

Du brauchst diese Komponenten:

- 1 Bühnenbild (z. B. "Jurassic")
- 1 Katze ("Cat1", "Figur1" oder andere Namen)



Tip

Schriftzüge kannst du hinzufügen, indem du eine neue Figur zeichnest und das Textwerkzeug benutzt.



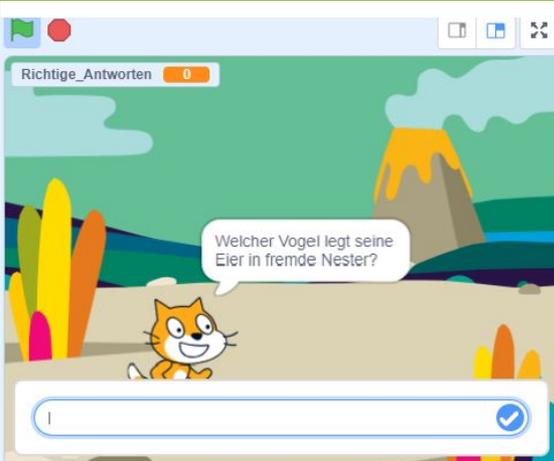
Aufgabe

1. Gib scratch.mit.edu in deinen Browser ein und klicke auf "Entwickeln"  Entwickeln
→ Dort kannst du programmieren.
Hinweis: Um deine Projekte abzuspeichern, musst du dich mit Benutzernamen und Passwort anmelden. Wie das funktioniert, erfährst du von deiner Lehrkraft.
2. Ändere die **Sprache** bei Bedarf auf **Deutsch**. 
3. Links siehst du das Design des Spiels und die Komponenten, die du benötigst.
 - a. Wähle ein **Bühnenbild** aus der Bibliothek als Hintergrundbild aus.
 - b. Überlege dir ein **Thema für dein Quiz**, z. B. "Tiere". Füge deinen **Quiznamen** als Schriftzug hinzu.
4. Lass die Katze **bei Spielstart sagen**: "Los geht's! Was weißt Du alles über [Thema]?"



Nächste Aufgabe

- Erstelle eine Variable "Richtige_Antworten".
- Erstelle selbst einen neuen Block und nenne diesen "Quiz_Teil1". Führe den Block aus, nachdem die Katze gesprochen hat.
Tip: Eigene Blöcke kannst du im Bereich "Meine Blöcke" erstellen.
- Definiere deinen Block: Setze die Variable auf 0 und stelle eine erste Quizfrage.

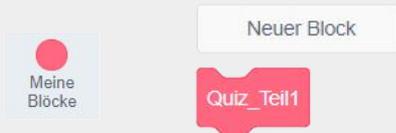


Du brauchst diese Komponenten:



Tipp

Im Reiter "Skripte" kannst du unter "Meine Blöcke" neue, eigene Programmierblöcke erstellen, definieren und danach wiederverwenden.



Aufgabe

1. Erstelle eine **Variable** "Richtige_Antworten".
2. Erstelle selbst einen neuen **Block** und nenne diesen "Quiz_Teil1".
3. **Definiere** deinen Block:
 - a. Setze die **Variable** "Richtige_Antworten" auf **0**.
 - b. **Stelle** deine erste **Quizfrage**, z. B. "Welcher Vogel legt seine Eier in fremde Nester?".

Hinweis: Nachdem eine Frage gestellt wurde, erscheint immer automatisch ein Eingabefeld für eine Antwort.
4. **Führe den Block "Quiz_Teil1" aus**, nachdem die Katze gesprochen hat.



Nächste Aufgabe

Überprüfe die eingegebene Antwort. Je nachdem, ob die Antwort richtig oder falsch ist, soll die Katze unterschiedlich reagieren. Benutze dazu eine bedingte Anweisung (falls...dann...sonst) und mathematische Gleichungen, die du unter "Operatoren" findest.



Du brauchst diese Komponenten:



Tipp

Es gibt einen Trick, wie du weitere Fragen sehr einfach hinzufügen kannst: Klicke mit Rechtsklick auf den Programmierblock (hier: "falls...") und wähle **duplizieren**. Jetzt musst du nur noch die Fragen und Antworten anpassen.

Aufgabe

- Überprüfe die eingegebene Antwort. Je nachdem, ob die Antwort richtig oder falsch ist, soll die Katze unterschiedlich reagieren:
 - Wenn** die eingegebene **Antwort richtig ist**, spiele einen **Klang** ab, lass die Katze **"Richtig!" sagen** und **ändere** die **Variable "Richtige_Antworten" um 1**.
 - Sonst** (wenn die eingegebene Antwort falsch ist), spiele einen anderen **Klang** ab und lass die Katze **sagen: "Leider falsch!"**.

Hinweis: Benutze eine bedingte Anweisung (falls... dann... sonst) und einen Gleich-Operator (=), um zu prüfen, ob die Antwort richtig ist.
- Füge zwei weitere **Fragen** hinzu und überprüfe jeweils die eingegebenen Antworten.
- Teste** dein Quiz, indem Du es mit der grünen Flagge startest und stoppst.



Nächste Aufgabe

Das Quiz soll zwei Schwierigkeitslevel haben: Quiz Teil 1 und Teil 2. Erstelle dafür ein weiteren Block "Quiz_Teil2".



Du brauchst diese Komponenten:

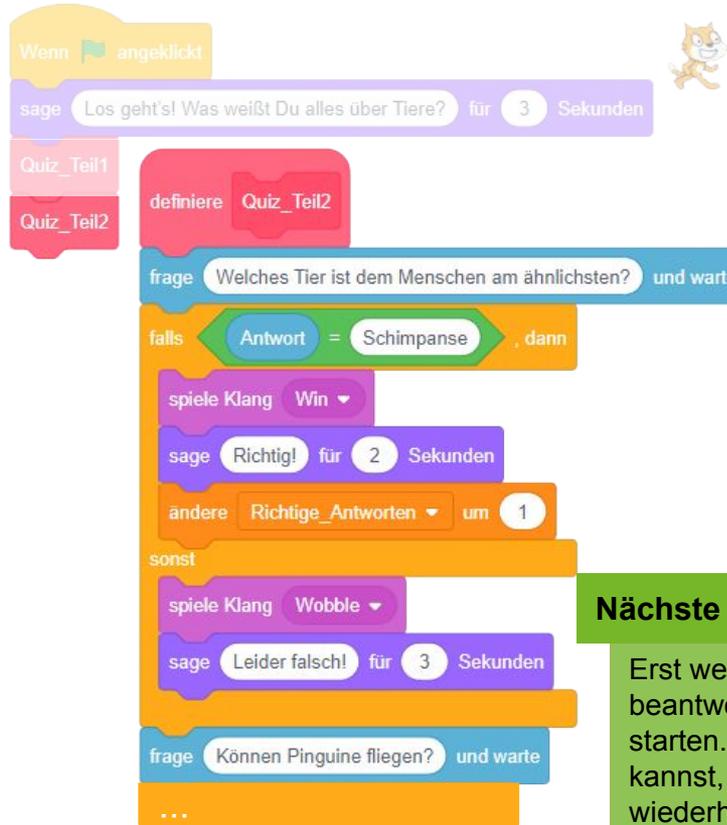


Hinweis

Klicke mit Rechtsklick auf den ersten "frage"-Block in Quiz_Teil1 und wähle **duplizieren**. Füge die Blöcke bei "definiere Quiz_Teil2" an und passe dann jeweils die Fragen und Antworten an.

Aufgabe

1. Das Quiz soll zwei Schwierigkeitslevel haben: Erstelle einen neuen **Block "Quiz_Teil2"** und **rufe ihn** nach Teil 1 **auf**.
2. Füge (wie im ersten Teil) **drei neue Fragen** hinzu und überprüfe danach die eingegebene Antworten.



Nächste Aufgabe

Erst wenn alle Fragen aus Teil 1 richtig beantwortet wurden, soll Teil 2 des Quiz starten. Überlege Dir, wie Du überprüfen kannst, ob der Spieler die Fragen aus Teil 1 wiederholen muss oder mit den Fragen aus Teil 2 weitermachen darf.



Du brauchst diese Komponenten:



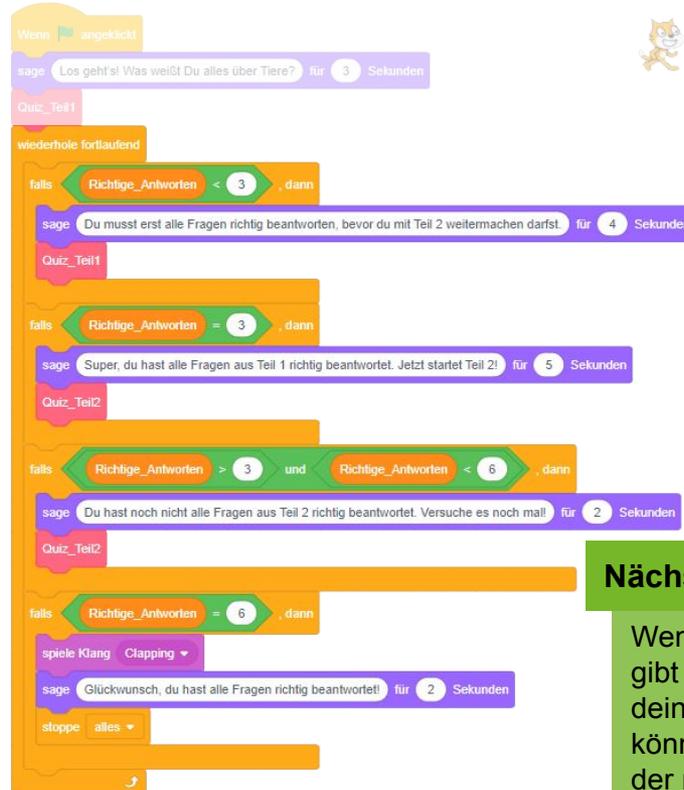
Tip

Benutze mathematische Gleichungen unter "Operatoren", um zu prüfen, wie viele Antworten bereits richtig beantwortet wurden. Du kannst die Blöcke auch ineinander schachteln.



Aufgabe

1. Erst wenn alle Fragen aus Teil 1 des Quiz richtig beantwortet wurden, soll Teil 2 starten: **Überprüfe**, wie viele Antworten bereits **richtig beantwortet** wurden und **entscheide** entsprechend, ob der Spieler die Fragen aus **Teil 1** wiederholen muss oder mit den Fragen aus **Teil 2** weitermachen darf.
2. **Wenn** der Spieler alle Fragen korrekt beantwortet hat, **beglückwünsche** ihn.
3. **Teste** dein Quiz, indem du es mit der grünen Flagge startest. Gib auch mal falsche Antworten ein, um zu testen ob dein Quiz so funktioniert wie gedacht.



Nächste Aufgabe

Wenn der Spieler einen Fehler in Teil 2 macht, gibt es einen Fehler. Diskutiere zusammen mit deinem Sitznachbarn was der Fehler sein könnte und wie er behoben werden kann. Auf der nächsten Seite ist eine mögliche Lösung.



Du brauchst diese Komponenten:



Info

Beim Programmieren gibt es meistens mehr als eine richtige Lösung. Das ist auch hier der Fall.

Alternativ können wir zum Beispiel statt `Richtige_Antworten = 6` die Bedingung so ändern:



```
Richtige_Antworten = 6 oder Richtige_Antworten > 6
```

Aufgabe

1. **Setze die Variable "Richtige_Antworten"** in Quiz_Teil 2 **auf 3**.
Hinweis: Der Wert muss der Anzahl der Fragen entsprechen. Wenn du in den Quizteilen mehr als 3 Fragen gestellt hast, musst du das hier und bei den falls-dann-Anweisungen (Lernkarte 4.4) anpassen.
2. **Teste** dein Quiz, indem Du es mit der grünen Flagge startest.

```
definiere Quiz_Teil2
setze Richtige_Antworten auf 3
frage Welches Tier ist dem Menschen am ähnlichsten? und warte
falls Antwort = Schimpanse dann
  spiele Klang Win
  sage Richtig! für 2 Sekunden
  ändere Richtige_Antworten um 1
...
```



Erläuterung: Warum müssen wir die Variable `Richtige_Antworten` in Quiz_Teil2 auf den Wert = 3 setzen?

Wenn der Spieler in Teil 2 einen Fehler macht, hat er im ersten Versuch zum Beispiel 5 Punkte. Im zweiten Versuch wird die Variable wie im Code vorgegeben weiter hochgezählt, sodass der Spieler (wenn er alle Fragen korrekt beantwortet) 8 Punkte hat. Die Bedingung für die Beglückwünschung und das Ende des Quiz ist, dass die Variable = 6 ist. Die Bedingung trifft also nicht ein, wenn wir die Variable nicht zurücksetzen.

Nächste Aufgabe

- Erstelle eine neue Variable, die das aktuelle Level anzeigt.
- Ändere den Hintergrund, wenn der Spieler Teil 2 des Quiz erreicht.



Du brauchst diese Komponenten:

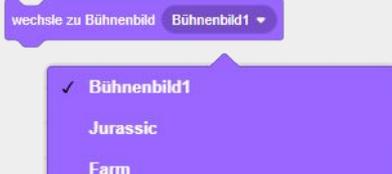


Tip

Weitere Bühnenbilder kannst du zunächst über den Button **“Bühnenbild wählen”** hinzufügen.



Anschließend kannst du die Bühnenbilder beim **“wechsle zu Bühnenbild”-Block** auswählen.



Aufgabe

1. Erstelle eine neue **Variable “Level”**, die anzeigt, ob sich der Spieler im ersten oder zweiten Teil des Quiz befindet. **Setze die Variable** entsprechend dem jeweiligen Level auf 1 bzw. 2.
2. **Ändere das Bühnenbild**, wenn der Spieler Teil 2 des Quiz erreicht.
Hinweis: Definiere das Bühnenbild im Code sowohl für Teil 2 als auch für Teil 1.



Nächste Aufgabe

Hast du noch weitere Ideen, wie du das Quiz erweitern kannst? Dann kannst du das jetzt ausprobieren. Viel Spaß dabei!