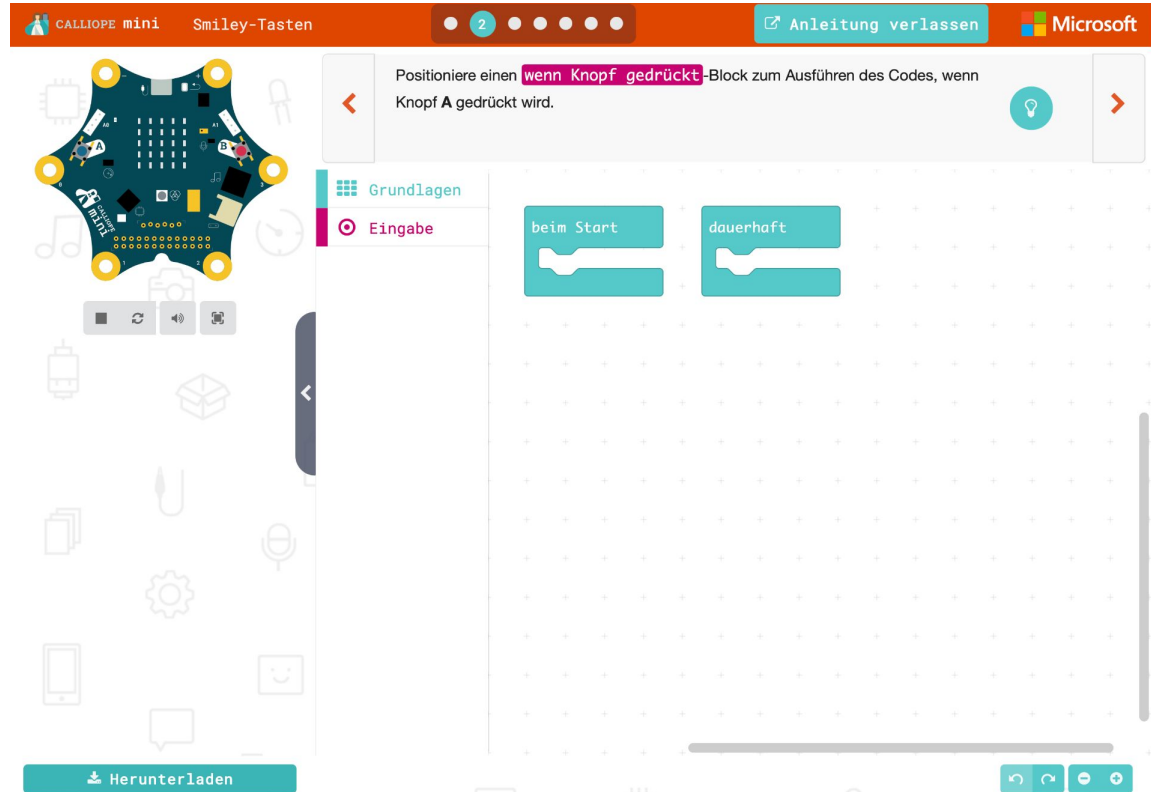


BEISPIEL

Tutorials können individuell erstellt und bereitgestellt werden.

Aufgaben können im eigenen Tempo bearbeitet, gelöst und überprüft werden.

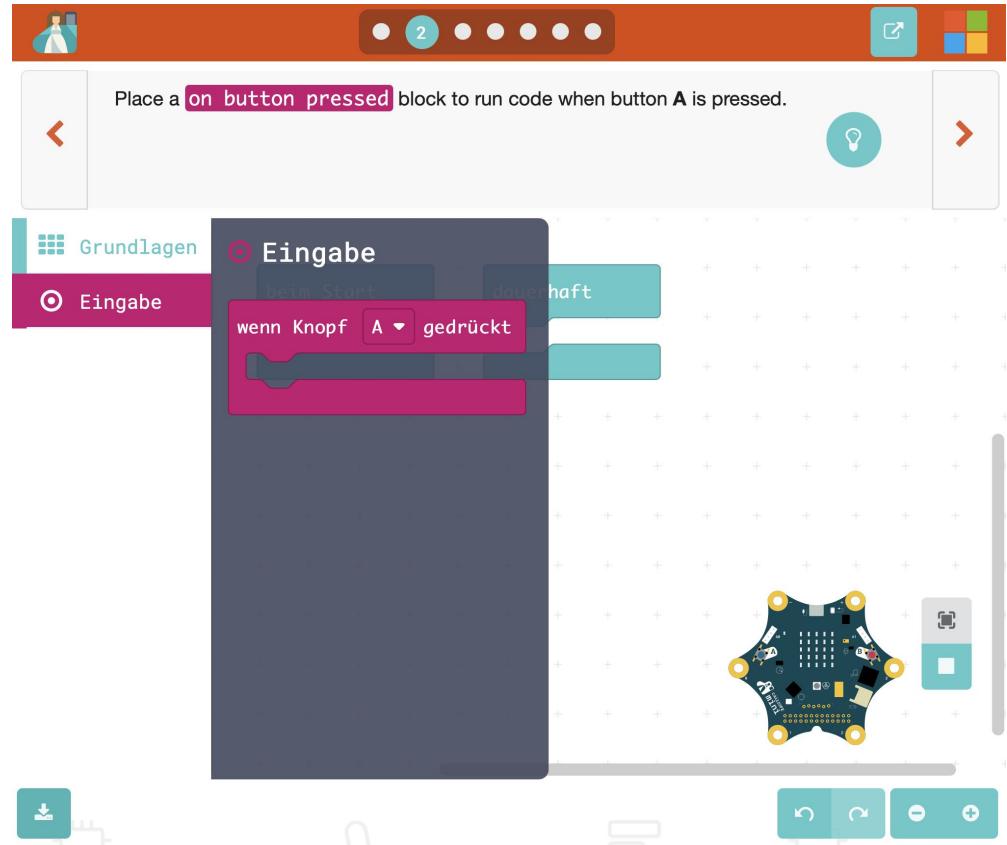
Das erstellte Programm lässt sich weiterentwickeln.



VERWENDETE CODEBLÖCKE

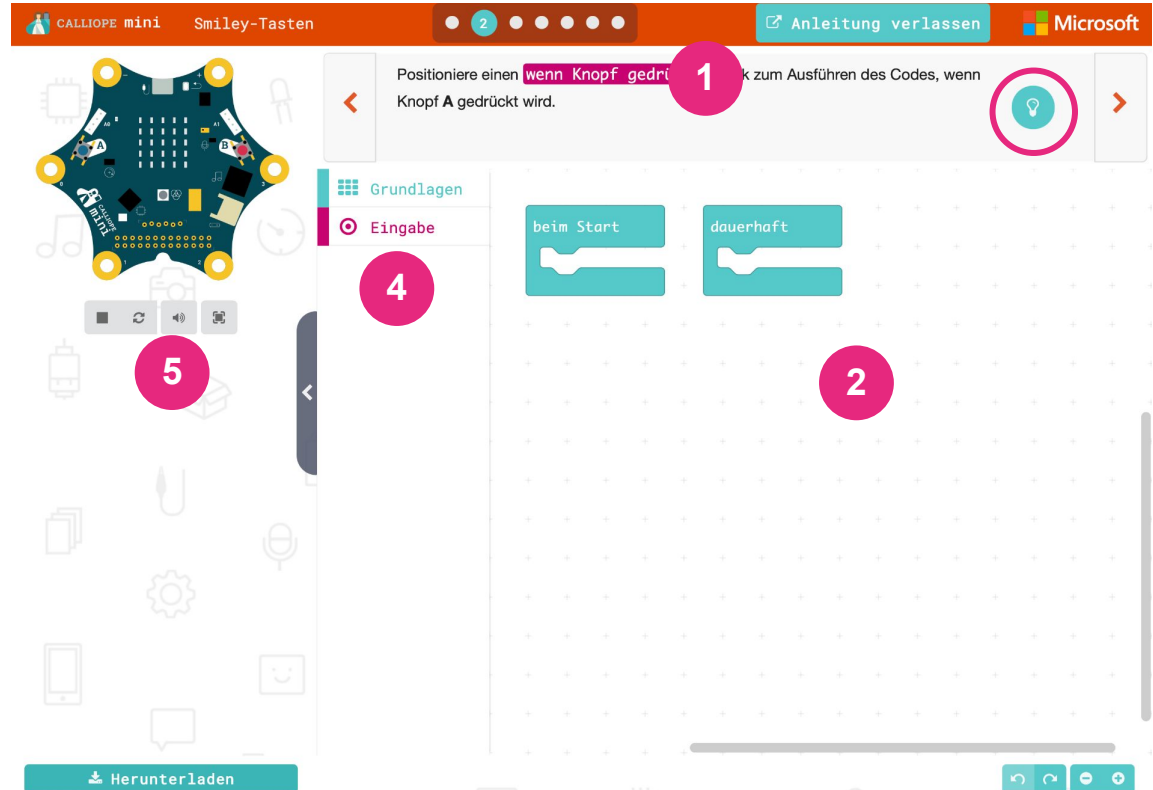
Tutorials haben außerdem den Vorteil, dass nur die verwendeten Blöcke angezeigt werden.

Die Kinder lernen so übersichtlich, in welchen Kategorien sich die einzelnen Programmierblöcke befinden.



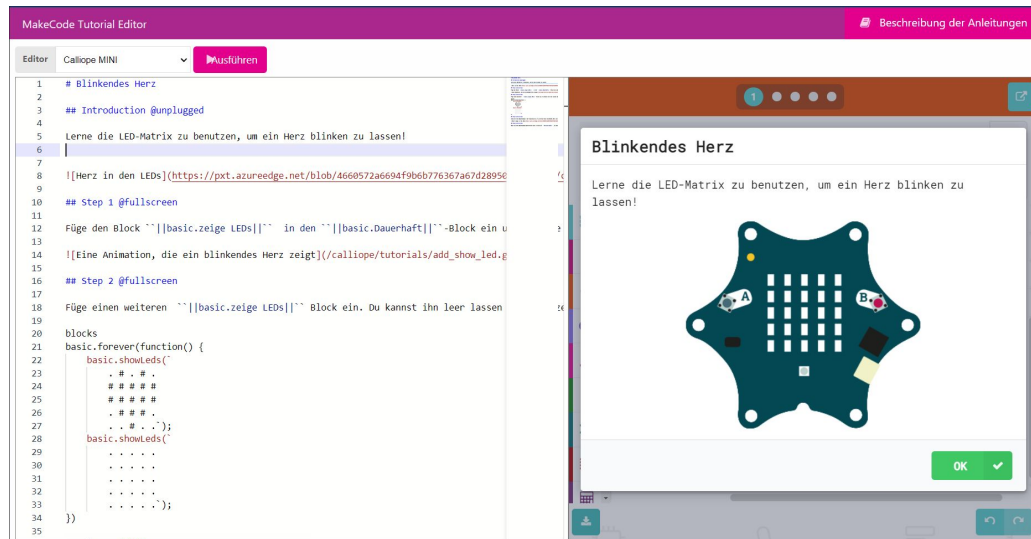
ÜBERBLICK

- 1 - Anweisungen und Tipps
- 2 - Programmierbereich
- 3 - Schritt-für-Schritt Steuerung
- 4 - Benötigte Codeblöcke
- 5 - Simulator



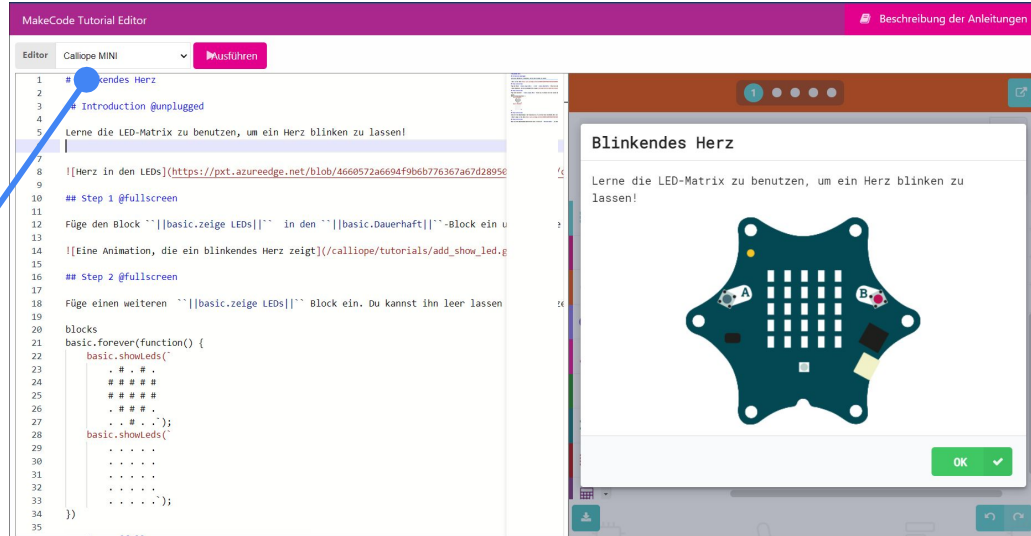
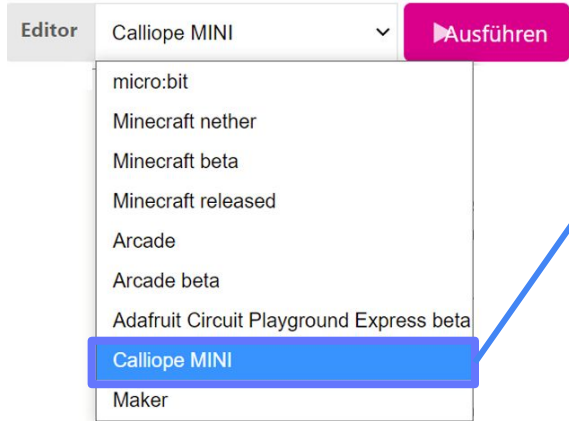
TUTORIAL TOOL

1. Überblick
2. Markdown Syntax
3. Schritte erstellen
4. Dialog Optionen
5. Codeblock Auszeichnung
6. Tipps
7. Editor Optionen
8. Templates
9. Teilen & Speichern
10. Github (Speichern & Teilen)



ÜBERBLICK

Bevor ein Tutorial für den Calliope mini erstellt wird, sollte der richtige Editor ausgewählt werden:



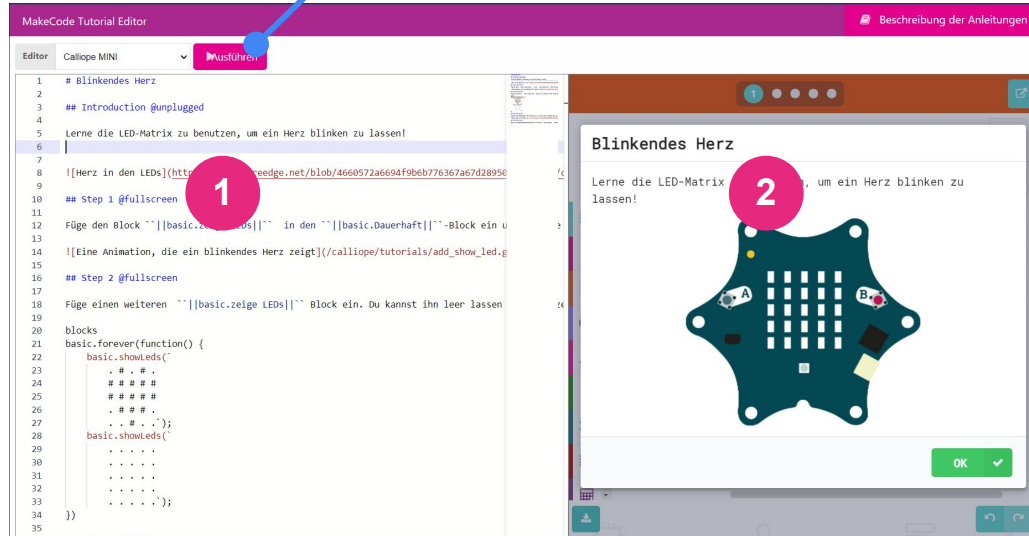
ÜBERBLICK

Die Tutorials werden in der Auszeichnungssprache Markdown geschrieben und anschließend zu Tutorials "gerendert". Markdown hat eine einfache Syntax, um Text zu formatieren und zu gestalten.

Das Tutorial-Tool hat zwei Ansichten:

- 1 - Markdown Text-Editor
- 2 - Ansicht des Tutorials

Über den Button **Ausführen** kann das Tutorial getestet und aktualisiert werden



MARKDOWN

Headline 1

Headline 2

Headline 3

fetter Text

kursiver Text

``code-block/text in monospace``

[\[link öffnen\] \(calliope.cc\)](#)

[!\[Alternativer Text\] \(Pfad_oder_URL_des_Bildes.png\)](#)



CALLIOPE

Headline 1

Überschrift des Programms

Headline 2



Erzeugt auch die
Zwischenschritte des
Tutorials

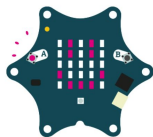
Headline 3

fetter Text

kursiver Text

`code-block/text in monospace`

[link öffnen](#)



SCHRITTE ERSTELLEN

Die Struktur und Schritte des Tutorials werden mit der ## zweiten Überschrift dargestellt.

Texte, Bilder und Codebeispiele können als Anweisung unterhalb der Schrittüberschrift eingefügt werden und werden im jeweiligen Schritt angezeigt

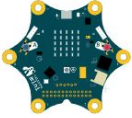
Anzeige der Schritte

1

Füge einen **wenn Knopf A gedrückt**-Block ein, um den Code auszuführen, wenn die Schaltfläche **A** gedrückt wird.

beim Start

dauerhaft



SCHRITTE ERSTELLEN

Smiley Tasten

Überschrift des Programms

Schritt 1

1. Schritt

Füge einen ```|input:wenn Knopf A gedrückt|```-Block ein, um den Code auszuführen, wenn die Schaltfläche ****A**** gedrückt wird.

Anweisung 1

Schritt 2

2. Schritt

Platziere einen ```|basic:zeige LEDs|```-Block in ```|input:wenn Knopf A gedrückt|```, um einen Smiley auf dem Bildschirm anzuzeigen. Drücke den ****A**** Knopf im Simulator, um den Smiley zu sehen.

Anweisung 2



CALLIOPE

Anzeige der Schritte

1

Füge einen **wenn Knopf A gedrückt**-Block ein, um den Code auszuführen, wenn die Schaltfläche **A** gedrückt wird.



DIALOG OPTIONEN

Einen Schritt kann man außerdem die Optionen geben, wie der Dialog dargestellt werden soll:

Tutorial Name

Einführung @unplugged

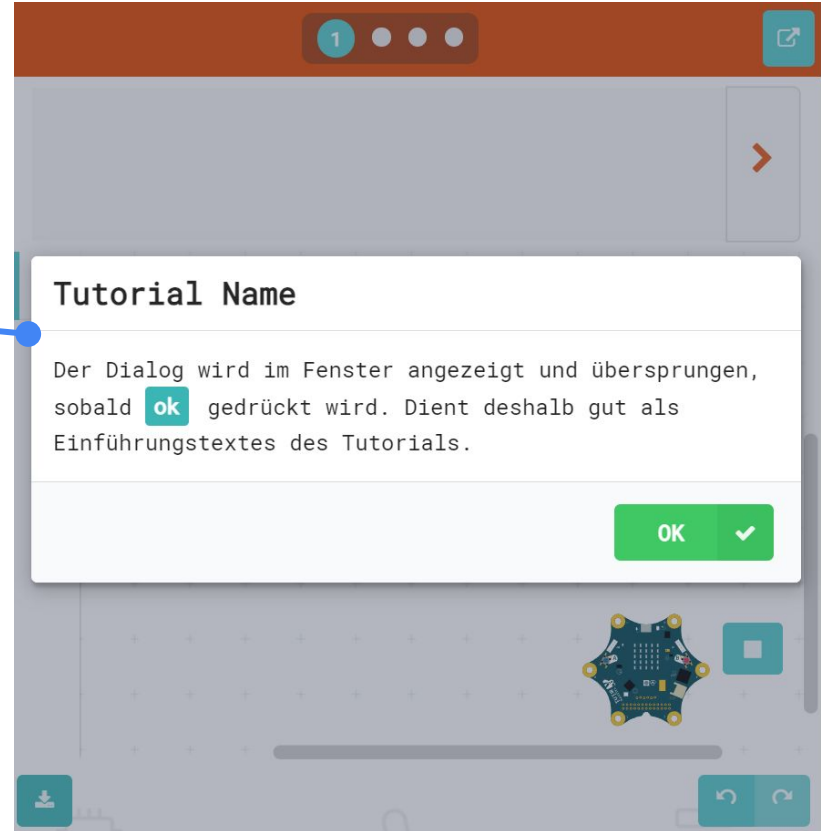
Der Dialog wird im Fenster angezeigt und übersprungen, sobald `ok` gedrückt wird.
Dient deshalb gut als Einführungstext des Tutorials.

Geöffneter versteckter Dialog @fullscreen

Der versteckte Dialog sollte direkt geöffnet sein



CALLIOPE



DIALOG OPTIONEN

Einen Schritt kann man außerdem die Optionen geben, wie der Dialog dargestellt werden soll:

```
# Tutorial Name
```

```
## Einführung @unplugged
```

Der Dialog wird im Fenster angezeigt und übersprungen, sobald ``|ok|`` gedrückt wird.

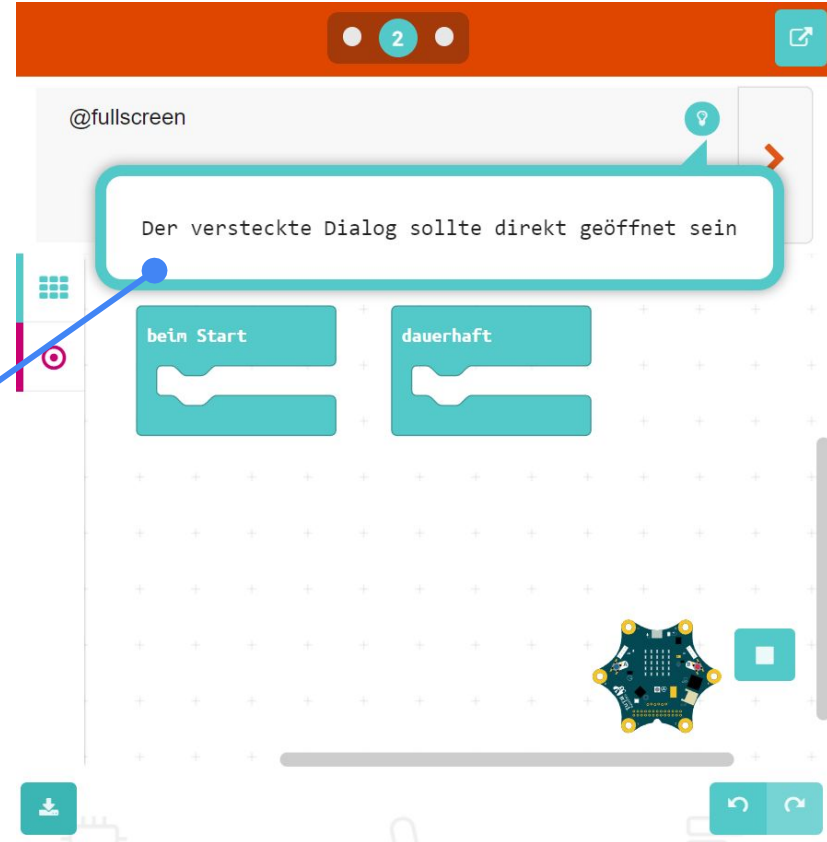
Dient deshalb gut als Einführungstext des Tutorials.

```
## Geöffneter versteckter Dialog @fullscreen
```

```
Der versteckte Dialog sollte direkt geöffnet sein
```



CALLIOPE



CODEBLOCK AUSZEICHNUNG

Neben den Standard Markdown-Auszeichnungen können Codeblöcke in der Farbe der Kategorie im Editor dargestellt werden.

Es ist nicht notwendig, dass es den geschriebenen Block tatsächlich gibt.

Um den englischen Namen für die Kategorie herauszufinden, kann man in die textbasierte Javascript-Ansicht wechseln.

englischer Kategorienname für
Farbzuweisung

beliebiger Name des
gekennzeichneten Blocks

```|input.Eingabe|```



CALLIOPE

## Tutorial



## Javascript-Ansicht



# TIPPS

Es kann ein Tipp hinterlegt werden, der nicht sofort im Dialogfeld erscheint.

Sobald der Text in 3 Gravis/Akzent-Zeichen eingefügt wurde, wird der Text versteckt und es erscheint die Glühbirne automatisch im Tutorial.

```

Wenn noch "blocks" davor geschrieben wird,
dann wird das Javascript in Blöcke dargestellt.

```

```blocks

```
input.onButtonPressed(Button.A, function () {  
    basic.showIcon(IconNames.Happy)  
})
```

```



CALLIOPE

The screenshot shows the Calliope3 app interface. At the top, there is an orange header bar with a '1' in a circle and a share icon. Below the header, a text box contains the instruction: 'Sobald der Text in 3 Gravis/Akzent-Zeichen eingefügt wurde, erscheint die Glühbirne automatisch im Tutorial.' A light blue callout box with a lightbulb icon contains the text: 'Wenn noch "blocks" davor geschrieben wird, dann wird das Javascript in Blöcke dargestellt.' Below this, a code block is shown with the following code: 

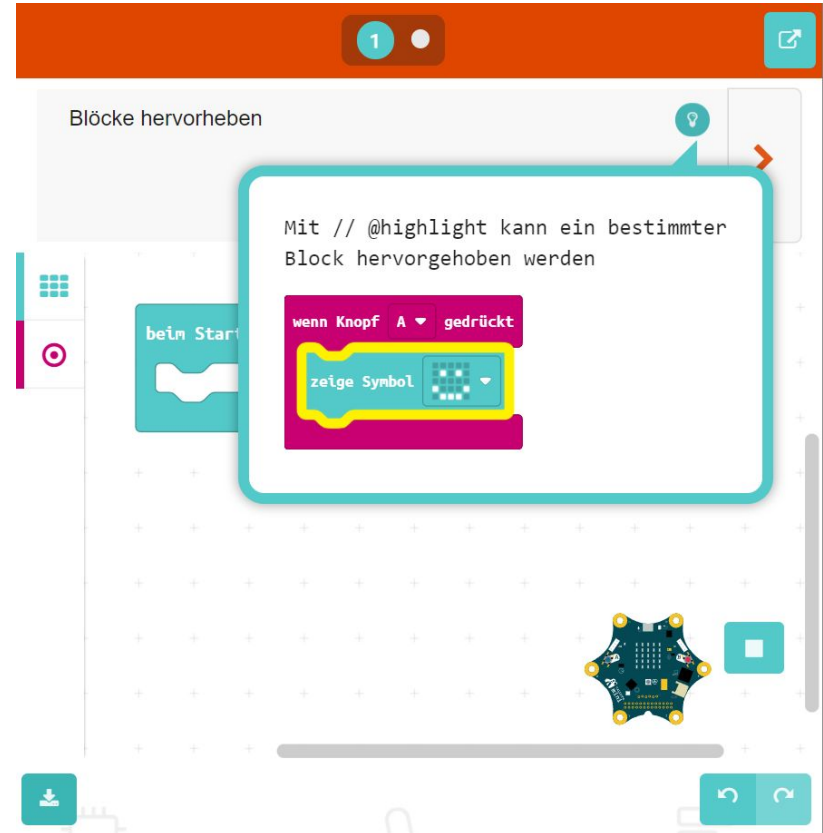
```
wenn Knopf A gedrückt
 zeige Symbol [Glühbirne]
```

 A blue callout box points to the code block with the text: 'Es werden nur die Kategorien angezeigt, die im Tipp vorzufinden sind!'. At the bottom right, there is a Calliope3 hardware component and a 'stop' button icon. The bottom of the screen features a navigation bar with a download icon, a home icon, and a refresh icon.

# TIPPS

Mit `// @highlight` kann ein bestimmter Block hervorgehoben werden

```
```blocks
input.onButtonPressed(Button.A, function () {
  // @highlight
  basic.showIcon(IconNames.Happy)
})
```
```



# EDITOR OPTIONEN

Es gibt ein paar Einstellungen die direkt nach dem Titel des Editors vorgenommen werden können. Diese werden mit `###` definiert und über einen logischen Wert `true` oder `false` aktiviert bzw. deaktiviert:

# Blinkendes Herz

```
@explicitHints true
@hideIteration false
@flyoutOnly true
@diffs true
```

**@diffs**  
stellt nur die im Schritt neu hinzugefügten Blöcke farblich dar

**@flyoutOnly**  
die Blöcke werden alle ausgeklappt angezeigt

**@hideIteration**  
wird diese auf false gesetzt  
verschwindet die Schrittanzeige

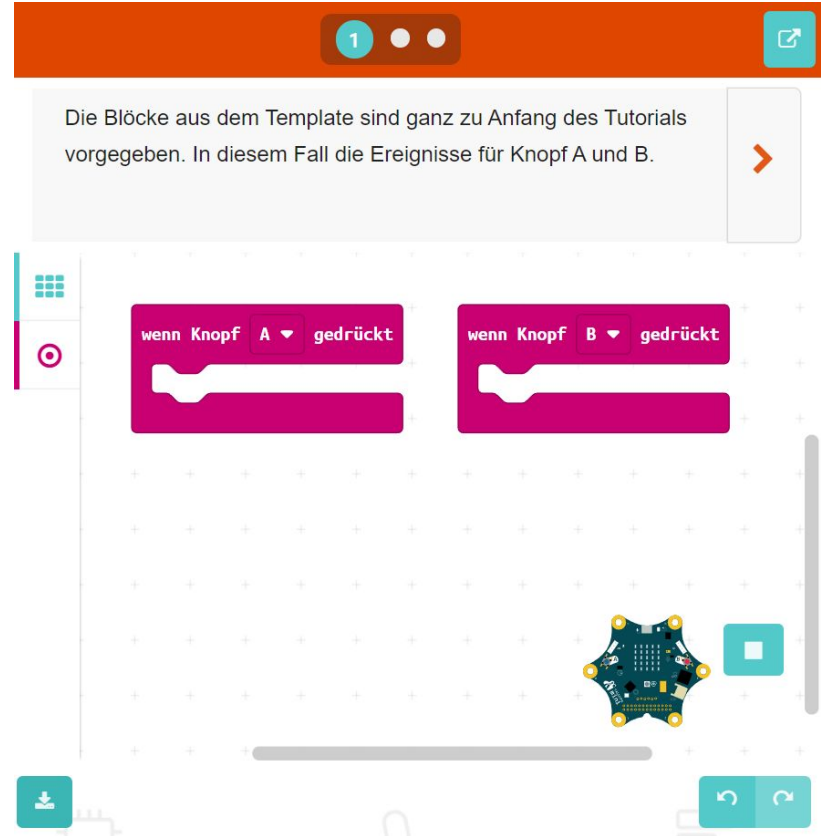
**@explicitHints** die Tipps werden nicht mehr versteckt, sondern direkt in die Dialogbox eingefügt



# TEMPLATES

Für manche Programme, die kein Startfunktion oder Dauerhafter Schleife beinhaltet, mag es sinnvoll sein ein anderes Template auszuwählen, dass die Blöcke zu Beginn des Tutorial vorgibt:

```
```template
input.onButtonPressed(Button.A, function () {
})
input.onButtonPressed(Button.B, function () {
})
```
```





# TEILEN & SPEICHERN

Zum Testen klicke auf Teilen und kopieren den Link in deinen Browser.



<https://makecode.calliope.cc/#tutorial:51014>

Der Programmcode des Tutorials kann lokal in einer Text/Markdowndatei auf dem Computer gespeichert werden.

Über kopieren und einfügen lässt sich das Tutorial wieder im Tutorial Tool bearbeiten.



# GITHUB

Um an einem Tutorial gemeinsam weiterzuarbeiten oder es zu teilen kann das Tutorial auch im Makecode-Editor erstellt und auf Github als Repository angelegt werden.

- Erstelle dazu ein neues Projekt im Makecode-Editor
- Erstelle ein Github Repository



- Melde dich bei Github an
- Gib dem Repository/Tutorial einen Namen

## Github-Repository erstellen



Hoste deinen Code auf GitHub und arbeite mit Freunden zusammen. ?

Repositoryname

blinkendes-herz

Repositorybeschreibung

Tutorial, um ein blinkendes Herz auf dem Display anzuzeigen.

Öffentliches Repository, jeder kann Deinen Code sehen. ▼

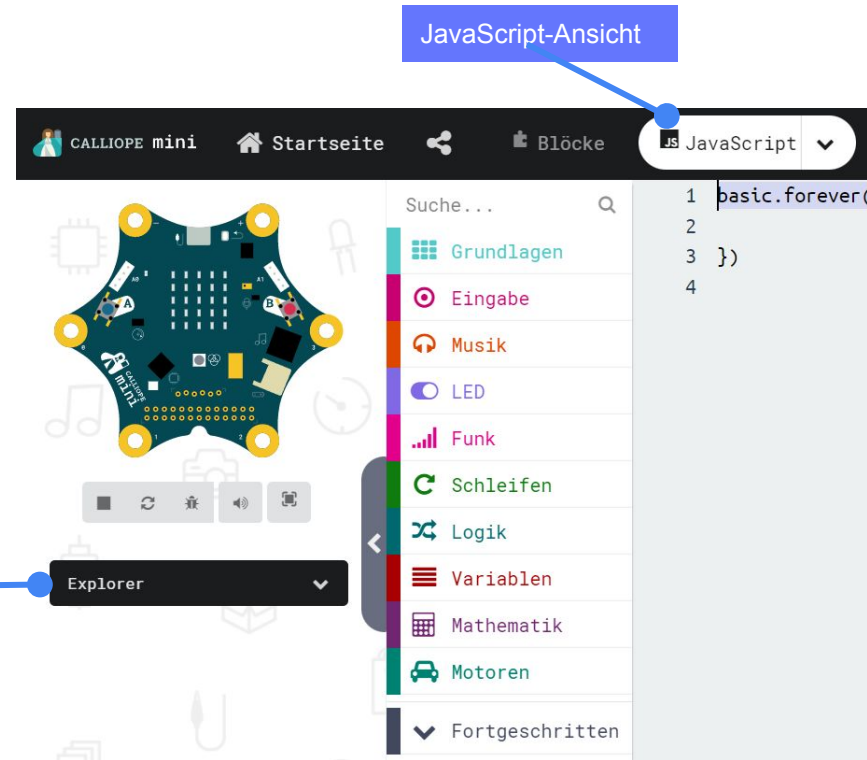
Los geht's!



# GITHUB

Wenn in die Javascript-Ansicht gewechselt wird, dann erscheint der Dateieexplorer, der alle Dateien anzeigt, die für das Programm auf Github angelegt wurden. Die main.ts ist die Datei in dem das Programm in Typescript/Javascript liegt und dessen Quellcode im Editor angezeigt wird.

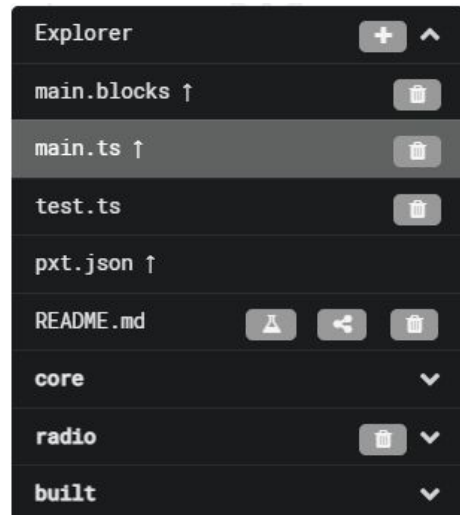
Unterhalb des Simulators befindet sich der Datei-Explorer



# GITHUB

Um eine neues Tutorial in Markdown zu erstellen, führe die folgenden Schritte aus:

- klicke auf das + um eine neue Datei hinzuzufügen.



# GITHUB

Um eine neues Tutorial in Markdown zu erstellen, führe die folgenden Schritte aus:

- klicke auf das + um eine neue Datei hinzuzufügen.
- benenne die Datei, um in den Tutorialnamen mit der Markdown-Endung .md

Eine neue Datei hinzufügen?



Bitte gib einen Namen für deine neue Datei an.

Los geht's!

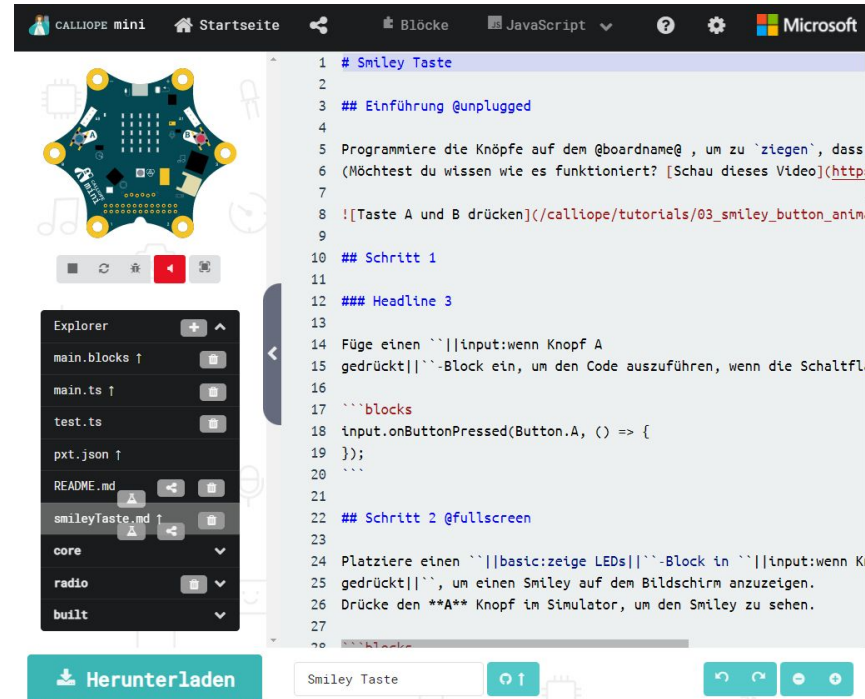


# GITHUB

Um eine neues Tutorial in Markdown zu erstellen, führe die folgenden Schritte aus:

- klicke auf das + um eine neue Datei hinzuzufügen.
- benenne die Datei, um in den Tutorialnamen mit der Markdown-Endung .md
- schreibe das Tutorial

Es empfiehlt sich das Tutorial im Tutorial-Tool zu schreiben und das Markdown, dann in den Editor zu kopieren!



# GITHUB

Um eine neues Tutorial in Markdown zu erstellen, führe die folgenden Schritte aus:

- klicke auf das + um eine neue Datei hinzuzufügen.
- benenne die Datei, um in den Tutorialnamen mit der Markdown-Endung .md
- schreibe das Tutorial
- Übermittle die Änderungen auf Github



Herunterladen

Smiley Taste



The screenshot shows the CALLIOPE mini IDE interface. At the top, there's a navigation bar with 'CALLIOPE mini', 'Startseite', 'Blöcke', 'JavaScript', and a Microsoft logo. Below this, there's a visual representation of the CALLIOPE mini board. To the left is a file explorer titled 'Explorer' showing a directory structure: 'main.blocks', 'main.ts', 'test.ts', 'pxt.json', 'README.md', 'smileyTaste.md' (highlighted), 'core', 'radio', and 'built'. To the right is a code editor titled 'smileyTaste.md' with a 'Zurücksetzen' button. The code in the editor is as follows:

```
@@ -1,0 +1,85 @@
1 + # Smiley Taste
2 +
3 + ## Einführung @unplugged
4 +
5 + Programmiere die Knöpfe auf dem @boardname@ , um zu
6 + `ziehen`, dass er glücklich oder traurig ist.
7 + (Möchtest du wissen wie es funktioniert? [Schau dieses
8 + Video](https://youtu.be/t_Qujjd_38o)).
9 +
10 + ![[Taste A und B drücken]]
11 + (/calliope/tutorials/03_smiley_button_animation.gif)
12 +
13 + ## Schritt 1
14 +
15 + ## Headline 3
16 +
17 + Füge einen ``|input:wenn Knopf A
18 + gedrückt|``-Block ein, um den Code auszuführen, wenn die
19 + Schaltfläche **A** gedrückt wird.
20 +
21 + ```blocks
22 + input.onButtonPressed(Button.A, () => {
23 + });
24 + ```
25 +
26 + ## Schritt 2 @fullscreen
27 +
28 + Platziere einen ``|basic:zeige LEDs|``-Block in
29 + ``|input:wenn Knopf A
30 + gedrückt|``, um einen Smiley auf dem Bildschirm
31 + anzuzeigen.
```



CALLIOPE

# GITHUB

Um eine neues Tutorial in Markdown zu erstellen, führe die folgenden Schritte aus:

- klicke auf das + um eine neue Datei hinzuzufügen.
- benenne die Datei, um in den Tutorialnamen mit der Markdown-Endung .md
- schreibe das Tutorial
- Übermittle die Änderungen auf Github
- Teile das Projekt über die URL, die sich aus der Editor- und Repository-URL zusammensetzt:

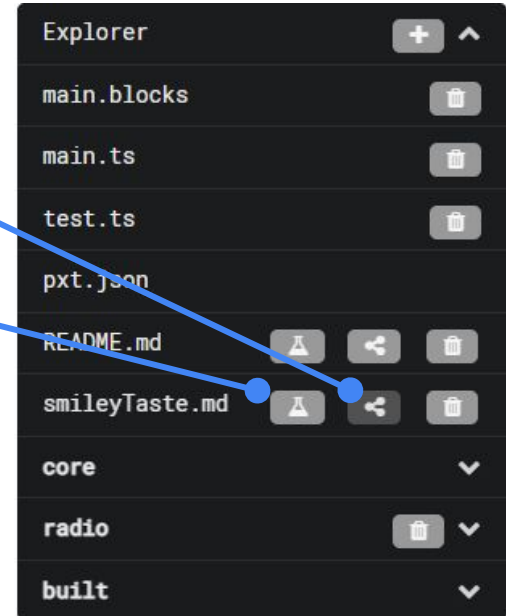
Editor URL

Github Pfad ohne .md-Endung

`https://makecode.calliope.cc/#tutorial:github:account/repository/smileyTaste`

Tutorial Link zum Teilen

Tutorial Vorschau







**CALLIOPE**

Bei Rückfragen stehen wir gerne zur Verfügung.  
Du kannst dich unter +49 (0)30 4849 2030 an uns wenden  
oder per E-Mail: [experts@calliope.cc](mailto:experts@calliope.cc)

**CALLIOPE.CC**

# LINKS

Informationen zum MakeCode Editor

<https://calliope.cc/programmieren/editoren/makecode>

Tutorial Tool

<https://makecode.com/tutorial-tool>

Tutorial Dokumentation

<https://makecode.com/writing-docs/tutorials>

MakeCode Editor

<https://makecode.calliope.cc/>

Tutorials veröffentlichen

<https://makecode.com/writing-docs/user-tutorials>

[https://www.youtube.com/watch?time\\_continue=1&v=XsM8hp7eysA&feature=emb\\_title](https://www.youtube.com/watch?time_continue=1&v=XsM8hp7eysA&feature=emb_title)

Markdown Syntax

<https://www.markdownguide.org/basic-syntax/>

