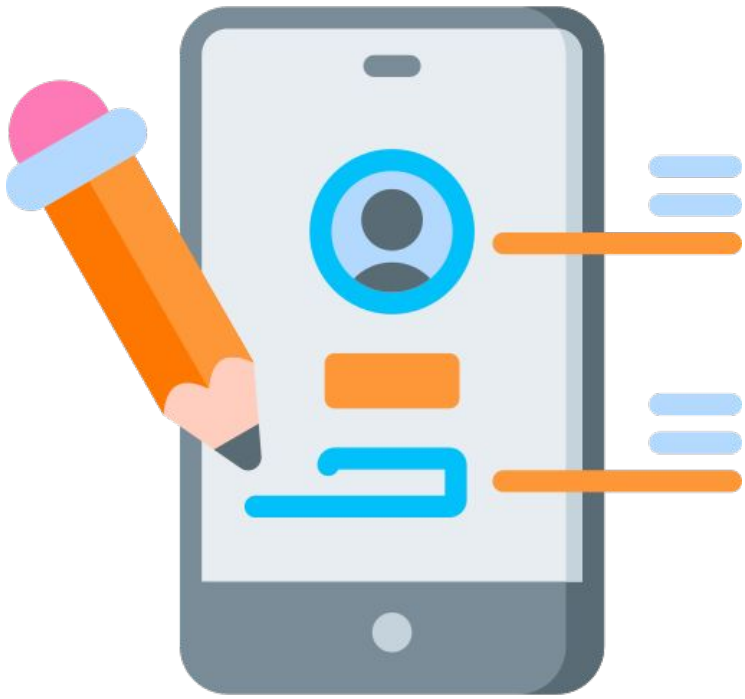
APP DESIGN



App Design mit thunkable

Lernkarten-Booklet

APP DESIGN

Richte die Elemente auf deinen **Screens** horizontal und vertikal aus.

(Layouting)

Stelle die **Abstände** zwischen Komponenten richtig ein.

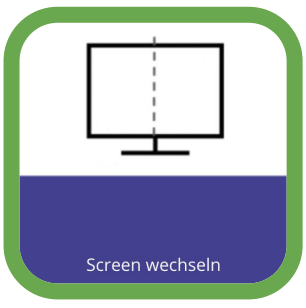
(Layouting)

Erstelle mit **Zeilen & Spalten** das perfekte Layout.

Layouting

So findest du Hilfe zu den Komponenten in der **Dokumentation**.

Basic



So kannst du in deiner App zwischen **Screens wechseln**.

Basic

Lerne, wie du **Variablen** verwendest und wofür du sie brauchst.

Basic

APP DESIGN

Baue **Google Maps-Karten** in deine App ein und zeige Marker an.

Basic

Zeige mit der **Webviewer**-Komponente eine Webseite in deiner App an.

Basic

Baue mit der **Übersetzer-Komponente** eine Übersetzungs-App.

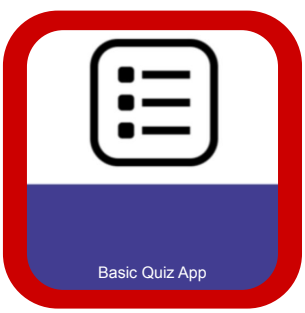
Basic

Data-Viewer - Zeige Elemente einer Datenbank als Kacheln oder Liste an

Advanced

Datasources & List Viewer - Daten hinzufügen, entfernen und auflisten

Advanced

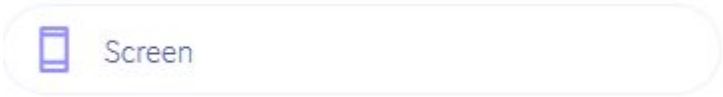


Basic Quiz App - Erstelle die Basis für dein eigenes Quiz

Advanced

Layouting mit Screens

Die Komponente **Screen**, die jede Seite auf der App darstellt, bietet einige grundlegende Layouting-Funktionen.

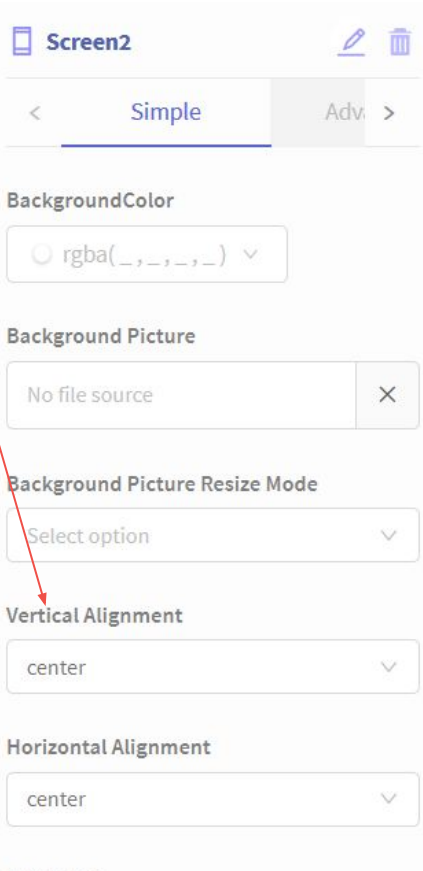


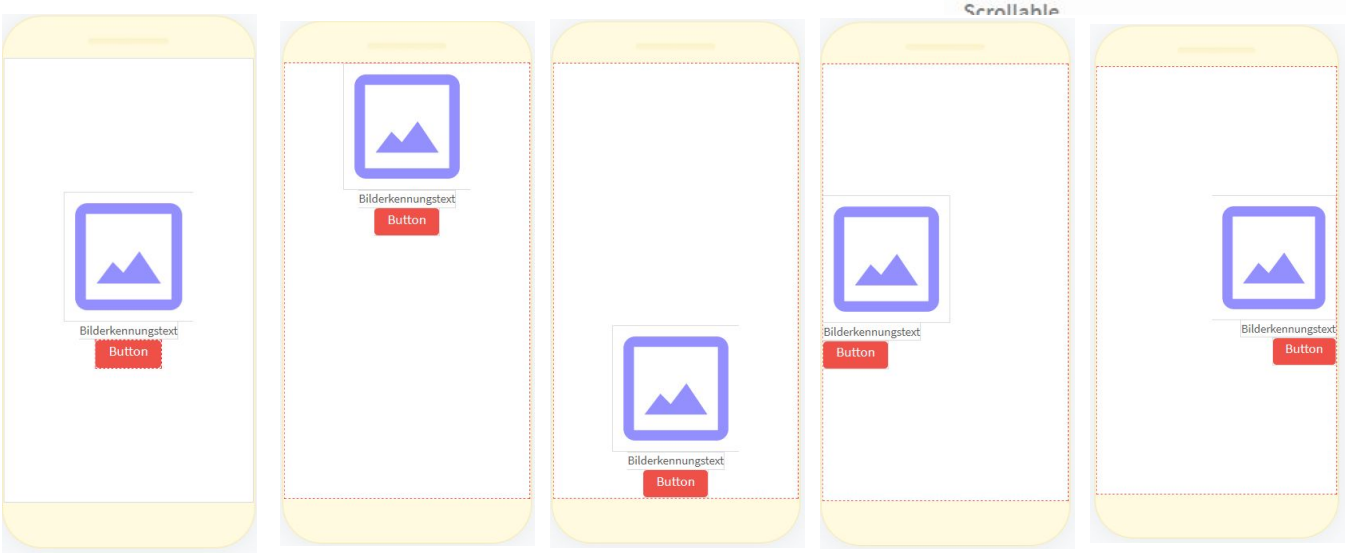
Das Beispiel verwendet einen Screen mit den Komponenten **Image, Label** und **Button.**

Die Anordnung der Elemente im Screen findet immer mit **allen** Elementen statt. Die Einstellung zur Anordnung findet man rechts in den Einstellungen.

Die Komponenten lassen sich auf zwei **Achsen**, der **X**- und **Y**-Achse, ausrichten.

Die Ausrichtung auf der X-Achse heißt **Horizontal Alignment (HA).**
Die Ausrichtung auf der Y-Achse heißt **Vertical Alignment (VA).**





VA: center
HR: center

VA: top
HR: center

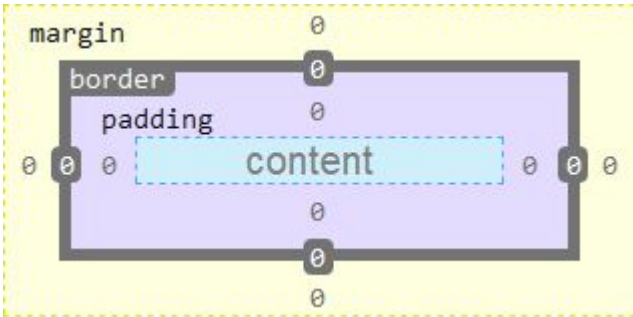
VA: bottom
HR: center

VA: center
HR: left

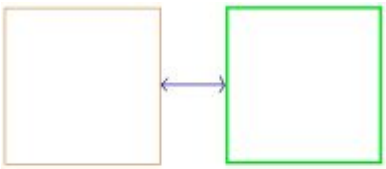
VA: center
HR: left

Layout Komponenten Finetuning

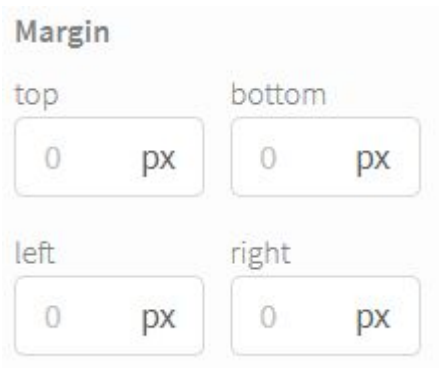
Jede sichtbare Komponente in Thunkable kann als ein Rechteck (**Container**) dargestellt werden. Dieses Rechteck hat einen Rand (**border**) und einen Inhalt (**content**). Für das Layouting sind zwei weitere Eigenschaften wichtig: **margin** und **padding.**



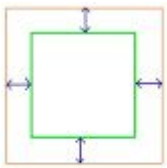
Margin beschreibt den Abstand vom **Rand** einer Komponente zum Rand einer **anderen** Komponente. Den Margin kann man über 4 Richtungen einstellen: Top, Right, Bottom und Left. Die Einstellungen für jede Komponente findet man im Fenster rechts:



Margin



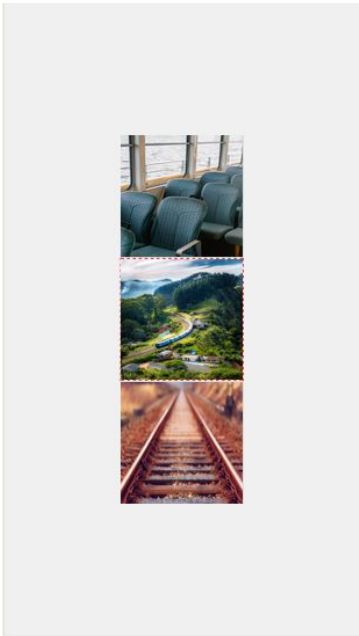
Padding beschreibt den Abstand vom **Rand** einer Komponente zu ihrem **Inhalt. Padding** kann man über 4 Richtungen einstellen: Top, Right, Bottom und Left.



Padding

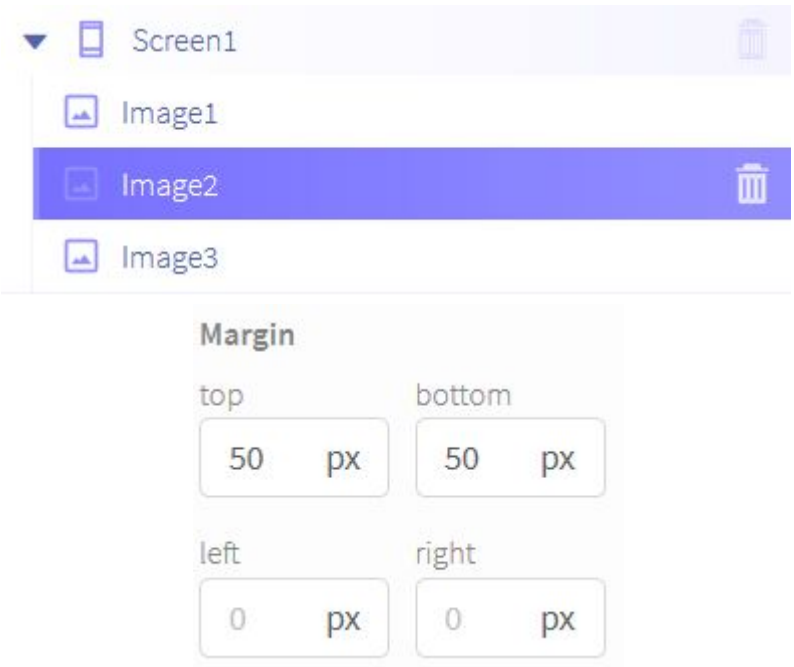
Layout Finetuning - Beispiele

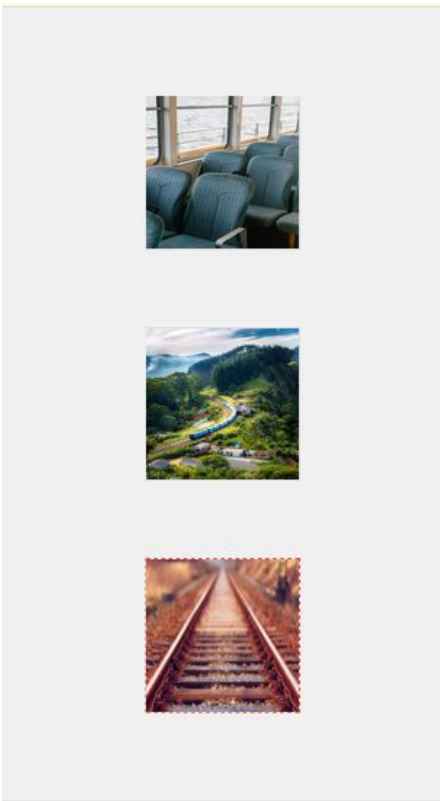
Als Grundlage dient eine App mit drei Bildern. Die Bilder wurden auf www.pexels.com gefunden und verwendet.



Beispiel 1

Damit die Bilder nicht so “aneinanderkleben” kannst du den **Margin** (Top und Bottom) vom mittleren Bild (Image2) um 50 px erhöhen. Dazu klickst du auf Image2 und änderst die **Margin-Einstellungen:**

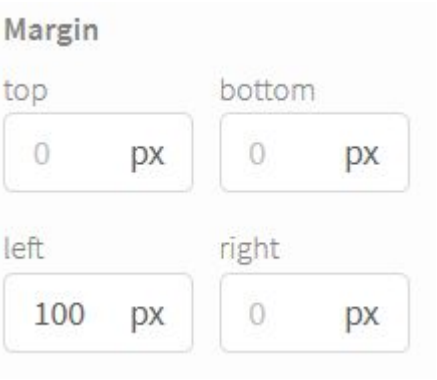




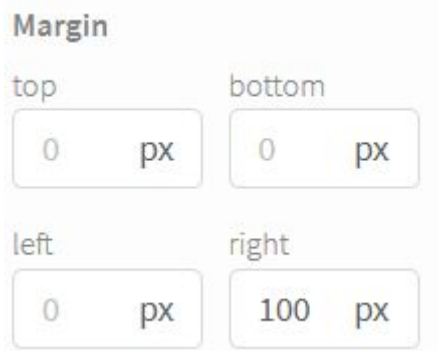
Layout Finetuning - Beispiele

Beispiel 2

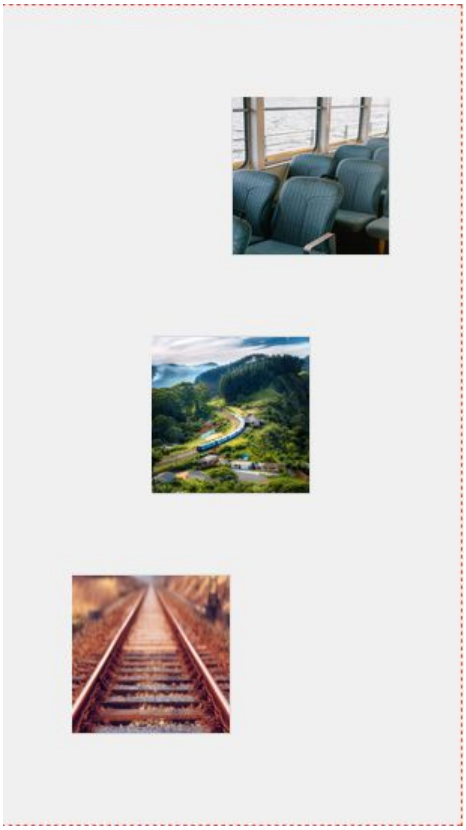
Als nächstes ordnest du die Images so an, dass sie stufenförmig erscheinen. Dazu änderst du von Image1 **Margin Left** auf 100 px und von Image3 **Margin Right** auf 100 px.



Margins von Image1



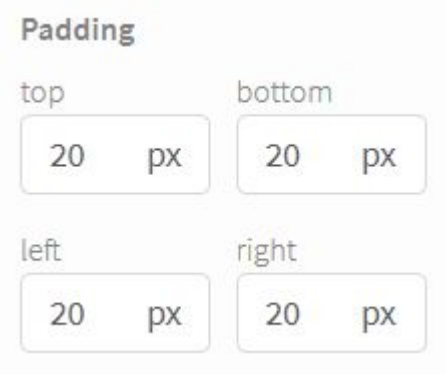
Margins von Image3



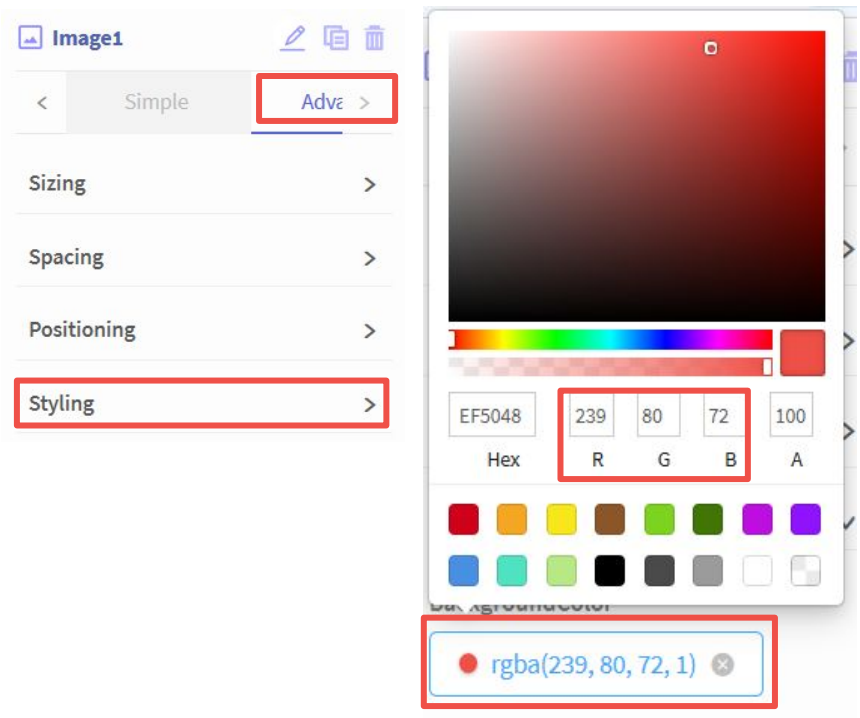
Layout Finetuning - Beispiele

Beispiel 3

Die Bilder sollen einen farbigen Rahmen bekommen. Ändere das **Padding** vom Bild auf 10 px (alle Richtungen).

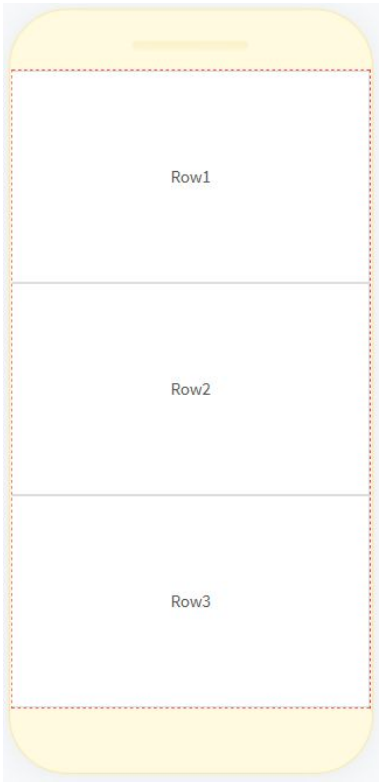


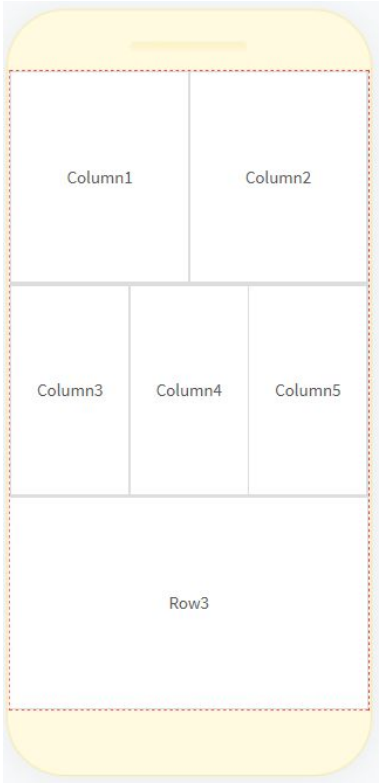
Als nächstes klickst du auf die **Advanced** Settings und dann auf **Styling**. Ändere Background Color auf **R 239, G 80** und **B 72.**



Layouting mit Rows und Columns

Thunkable bietet eine sehr einfache Möglichkeit, Layouts mit **Rows** (Zeilen) und **Columns** (Spalten) anzulegen.





In diese **Rows** and **Columns** kann man beliebige Komponenten platzieren. Somit kann man sehr schnell ein Layout für eine App erstellen.

Die Eigenschaften einer Row und einer Column sind dieselben wie die eines Screens (siehe Cheatsheet **Layouting mit Screens**). Das “Finetuning” passiert ebenso mit **Margin** und **Padding** (siehe Cheatsheet **Layout Finetuning**).

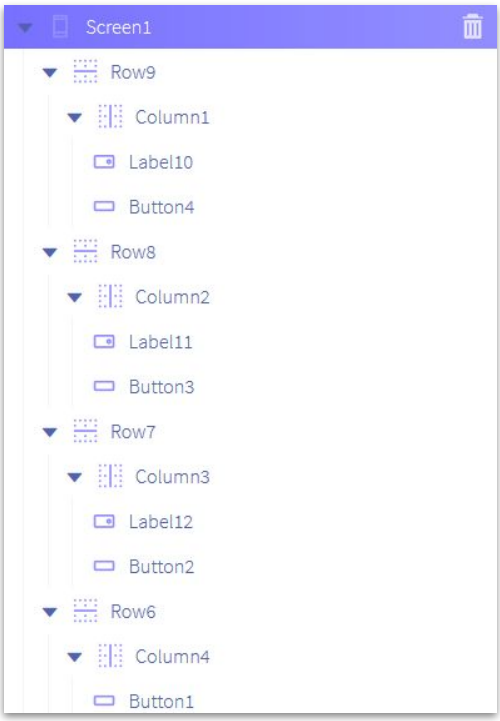
Layouting mit Reihen und Spalten

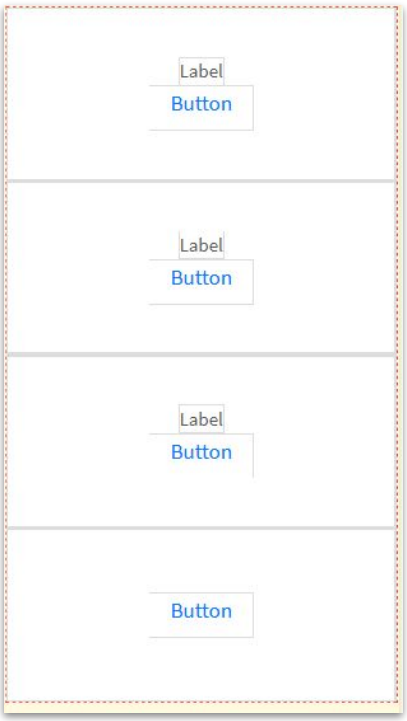
Layouting mit Rows und Columns

Beispiel - Menü

Rows und Columns werden verwendet, um ein Hauptmenü zu gestalten. Das Menü besteht aus 4 Rows zu jeweils 1 Column.

Wichtig: Achte auf die richtige **Ordnung** und **Unterordnung** der Elemente! Columns sind Rows immer **untergeordnet!**





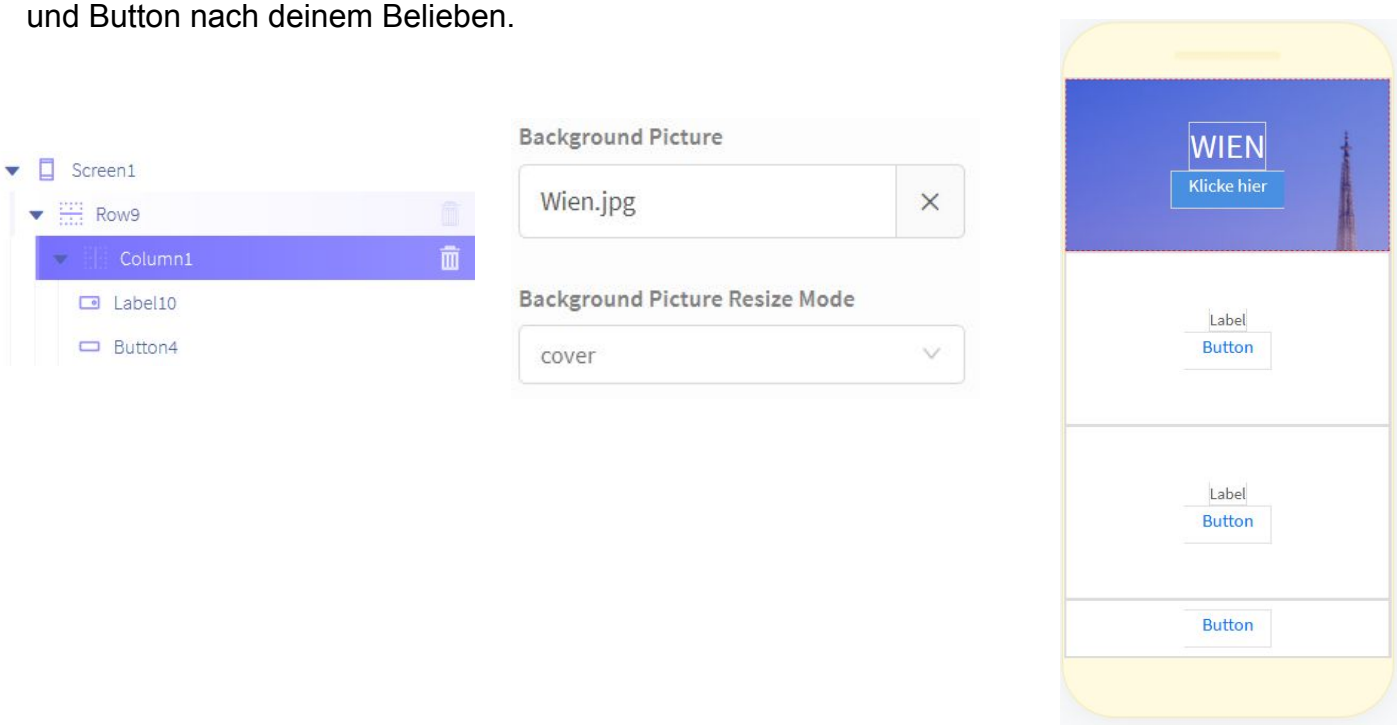
Layouting mit Reihen und Spalten

Layouting mit Rows und Columns

Beispiel - Fortsetzung

Als nächstes veränderst du die Höhe der untersten **Row** damit sie nicht so viel Platz einnimmt und somit mehr Platz für die Menüpunkte ist. Man ändert die Höhe auf 20 % des Bildschirms.

Wähle nun die **Column** in der **ersten Row** aus und füge ihr ein **Hintergrundbild (selbst zu gestalten)** hinzu und setze den **Background Picture Resize Mode** auf **cover.** Ändere das Label und Button nach deinem Belieben.

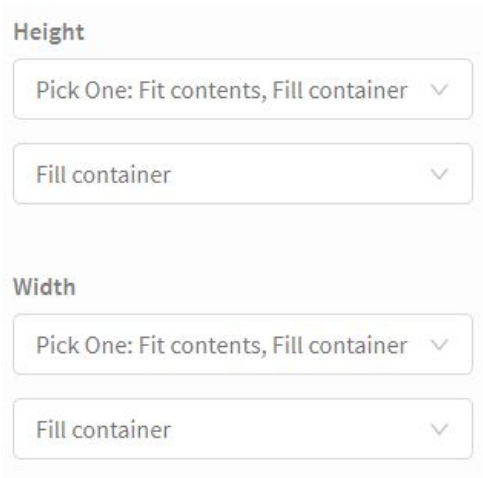


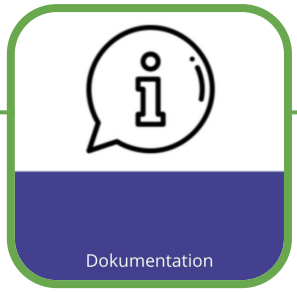
Wiederhole diese Schritte für die zwei weiteren Rows.

Layouting mit Rows und Columns

Beispiel - Fortsetzung

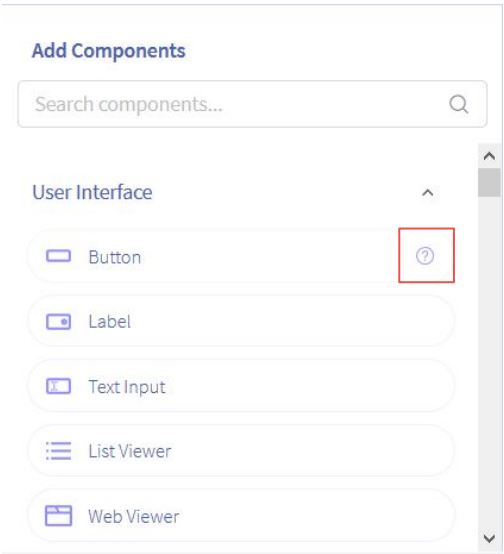
Als letztes fehlt noch die Gestaltung des letzen Buttons. Setze die Height und Width Einstellungen auf “Fill container”, damit der Button den gesamten Platz der Column befüllt.



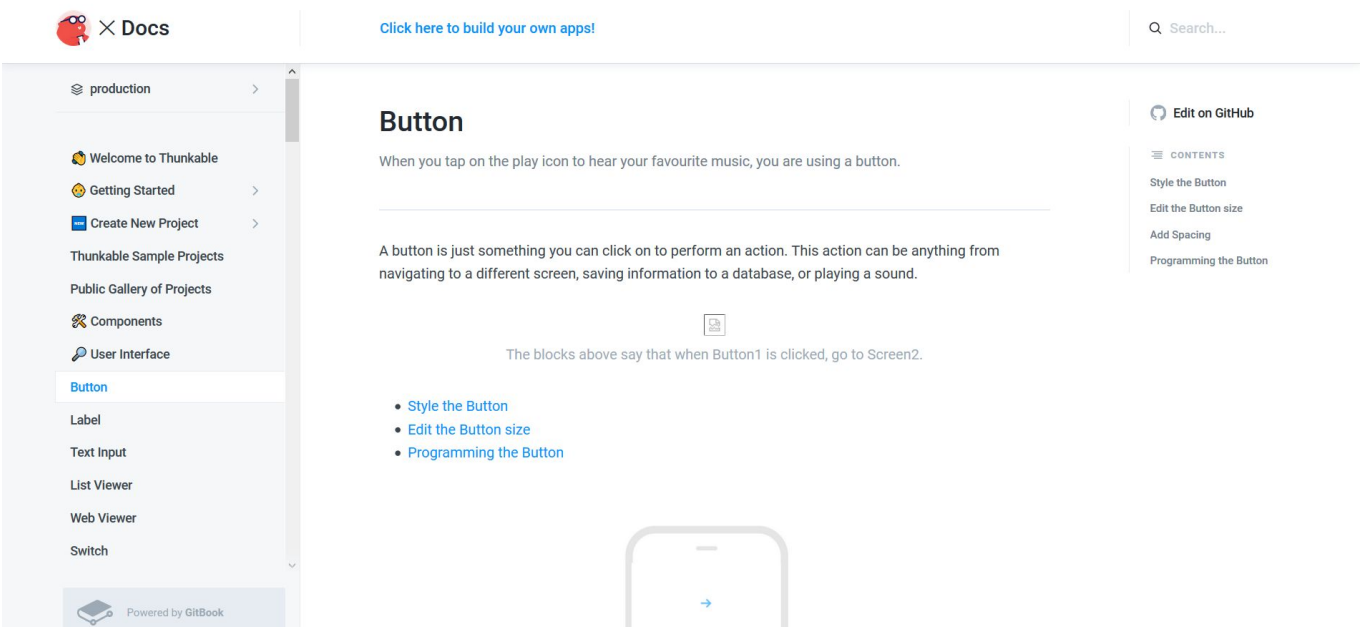


Dokumentation

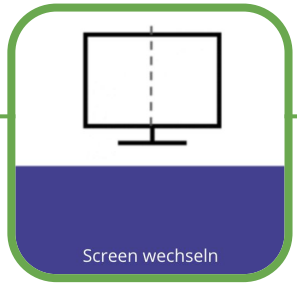
Jede Komponente in Thunkable hat einen **Eintrag** in der **Dokumentation.** Dazu musst du nur auf das **Fragezeichen neben** jeder Komponente klicken.



In der Dokumentation finden sich viele Beschreibungen und Beispiele zum Verwenden der Komponenten.



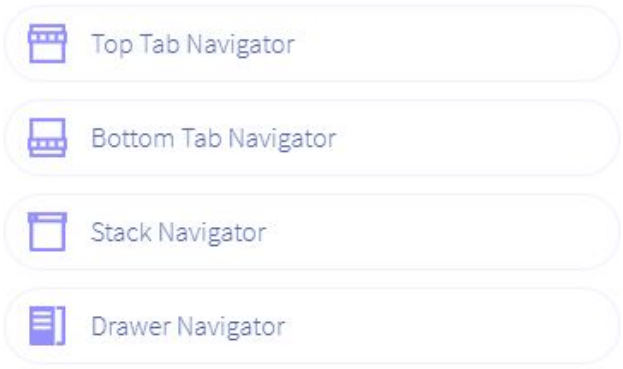
BASIC



Navigatoren: Zwischen Screens wechseln

1

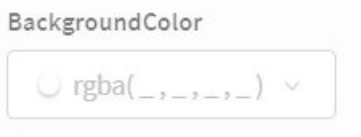
Verwende einen **Navigator**:



2

Du kannst mehrere Screens zu einem Navigator hinzufügen (Achte auf die **Einrückung**!)

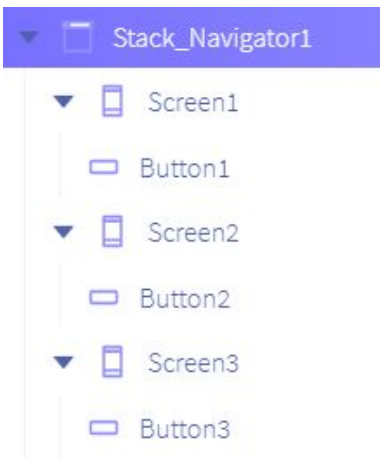
Gib den Screens eine **Farbe**, damit du den Wechsel zwischen den Screens sehen kannst.



3

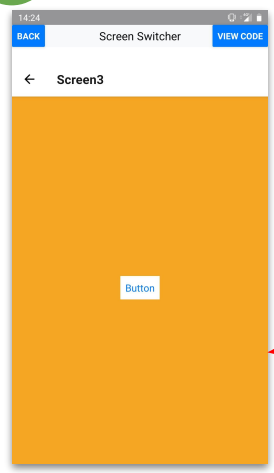
Beim **Stack Navigator** werden die Screens übereinander gelegt, du kannst nur den obersten Screen (der am “Stack” oben liegt) sehen:

Füge nun jedem Screen einen Button hinzu:



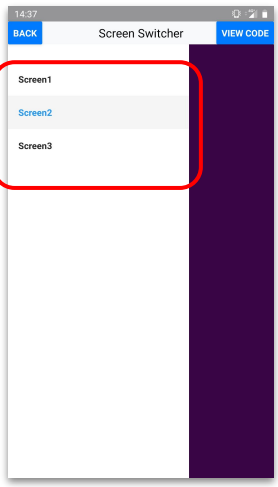
Die Buttons kannst du so **programmieren**, um zwischen den Screens zu **wechseln** (achte beim Programmieren darauf, welcher Screen gerade ausgewählt ist):

4

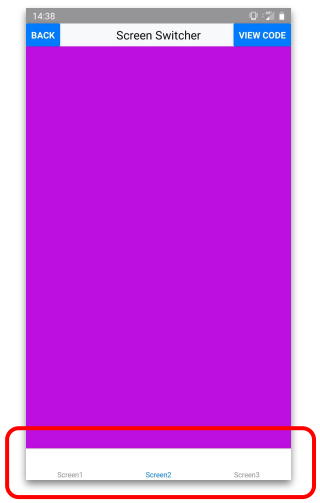


So sieht das z.B. beim **StackNavigator** aus. Mit jedem Klick auf den Button wird ein neuer Screen über den aktuellen gelegt. Mit dem Back-Button links oben kannst du zurücknavigieren.

Stack Navigator



Drawer Navigator



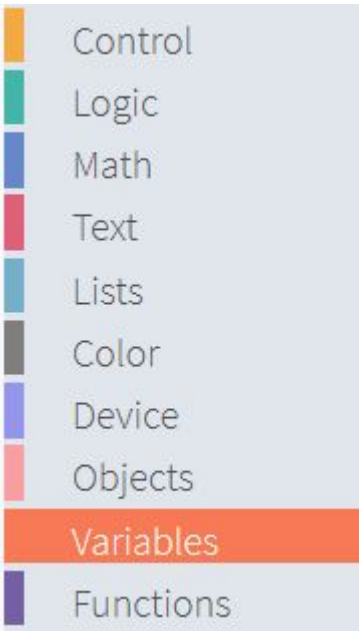
Bottom Tab Navigator

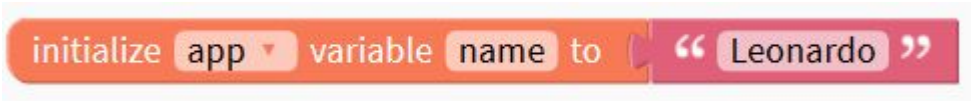
Variablen

Variablen (1/2)

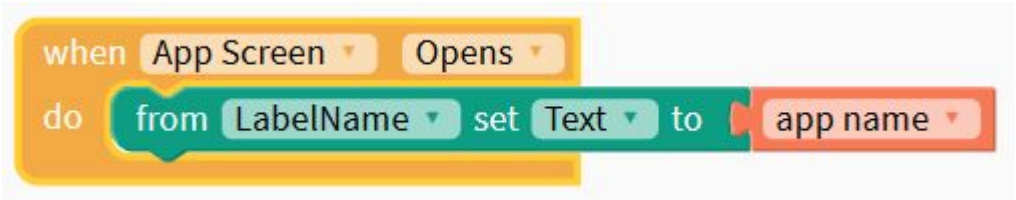
Eine **Variable** dient als eine Art **“Behälter”,** in dem man Werte **speichern** kann (zum Beispiel Zahlen oder Zeichenketten). Diese können dann an anderen Orten, zum Beispiel anderen Screens, **wiederverwendet** werden.

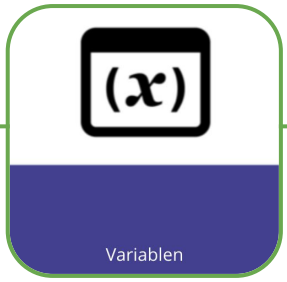
Anlegen einer Variable





Auslesen des Werts einer Variable in ein Label

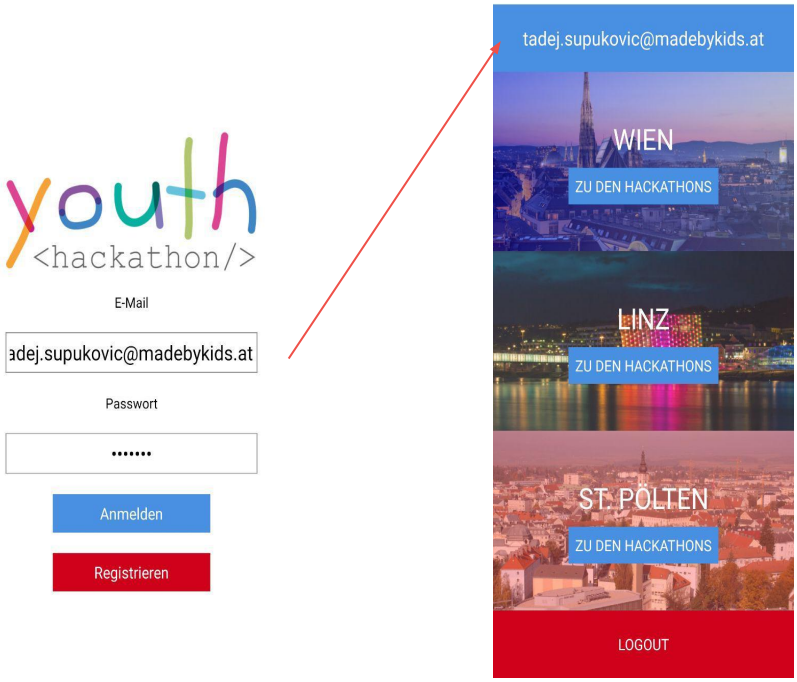


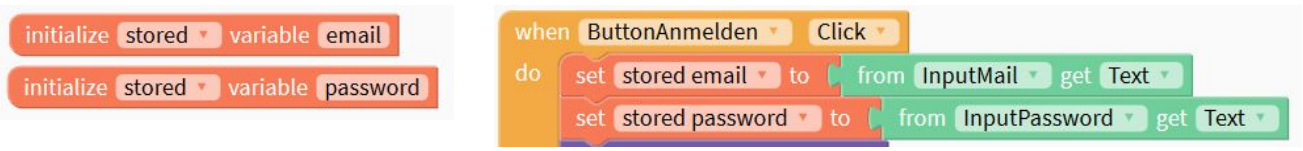


Variablen (2/2)

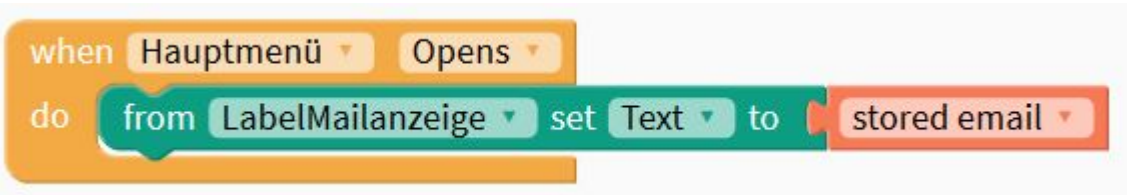
Beispiel: E-Mail-Adresse im Hauptmenü anzeigen

Das Beispiel zeigt eine im Login-Feld eingegebene Mail-Adresse im Hauptmenü als Label an.

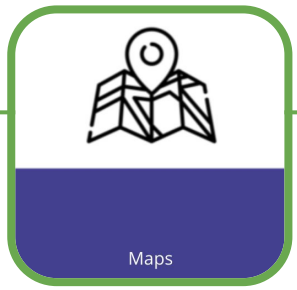




Initialisieren der Variablen und Wertzuweisung beim Einloggen.



Anzeigen der gespeicherten Mail-Adresse im Label.

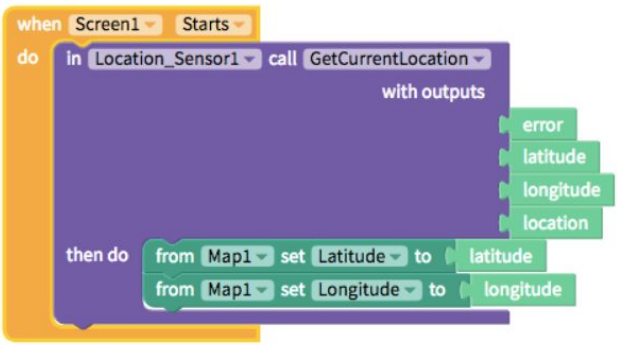


Karten (Maps) in deine App einbauen

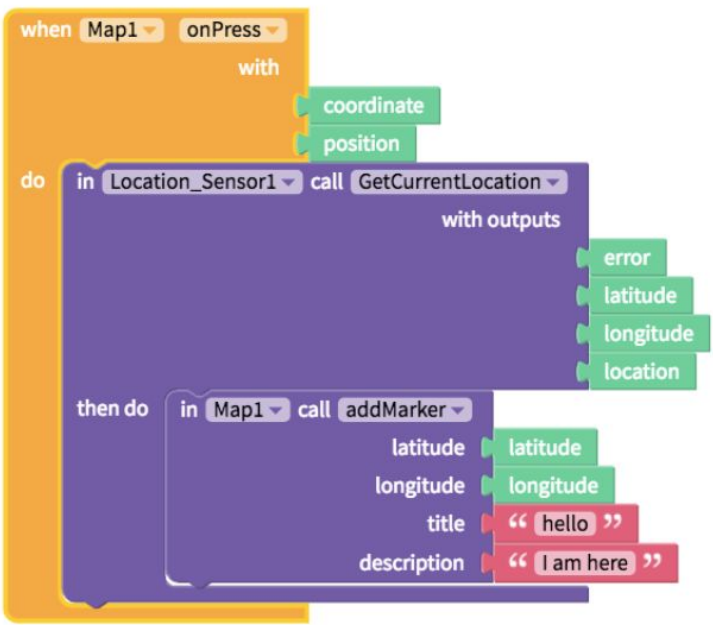
Mit der Google Maps-Komponente (sie lässt sich für iPhones auch auf Apple Maps umkonfigurieren) kannst du Karten in deine App einbauen. Ziehe dazu die Komponente **Map** auf die App-Vorschau:

Füge jetzt auch noch einen Location Sensor hinzu:

Mit diesem kannst du die Map jetzt auf deinen aktuellen Standort auto-zentrieren:

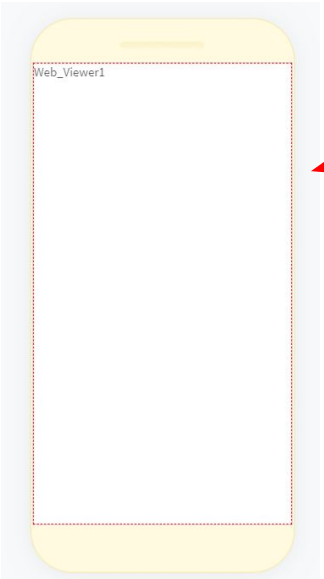


So kannst du auch noch einen **Marker mit Beschriftung** hinzufügen:



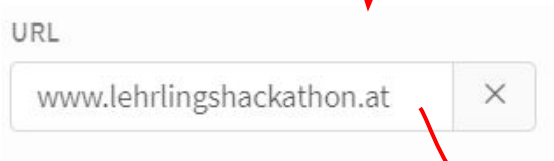
Tippe auf den Marker, um die **Beschriftung** zu sehen, die du bei *title* und *description* angegeben hast.

Webseiten mit Webviewer anzeigen

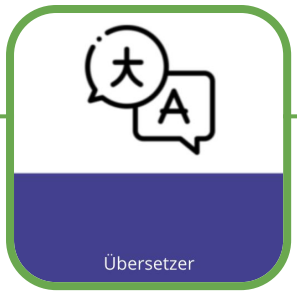


Mit der **Webviewer-Komponente** kannst du eine **Website** in deiner App anzeigen! Ziehe dazu als erstes die Komponente **Webviewer** auf die App-Vorschau in Thunkable.

In den **Properties** der Webviewer-Komponente kannst du einstellen, auf welche Seite der Webviewer gehen soll:



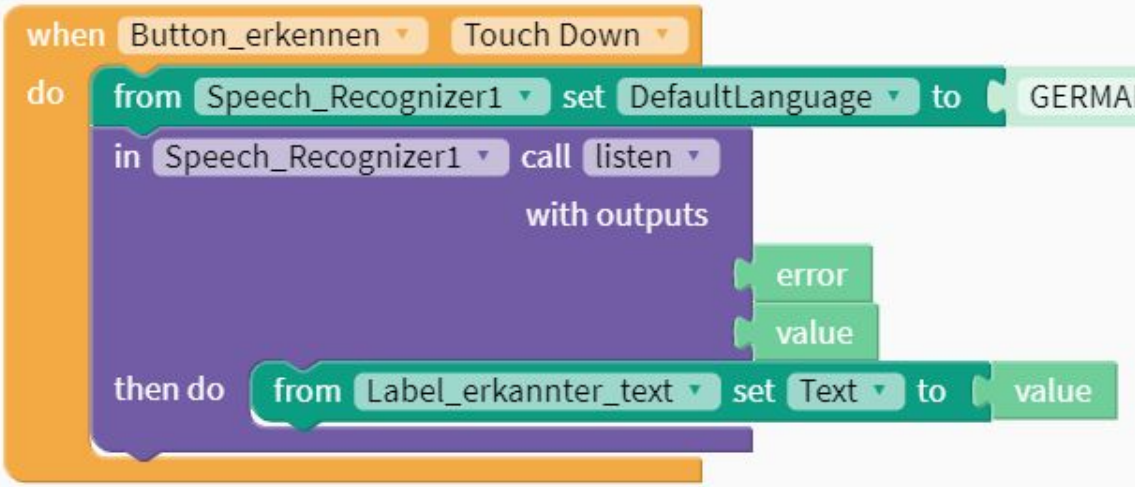
So kannst du, durch einen Button-Klick, auf eine andere Website wechseln:



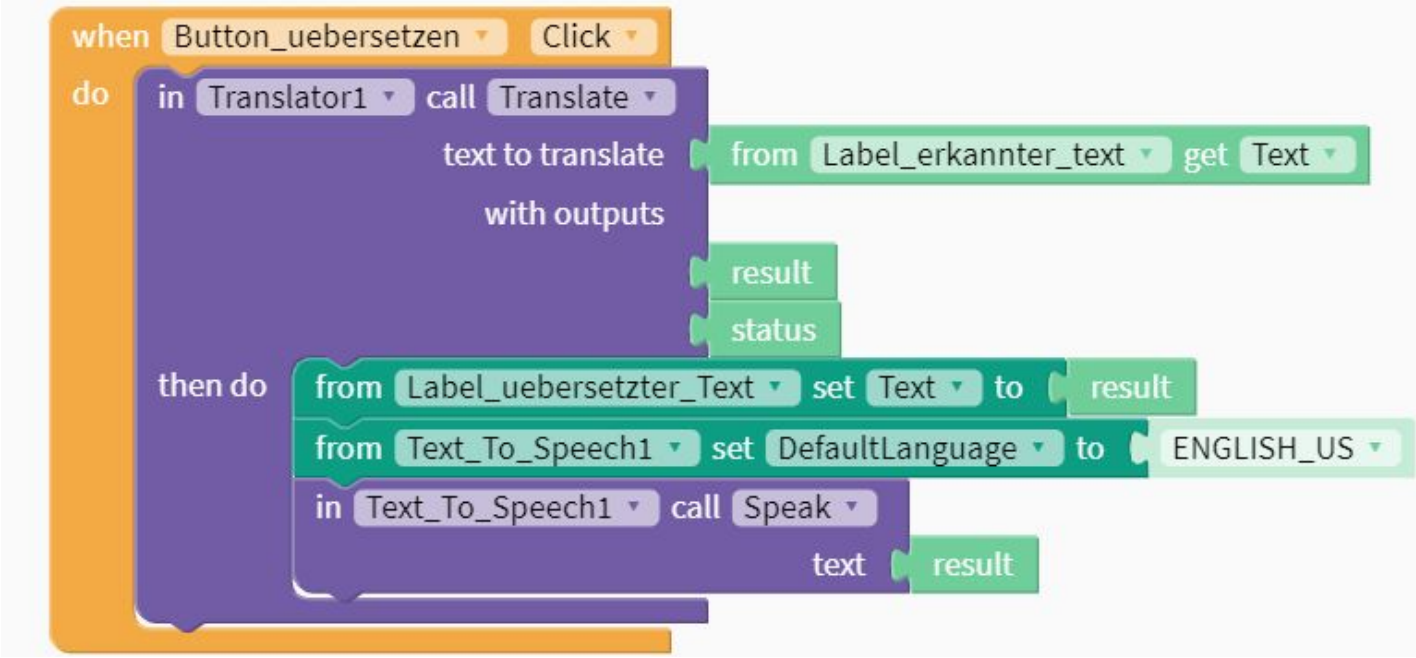
Einen Übersetzer bauen

Mit den Komponenten **Speech Recognizer**, **Translator** und **Text to Speech** kannst du eine Übersetzer-App bauen.

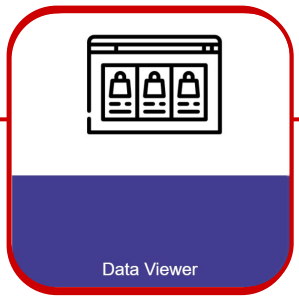
Wenn ein Button gedrückt wird, hört der **Speech Recognizer** zu - so lange der Button gedrückt bleibt (**Touch Down**) - und lädt den erkannten Text (**value** aus den Outputs) in ein Label:



Klickt man nun auf einen anderen Button, wird der erkannte Text an den **Translator** geschickt und übersetzt (in den Properties des Translators musss man die Ausgangs- und die Zielsprache angeben). Der übersetzte Text (**result** aus den Outputs) kann jetzt mit **Text to Speech** vorgelesen werden:



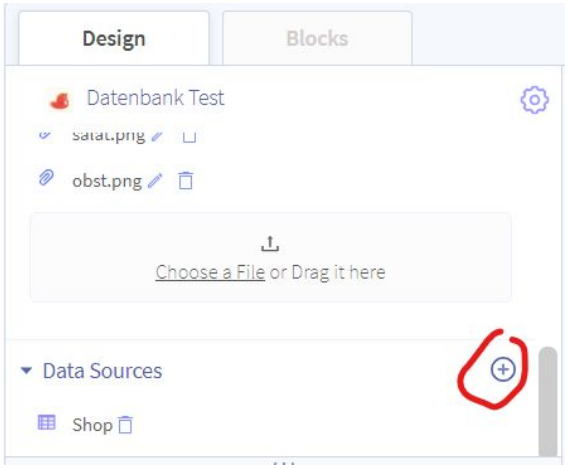
ADVANCED

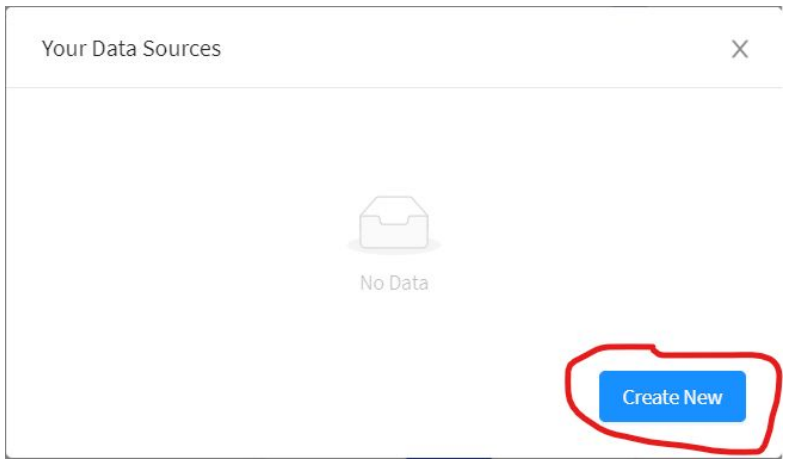


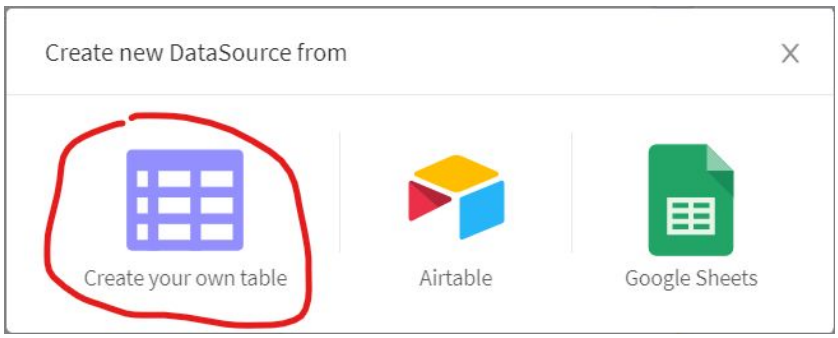
Tabellendaten in gestylter Liste anzeigen

In Thunkable lässt sich sehr gut mit Tabellen und Datenbanken arbeiten. Es gibt verschiedene Möglichkeiten, diese in seinem Projekt zu integrieren. Wir gehen hier auf eine der lokalen Funktionen ein - der "**Data Source**", die speziell für die **Data Viewer Komponenten** verwendet wird. Diese Komponenten sind sehr praktisch für Webshops oder ähnliche Prototypen.

Hinzufügen einer Tabelle:







Man kann in Thunkable auch über Cloud-Plattformen auf Tabellen zugreifen, doch für die Erstellung eines Prototyps ist es einfacher und es reicht vollkommen aus, eine lokale Tabelle zu verwenden.

Fülle die Tabelle mit deinen Daten

Hier kannst du nun deine Daten für die Weiterverarbeitung einfügen. Es ist auch möglich, die Daten aus einer Excel- oder Google-Tabelle hinein zu kopieren.

Shop

Table 1

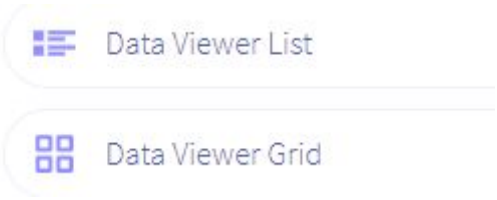
Produkt × | Preis × | Bild × | + New Column

	Produkt	Preis	Bild
1	Semmel	2	semmel.png
2	Salat	3	salat.png
3	Obst	1	obst.png
4			

ADVANCED

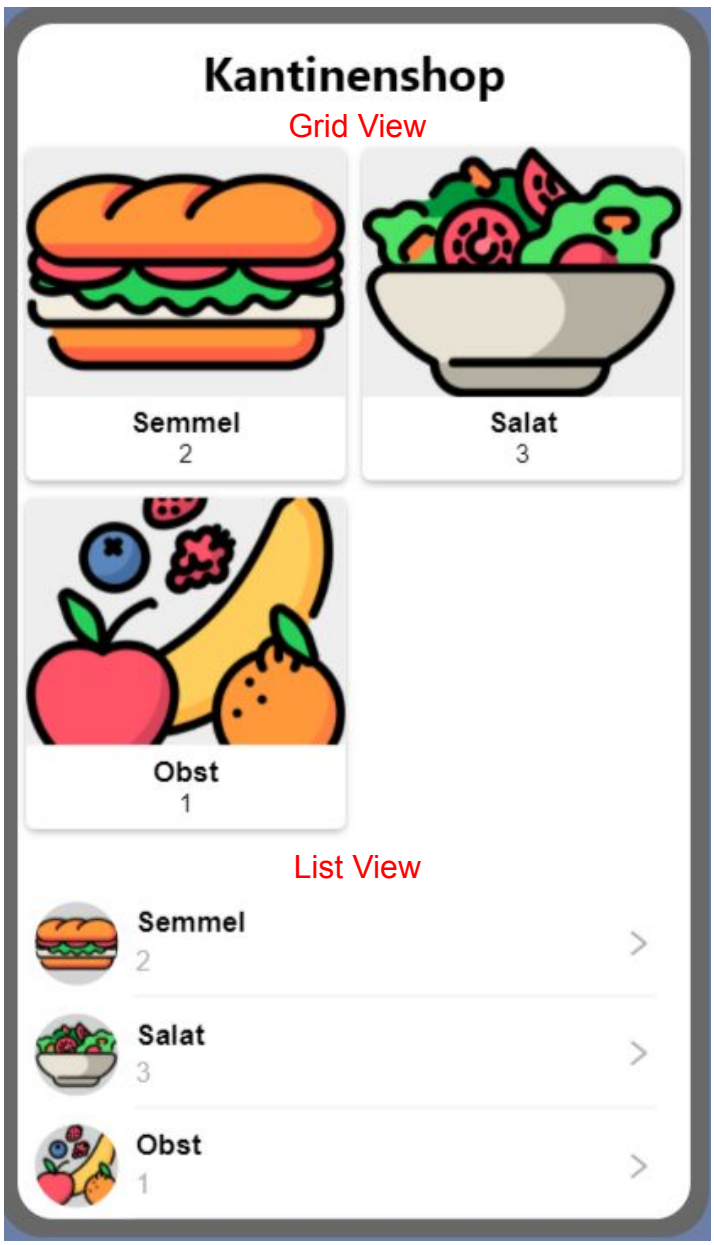
Data Viewer

Data Viewer List & Grid

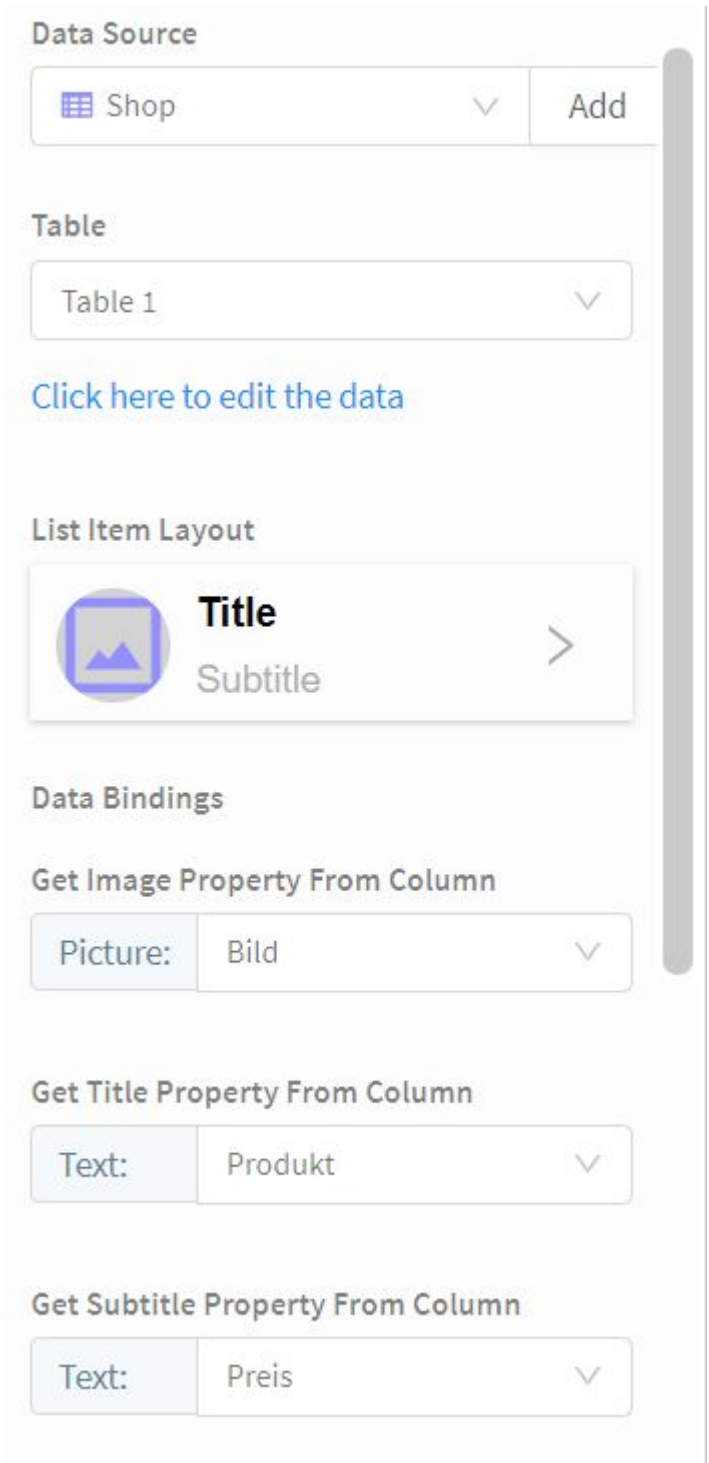


Mit der Komponente “Data Viewer List”, werden die Daten untereinander in einer Liste angezeigt, in “Data Viewer Grid” als Kacheln (siehe Beispiel unten links).

Wähle die Komponente aus, die du verwenden möchtest. In den Eigenschaften wählst du deine Datenquelle (Data Source), ein bevorzugtes Layout, und verbinde die entsprechenden Elemente deiner Daten mit den Properties, die angezeigt werden sollen.



Komponenten Eigenschaften

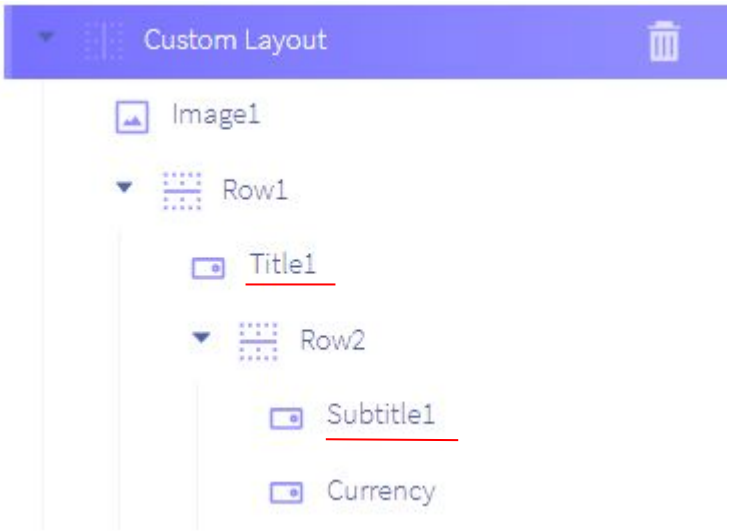


ADVANCED

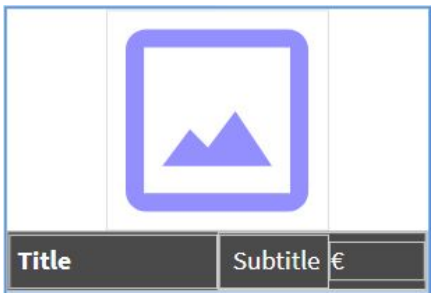
Data Viewer

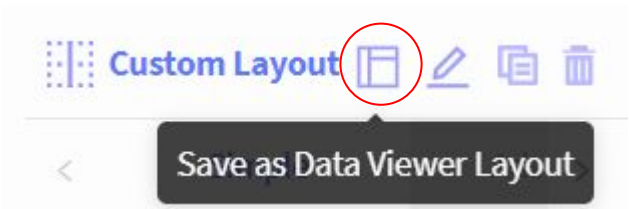
Individuelle Layouts erstellen für Data Viewer

Man kann **eigene Layouts** designen und dann abspeichern, damit sie in der Layout-Auswahl für deine Elemente aufscheinen. Dafür musst du aus Rows und Colums, sowie den Komponenten für die Properties (Labels für Text, Images für Bilder etc.) als einen Prototyp zusammenbauen und anpassen (➡ Siehe Layout-Lernkarten für Hilfe).



Hier siehst du die Komponenten in einer Column gesammelt, die für ein neues Grid Layout gedacht sind. **Benenne die Komponenten**, die für die Properties zuständig sind, am besten entsprechend um.

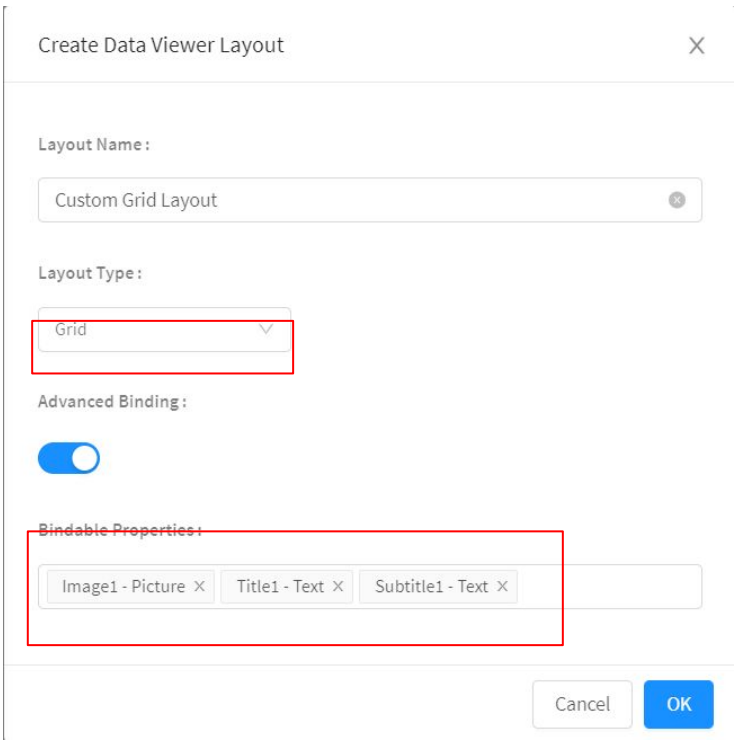




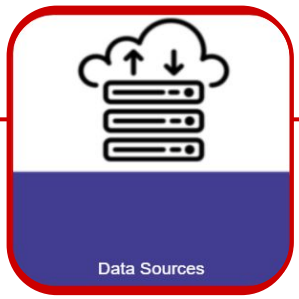
Klicke hier um dein fertiges Layout für den Data Viewer zu **speichern.**

Gib deinem Layout einen Namen, lege fest, ob es für **Grid oder List** konzipiert ist und **Verbinde** deine visuellen **Komponenten mit den Datenquellen**, die zur Verfügung stehen sollen. Hier wurde Picture, Title und Subtitle gewählt, du kannst aber beliebig viele Komponenten wählen.

Ergebnis



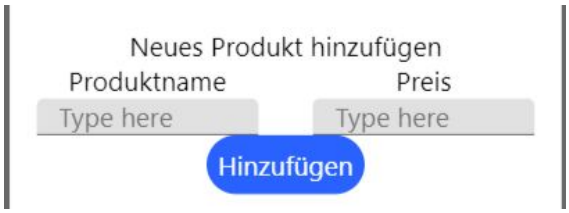
ADVANCED

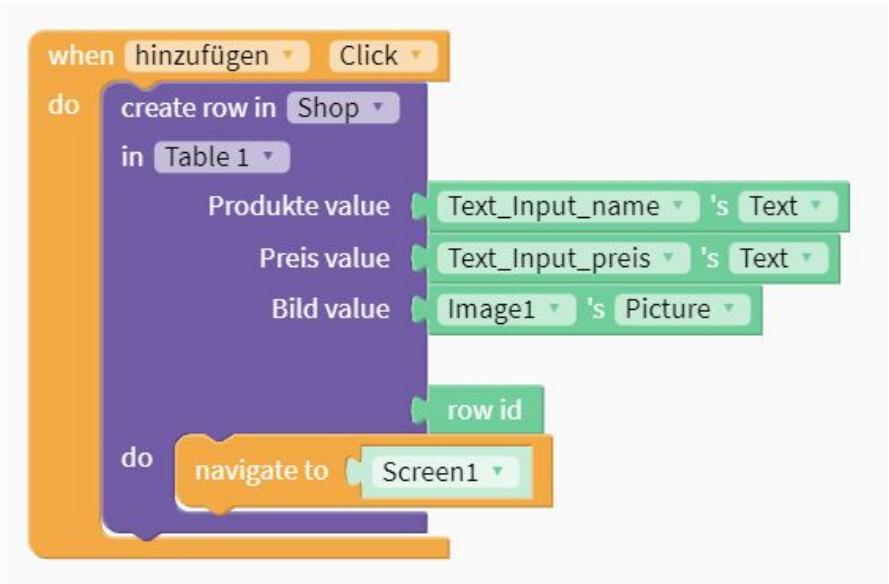


Tabellenreihen hinzufügen und entfernen

Mit den Blöcken von **Data Sources** kann man in seiner Tabelle neue Reihen hinzufügen. Zuvor muss man eine Eingabemöglichkeit geben für die neuen Inhalte, z.B. mit **Text-Input**-Feldern.

Hinzufügen mithilfe von Text-Inputs:



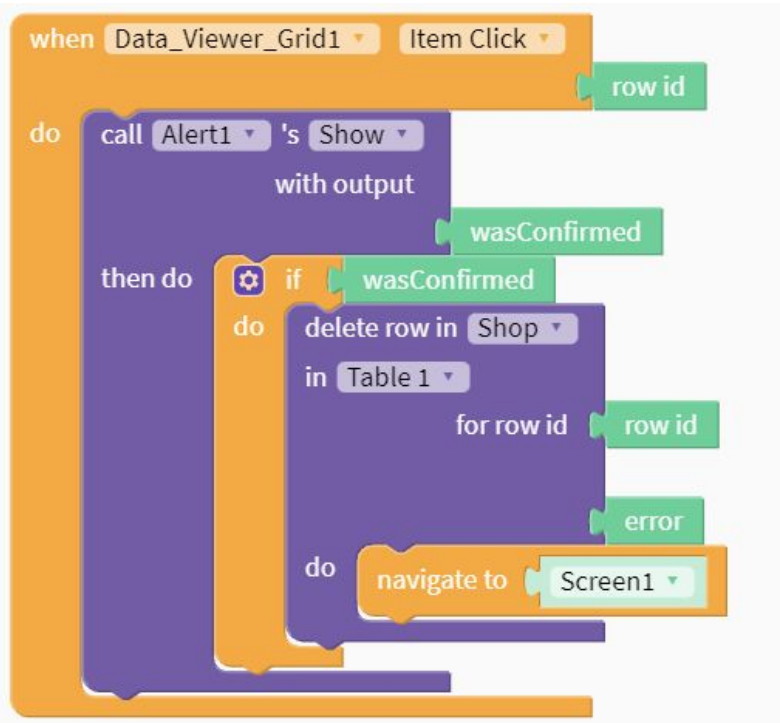


Reihe hinzufügen

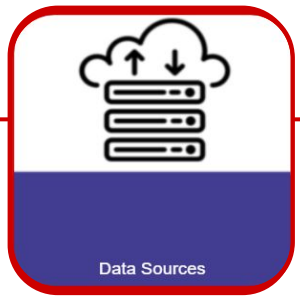
Name und Preis werden in die nächste freie Reihe in der Tabelle eingefügt (das Bild bleibt derzeit leer). Im Anschluss wird der Screen neu geöffnet, um die Änderungen sichtbar zu machen.

Reihe entfernen

Mit diesen Blöcken wird, wenn ein Item im Data-Viewer angeklickt wird, dieses aus der Tabelle gelöscht. Bevor der Eintrag entfernt wird, wird mit einer "**Alert**"-Komponente abgefragt, ob das Element wirklich gelöscht werden soll.



ADVANCED



List Viewer

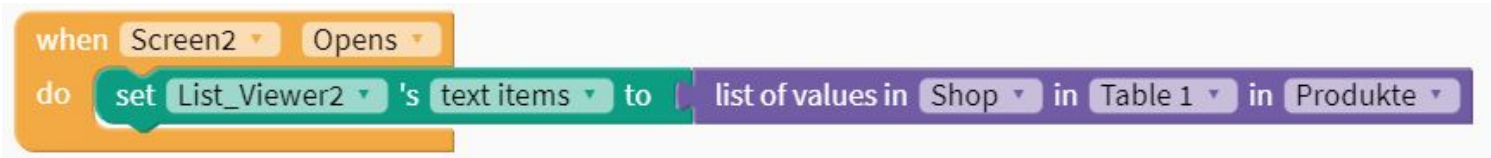
Mit dem **List Viewer** kann man eine Liste mit klickbaren Items anzeigen. **Screenshot, Produktliste, Screenshots beziehen sich auf eine App.**

Anlegen einer Liste mit fixen Werten

Auslesen eines geklickten Wertes in ein Label

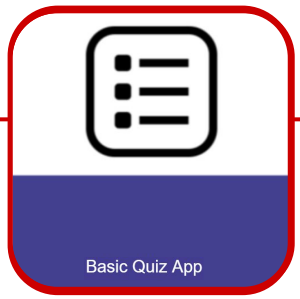
Beispiel: Dynamische Liste (Data Sources)

Zeige die Einträge einer Spalte deiner Tabelle im List Viewer an:



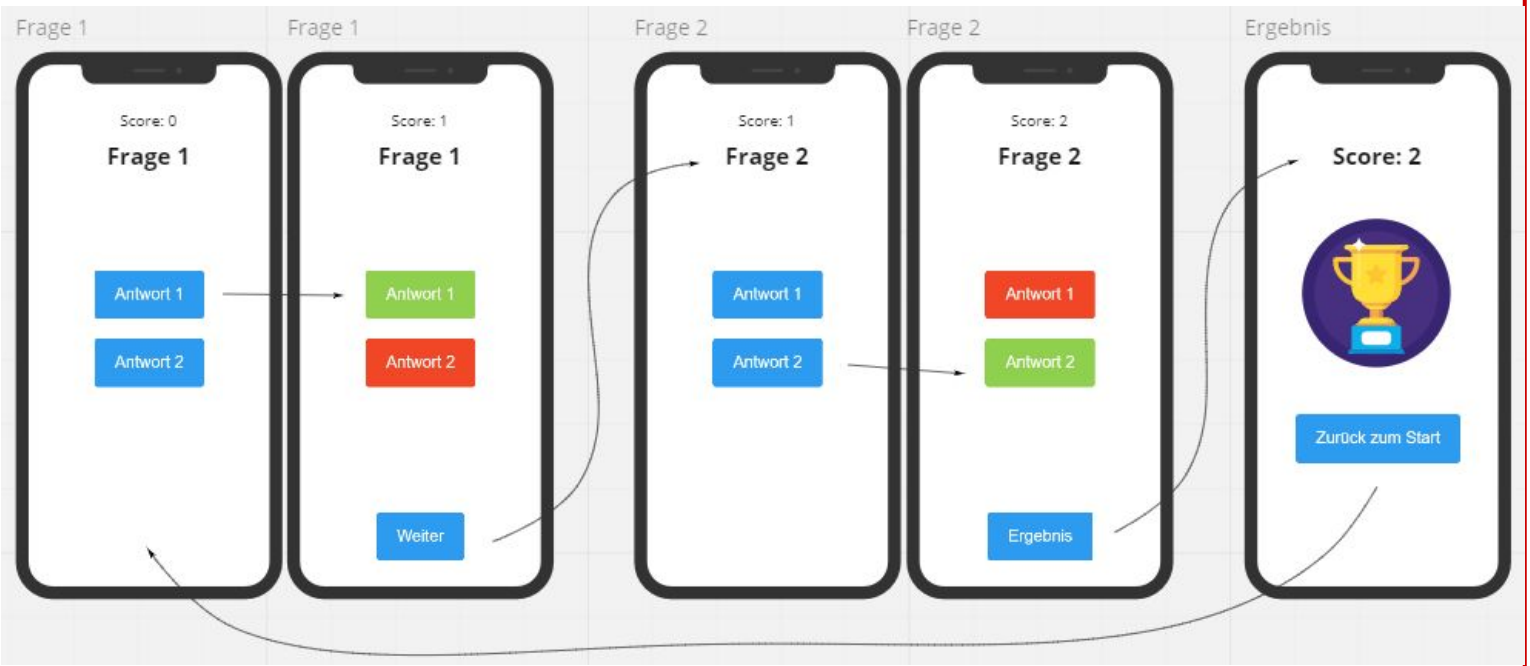
	Produkte
1	Semmel
2	Salat
3	Obst
4	

ADVANCED



Erstellen einer einfachen Quiz-App

Hier findest du Anleitungen, wie du in Thunkable eine einfache Basis für eine Quiz-App erstellst, die du beliebig designen und erweitern kannst.
Das Wireframe-Mockup für die App sieht so aus:



Für die App verwenden wir die Komponenten:

- Columns
- Labels
- Buttons
- Images

Außerdem verwenden wir bei den Blöcken, neben den Komponenten-Blöcken, die meisten aus folgenden Kategorien:

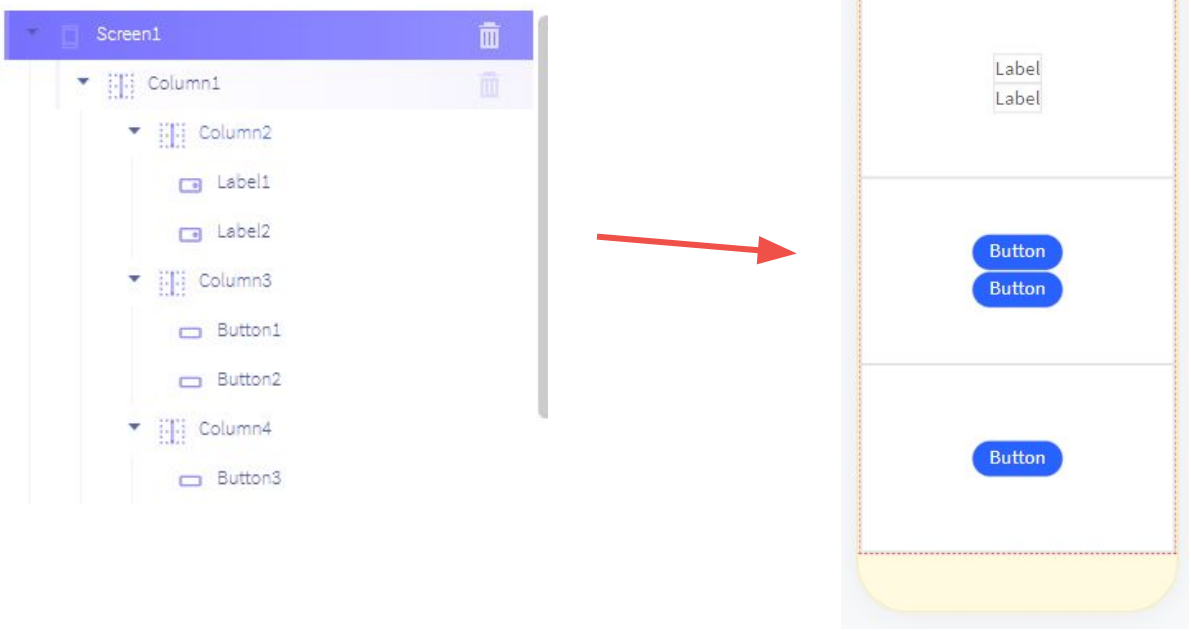
- Variables
- Functions
- Any Components

Solltest du bei dieser App Schwierigkeiten haben, findest du bei anderen Cheatsheets mehr Informationen zu den einzelnen Komponenten und auch Hilfe für das erweiterte Design deiner Quiz-App.

ADVANCED

Basic Quiz App

Komponenten für die erste Quizfrage anordnen

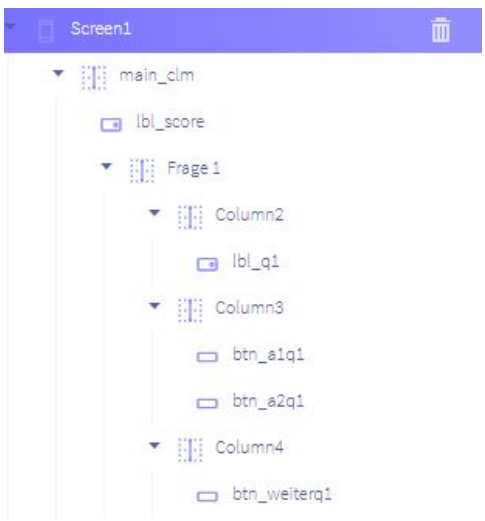


Für die Fragen brauchen wir Labels und Buttons, die in Columns angeordnet sind, damit wir ein ansprechendes, responsives Design bekommen.

Wichtig: Bringe das gesamte Design der Frage in eine übergeordnete Column (siehe unten “Frage 1”, das erleichtert dir später die Erweiterung des Quizzes.

Passe das Design der Komponenten an (siehe Cheatsheet Layouting) und gib die entsprechenden Texte ein.

Tipp: Benenne deine Komponenten um, damit dir das Programmieren im Anschluss leichter fällt. Wenn du alle Buttons der Frage 1 mit der Zahl 1 am Ende benennst (wie im Bild), kann Thunkable bei weiteren Fragen automatisch die Zahl richtig fortführen-



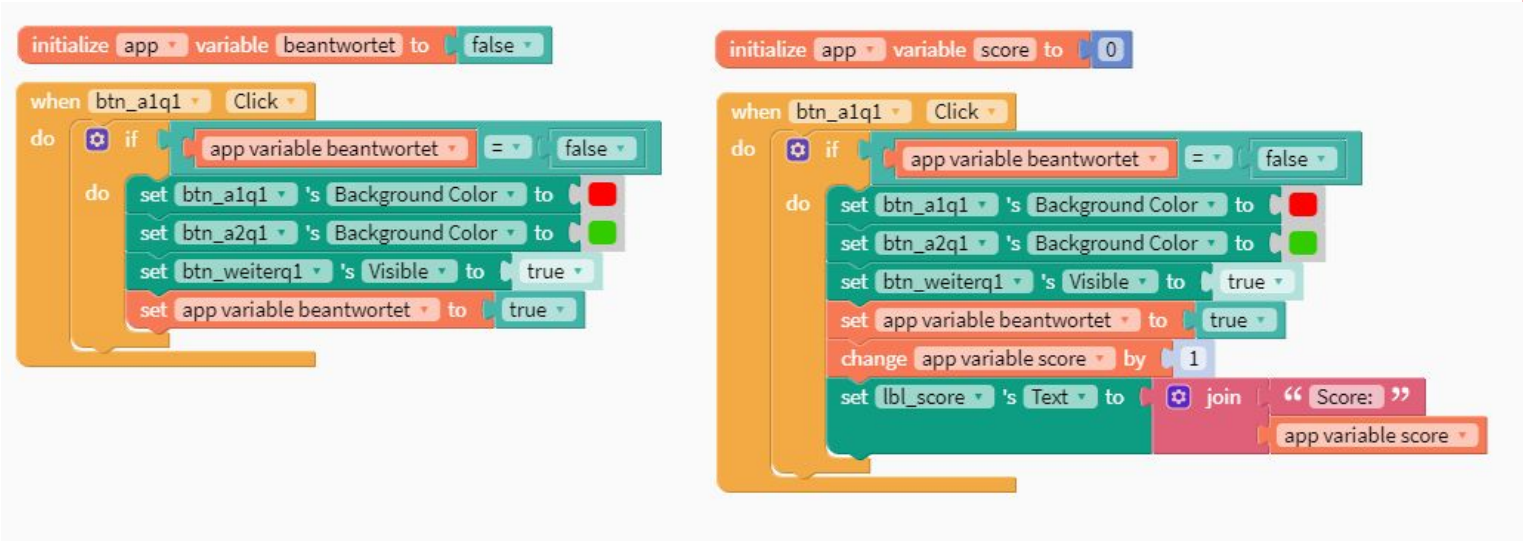
ADVANCED

Mit Funktionen einfach programmieren

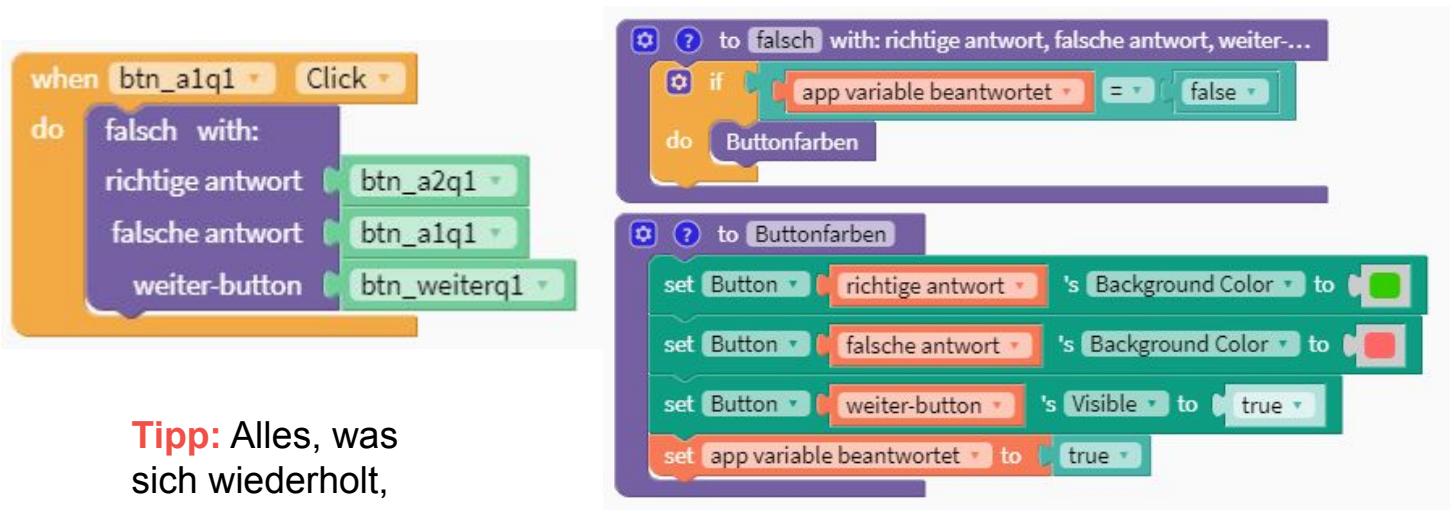
Basic Quiz App

Ablaufschritte der App

Was bei dieser App passieren soll: Der User drückt auf einen der beiden Antwort-Buttons. Danach verändern diese ihre Farben: die richtige Antwort wird grün, die falsche rot. Bei einer richtigen Antwort erhält man einen Punkt. Nachdem man einen Button gedrückt hat, kann man ihn nicht wieder drücken um sein Ergebnis zu verändern, und es erscheint ein neuer Button, der den User zur nächsten Frage leitet.



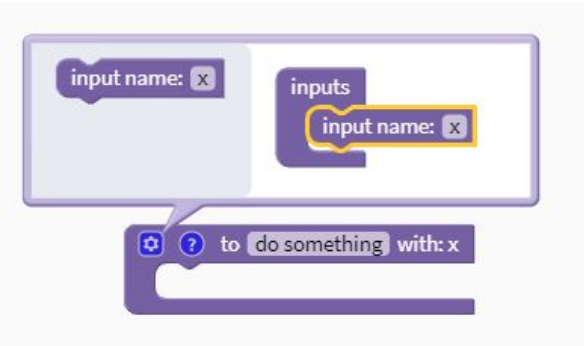
Dieser Prozess kann sich beliebig oft wiederholen. Damit wir nicht jedes mal die gleichen Blöcke programmieren müssen, verwenden wir Funktionen, bei denen nur die entsprechenden Komponenten als Input angegeben werden müssen.

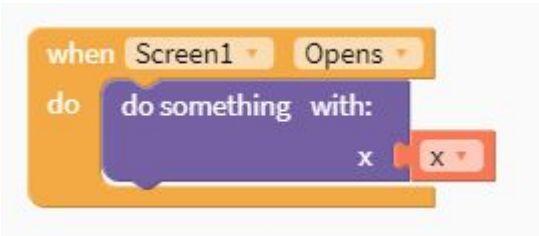


Tipp: Alles, was sich wiederholt, kann in eine Funktion gegeben werden

Funktionen mit Inputs

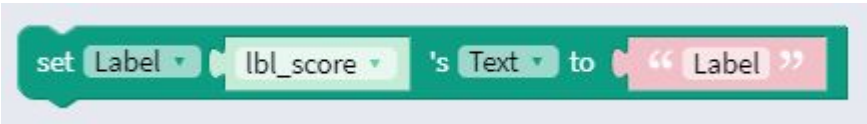
Funktionen können Inputs erhalten, so kannst du sehr einfach mit nur einer Funktion auf unterschiedliche Komponenten zugreifen, indem du diese als Input angibst:

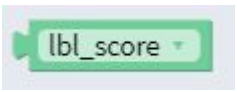




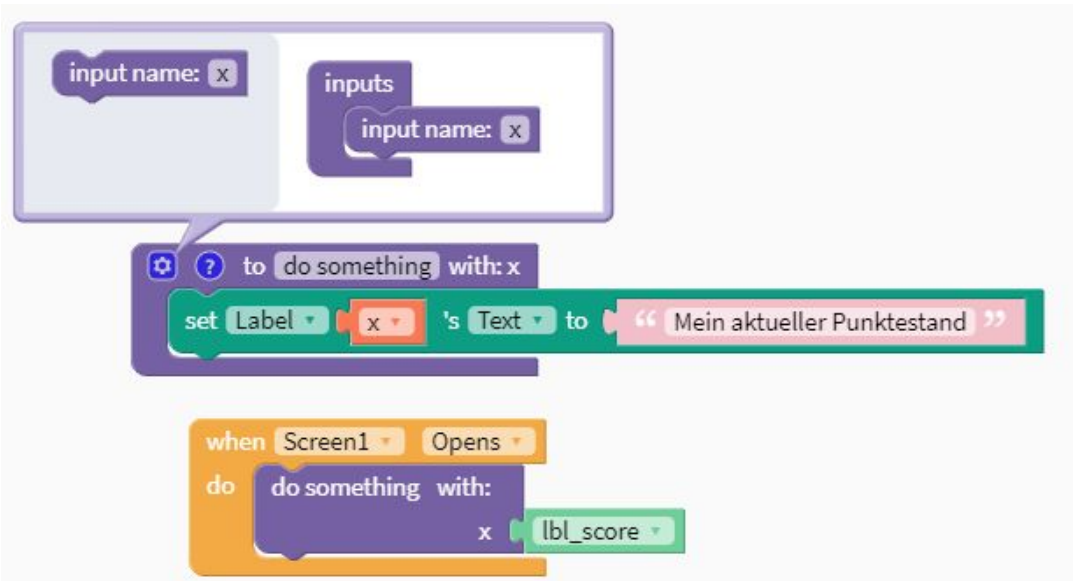
Any Components -Blöcke

Um diese unterschiedlichen Komponenten innerhalb der Funktion zu verändern, kannst du das mit diesen Blöcken machen:

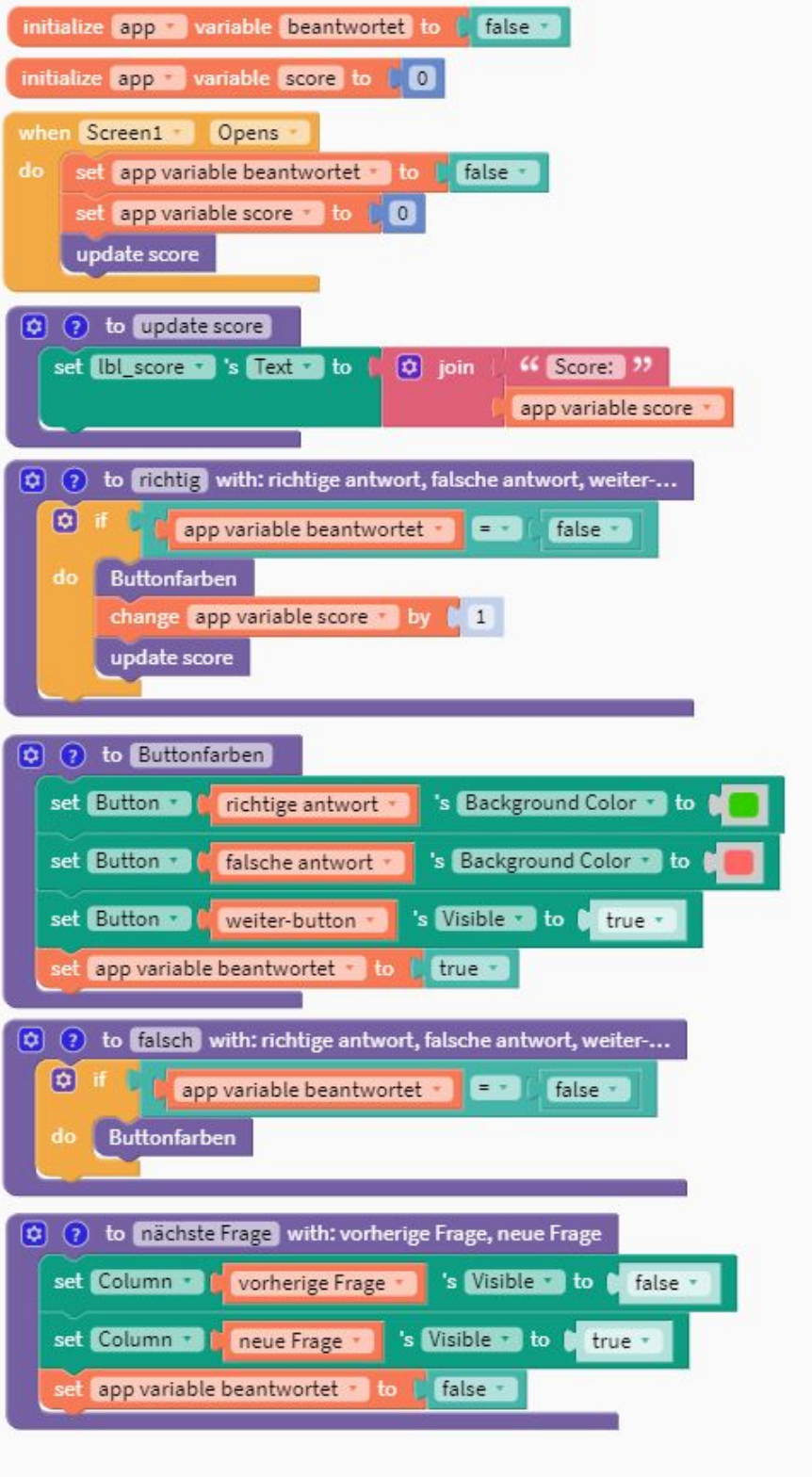


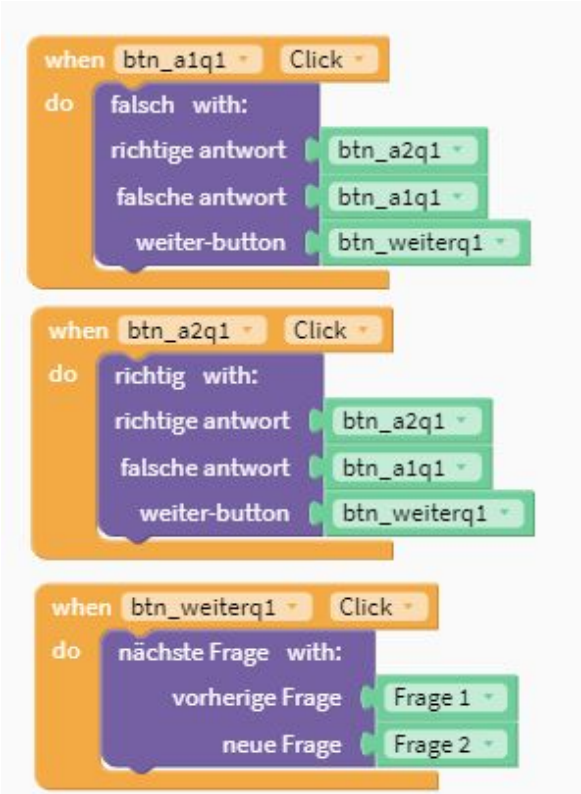


Du kannst den passenden Komponenten-Typ und dessen Eigenschaften anpassen



Blöcke für Frage 1



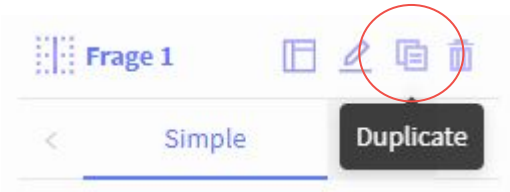


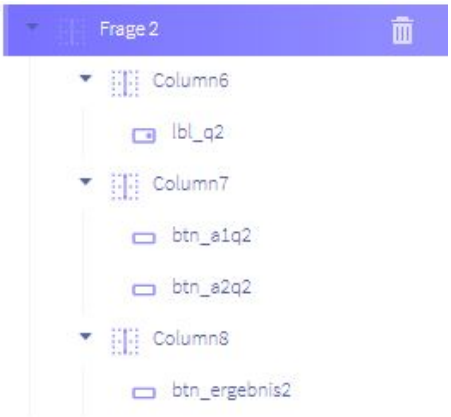
Für Frage 2 müssen am Ende nur die Blöcke für die Buttons hinzugefügt werden, die Grundfunktionen bleiben die selben.

Bevor das jedoch passieren kann, müssen wir die Komponenten für die zweite Frage hinzufügen.

Komponenten duplizieren

Wir kehren in den Design-Tab zurück und machen dort eine Kopie von Frage 1:





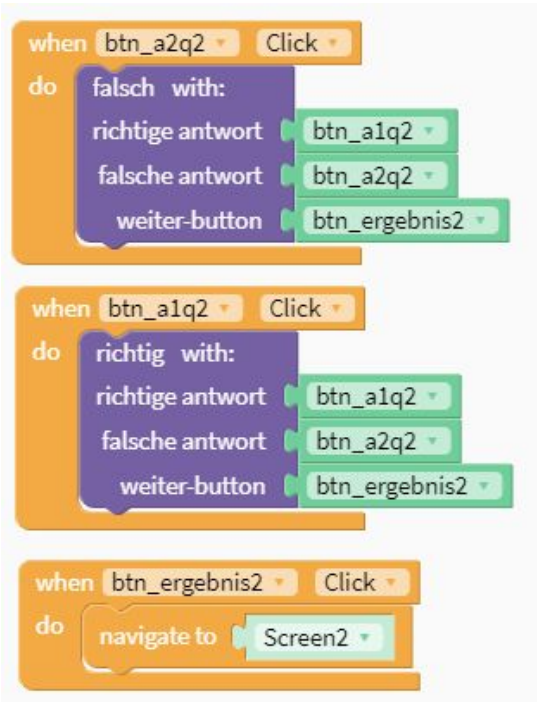
Wenn zuvor alle Buttons richtig benannt wurden, sodass am Ende 1 stand, erhalten sie beim duplizieren automatisch nun den selben Namen mit 2 am Ende.

Die Column mit der Frage 2 muss zu Beginn des Quizzes auf Visible = false gestellt werden

Nun müssen nur noch die Blöcke für die Buttons von Frage 2 gesetzt werden, und die Inputs der Funktionen mit den richtigen Komponenten gefüllt werden.

Dieser Prozess kann nun für beliebig viele weitere Fragen wiederholt werden.

Am Ende des Quizzes ersetze den Weiter-Button mit einem Ergebnis-Button und navigiere zu einem neuen Screen, der das Endergebnis abbildet und dich zum Start zurückschicken kann.



APP DESIGN

Apps testen

Um deine App zu testen, musst du auf einem Smartphone oder Tablet die App “**Thunkable Live**” (verfügbar im **Google Play Store** und **Apple AppStore**) installieren.

Wichtig: In der App musst du dich mit dem **selben Google Account einloggen**, wie auf www.thunkable.com

Nun kannst du in Thunkable (im Browser) rechts oben auf den Button “**Live Test**” klicken:

Jetzt kannst du die App auf deinem Smartphone testen! (**Tipp:** Sollte das nicht gleich funktionieren, hilft es manchmal, Thunkable auf dem Handy über den App Switcher zu **killen** und **neu zu starten**.)

Apps teilen

Erstelle zum Schluss einen **Link zu deiner App**, indem du auf “**Share**” und dann auf “**Generate Link**” klickst. Diesen Link kannst du kopieren und an deine Freunde versenden. Jede/r der/die einen Thunkable-Account hat und eingeloggt ist, kann diesen Link - und damit dein Projekt - öffnen.

Wie geht's weiter?

Beim Erstellen von Apps sind deiner Fantasie keine Grenzen gesetzt! Allerdings braucht es manchmal Zeit und Geduld bis man herausgefunden hat, wie sich die eigenen Ideen umsetzen lassen.

Deshalb haben wir eine kleine Auswahl an Links für dich zusammengestellt, die dir bei deinen Apps weiterhelfen können:

- **Thunkable Docs**
  https://docs.thunkable.com/
  Hier findest du detaillierte Erklärungen zu allen Thunkable **Komponenten** (z.B. Button, Label, Text Input…) und **Programmierblöcken**. Es lohnt sich, sich diese Seite genauer anzusehen, insbesondere, wenn du verstehen möchtest, wofür eine bestimmt Komponente oder ein bestimmter Programmierblock gut ist.

  - **Thunkable Spiele**
    https://docs.thunkable.com/gaming
    In den Docs findest du auch Spiele, die mit Thunkable programmiert wurden. Du kannst dir anschauen, wie sie programmiert wurden und sie sogar remixen, um ein eigenes Spiel zu erstellen.

- **Thunkable Community**
  https://community.thunkable.com/
  Bei Problemen zu allen möglichen Sachen kann dir die Thunkable Community weiterhelfen.

  - **Tutorials auf der Thunkable Community Website**
    https://community.thunkable.com/t/thunkable-x-beginner-tutorials/40418
    Auf der Website der Thunkable Community gibt es Tutorials für Einsteiger/innen, erstelle z.B. einen Generator für Zufallsantworten, nutze Text-to-Speech, erstelle einen Übersetzer oder erstelle eine Schnitzeljagd-App.

- **YouTube-Kanal: Thunkable X**
  https://www.youtube.com/channel/UCTVZRyybOCDBL2zLXSeQVsw/videos
  Der offizielle YouTube-Kanal von Thunkable. Hier gibt es Anleitungen, zu allem, was das Herz begehrt. Beispiele:
  - Currency Converter with API
    https://www.youtube.com/watch?v=bNoz9hl_w54
  - How to use the Thunkable Web API: Weather App
    https://www.youtube.com/watch?v=V-OO0wEYcrs
  - How to use Thunkable Web API Component: QR Code Scanner App
    https://www.youtube.com/watch?v=MfMGA6S5YvE
  - und viele mehr...

- **YouTube-Kanal: Thunkable X Tutorials**
  https://www.youtube.com/channel/UCHDDjy-6nbgwdrJpSZIfCOA
  Hier findest du zahlreiche Video-Tutorials. Beispiele:
  - 50 Thunkable X Tips and Tricks
    https://www.youtube.com/watch?v=GsF8XtnAHUs
  - How to Take a Photo in Thunkable X with the Camera Component
    https://www.youtube.com/watch?v=rik-A44itK8
  - und viele mehr…