

Einstieg in den App Inventor

In diesem Tutorial erfährst du, wie du Apps für Smartphones mit Android-Betriebssystem programmieren kannst. Du nutzt dafür das Programm **App Inventor**, das nur online im Browser (z. B. Firefox oder Chrome, nicht Internet Explorer) aufgerufen werden kann.

1 Anmeldung und Erstellung eines Projekts

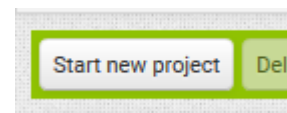
Um die Programmierung zu starten, musst du den App Inventor aufrufen. Dafür benötigst du ein Benutzerkonto (Account) bei Google, beachte hier bitte die Hinweise deiner Lehrkraft. Sie wird dir sagen, mit welchem Benutzerkonto du dich anmelden kannst.

1. Rufe im Browser folgende Internetseite auf: <http://appinventor.mit.edu>

2. Klicke auf „Create apps!“ (oben rechts auf der Webseite). Danach musst du dich mit dem Google-Account anmelden.

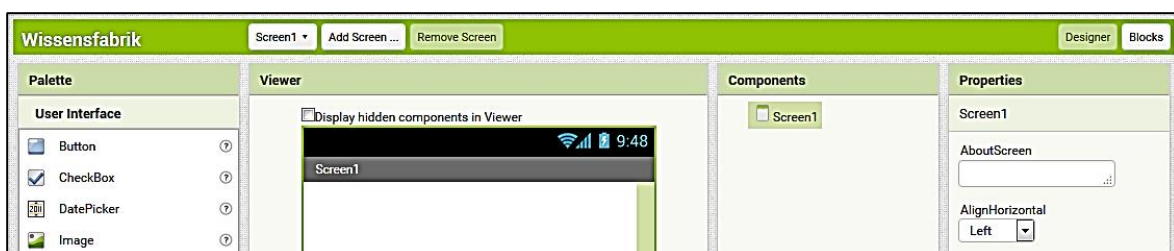


3. Sobald du eingeloggt bist, kannst du auf „Start new project“ klicken, um ein neues Projekt anzulegen. Gib nun einen Namen für das Projekt an, zum Beispiel „Meine erste App“. Danach solltest du die Entwicklungsumgebung vom App Inventor sehen.



2 Entwicklungsumgebung – Design Editor

Die Entwicklungsumgebung des App Inventor umfasst zwei Editoren, die jeweils in mehrere Bereiche aufgeteilt sind. Den ersten Editor siehst du bereits. Hierbei handelt es sich um den **Design Editor**. Im Bild unten siehst du die Aufteilung in *Palette*, *Viewer*, *Components* und *Properties*. Einen Gesamtüberblick mit Erläuterungen zum Design Editor findest du auf den letzten Seiten dieses Tutorials.



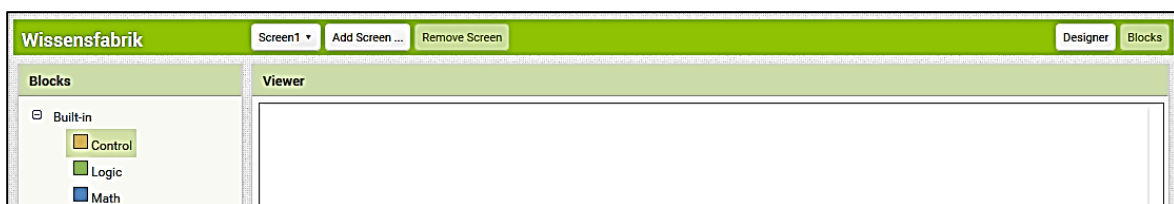
In **Palette** findest du alle Komponenten, die dir für deine App zur Verfügung stehen. Unter *User Interface* (Benutzerschnittstelle) beispielsweise *Buttons* (Knöpfe), unter *Media* die *Camera* und unter *Drawing and Animation* (Zeichnen und Bewegung) das *Canvas* (Leinwand). Die einzelnen Komponenten kannst du in den **Viewer** ziehen. Im **Viewer** (Betrachter) siehst du, wie deine App später auf dem Handy aussehen wird. Unter **Components** werden die Komponenten angezeigt, die du schon in deine App eingefügt hast. Hier kannst du Komponenten auch umbenennen und löschen. Unter **Properties** werden die Eigenschaften der Komponente angezeigt, die du gerade unter **Components** oder im **Viewer** angeklickt hast.

Hier kannst du zum Beispiel Texte von Buttons ändern oder Hintergrundfarben für ein *Canvas* auswählen.

3 Entwicklungsumgebung – Blocks Editor

Im Design Editor kannst du grundlegende Komponenten für deine Apps einbauen. Damit diese jetzt aber auch tun, was sie sollen, musst du es ihnen beibringen.

Die einzelnen Komponenten der App müssen also mit Funktionen ausgestattet werden. Dies geschieht im **Blocks Editor**, der für jede Komponente verschiedene Funktionsbausteine zur Verfügung stellt, die wie ein Puzzle zusammengebaut werden. Zwischen dem **Design** und dem **Blocks Editor** schaltest du mit den Buttons *Designer* und *Blocks* im oberen rechten Bereich der Webseite um. Sobald du zum Blocks Editor gewechselt bist, siehst du, wie im Bild unten, eine Aufteilung in *Blocks* und *Viewer*.



Unter *Blocks* findest du Kategorien wie *Control* (Kontrolle), *Logic* (Logik) und *Math* (Mathematik) mit vordefinierten Bausteinen. Die Farbkodierung sollte dir helfen, die richtigen Bausteine zu finden. Weiter unten findest du in diesem Bereich auch die Komponenten, die du vorher im Design Editor hinzugefügt hast. In den *Viewer* werden die Bausteine gezogen und miteinander verbunden. Hier wird der App gesagt, was wann zu tun ist.

4 Testen einer App

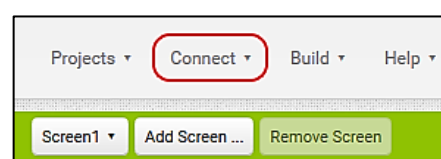
App Inventor bietet mehrere Möglichkeiten zum Testen der erstellten Apps. So kann die App über WIFI oder USB-Kabel auf das Smartphone geladen werden. Sofern kein Smartphone vorhanden ist, kann auch ein Programm (ein Emulator) zum Testen verwendet werden.

Verbindung über WLAN

1. Installiere zunächst die **MIT AI2 Companion-App** aus Google Play auf deinem Android-Gerät. Zu der App gelangst du unter anderem über folgenden Link:

<https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>

2. Melde den PC und das Android-Gerät im selben WLAN an.
3. Starte die Companion App auf deinem Android-Gerät.
4. Klicke im App Inventor auf *Connect*.
5. Wähle *AI Companion* aus, scanne den QR-Code ein, und schon kannst du die App testen.



Verbinden über USB-Kabel

1. Installiere zunächst die **MIT AI2 Companion-App** aus Google Play auf deinem Android-Gerät. Zu der App gelangst du unter anderem über folgenden Link:

<https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>

2. Installiere die **App Inventor-Software** auf dem Computer (**auf das richtige Betriebssystem achten; unter Windows 7 und aufwärts als Administrator**). Zu dem Programm gelangst du über folgenden Link (siehe Step 1):

<http://explore.appinventor.mit.edu/ai2/setup-device-usb>

3. Öffne das Programm **aiStarters** auf dem Computer.
4. Verbinde das Android-Gerät und den Computer über ein USB-Kabel.
5. Aktiviere auf dem Android-Gerät **USB-Debugging**, sofern dies nicht aktiv ist (unter Einstellungen > Entwickleroptionen zu finden).
6. Klicke im App Inventor auf **Connect**.
7. Wähle USB aus und warte kurz.
8. Nun kannst du deine App testen.



Verbindung zum Emulator

1. Installiere die **App Inventor-Software** für das entsprechende Betriebssystem auf dem Computer (**auf das richtige Betriebssystem achten, unter Windows 7 und aufwärts als Administrator**). Zu dem Programm gelangst du über folgenden Link (siehe Step 1):

<http://explore.appinventor.mit.edu/ai2/setup-device-usb>

2. Öffnen das Programm **aiStarters** auf dem Computer.
3. Klicke im App Inventor auf **Connect** und wähle **Emulator** aus.
4. Warte bis die Companion App sich im Emulator öffnet. (Dies kann etwas länger dauern! Vorher bitte nichts im Emulator machen ...)
5. Wenn eine Update-Meldung im App Inventor (Browser) erscheint, dann musst du den Emulator vorher noch updaten. Folger dazu folgender Schritt für Schritt Anweisung:
 - a. In der *Replace application* Meldung auf **OK** klicken
 - b. Klicke dann auf **Install**
 - c. Nach der Installation muss du den Emulator schliessen und wieder bei Schritt 2 beginnen
6. Warten bis der Emulator die Daten der App erhalten hat und diese anzeigt.
7. Nun kannst du deine App testen.



Hinweis: Die Verwendung des Emulators ist zwar der Vollständigkeits halber erwähnt, jedoch wird die Verwendung der **Companion-App** mittels WLAN empfohlen.

Übersicht Design Editor

Hier befinden sich alle Komponenten, die man zum Bauen einer App braucht. Am wichtigsten sind hier die Kategorien: *User Interface*, *Layout*, *Media* und *Drawing and Animation*.

Unter *Connect* findest du die Optionen zum Testen deiner App und unter *Build* kannst du die App auf deinem Smartphone fest installieren.

Wie die App später auf eurem Smartphone aussieht, das kannst du hier sehen. Um neue Elemente hinzuzufügen, kannst du diese ganz einfach aus der *Palette* dort hinziehen, wo sie später sein sollen.

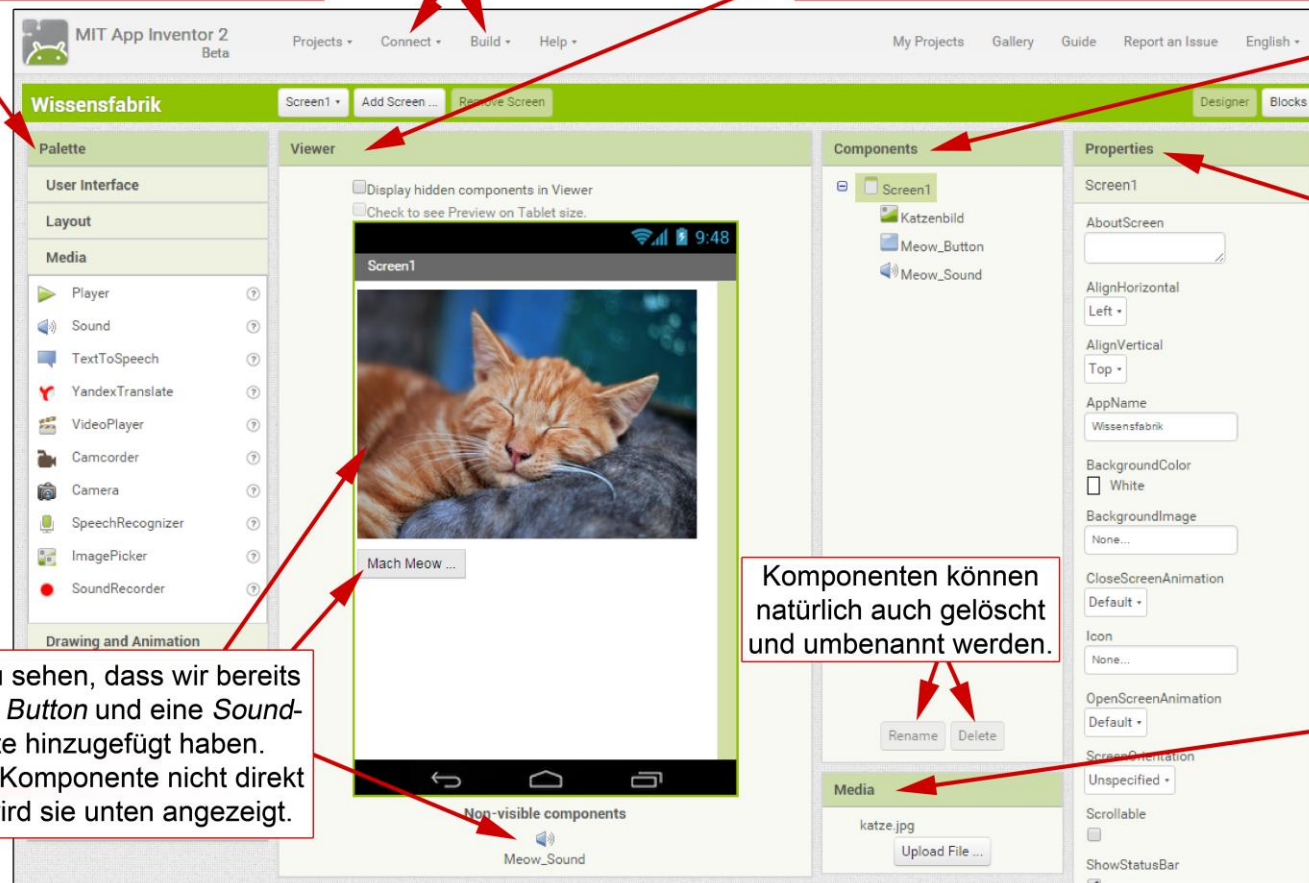
Die verwendeten Komponenten werden in diesem Bereich angezeigt

Die einzelnen Eigenschaften (wie z. B. Größe, Schrift oder Farben) der verschiedenen Komponenten können in diesem Bereich verändert werden.

Möchtest du eigene Bilder oder Musik verwenden, kannst du die Dateien hier hochladen.

Komponenten können natürlich auch gelöscht und umbenannt werden.

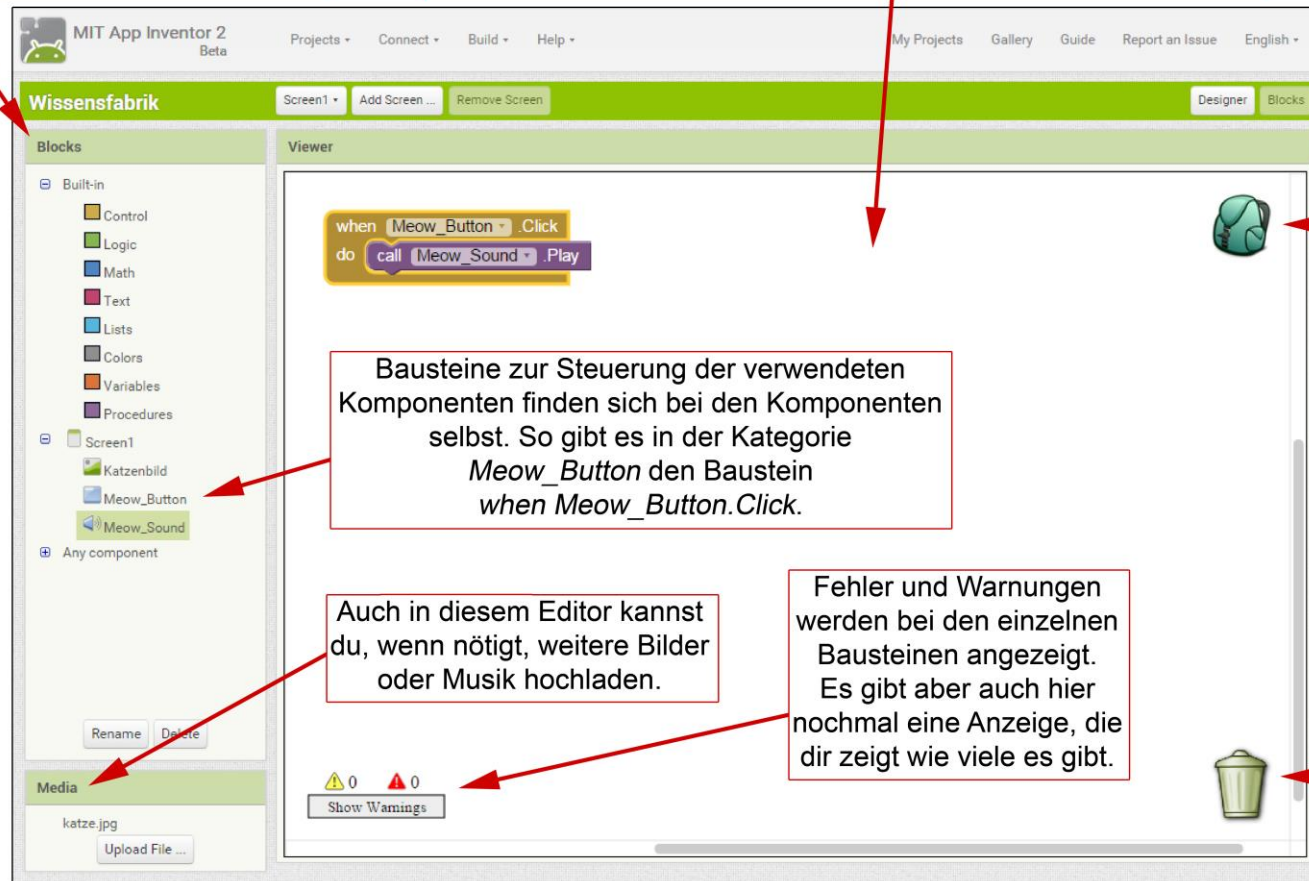
Hier kannst du sehen, dass wir bereits ein Bild, einen *Button* und eine *Sound*-Komponente hinzugefügt haben. Da die *Sound*-Komponente nicht direkt sichtbar ist, wird sie unten angezeigt.



Übersicht Blocks Editor

In diesem Bereich befinden sich alle Bausteine, die du zum Programmieren deiner App benötigst. Die Farbkodierung der Bausteine sollte dir bei der Suche nach den richtigen Bausteinen helfen.

Dies ist die Arbeitsfläche, in die du die zur Programmierung benötigten Bausteine einfach hineinziehen kannst. Mit Rechtsklick auf die Arbeitsfläche oder auf einzelne und verkettete Bausteine ergeben sich nützliche Optionen, die dir bei der Entwicklung deiner App behilflich sein können.



Mit dem Rucksack kannst du Bausteine zwischen den verschiedenen Projekten hin und her schieben.

Bausteine zur Steuerung der verwendeten Komponenten finden sich bei den Komponenten selbst. So gibt es in der Kategorie *Meow_Button* den Baustein *when Meow_Button.Click*.

Auch in diesem Editor kannst du, wenn nötig, weitere Bilder oder Musik hochladen.

Fehler und Warnungen werden bei den einzelnen Bausteinen angezeigt. Es gibt aber auch hier nochmal eine Anzeige, die dir zeigt wie viele es gibt.

Wenn du Bausteine löschen möchtest, kannst du sie ganz einfach in den Mülleimer ziehen.