



Kunst Interaktiv



– Die intelligente Museumsapp –

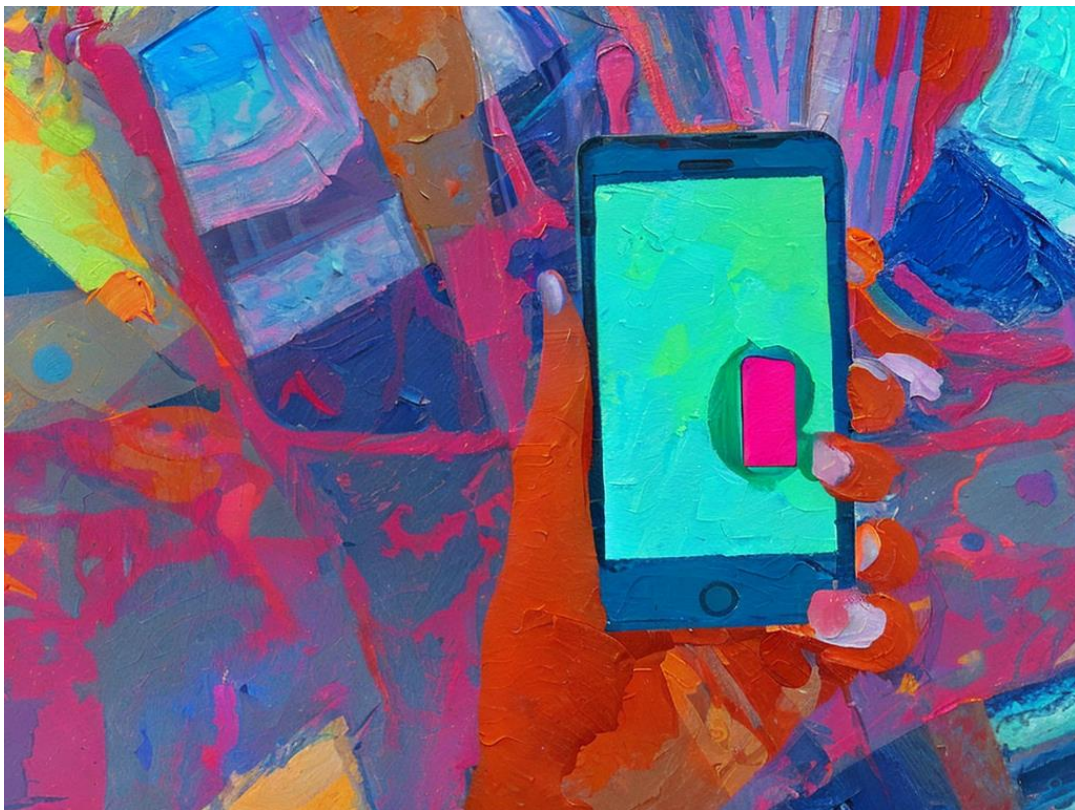


Bild KI-generiert mit [Disco-Diffusion](#) ([Disco Diffusion License](#))

BILDERKENNUNG

Sicher warst du schon einmal in einem Museum und hast dir einen **Audioguide** ausgeliehen. Damit erhält man beim Betreten eines entsprechenden Raumes oder beim Drücken einer Taste **Informationen**. Wir wollen eine App mit ähnlichem Prinzip erstellen. Diese funktioniert jedoch nicht auf Tastendruck, sondern kann auf **Bildern** das gesuchte Exponat **selbstständig** erkennen.

DAS LERNST DU

In diesem Lernmodul erfährst du, wie dein eigenes **Machine Learning Modell** in eine **App** eingebettet werden kann und wie du sie mit dem Smartphone oder Tablet benutzt.

AUFGABE

Zum Einrichten der Programmierumgebung **öffne** zuerst <http://code.appinventor.mit.edu/login/> in deinem Browser und **melde** dich **ohne Google-Account** an (siehe Bild).

Welcome to MIT App Inventor!

[Continue Without An Account](#)

or

Your Revisit Code:

[Enter with Revisit Code](#)

HINWEIS

Wenn du auch zuhause auf dein Projekt zugreifen oder damit weiterarbeiten möchtest, **notiere** dir **den Code**, der dir nach dem Einloggen in einem grünen Feld angezeigt wird. Klicke anschließend auf *Continue*.

Welcome to App Inventor!

You are logged in without an account.
You can use the code below to re-enter MIT App Inventor and work on your projects again.

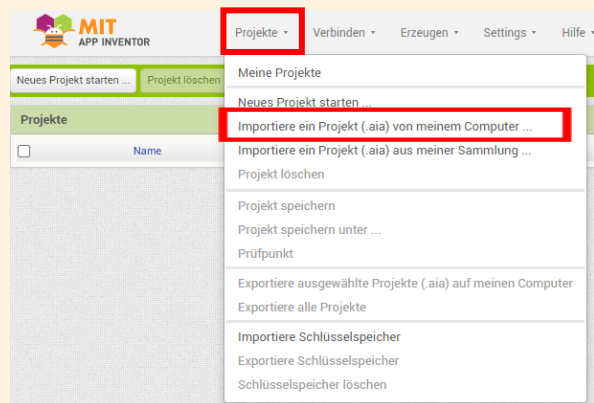
Your Code is: CHEN-GIRD-JAKE-FRY

[Continue](#)



AUFGABE

- Ändere bei Bedarf die **Sprache** oben rechts auf **Deutsch**.
- Importiere dir das bereits vorgefertigte App-Projekt über „**Projekte**“ von deinem Computer, wie im Bild gezeigt. Deine Lehrkraft stellt dir die Datei zur Verfügung.



AUFBAU VON APPINVENTOR

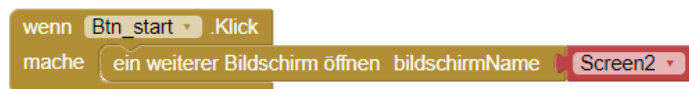
Wenn du das Projekt erfolgreich importieren konntest, geht es weiter. Nun wollen wir uns den Aufbau und die Funktion der **Programmierungsumgebung „AppInventor“** anschauen. Auf dem Startbildschirm sehen wir ein kleines Logo und eine Beschreibung, was die App tun soll.



AUFGABE

Wechselt mit den Buttons **oben rechts** **Designer** **Blöcke** in den **Blöcke**-Modus, um den bereits geschriebenen Code im Blockformat anzusehen.

Dort sehen wir, dass durch den Klick auf den Button *Start* ein anderer Bildschirm **Screen2** geöffnet wird. Weitere Hinweise erhältst du in der grünen Box auf der nächsten Seite.



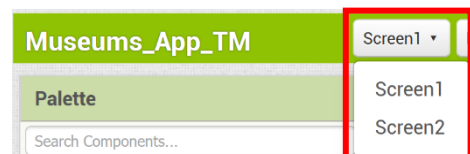
HINWEIS

Mit dem *Designer* wird die **App gestaltet**. Mit Hilfe des Auswahlmenüs links können verschiedene Texte, Bilder usw. durch „Drag and Drop“ gestaltet werden.

Im Modus *Blöcke* kann die **Oberfläche** mithilfe der **Programmierbausteine links** durch „Drag and Drop“ beliebig **programmiert** werden.



Zwischen **Screens** kann man wechseln, indem man oben links **Screen1** anklickt.



AUFGABE

Wechsle nun wieder in den *Design*-Modus mit den Buttons **oben rechts**



Schauen wir uns **Screen 2** an, sehen wir das Herzstück der App. Die Funktion der App ist es, mittels des Buttons „**Klassifizieren**“ ein Bild aufzunehmen. Anschließend wird das Ergebnis ausgegeben, welches mit der **höchsten Wahrscheinlichkeit** erkannt wurde. Dann machen wir uns mal an die Umsetzung!

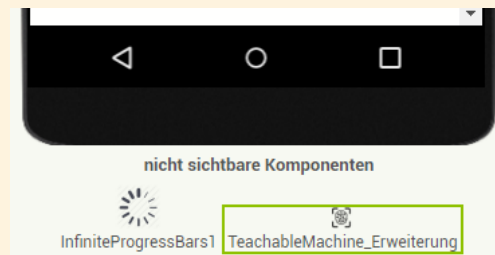
TEACHABLE MACHINE EINBINDEN

Fangen wir damit an, unser Teachable Machine-Modell (TM) einzubinden, welches wir bereits erstellt und trainiert haben.

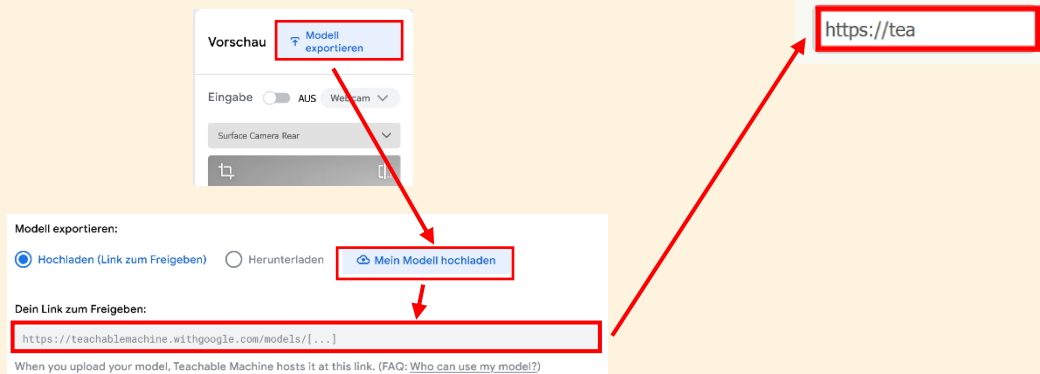


AUFGABE 4

- a) Wähle im *Designer* von **Screen2** die *TeachableMachine_Erweiterung* unter **nicht sichtbare Komponenten** aus.



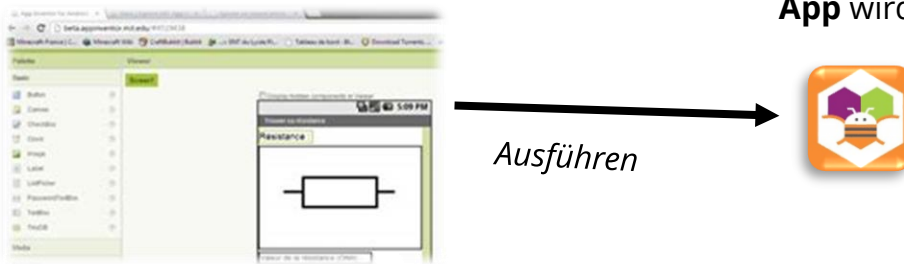
- b) Jetzt müssen wir unser TM-Modell mit der Erweiterung verbinden. Trage bei **Eigenschaften** (rechts) unter **URLModel** den Link zum Teachable Machine-Modell ein.




HINWEIS

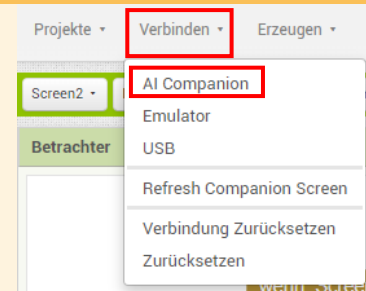
Unterhalb des **Betrachters** im *Designer* sind die **nicht sichtbaren Komponenten** des jeweiligen Screens zu sehen. Beim **Anklicken** öffnet sich **rechts** ein **Fenster**, in dem **Eigenschaften** geändert werden können.

Um die App auf Tablet oder Smartphone zu testen, benutzen wir die **MIT AI2 Companion-App**. Auf dem Tablet wird der Code „zum Leben erweckt“ und die **App wird sicht- und nutzbar**.



AUFGABE 5

- Wähle oben links unter **Verbinden** → **AI Companion** aus.
- Startet auf deinem Tablet oder Smartphone die *AI Companion App*  und **scanne** den **QR-Code**.



Wenn alles funktioniert hat, solltest du nun ein Bild von einem Exponat machen können und anschließend den Namen des Exponates angezeigt bekommen.

HINWEIS

Wenn die App noch nicht zuverlässig die Exponate erkennt, musst du dein TM-Modell noch einmal **verbessern**. Wechsle dazu zurück zu Teachable Machine und verbessere dein Modell mittels weiterer Trainingsdaten.

ACHTUNG

Vergiss nicht, das Modell wieder neu zu **exportieren**!

BESCHREIBUNG EINFÜGEN

Die App funktioniert zwar, aber wir bekommen noch keine Beschreibungen für die Exponate angezeigt. Das wollen wir nun ändern. Als erstes brauchen wir dafür die **Beschreibungstexte**.



AUFGABE 6

Denk dir eine kurze Beschreibung zu jedem Exponat aus und notiere sie auf einem **Blatt oder neuen Dokument**. Das können zum Beispiel interessante oder lustige Fakten sein.

Um die Texte in den Code einzupflegen, müssen wir erst einmal wissen, wie die App funktioniert.



AUFGABE 7

Wechsle auf **Screen 2** in den *Blöcke*-Modus und schau dir den Code an. Was passiert, wenn dieser Code ausgeführt wird?

Mithilfe des `aufrufen TeachableMachineImageClassifier1 .ClassifyVideoData` - Befehls wird dem Modell ein Bild übergeben. Anschließend wird dieses durch das Modell ausgewertet.

Unter dem Befehl

```
wenn TeachableMachineImageClassifier1 .GotClassification
  Ergebnis
  mache
```

kann das Ergebnis

dieser Klassifikation verarbeitet werden. Das **Ergebnis** ist als Variable nutzbar. Sie hat die Form eines **Dictionaries**.



DICTIONARIES

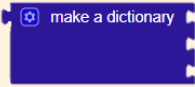
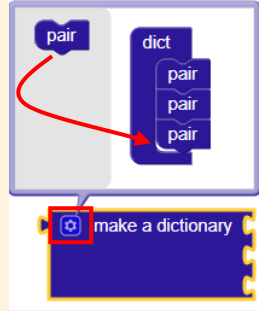
Dictionaries sind eine besondere Form der Datenstruktur. Hier werden Daten nicht wie bei einem Array an einem bestimmten Platz gespeichert, sondern als Kombination aus einem **Schlüssel** und dem zugehörigen **Wert**.

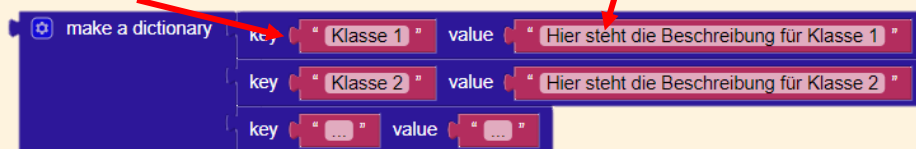
Haben wir zum Beispiel ein Modell zum Erkennen von Schere, Stein, Papier erstellt, könnte ein Dictionary mit einem erkannten Ergebnis so aussehen →

Schlüssel		Wert
Schere	→	0,08
Stein	→	0,03
Papier	→	0,89

Nun wollen wir die Beschreibungen der Objekte hinzufügen, indem wir ebenfalls ein *Dictionary* verwenden.

AUFGABE 8

- Wähle unter **Variablen** den Block `global initialisieren (Name) auf`. Klicke auf **Name** und bezeichne dein Dictionary sinnvoll (zum Beispiel *beschreibungenExponate*)
- Wähle nun unter **Dictionary** den Block und füge ihn an den Block „global initialisieren“ an .
- Wenn du auf das Zahnrad klickst, kannst du für jedes Exponat einen Eintrag im Dictionary anlegen, indem du jeweils einen `pair`-Block für jedes Exponat in die Liste rechts daneben ziehst. 
- Nimm für jedes Exponat aus **Dictionary** den `key` - `value` - Block und füge ihn am Dictionary hinzu.
- In die freien Stellen fügst du aus **Text** die `key` - `value` - Blöcke ein und gibst unter *key* die Klassennamen und unter *value* die Beschreibungen ein.



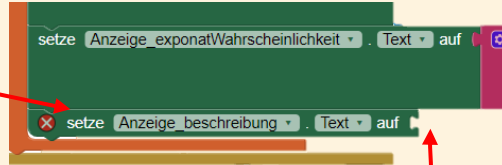
Super, die **Beschreibungen** sind jetzt in der App! Jetzt müssen wir nur noch dafür sorgen, dass sie **angezeigt** werden, wenn das entsprechende Exponat erkannt wurde. Das passiert nach der **Erkennung**. Wir arbeiten ab jetzt in der

wenn `TeachableMachineImageClassifier1` `.GotClassification`
`Ergebnis`
`mach` -Methode.

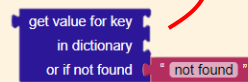


AUFGABE 9

- a) Scrolle links im *Blöcke-Abschnitt* runter und wähle unter *Screen 2* das Element **Anzeige_beschreibung** aus. Wähle nun den Block **setze Anzeige_beschreibung . Text auf** aus und setze ihn unter den anderen **grünen**.



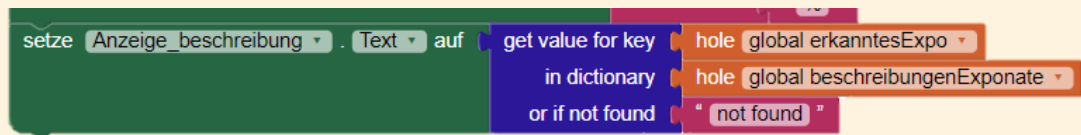
- b) Füge nun aus **Dictionaries** den **get value for key in dictionary or if not found** - Block an.



- c) Für die obere freie Stelle benötigen wir den Schlüssel, also den Namen des erkannten Exponats. Diesen bekommen wir mit dem Block

hole global erkanntesExpo unter **Variablen**.

- d) Als letzten Schritt müssen wir nur noch das Dictionary angeben, in dem die Beschreibung liegen. Also genau das, welches du in Aufgabe 8 a) benannt hast.



Super, du hast es geschafft! Nun kannst du deine fertige App wieder mit der **AI Companion App** verbinden und ausprobieren.