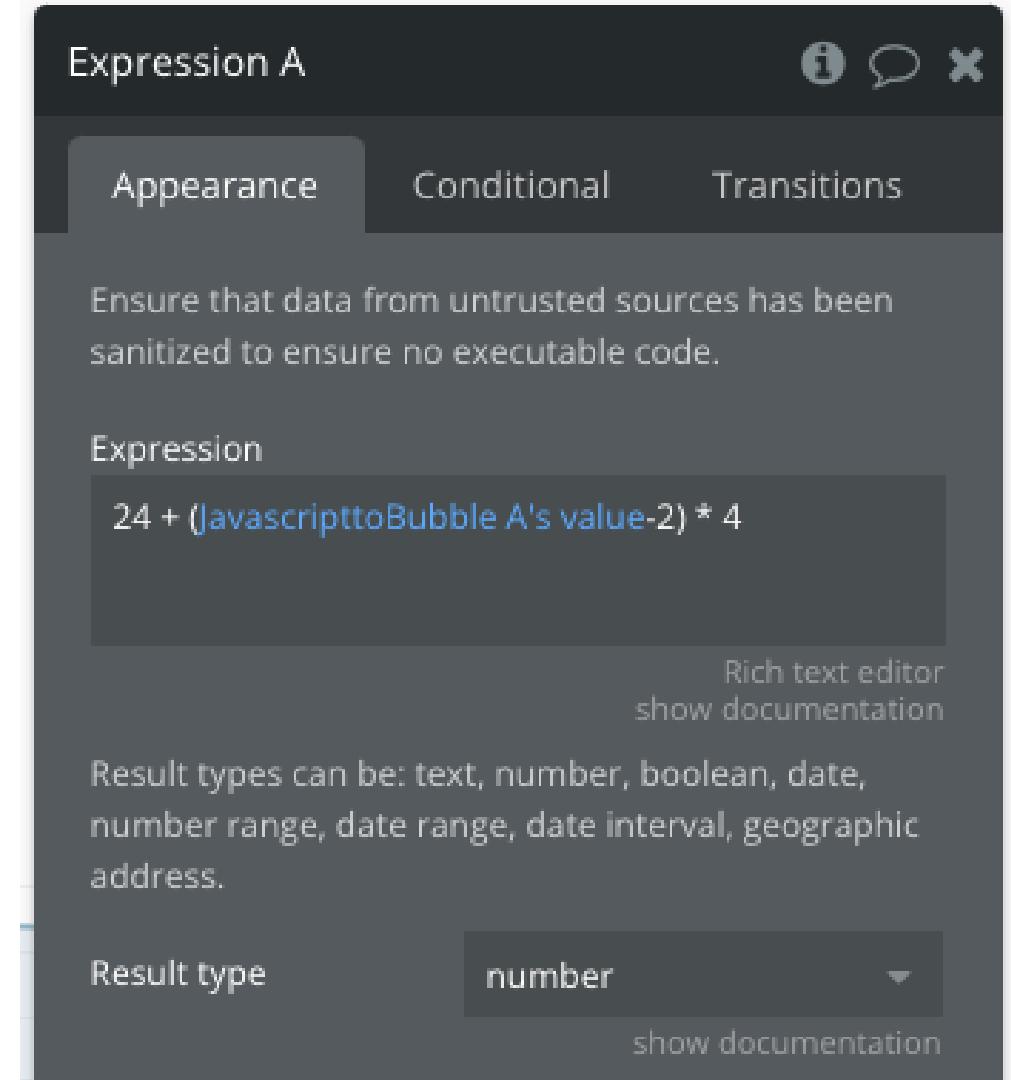
- Visual elementsから Expression を選択して、先ほどの Javascript to Bubble の横辺りに配置する
- Expressionに 24 + (と入力する
- insert dynamic data で JavascripttoBubble A > 's value を差し込む
- 続気に -2) * 4 と入力する
- Result typeに number と指定する



<Advanced>

表示の設定をします。

- Ageのラベルを犬猫年齢も含むことがわかりやすいように、Age(as Dog/Cat)と変更する
- Ageのテキストの中身に元々入力されているものの後に(と入力する
- insert dynamic dataで Expression A > 's valueを差し込む
-)と入力する



<Advanced>

プレビューしてみましょう



Name (Initial)

pochi(P)

Birthday

2020年8月18日

Age (as Dog/Cat)

2(24)

Gender

男の子

Latest Weight

9kg



画面を権限によって切り替える

画面を権限によって切り替える

ここまで、画面操作へのフィードバックやデータの抽出・加工など部分部分でのロジックの組み込みを説明してきました。

次は、複数機能にまたがるようなロジックをいれていくうと思います。

以下のことを行います。

- ユーザーをペットの飼い主（Pet Owner）とペットの飼育アドバイザー（Pet Advisor）に分ける
- 飼い主は今まで作ってきた画面・機能を利用できる
- アドバイザーはアドバイザー専用の画面・機能を利用できる

開発の流れとしては以下の順番に作り混んでいきます。

- ユーザー情報に飼い主かアドバイザーかを判別できるフィールドを追加する
- ユーザー登録時に、飼い主かアドバイザーかを選択して登録できるようにする
- アドバイザーの一覧画面、詳細画面を作成する
- 飼い主かアドバイザーかによって、ログイン後・サインアップ後の画面遷移先を切り替える

手順は多くかかりますが、複数のユーザー種別を扱うプロダクトはよくありますので、ぜひ身につけていきましょう

ユーザーを判別できるフィールドを追加する

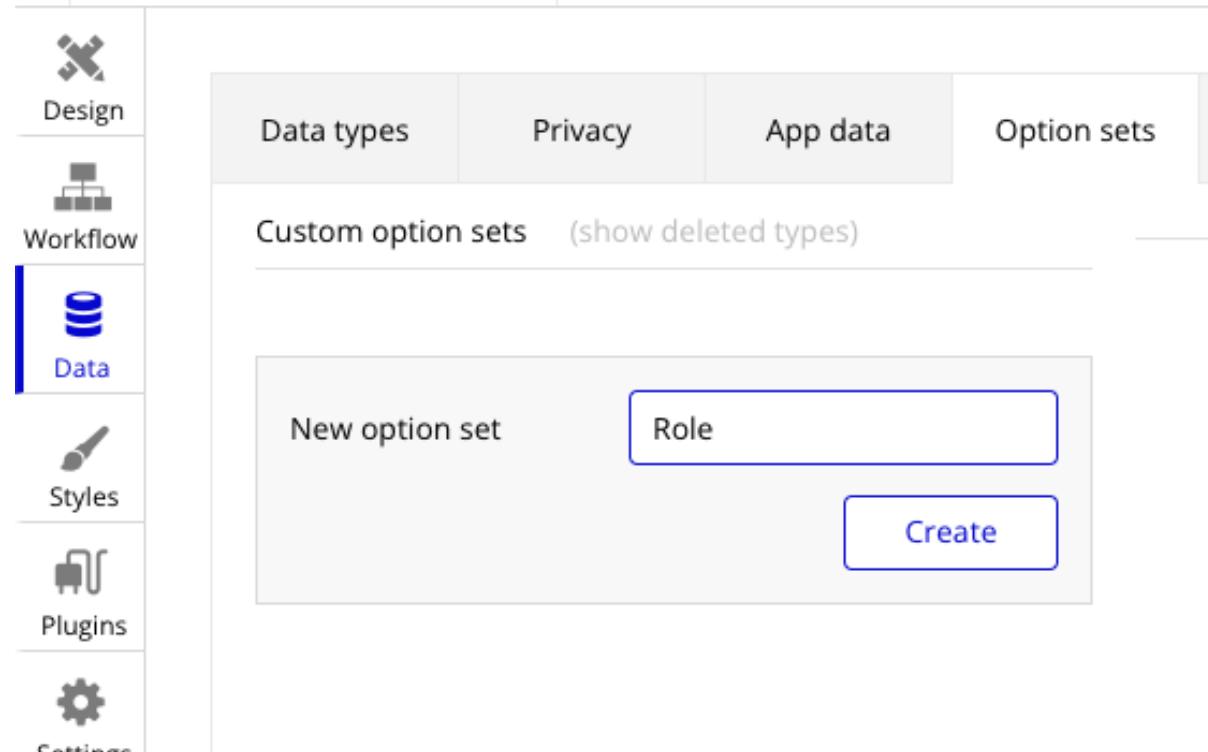
まずは飼い主かアドバイザーかという役割の違いをデータ上保持できるようにします。

ペットのオススメの時のようにテキストで保持してもよいのですが、決まった選択肢の中から指定するような値は、選択肢をあらかじめ定義して利用することで扱いやすくなります。

BubbleからOption setという仕組みが提供されているので、それを使ってみましょう。

Optionsを設定してみましょう

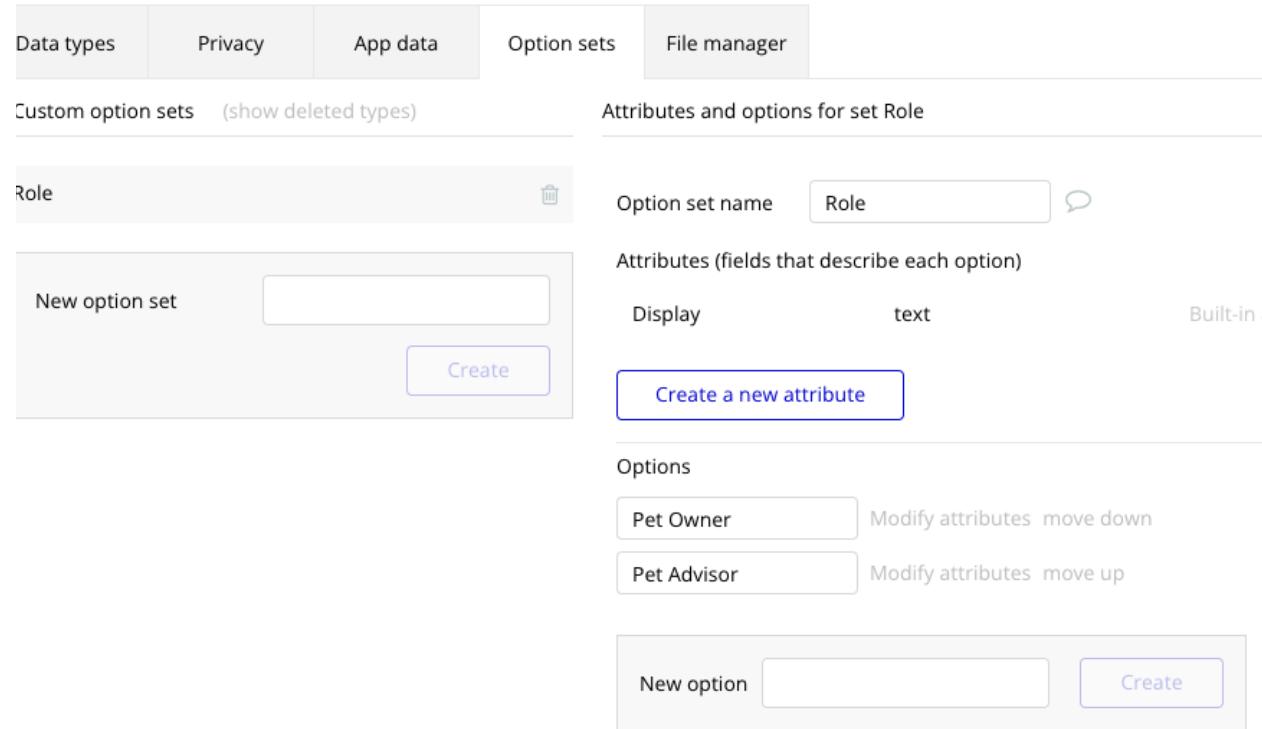
- 左メニューの Data > タブの Option sets と移動する
- New Option set に Role と入力して、Createボタンを押す
- 新たなOption setとしてRoleが作成されます



RoleというOption set（選択肢のセット）に、具体的なOption（選択肢）を追加していきます。今回は、Pet OwnerとPet Advisorを作成します。

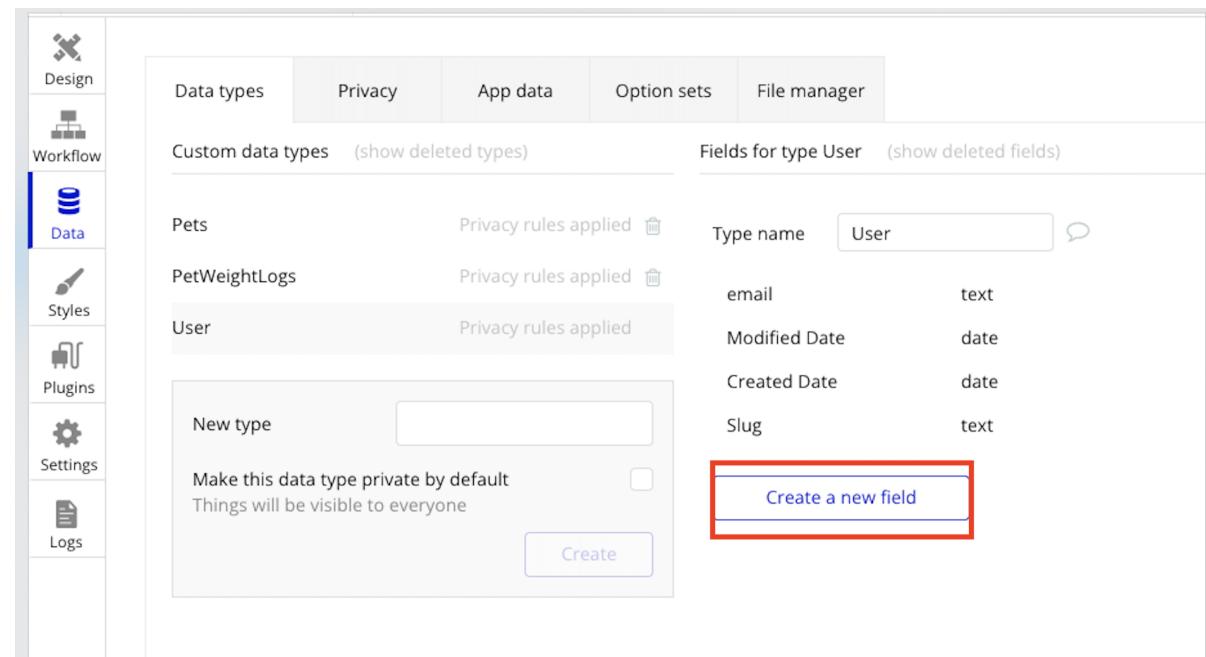
- 画面右下のNew OptionにPet Ownerと入力して、Createボタンを押してください
- 同様にNew OptionにPet Advisorと入力して、Createボタンを押してください

以上で設定完了です



次はユーザーの属性として、ロールを追加しましょう

- 左メニューの **Data** > タブの **Data types** と移動する
- User** を選択する
- 画面右下の **Create a new field** ボタンをクリックする



- Field name に Role と入力する
 - これはわかりやすいものであれば、どういう名前でも構いません
- Field type に Role を選択する
 - ここで指定しているのは先ほど作成した Option set としての Role になります。
- Createボタンを押す

Create a new field

Field name

Field type ▾

This field is a list (multiple entries)

CREATE Cancel

新しくフィールドを追加したので、既に作成してしまっているユーザーについてはRoleが空になってしまいます。後々、不整合を招きますので、既存データにパッチ（データ補正）をあてておきましょう。

- App Data タブ荷移動して、All Users を選択する
- 表が表示されるので、表の左端のペンアイコンをクリックして、1件ずつ編集する
 - 今作成されているユーザーはすべて飼い主のはずなので、Role に Pet Owner を指定する

Modify an existing database entry

Type of thing	User	Email	Created Date	Modified Date
Role	Pet Owner	kim+2@guildworks.jp	Nov 19, 2021 7:25 am	Nov 19, 2021 7:25 am
Slug		Unique id	1637274311987x540151212983663400	

SAVE **Cancel**

All PetWeightLogs Run as → kim@guildworks.jp Nov 16, 2021 7:19 pm Nov 19, 2021 6:25 am

Usersの中のすべての行がの Role が Pet Owner になっていればOK

The screenshot shows the Bubble app's interface with the 'Data' tab selected from the left sidebar. The main area displays a table titled 'Application data - All Users - Development version'. The table has columns for Email, Role, Created Date, and Modified Date. All seven entries in the 'Role' column are highlighted with a red box, indicating they are all set to 'Pet Owner'. The table also includes a search bar, a delete button (Delete (0)), and various export options.

		Email	Role	Created Date	Modified Date
<input type="checkbox"/>	Run as →	kim+2@guildworks.jp	Pet Owner	Nov 19, 2021 7:25 am	Nov 20, 2021 6:28 am
<input type="checkbox"/>	Run as →	kim+advisor2@guildworks.jp	Pet Owner	Nov 19, 2021 6:32 am	Nov 20, 2021 6:28 am
<input type="checkbox"/>	Run as →	kim+advisor@guildworks.jp	Pet Owner	Nov 19, 2021 5:35 am	Nov 20, 2021 6:28 am
<input type="checkbox"/>	Run as →	kim@guildworks.jp	Pet Owner	Nov 16, 2021 7:19 pm	Nov 20, 2021 6:28 am
<input type="checkbox"/>	Run as →	kyogoku+bubble_test2@guildw	Pet Owner	Nov 12, 2021 10:10 am	Nov 20, 2021 6:28 am
<input type="checkbox"/>	Run as →	kyogoku+bubble_test@guildw	Pet Owner	Oct 31, 2021 8:51 pm	Nov 20, 2021 6:28 am

ユーザー登録時にロールを指定できるようにする

では、次はユーザー登録時に飼い主なのかアドバイザーなのか指定して登録できるようにします。

登録画面はBubbleが用意しているものを使い回してきましたが、そこに手を入れます。

- ログインページ `index` に移動する
- `Re-enter Password` のラベルをコピーして、`Role` というラベルを配置する
- `Design` メニューの `Input forms` の中から `Dropdown` を選択して、パスワード入力欄の下に配置する

PetLog

Name

Email

Password

Re-enter Password

Role

Sign Up

Already have an account? [Log In](#)

- Dropdownの設定は以下のとおりとする
 - Element名 : Dropdown Role
 - Placeholder: Choose a role...
 - Choice style : Dynamic choices
 - Type of choices : Role
 - Choices source : All Role
 - Option caption : Current option > 's Display
 - Default value : Pet Owner
 - This input should not be empty : チェック

※画面イメージは次のページ

入力後イメージ

PetLog

Name
John Doe

Email
johndoe@gmail.com

Password

Re-enter Password

Role
Dropdown Role

Choose an role...

Sign Up

Dropdown Role

Appearance Layout Conditional

Placeholder Choose an role...

Choices style Dynamic choices

Type of choices Role

Choices source All Role

Option caption Current option's Display

Enable auto-binding on parent element's thing

Default value Pet Owner

This input should not be empty

This input is disabled

Style Attribute Standard Dropdown

Edit style Detach style

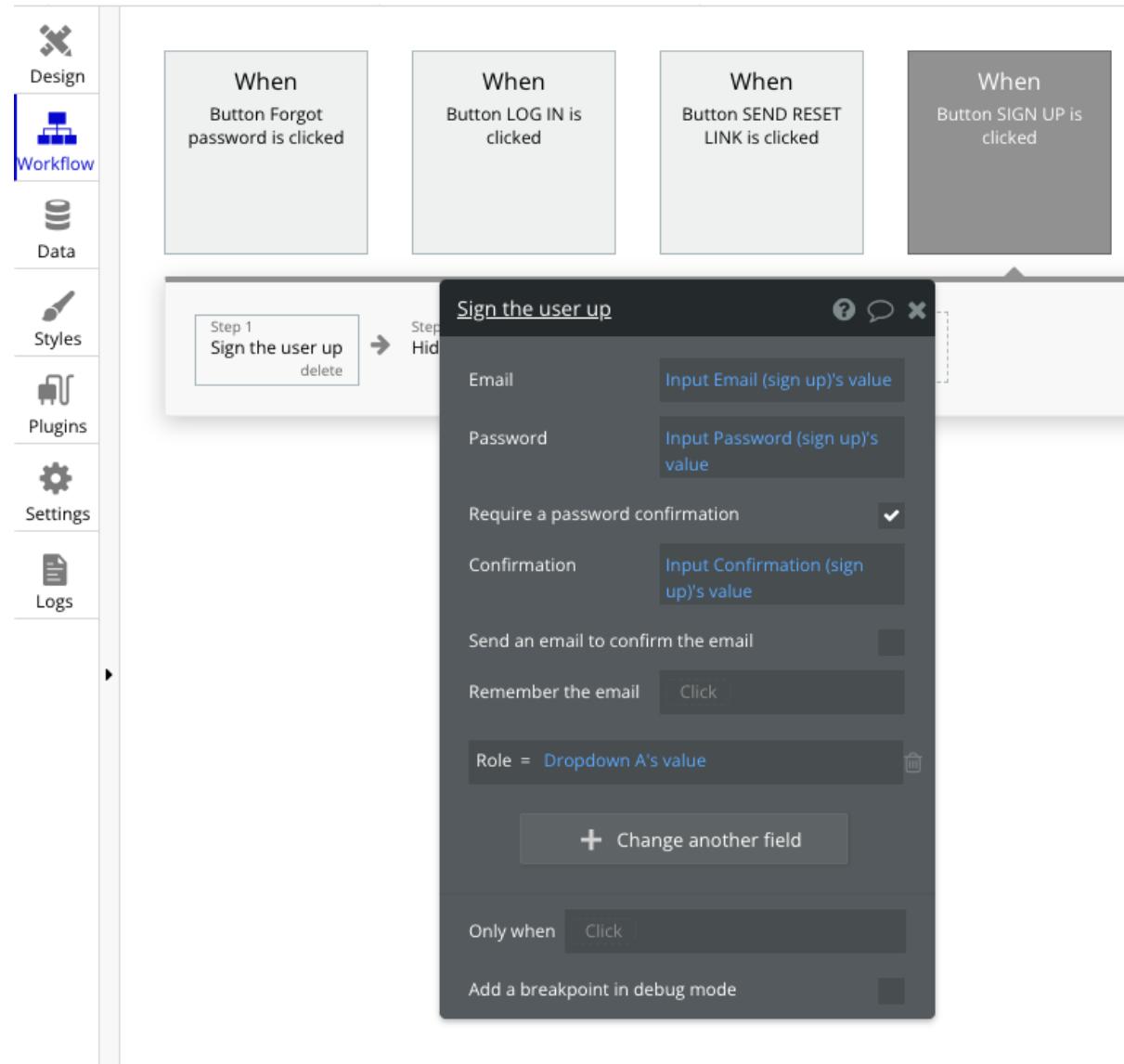
Appearance Settings

Opacity 100 %

App Font (Open Sans) 400

つづいて、入力されたRoleがユーザー登録時に設定されるようにします

- 左メニューから Workflow > 並んでいる正方形の中から Button Sign up is clicked > 並んでいるActionから Sign the user up の順に移動する
- Actionの設定画面内の Change another field ボタンをクリックする
- 入力欄があるので、Role = Dropdown Role 's value と選択する



プレビューと動作確認をしましょう

アドバイザーとして、アカウント登録ができた！（まだ専用画面なし）

PetLog

Name

Email

Password

Re-enter Password

Role

Sign Up

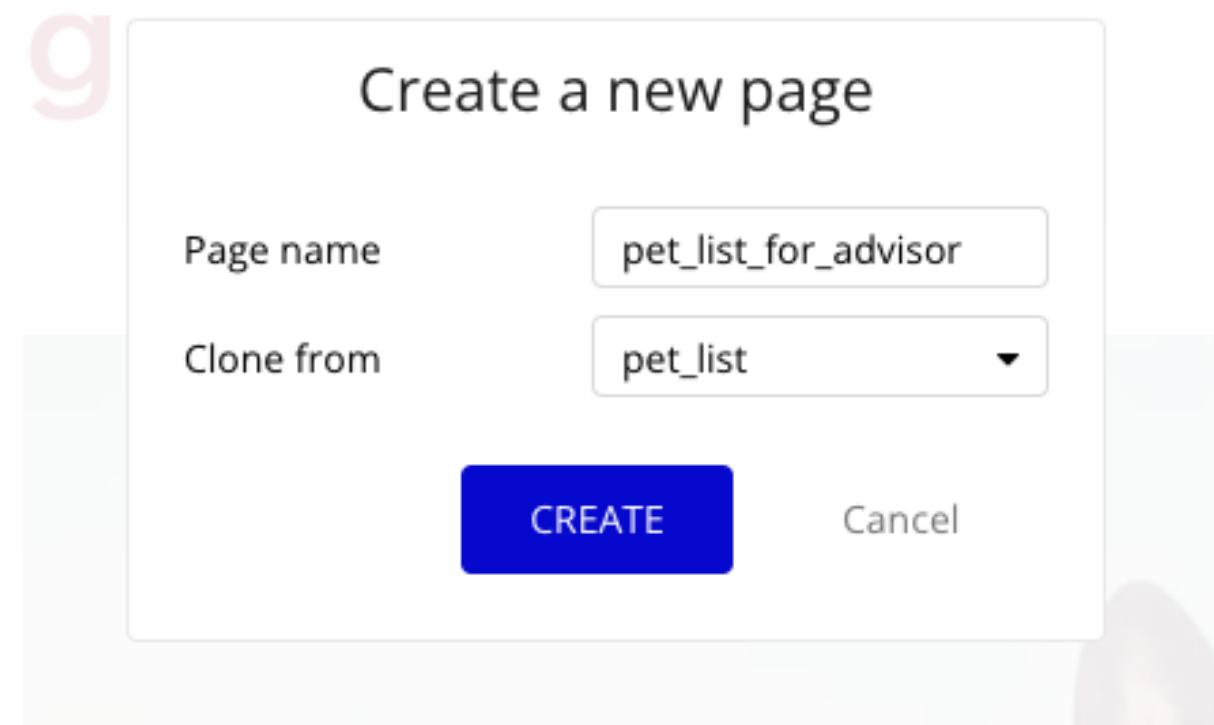
Already have an account? [Log In](#)

		Email	Name	Role
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Run as → kim+advisor@guildworks.jp	Sanhe Kim (as advisor)	Pet Advisor

アドバイザーの一覧画面を作成する

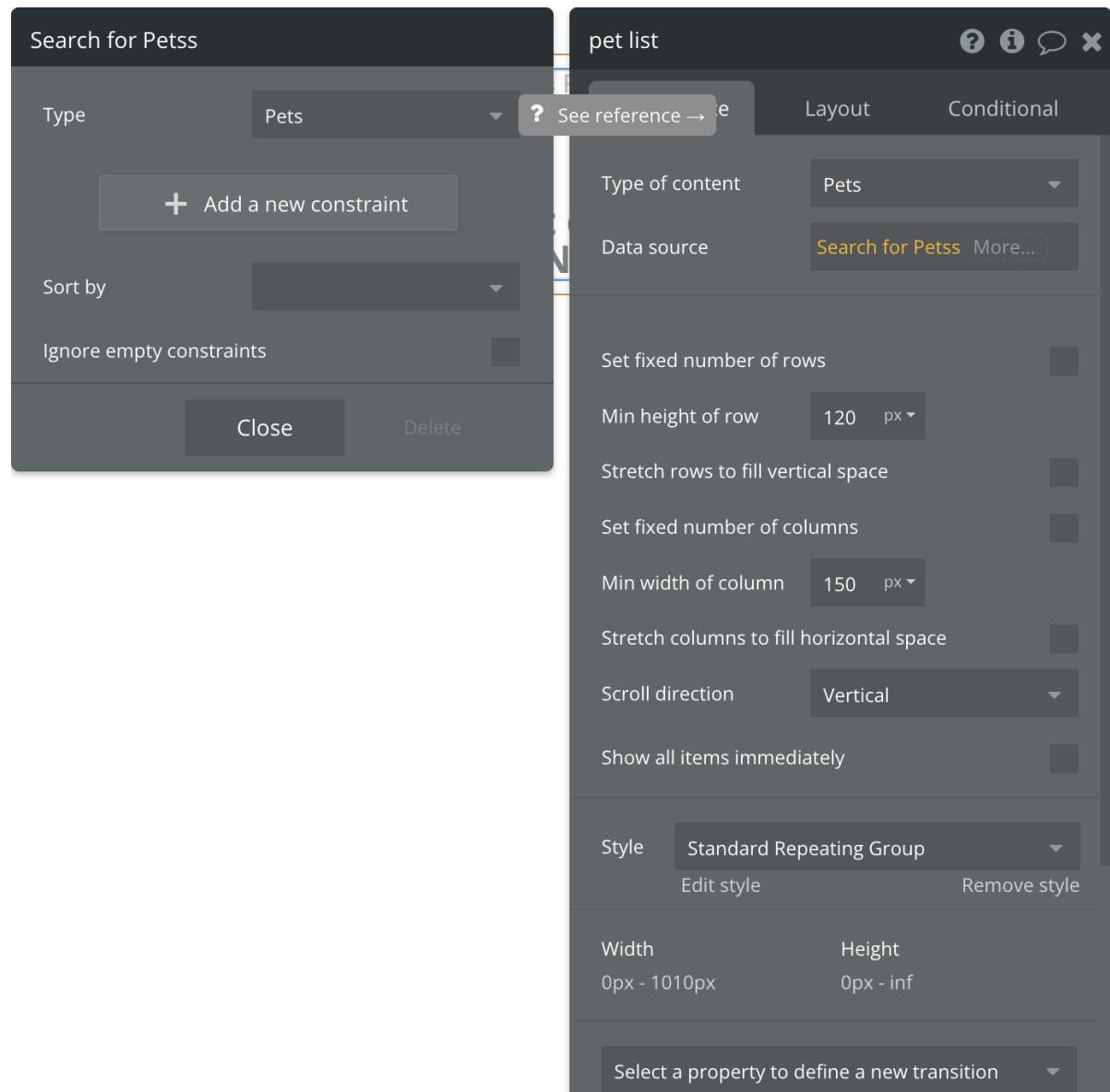
アドバイザーの一覧画面を作りましょう

- ① ロゴ横のメニューを開いて、
`Add a new page...`
- ② `Page name` に
`pet_list_for_advisor` と入力する
- ③ `Clone from` で `pet_list` を選ぶ
- ④ 新しく画面が作成される



アドバイザーは登録されたすべてのペットを参照できるようにします。

- もとの検索条件を削除する
 - Data source の Do search for をクリックする
 - Created By = Current User と条件指定されている部分をゴミ箱アイコンをクリックして削除
- アドバイザーはたくさんのペットを閲覧する必要があるので、一つ一つのセルのサイズは小さくする。
 - Min width に150pxと Min height に120pxを指定



プレビューしましょう

まだログイン後の遷移先は通常のペットリストになっているので、Previewボタンから直接アドバイザー用のペットリストを開きます。

...何も出ない！？なんで？？

PetLog [Pet Register](#) [Log out](#)

権限がないからです。

Bubbleでの権限制御

今まで意識しなくてよかったですが、
BubbleではDataへのアクセスが厳密に
制限されています。

左メニューの Data > タブの Privacy

と移動してください。

初期状態では、データは作成者しかア
クセスできなくなっています。

当然と言えば当然ですね。

The screenshot shows the Bubble Privacy settings interface. At the top, there are three items listed: 'Pets' (Privacy rules applied), 'PetWeightLogs' (Privacy rules applied), and 'User' (Privacy rules applied). Below this, a specific rule is defined:

Name	Visible to creator	
When	This Pet's Creator is Current User	
Users who match this rule can...		
View all fields	<input checked="" type="checkbox"/> Find this in searches	<input checked="" type="checkbox"/> View attached files

Below the rule definition, there is a section titled "Everyone else (default permissions)" with the following options:

View all fields	<input type="checkbox"/>	Birthday	<input type="checkbox"/>	Gender	<input type="checkbox"/>	Image
		Modified Date	<input type="checkbox"/>	Slug	<input type="checkbox"/>	Created By
Find this in searches	<input type="checkbox"/>	View attached files	<input type="checkbox"/>	Allow auto-binding	<input type="checkbox"/>	

A blue button labeled "Define a new rule" is located at the bottom right.

では、アドバイザーだったら、すべてのデータを見れるという権限を追加していきます。

- Data タブから、さらに Privacy タブの中で、Pets を選択する
- Define a new rule ボタンをクリックする
- Rule name に Visible to advisor と入力する
- When に Current User 's Role is Pet Advisor と選択する
 - ユーザーがアドバイザーだった場合という条件です

これでアドバイザーだったら、すべてのペットのデータが見れます。

ルールごとに参照できるフィールドを限定することができますが、今回は利用しません。

*画面イメージは次のページ

Data types Privacy App data Option sets File manager

Custom data types

Pets Privacy rules applied

PetWeightLogs Privacy rules applied

User Privacy rules applied

Data rules for type Pets

Name: Visible to advisor 讲话图标 清除图标

When: Current User's Role is Pet Advisor

Users who match this rule can...

View all fields Find this in searches View attached files Allow auto-binding

Name: Visible to creator 讲话图标 清除图标

When: This Pet's Creator is Current User

Users who match this rule can...

View all fields Find this in searches View attached files Allow auto-binding

Everyone else (default permissions)

View all fields Birthday Gender Image Name Created Date

Birthday Gender Image Name Created Date

Modified Date Slug Created By

Find this in searches View attached files Allow auto-binding

[Define a new rule](#)

Define a new rule

では、同じように PetWeightLogs にもルールを追加してください。
これでアドバイザーはすべてのデータを見れるようになっているはずです。

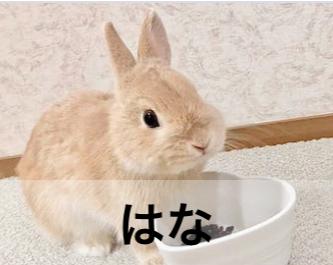
プレビューしましょう

わーい

PetLog

[Pet Register](#)[Log out](#)

もふこ



はな



ぴょんきち



うさまる



pochi



shiro



kuro
ups



pochi2

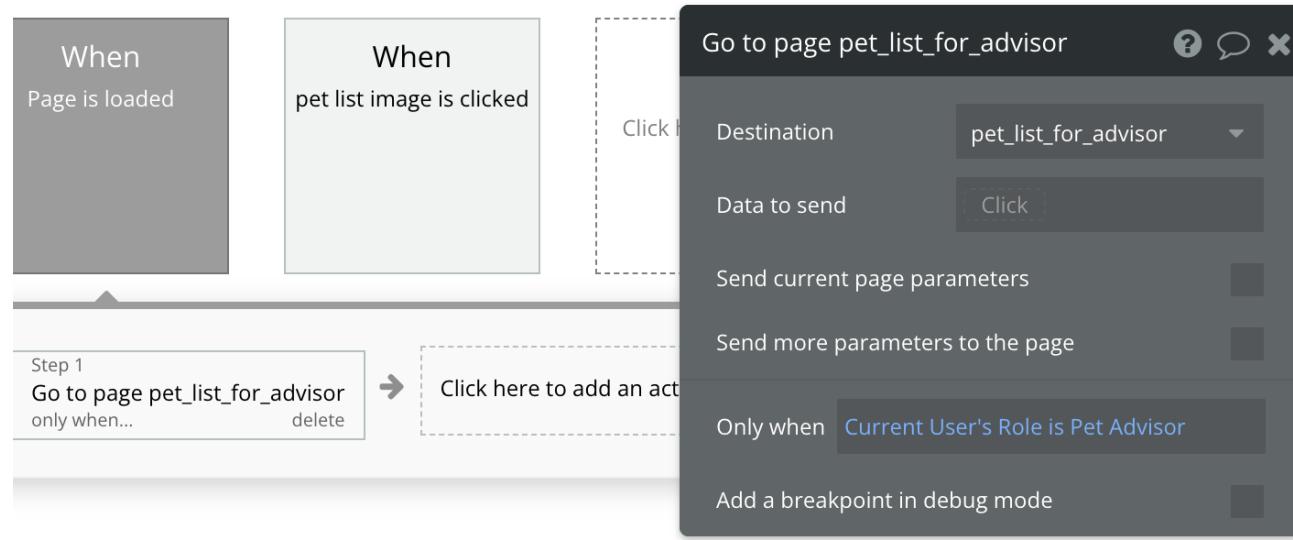


pochi

次に、ログイン時の遷移先を制御します。

アドバイザーの場合は、pet_listに遷移した際に、pet_list_for_advisorに遷移するというActionを追加します

- indexページのWorkflowをひらく
- Click here to add an event.. をクリックする
- General > Page is loaded を選択
- Click here to add an action.. をクリックする
- Navigation > Go to page.. をクリックする
- 設定が開くので、Destinationに pet_list_for_advisor を選択する
- Only when に Current User 'sRole is Pet Advisor を選択する



プレビュー&動作確認しましょう

アドバイザーでログインしたら

PetLog

Pet Register

Log out



飼い主でログインしたら

PetLog

[Pet Register](#)[Log out](#)

よしよし

<Advanced>

アドバイザーで勝手にアカウント作られて、

勝手に情報見られていの？

<Advanced>

システム管理者が承認しないと、利用開始出来ないようにしよう

<Advanced>

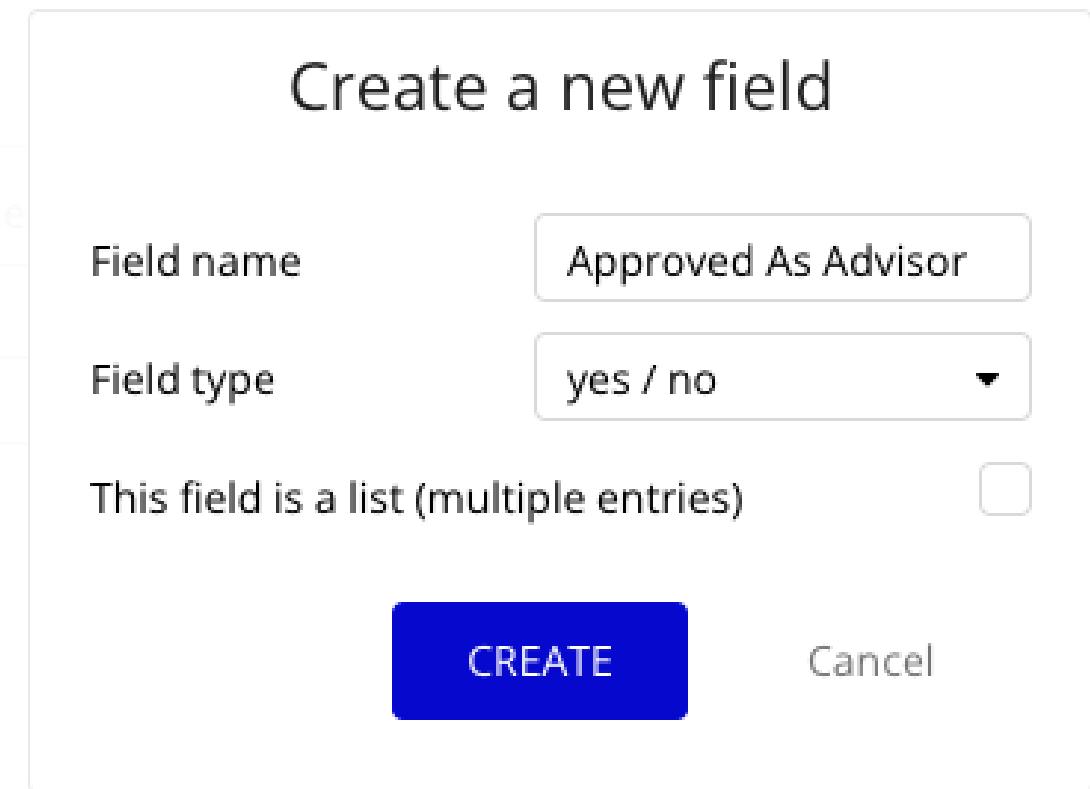
以下をやっていきます

- ユーザー情報にアドバイザーとして承認されているかを示すフィールドを追加する
- データへのアクセス権限はアドバイザーであることと同時に、承認されていることを条件に加える
- アドバイザー用のペットリストには、未承認だったら審査中ですというメッセージをだす

<Advanced>

ユーザー情報にアドバイザーとして承認されているかのフィールドを追加する

- 左メニューから Data >タブの Data types >中の User とクリックしていく
- 画面右下の Create a new field ボタンをクリックする
- Field nameに Approved As Advisor と入力する
- Field typeに yes/no を選択する
- Createボタンをクリック



<Advanced>

追加されたフィールドに `default` という欄があるので、`no`（もしくは `いいえ`）を設定しておく。

作成されたタイミングでは、未承認状態となる。

The screenshot shows the 'Fields for type User' configuration page. On the left, there's a sidebar with 'Data types', 'Privacy', 'App data', 'Option sets', and 'File manager'. Below that are sections for 'Custom data types' (listing 'Pets', 'PetWeightLogs', and 'User') and a 'New type' input field. A note says 'Make this data type private by default' with a checkbox and a message 'Things will be visible to everyone'. A 'Create' button is at the bottom of this sidebar.

The main area shows fields for the 'User' type. It includes:

- Type name: User
- Approved As Advisor: yes / no (highlighted with a red box)
- Role: Role (default dropdown set to 'no')
- email: text (Built-in field)
- Modified Date: date (Built-in field)
- Created Date: date (Built-in field)
- Slug: text (Built-in field)

At the bottom is a 'Create a new field' button.

<Advanced>

既存のユーザーについては、 Approved As Advisor はすべて no にしておく
(めんどうで、しょんぼり。。。だけど、大事！)

	Email	Approved As Advisor	Role	
<input type="checkbox"/>	Run as → kim+advisor3@guildworks.jp	no	Pet Advisor	Ni
<input type="checkbox"/>	Run as → kim+2@guildworks.jp	no	Pet Owner	Ni
<input type="checkbox"/>	Run as → kim+advisor2@guildworks.jp	no	Pet Owner	Ni
<input type="checkbox"/>	Run as → kim+advisor@guildworks.jp	no	Pet Owner	Ni
<input type="checkbox"/>	Run as → kim@guildworks.jp	no	Pet Owner	Ni
<input type="checkbox"/>	Run as → kyogoku+bubble_test2@guildw	no	Pet Owner	Ni
<input type="checkbox"/>	Run as → kyogoku+bubble_test@guildwo	no	Pet Owner	Oi

<Advanced>

データへのアクセス権限は承認されているかも見る

- 左メニューから Data > タブの Privacy > 中の Pets とクリックしていく
- Visible to advisor の When の条件が記載されている部分の末尾の Pet Advisor をクリックする
- More があらわれるので More をクリックする
- and Current User 's Approved As Advisor is "yes" と選択する
- PetWeightLogs も同様にする

<Advanced>

動作確認してみよう

Approved As Advisor が no のユーザーでログインしてみる

PetLog

Pet Register

Log out

よし

<Advanced>

Approved As Advisor が yes にしたら？

PetLog

Pet Register

Log out

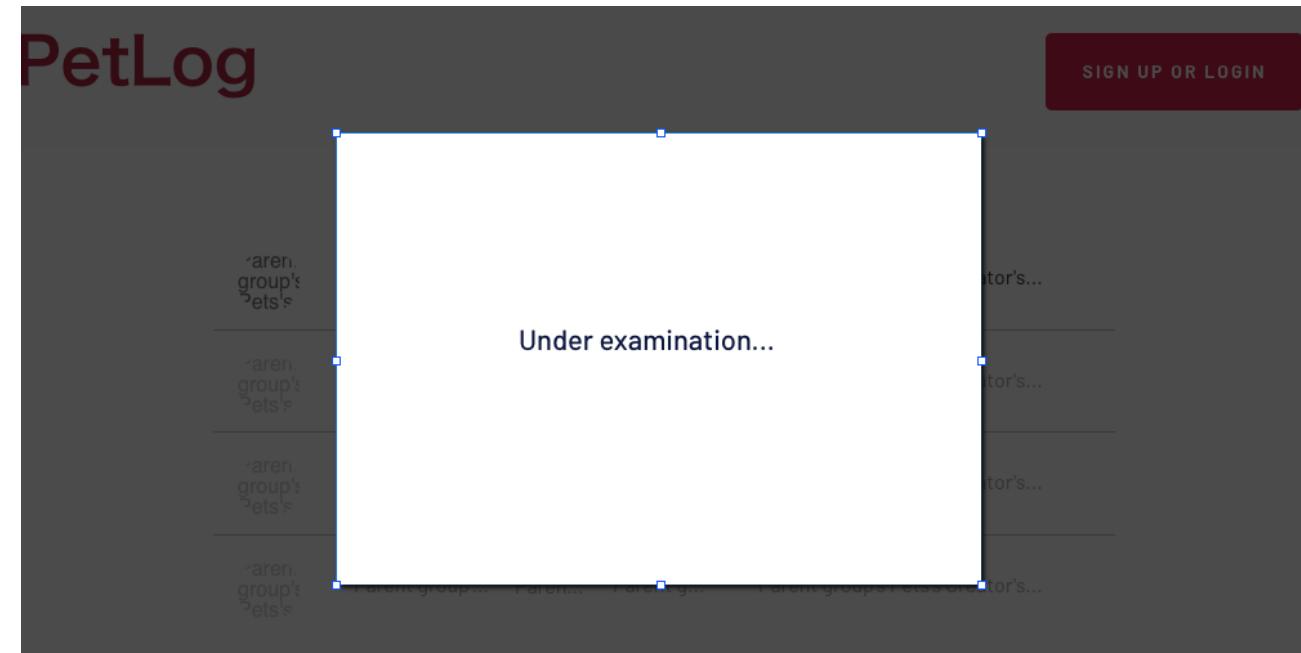


よし

<Advanced>

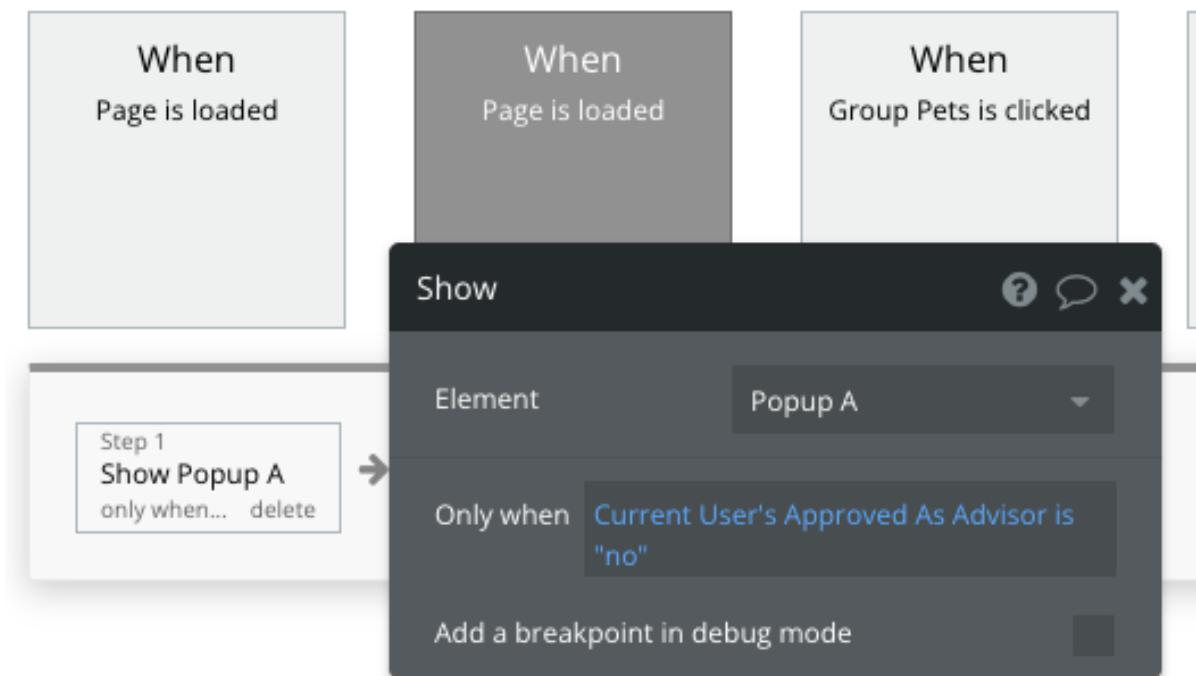
未承認だったら審査中ですと
いうメッセージをだす

- pet_list_for_advisorの画面で
Design メニューを開く
- Popup を追加する
- Popup の上にテキストElementを
追加して、審査中である旨のメッ
セージを記載する



<Advanced>

- メニューからWorkflowに移動する
- Click here to add an event... > Page is loaded とクリックする
- Click here to add an action... > Element Actions > Show とクリックする
- Elementに Popup A を指定する
- Only whenに、 Current User 's Approved As Advisor is "no" を指定する



動作確認してみよう

yesのアドバイザーなら

PetLog

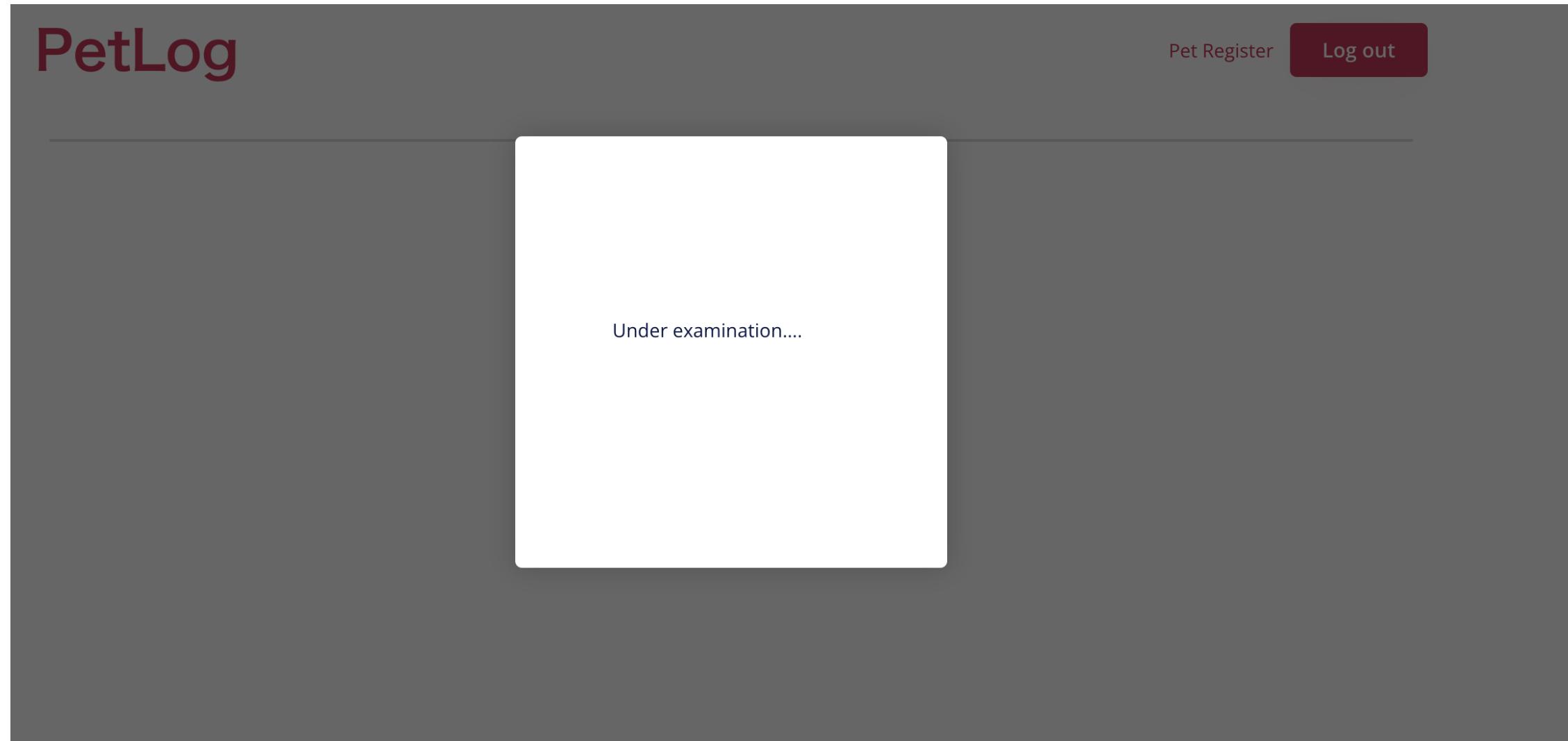
Pet Register

Log out



<Advanced>

noなら



よし

<Advanced>

システム管理者はどうやって気づくの？

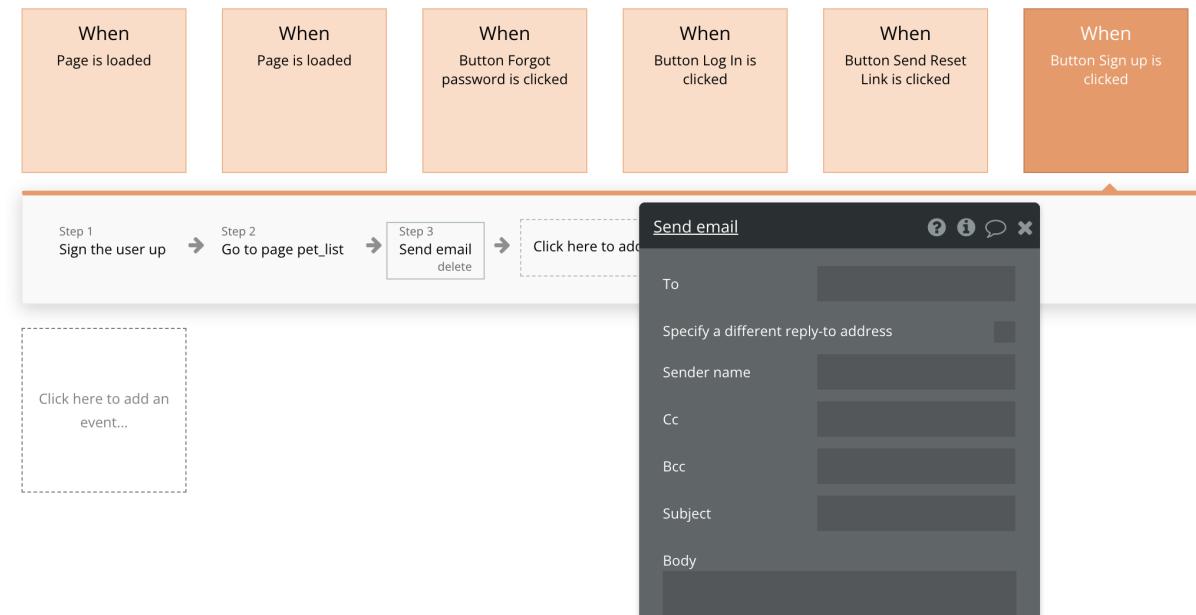
<Advanced>

アドバイザーの登録がされたら、システム管理者にメール通知されるようしょう

<Advanced>

システム管理者にメール通知 されるようにしよう

- index ページを開く
- メニューからWorkflowに移動して、Button Sign up is clicked を選ぶ
- Click here to ad an action... > Email > Send Email とクリックする
- Actionの位置を Go to page pet_list の前にドラッグして移動する

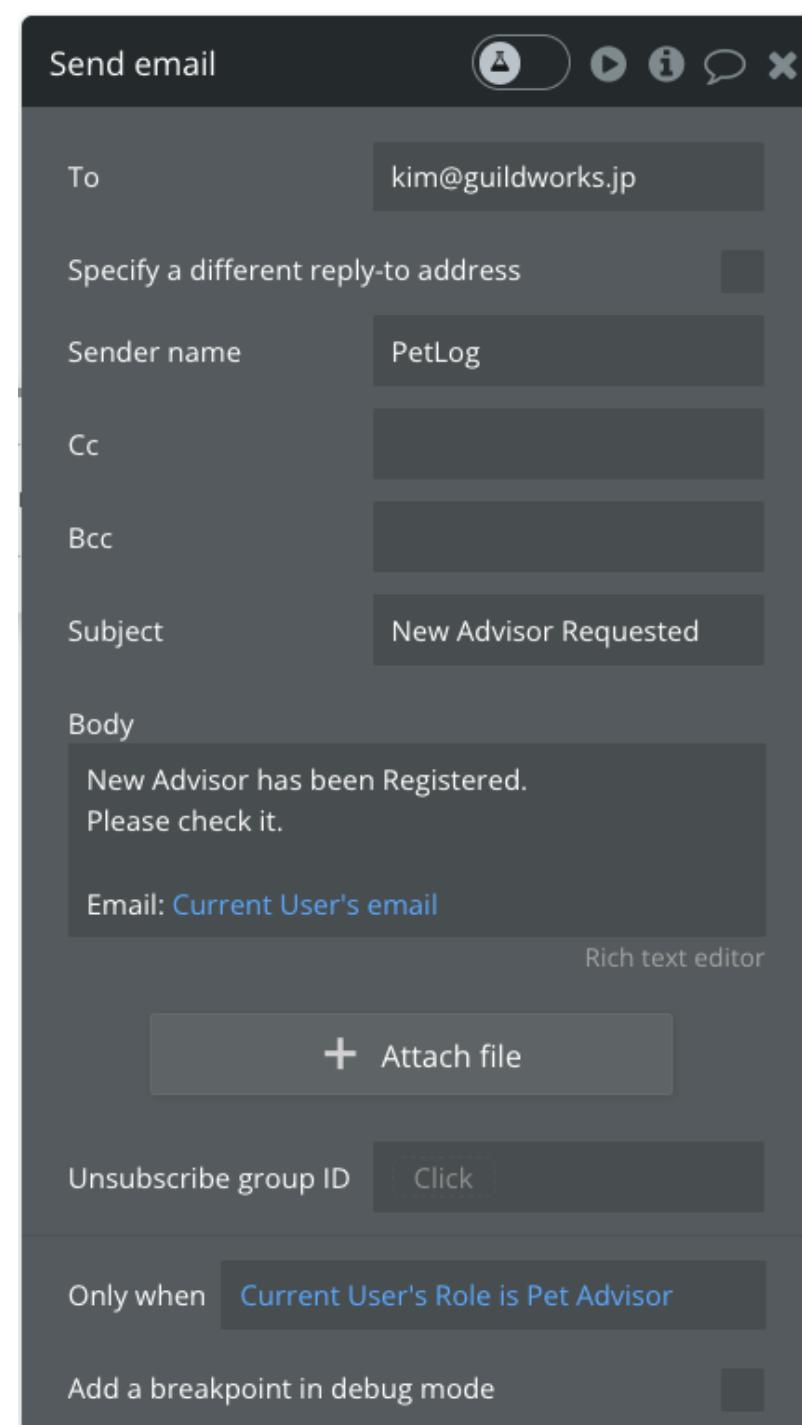


<Advanced>

- Toに自分のメールアドレスを設定する
- Sender nameは PetLog
- Subjectは New Advisor Requested
- Bodyは以下の本文の末尾に dynamic data insert で Current User 's email を選択

New Advisor has been Registered.
Please check it.

Email:



<Advanced>

- Only when に、 Current User 's Approved As Advisor is "yes" を指定する

<Advanced>

動作確認してみよう

アドバイザーでサインアップしたら

PetLog

New Advisor Registered

宛先: kim@guildworks.jp,

返信先: titech-bubble-2-suburi-2021119-no-reply@bubbleapps.io

New Advisor has been Registered.
Please check it.

Email: kim+advisor4@guildworks.jp

<Advanced>

飼い主なら

うん、こない。やったー

<Advanced>

演習

ここまでで学んだ、デザインやロジックの作り込み方を使って、機能を追加してみよう。

例えば、アドバイザーから飼い主、もしくは飼い主からアドバイザーにアプローチできる機能をつくって見よう

- 例：飼い主にアドバイスを送れる
- 例：広告を掲載できる
- 例：アドバイザーに相談を投げられる などなど

ここまでのおさらい

デザインを作り込みました

- ディスプレイサイズに合わせた画面をつくりました
 - レスポンシブウェブデザインという手法を使って、以下のようなルールを用いて、ディスプレイサイズに合わ見た目を制御しました。
 - 親要素内の配置ルール
 - 要素のサイズ決定ルール
 - 表示の有無ルール
- Styleを使ってみました
 - Styleを編集・追加したり、個別にスタイルをあてました

ロジックを作りこみました

- 画面操作に対するフィードバックを返しました
- データを抽出、加工しました
- 画面を権限によって切り替えました

Bubbleで様々なところにロジックを埋め込めるのを見ていきました

ここまでのかえり

以上で、Bubbleの基礎講義は終わりになります。

このあとはBubbleの応用編になります。

触れられなかった機能もたくさんありますが、

Bubbleはマニュアルとリファレンスが充実していますので、

もしBubbleを採用するならぜひ活用してください。

マニュアルはこちらです。

<https://manual.bubble.io/>

リファレンスは画面上でわからないものにカーソルを合わせると現れます。

だいたいの機能についてリファレンスへのリンクが浮き出できます。

では、次は応用編に参ります