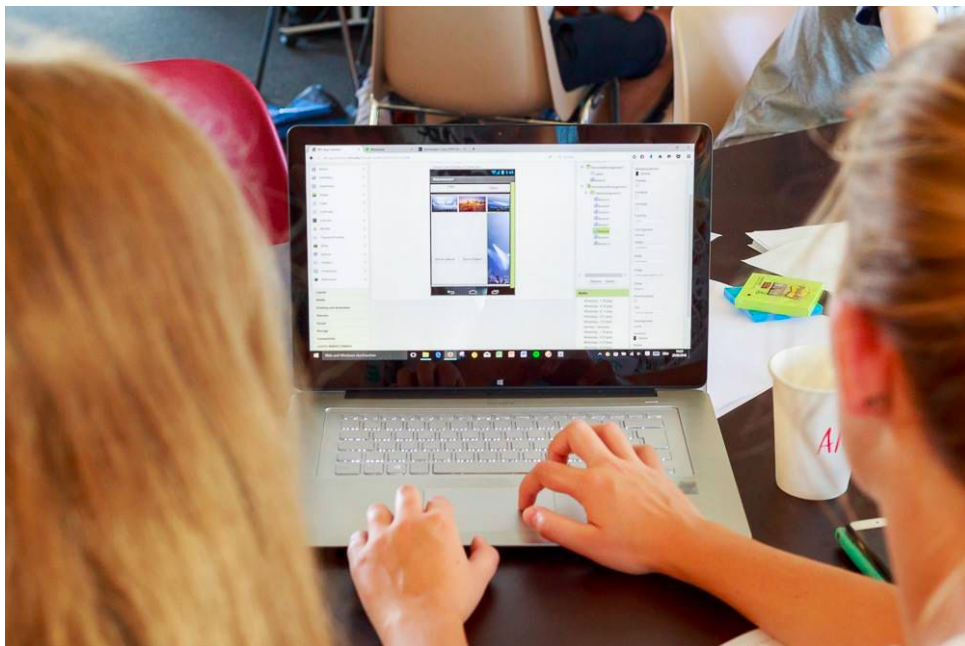


Unterrichtsmaterial "App Entwicklung"
Info für Lehrkräfte

< Konzepte & Grundlagen der Programmierung />



Schüler fürs
PROGRAMMIEREN
begeistern

< Konzepte der Informatik />

Bei der Erstellung von Apps mit den Unterrichtsmaterialien zum Thema “App Entwicklung” lernen Kinder und Jugendliche spielerisch grundlegende Konzepte der Informatik und Programmiermethoden.

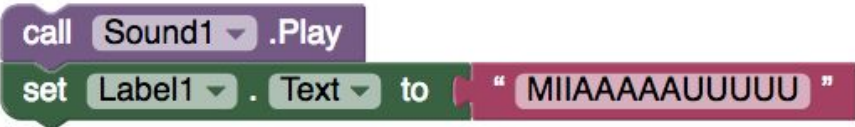

Zudem wird logisches Denken gefördert und ein fundiertes Verständnis für Programme und Anwendungen geschaffen, welche wir im Alltag täglich nutzen. Der Kurs zum Thema “App Entwicklung” verbessert auch die Problemlösungs- und Projektkompetenz.

Die Jugendlichen lernen und üben:


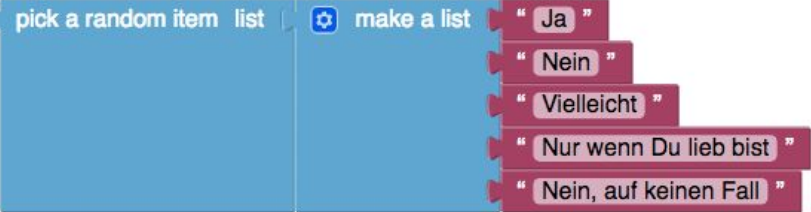
- Konzepte der Informatik und Programmiermethoden
- Besseres Verständnis für digitale Anwendungen
- Ideen von der ersten Konzeption bis zum abgeschlossenen Projekt umsetzen
- Logisches Denken
- Probleme erkennen und lösen

Einige grundlegende Konzepte der Informatik, welche in den Sitzungen dieses Kurses vermittelt werden, möchten wir nachfolgend kurz erläutern.

Basiskurs: App 1 Katze

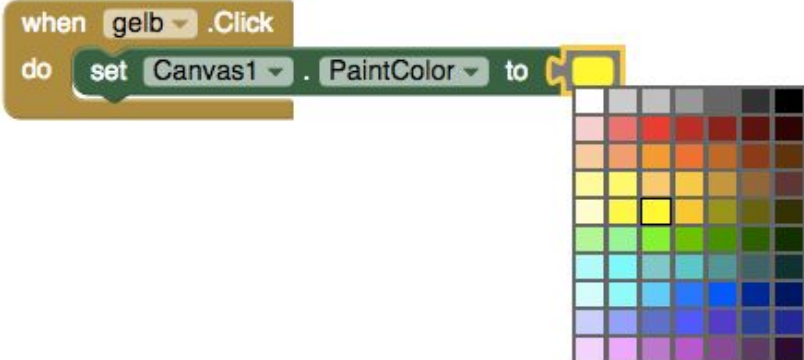
Konzept	Beschreibung
<p>Befehlsfolgen (Algorithmen)</p>	<p>Um ein Computerprogramm zu erstellen, muss man dem Computer genau sagen was zu tun ist - Schritt für Schritt. Dabei helfen Befehlsfolgen, ähnlich wie bei einem Kuchenrezept. Sie definieren, welche Anweisungen in einer festgelegten Reihenfolge durchgeführt werden.</p> <p>Beispiel</p>  <pre> call Sound1 .Play set Label1 . Text to "MIIAAAAUUUUU" </pre>
<p>If-Else (Bedingungen und bedingte Anweisungen)</p>	<p>Bedingte Anweisungen überprüfen eine Bedingung. Dadurch können Fallunterscheidungen vorgenommen werden: Bestimmte Anweisungen werden nur ausgeführt, wenn eine bestimmte Bedingung eintritt.</p> <p>Beispiel</p>  <pre> if (Button1 . Image = "dog.png") then set Button1 . Image to "kitty.png" set Label1 . Text to "MIAUUU" call Sound_Katze .Play else if (Button1 . Image = "kitty.png") then set Button1 . Image to "dog.png" set Label1 . Text to "WUFF" call Sound_Hund .Play </pre>

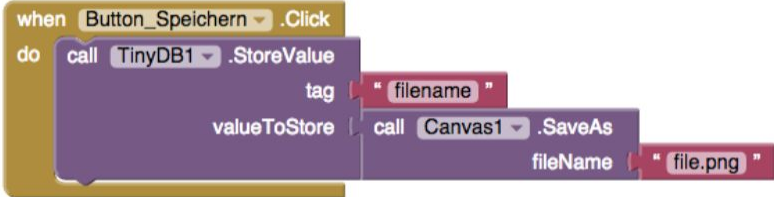
Basiskurs: App 2 Wahrheitskugel

Konzept	Beschreibung
<p>Listen</p>	<p>Listen sind eine wichtige Datenstruktur in der Informatik, um zusammenhängende Daten zu sortieren und zu verwalten. Listen können unterschiedliche Elemente beinhalten, wie z.B. Zahlenwerte oder auch Textelemente. Man kann Listen nach bestimmten Kriterien sortieren, z.B. alphabetisch oder bei Zahlen aufsteigend oder absteigend.</p> <p>Listen haben außerdem Eigenschaften, wie z.B. die Listenlänge, die die Anzahl der Elemente in einer Liste definiert.</p> <p>Beispiel</p> 
<p>Random (Zufallswerte)</p>	<p>Der Programmier-Block "pick a random item list" wählt ein zufälliges Element innerhalb einer Liste aus.</p> <p>Beispiel</p> 

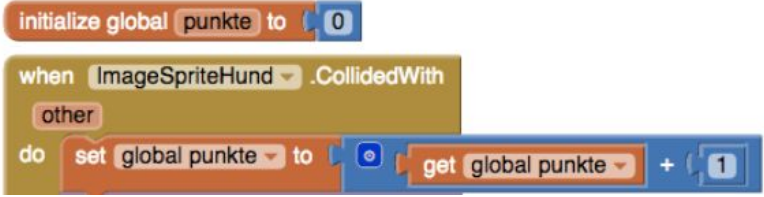
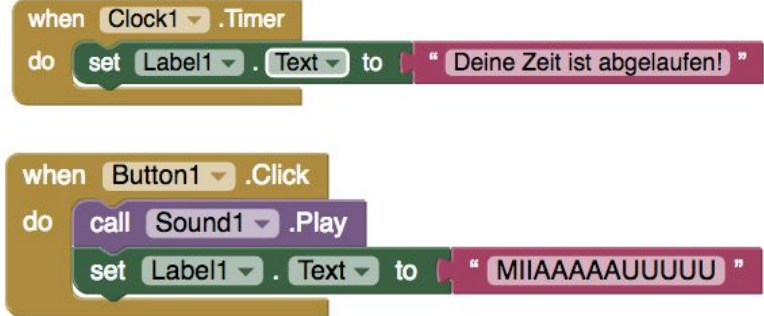
Konzept	Beschreibung
Text-to-speech (Sprachausgabe)	<p>Der Programmier-Block "text-to-speech" ermöglicht die Sprachausgabe über die Lautsprecher des Smartphones. So können z.B. Textelemente "vorgelesen" werden.</p> <p>Beispiel</p>  <p>The image shows a Scratch code block for text-to-speech. It is a purple block with the text "call TextToSpeech1 .Speak" and a "message" input field. The input field contains "Label_Antwort . Text".</p>
Play (Soundausgabe)	<p>Der Programmier-Block "call (Soundname) play" ermöglicht die Soundausgabe über die Lautsprecher des Smartphones. So können Töne, Geräusche und Soundfiles (z.B. mp3-Datei) vom Smartphone abgespielt werden.</p> <p>Beispiel</p>  <p>The image shows a Scratch code block for playing a sound. It is a purple block with the text "call Sound1 .Play".</p>

Basiskurs: App 3 Zeichenapp

Konzept	Beschreibung
<p>Farbmischung (RGB-Werte)</p>	<p>Mit den Programmier-Blöcken unter "Colors", können verschiedene Farben ausgewählt und zugewiesen werden. Es können aber auch eigene Farben definiert werden:</p> <p>Farben bestehen auf Bildschirmen immer aus drei Farbwerten für Rot, Grün und Blau (=RGB). Jede Farbe kann dabei einen Wert von 0 bis 255 zugewiesen bekommen. "Lila" ist zum Beispiel eine Mischung aus Rot und Blau und hat deswegen folgende RGB-Werte: R=100, G=0, B=100.</p> <p>Beispiel</p>  <p>Zufallsfarbe:</p> 

Konzept	Beschreibung
<p>Datenbanken</p>	<p>Datenbanken helfen uns dabei, größere Datenmengen zu verwalten. Mit Hilfe einer Datenbank können wir Daten und auch ganze Dateien speichern, strukturieren, suchen und finden.</p> <p>Im App Inventor kann man mit dem Element "TinyDB" eine Datenbank auf dem Handy bzw. dem Emulator angelegen. Über das „tag“ (hier: filename) wird definiert, wie du später auf die Datei zugreifen kannst.</p> <p>Beispiel</p>  <pre> when Button_Speichern .Click do call TinyDB1 .StoreValue tag "filename" valueToStore call Canvas1 .SaveAs fileName "file.png" </pre>

Basiskurs: App 4 Spiel

Konzept	Beschreibung
Variablen	<p>Variablen sind Platzhalter für veränderliche Werte, wie z.B. der Punktestand in einem Spiel. Sie speichern also bestimmte Werte, die sich verändern können.</p> <p>Unter "Variables" befinden sich Programmier-Blöcke, um Variablen zu erstellen und sie im Programm zu verwenden.</p> <p>Beispiel</p>  <pre> initialize global punkte to 0 when ImageSpriteHund .CollidedWith other do set global punkte to + get global punkte + 1 </pre>
Events / Ereignisse	<p>Events helfen uns bestimmte Befehlsfolgen immer dann auszuführen, wenn eine bestimmte Begebenheit eintritt. Ereignisse können z.B. Benutzereingaben (Button-Klick, Maus-Klick, Taste,...) oder Systemereignisse (Zeitpunkt, Zustandsänderung,...) sein.</p> <p>Beispiel</p>  <pre> when Clock1 .Timer do set Label1 .Text to "Deine Zeit ist abgelaufen!" when Button1 .Click do call Sound1 .Play set Label1 .Text to "MIIAAAAUUUUU" </pre>