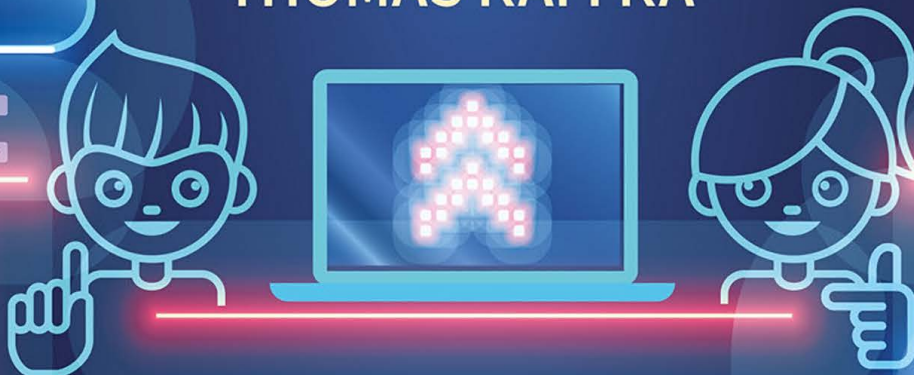


THOMAS KAFFKA



KÜNSTLICHE INTELLIGENZ

einfach visuell
programmieren

EIGENE KI-SPIELE & CO.
MIT SCRATCH UND PYTHON



INHALTSVERZEICHNIS

EINLEITUNG	9
Was dich in diesem Buch erwartet	9
Downloads zum Buch	12
TEIL I: SCRATCH	13
1 MIT SCRATCH IN PICTOBLOX PROGRAMMIEREN	15
1.1 PictoBlox starten oder installieren	15
1.2 Die Figuren	16
1.3 Programme erstellen	18
1.4 Die Bühne	19
1.5 Daten speichern	19
1.6 Nachrichten senden	20
2 NEURONALE NETZE	23
2.1 Das menschliche Gehirn als Vorbild	23
2.2 Künstliche Intelligenz	24
2.3 Lernende Programme	25
3 WELCHES GEDICHT IST DAS?	27
3.1 Die Idee	27
3.2 Programm vorbereiten	28
3.3 Trainieren und testen	31
3.4 Das Programm	32
4 IST DAS EIN HUND ODER EINE KATZE?	35
4.1 Die Idee	35
4.2 Programm vorbereiten	35
4.3 Trainieren und testen	36
4.4 Das Programm	40

5	OBJEKTE AUF BILDERN ERKENNEN	45
5.1	Die Idee	45
5.2	Programm vorbereiten	45
5.3	Trainieren und testen	48
5.4	Das Programm	49
6	DAS SPIEL »MONDLANDUNG« MIT DEINER HAND GESTEUERT	53
6.1	Die Idee	53
6.2	Programm vorbereiten	53
6.3	Das Programm	56
7	DAS SPIEL »JUMP« MIT GERÄUSCHEN GESTEUERT	61
7.1	Die Idee	61
7.2	Programm vorbereiten	61
7.3	Trainieren und testen	64
7.4	Das Programm	65
8	DAS SPIEL »LABYRINTH« MIT GESTEN GESTEUERT	71
8.1	Die Idee	71
8.2	Programm vorbereiten	71
8.3	Trainieren und testen	73
8.4	Das Programm	75
9	WELCHE BLUME IST DAS?	77
9.1	Die Idee	77
9.2	Programm vorbereiten	78
9.3	Trainieren und testen	79
9.4	Das Programm	81
10	AMPEL-STEUERUNG MIT ARDUINO	83
10.1	Die Idee	83
10.2	Der Arduino Uno	83
10.3	Experiment mit blinkender LED	84
10.4	Die Ampelsteuerung	88
10.5	Trainieren und testen	90
10.6	Das Programm	92

TEIL II: PYTHON 95**11 MIT PYTHON IN PICTOBLOX PROGRAMMIEREN 97**

11.1	Erste Schritte	97
11.2	Programmschleifen	99
11.3	Programmbedingungen	103
11.4	Zeichenketten (Strings)	104
11.5	Zahlen	107
11.6	Listen	110
11.7	Funktionen	114
11.8	Programm: »Nimm-Spiel«	115

12 GESICHTSERKENNUNG 119

12.1	Die Idee	119
12.2	Das Programm	119

13 SITZT DIE KAPPE RICHTIG? 123

13.1	Die Idee	123
13.2	Programm vorbereiten	123
13.3	Trainieren und testen	125
13.4	Das Programm	127

14 WIE VIEL GELD KANN ICH SPÄTER VERDIENEN? 131

14.1	Die Idee	131
14.2	Programm vorbereiten	131
14.3	Trainieren und testen	133
14.4	Das Programm	135

15 WELCHE AUTOMARKE SOLLTE ICH MIR KAUFEN? 137

15.1	Die Idee	137
15.2	Programm vorbereiten	137
15.3	Trainieren und testen	139
15.4	Das Programm	140

16 WELCHER ROBOTER IST DAS? 143

16.1	Die Idee	143
16.2	Programm vorbereiten	145
16.3	Trainieren und testen	147
16.4	Das Programm	149

17	DAS SPIEL »SCHERE STEIN PAPIER« MIT DEINER HAND GESTEUERT	155
17.1	Die Idee	155
17.2	Programm vorbereiten	155
17.3	Trainieren und testen	157
17.4	Das Programm	158
	HINWEISE FÜR ELTERN UND LEHRER	161
	Das Ziel dieses Buches	161
	Fertigkeiten und Voraussetzungen	162
	Benötigte Materialien	162
	Wie wird dieses Buch verwendet?	163
	STICHWORTVERZEICHNIS	165

EINLEITUNG

Den Begriff *Künstliche Intelligenz* hast du bestimmt schon oft gehört, oder davon gelesen. Abgekürzt wird er auch durch die Buchstaben KI. KI ist momentan sehr aktuell. Im Internet findest du viele Anwendungen, die mit KI arbeiten. Aber was ist KI eigentlich genau?

Beginnen wir mit der Intelligenz. Intelligenz befähigt uns kurz gesagt zur Problemlösung. Wir können Informationen benutzen, um Aufgaben, die uns gestellt werden, zu bewältigen.

Aber was ist das »künstlich« bei der KI? Die Problemlösung wird eben nicht von einem Lebewesen, sondern von einer Maschine durchgeführt, nämlich von einem Computer. Computer können aber nur etwas tun, wenn sie programmiert werden. Und genau damit beschäftigen wir uns in meinem Buch. Wir erstellen Computerprogramme, die KI verwenden.

KI ist ein weites Feld von Technologien. Wir verwenden hier die sogenannten *Neuronalen Netze*. Mit einem neuronalen Netz ahmt ein Computer die Arbeitsweise des Gehirns nach. Er simuliert praktisch die Zusammenarbeit von verschiedenen Nervenzellen.

Wie du noch sehen wirst, muss ein neuronales Netz trainiert werden, um eine gestellte Aufgabe zu erledigen. Es muss lernen. Genauso wie du in der Schule neue Dinge lernst, bringst du dem neuronalen Netz etwas Neues bei. Und dabei ist es sehr wichtig, dass die Daten, die es lernen soll, qualitativ gut sind, damit es auch das Richtige lernt.

Die nötigen Programme entwickeln wir mit den Programmiersprachen Scratch und Python. Unsere neuronalen Netze trainieren wir mit der kostenlosen Coding-App PictoBlox.

Mit Scratch kannst du deine Programme mithilfe verschiedenfarbiger Blöcke mit der Maus zusammenklicken, es sind also keine Vorkenntnisse notwendig. Wenn du über dieses Buch hinaus noch mehr über Scratch lernen möchtest, kannst du dazu auch mein Buch »Spiele programmieren mit Scratch« verwenden.

WAS DICH IN DIESEM BUCH ERWARTET

Jetzt bist du bestimmt neugierig, welche Themen wir im Buch im Einzelnen behandeln werden.

Nachdem du dich in den ersten Kapiteln mit den Grundlagen von Scratch und neuronalen Netzen vertraut gemacht hast, geht es direkt ans Programmieren mit Scratch:

■ **Welches Gedicht ist das?**

Das neuronale Netz lernt die Wörter zweier Gedichte. Es kann danach entscheiden, zu welchem Gedicht ein bestimmtes Wort gehört.

■ **Ist das ein Hund oder eine Katze?**

Als Nächstes beschäftigen wir uns mit Bildern von Hunden und Katzen. Das neuronale Netz soll lernen, Hunde- und Katzenbilder zu unterscheiden.

■ **Objekte auf Bildern erkennen**

Die Bilder des vierten Kapitels stellen immer nur einen Hund oder eine Katze dar. Nun soll das neuronale Netz verschiedene Objekte, die auf einem Bild dargestellt werden, unterscheiden können. Ich verwende als Beispiel Fotos von Kugelschreibern und Marker-Stiften. Das neuronale Netz soll auf einem Bild jeweils die Kugelschreiber oder Marker identifizieren.

■ **Das Spiel »Mondlandung« mit deiner Hand gesteuert**

In diesem Kapitel lassen wir eine Rakete auf dem Mond landen. Dabei kann sie die Bremsraketen verwenden, um langsamer zu fliegen und letztlich auch sicher zu landen. Die Bremsdüse kannst du mit einer Handgeste an- und abschalten. Die Geste wird mit der Kamera deines Computers aufgezeichnet und an das neuronale Netz weitergeleitet. Das Spiel wird wieder mit Scratch programmiert.

■ **Das Spiel »Jump« mit Geräuschen gesteuert**

Du kannst hier mit Geräuschen einen laufenden Avatar steuern, der über Hindernisse springen kann oder unter ihnen hindurchgleitet. Die Geräusche werden mit dem Mikrofon deines Computers aufgezeichnet und vom neuronalen Netz verwendet. Das Scratch-Programm fragt das neuronale Netz ab und steuert den Avatar.

■ **Das Spiel »Labyrinth« mit Gesten gesteuert**

Du leitest eine Maus durch ein Labyrinth. Dazu verwendest du Gesten deines Körpers, die ein neuronales Netz gelernt hat. Das Spiel wird wieder mit Scratch programmiert.

■ **Welche Blume ist das?**

Wir beschäftigen uns hier mit der Unterscheidung von Blüten der Iris. Das neuronale Netz soll anhand der Abmessungen der Blütenblätter entscheiden, um welche Iris-Unterart es sich jeweils handelt. Dies ist ein Beispiel dafür, dass ein neuronales Netz auch mit Zahlen arbeiten kann.

■ Ampelsteuerung mit dem Arduino

Dies ist ein besonders spannendes Projekt. Dazu benötigen wir noch das Microcontroller-Board Arduino Uno. Wir bauen mit ihm eine Ampel, die durch ein Scratch-Programm gesteuert wird. Dann lernt ein neuronales Netz Fotos von vollen und leeren Straßen. Die Ampel soll nämlich intelligent sein und immer dann auf Grün schalten, wenn die Straße voll ist, also Autos hinter der Ampel stehen. Wenn kein Auto auf der Straße steht, schaltet die Ampel auf Rot.

Im zweiten Teil des Buches kommt die Programmiersprache Python zum Einsatz. Sie ist nicht so visuell wie Scratch, aber PictoBlox hilft dir, indem es automatisch ein Grundgerüst für dein Programm erstellt. Die Projekte in diesem Teil des Buches sind:

■ Gesichtserkennung

Ein neuronales Netz wurde so trainiert, dass es die Mimik eines Gesichts erkennen kann. Das ist das erste Programm, das in Python erstellt wurde.

■ Sitzt die Kappe richtig?

Ein neuronales Netz lernt, wie es aussieht, wenn eine Baseball-Kappe richtig oder falsch getragen wird. Ein Python-Programm kann dich dann beraten.

■ Wie viel Geld kann ich später verdienen?

Hier geht es darum, ob ein Mensch, der mit bestimmten Merkmalen gekennzeichnet wird, über oder unter 50.000 \$ verdienen kann. Das Programm wurde auch mit Python erstellt.

■ Welche Automarke sollte ich mir kaufen?

Wenn du bestimmte Vorstellungen hast, was dein späteres Auto können soll, kann dich ein neuronales Netz beraten, welche Automarke du dir zulegen solltest. Dies ist auch ein Python-Programm.

■ Welcher Roboter ist das?

Du erhältst im Download Fotos von drei Robotern, die das neuronale Netz zu unterscheiden lernt. Dann kann ein Python-Programm Testfotos analysieren, welche Roboter abgebildet sind.

■ Das Spiel »Schere Stein Papier« mit deiner Hand gesteuert

Du kennst bestimmt das Spiel »Schere, Stein, Papier«. Ein neuronales Netz lernt die entsprechenden Handgesten und du kannst dann das Spiel gegen ein Python-Programm spielen.

Wenn du die Projekte durchgeführt hast, wirst du eine Vorstellung davon haben, was KI und insbesondere neuronale Netze leisten können. Du wirst feststellen, dass sie eine Entscheidung immer nur mit einer gewissen Sicherheit treffen kön-

nen. Manchmal treffen sie auch die falsche Entscheidung und sagen beispielsweise, auf einem Foto ist ein Hund, dabei legst du ihnen ein Katzenfoto vor.

KI ist ein spannendes Thema mit viel Potenzial. Es wird verschiedene Bereiche revolutionieren und Dinge möglich machen, die bisher nicht möglich waren. Mit den Projekten meines Buches bekommst du einen Einblick und einen Einstieg in diese Technologie. Ich wünsche dir viel Spaß und Erfolg beim Programmieren.

DOWNLOADS ZUM BUCH

Wir verwenden in diesem Buch die Entwicklungsumgebung »PictoBlox«, um mit neuronalen Netzen in Scratch und Python zu arbeiten. Die neuronalen Netze, die ich in meinem Buch verwendet habe, werden immer mit den Daten und den Programmen gekoppelt gespeichert. Außerdem ist es bei der hier verwendeten Technologie so, dass die Ergebnisse sehr von der Kamera und dem Mikrofon des Computers und beispielsweise den verwendeten Fotos, die du auch selbst machen kannst, abhängen, daher ergibt es keinen Sinn, diese Programme zum Download anzubieten.

Ich stelle dir stattdessen im Download Daten in Form von Fotos oder Texten bzw. Tabellen zur Verfügung, damit du die Grundlagen für das Training der neuronalen Netze hast. Die Scratch- und Python-Programme werden jeweils abgedruckt und du kannst sie dann in den Editor übernehmen. Wenn du Programme selbst eingibst, ist das eine sehr gute Übung und du wirst vieles viel besser verstehen.

Die Downloads zum Buch findest du unter www.mitp.de/1087.

1 MIT SCRATCH IN PICTOBLOX PROGRAMMIEREN

Zunächst schauen wir uns an, was du mit Scratch in PictoBlox so alles machen kannst. Scratch ist eine Programmiersprache für Anfänger, die grafisch aufgebaut ist. Um mit Scratch programmieren zu können, benötigst du einen sogenannten *Editor* oder auch *Integrierte Entwicklungsumgebung* (IDE, Integrated Development Environment) genannt.

1.1 PICTOBLOX STARTEN ODER INSTALLIEREN

Du kannst PictoBlox im Internetbrowser aufrufen oder einen Editor auf deinem Computer installieren. Im Internet rufst du die Adresse <https://pictoblox.ai> auf. Wenn du dann auf die Kachel BLOCK-CODIERUNG klickst, startet der Editor.

Du kannst PictoBlox auch auf deinem Computer installieren. Gib dazu bei Google oder in einer anderen Suchmaschine »PictoBlox Download« ein und gehe auf die Seite von STEMPedia, lade das für dein Betriebssystem richtige Programm herunter und installiere es per Doppelklick.

Nach dem Start von PictoBlox siehst du das Fenster in Abbildung 1.1.



Abbildung 1.1: PictoBlox nach dem Aufruf

Klicke jetzt auf BLOCK-CODIERUNG und du landest im Editor, wie in der nächsten Abbildung zu sehen ist.

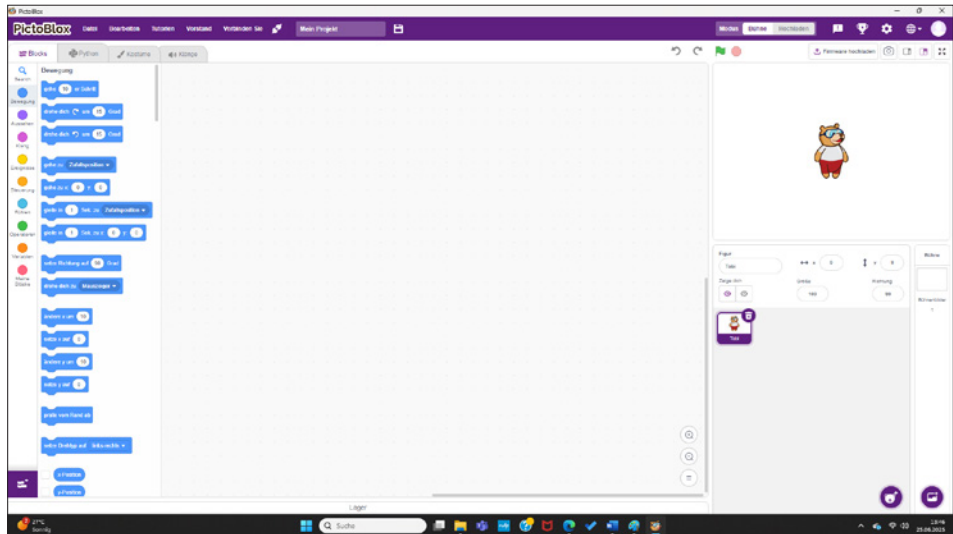





Abbildung 1.2: Der PictoBlox-Editor

In der linken Spalte werden sämtliche Scratch-Befehle aufgeführt. In der Mitte steht später dein Programm. Auf der rechten Seite befindet sich oben das Anzeigefenster, in dem in der Mitte ein Bärchen steht. In diesem Fenster werden die Programmergebnisse ausgegeben. Du kannst es vergrößern, indem du auf dem Button  darüber klickst.

Darunter befinden sich zwei weitere Fenster *Figur* und *Bühne*.

1.2 DIE FIGUREN

Im Figurenfenster werden die von dir in einem Programm verwendeten Figuren angezeigt. Du kannst eine Figur löschen, indem du auf den Löschbutton  oben rechts auf einer Figur klickst. Wenn du das Icon einer Figur anklickst, werden in dem mittleren Fenster die zu der Figur gehörenden Programme angezeigt.

Als Erstes wird immer ein kleiner Bär namens Tobi angezeigt. Du kannst natürlich auch eine andere Figur verwenden. Klicke dazu auf den Button  unten rechts im Fenster. Wenn du dann auf FIGUR WÄHLEN klickst, wird ein Fenster mit vielen Figuren angezeigt, die du auswählen kannst.

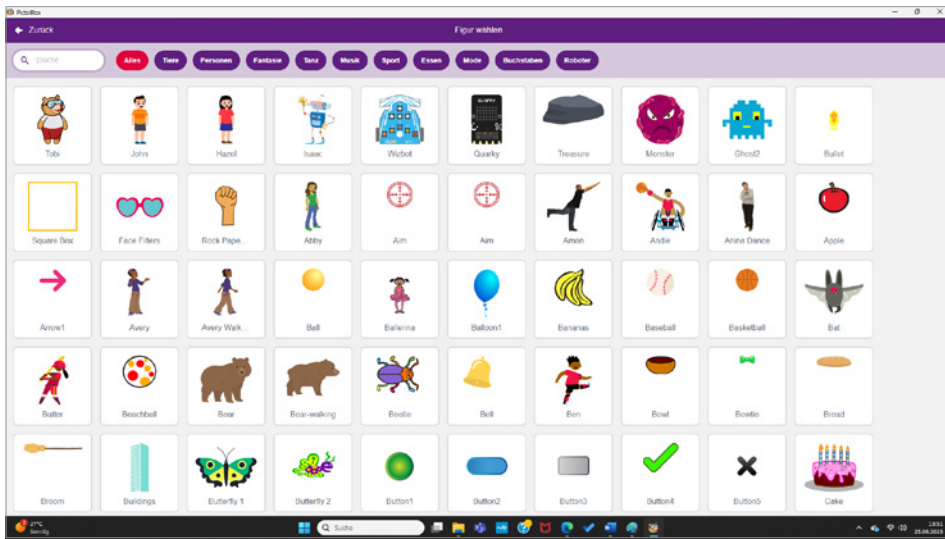


Abbildung 1.3: Figuren zum Wählen

Rette die Programme deiner Figuren

Wenn du deine Figur wechselst und für sie bereits Programme erstellt hast, kannst du diese Programme retten. Klicke ein Programm an und halte es fest. Fahre dann mit ihm über das Icon der Figur, die es erhalten soll, und lass los. Das Programm wurde in den Programmbereich der neuen Figur kopiert. Erst danach solltest du die alte Figur löschen, da deren Programme mit gelöscht werden.

Es ist auch möglich, eigene Figuren zu verwenden. Male am besten mit einem Grafikprogramm ein Bild im Format *png*. Danach kannst du es nach PictoBlox laden, indem du auf **FIGUR HOCHLADEN** klickst (Abbildung 1.4).

Wenn du ein Programm für eine Figur erstellen möchtest, klicke diese im Figurenfenster an und klicke oben links auf die Lasche **Blocks**. Dann kannst du die Programmbefehle aus der linken Spalte per Drag and Drop in das Programmfenster ziehen. Sie werden praktischerweise »zusammengeklickt«. Wenn du einen Befehl löschen möchtest, nimm ihn mit der Maus und bewege ihn nach links über die Liste der Befehle. Lass ihn dort los und er ist gelöscht.




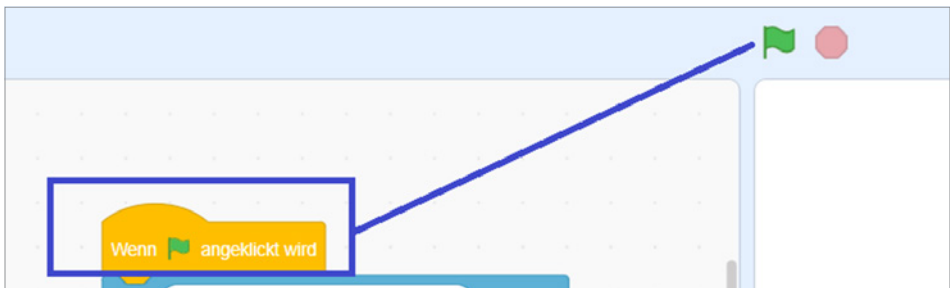
Abbildung 1.4: Hochladen einer eigenen Figur

1.3 PROGRAMME ERSTELLEN

In Scratch werden Programme aus Blöcken zusammengesetzt. Zumeist wird mit dem Befehl mit der Fahne begonnen.




Ein Programm startet, wenn du auf den Button mit der Fahne  über dem Anzeigefenster klickst. Mit dem Button daneben wird ein Programm beendet.



Die Programmbefehle in der linken Spalte sind farblich sortiert. Du kannst auch auf die farbigen Kreise klicken, um direkt zu den richtigen Befehlen zu gelangen.

Zu einem Programm gehört zunächst mindestens eine Figur. Oben links über den Befehlen ist ein Reiter **BLOCKS**, der das Befehlsfenster anzeigt, wenn du auf ihn klickst. Daneben ist der Reiter **KOSTÜME**, der die Figur anzeigt, die du gerade betrachtetest. In diesem Fenster kannst du die Figur verändern.

Dann gibt es noch einen Reiter KLÄNGE, der die Klänge anzeigt, die zu einer Figur gehören. Bei unserem Bärchen wird natürlich ein »Grunzen« angezeigt. Klicke auf den Button  zum Abspielen des Klangs und probiere ihn aus. Du kannst dort auch neue Klänge laden.

Der Reiter PYTHON bezieht sich auf die gleichnamige Programmiersprache. Mehr dazu erfährst du in Kapitel 11.

1.4 DIE BÜHNE

Ich habe dir noch ein Fenster unterschlagen, nämlich das, auf dem BÜHNE steht, ganz rechts unten. Mit ihm legst du den Hintergrund des Anzeigefensters fest. Damit kannst du das Bärchen beispielsweise auf eine richtige Bühne versetzen.

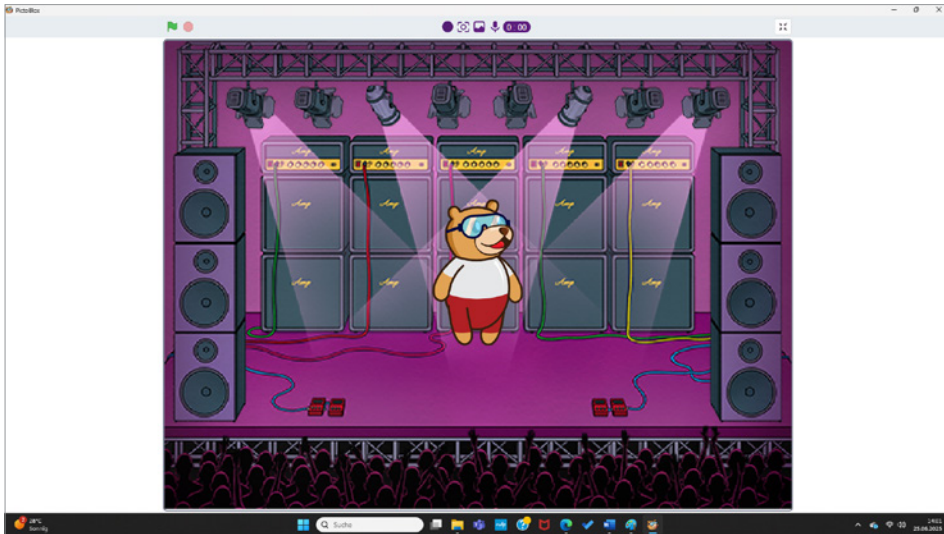


Abbildung 1.5: Das Bärchen auf der Bühne

Die Bühne kann übrigens auch Programme haben.

1.5 DATEN SPEICHERN

In deinen Programmen muss sich der Computer schon mal Daten merken. Dazu verwendest du »Variablen«. In einer Variablen kann zum Beispiel eine Zahl gespeichert werden. Die Variable erhält einen Namen, und wenn du später die Zahl verwenden möchtest, kannst du die Variable mit ihrem Namen nennen.

Variablen werden links unter den Befehlen angelegt, mit dem Button:

Neue Variable

Nach dem Anlegen erscheinen sie unter diesem Button in einer Liste. Du kannst vor einer Variablen ein Häkchen machen, dann wird diese Variable mit ihrem Wert im Anzeigefenster angezeigt. Das ist gut, wenn du den Wert einer Variablen kontrollieren möchtest. Wenn du den Button **NEUE VARIABLE** betätigst, erscheint ein Fenster, in dem du eine Variable anlegen kannst.

kap01_NeueVariableFenster_2.png

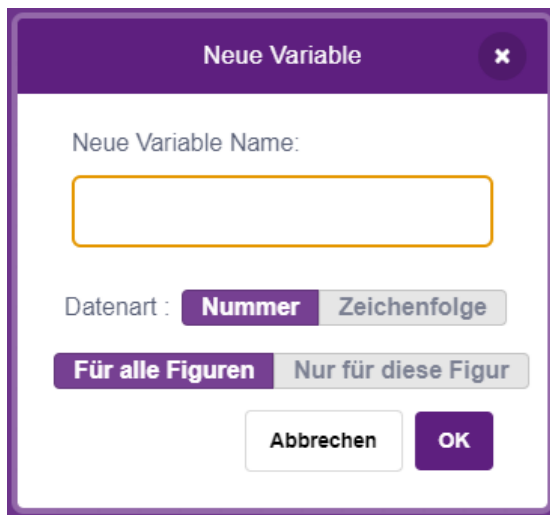


Abbildung 1.6: Fenster zum Anlegen einer Variablen

Wähle **FÜR ALLE FIGUREN**, wenn sich deine Variable auf alle Figuren bezieht. Wähle **NUR FÜR DIESE FIGUR**, wenn die Variable nur zur aktuellen Figur gehören soll.

Variable kontrollieren

Du kannst später übrigens kontrollieren, ob sich eine Variable nur auf eine Figur bezieht, indem du das Häkchen vor der Variablen setzt. Im Anzeigefenster wird dann der Name der Figur vor dem Variablennamen angezeigt. Das kann manchmal wichtig sein.

1.6 NACHRICHTEN SENDEN

Du kannst innerhalb deiner Programme Nachrichten von einer Figur zu einer anderen oder von der Bühne zu einer Figur senden. Dazu gibt es unter anderem zwei Befehle.



Abbildung 1.7: Befehle für Nachrichten

Möchtest du eine Nachricht versenden, verwendest du den Befehl »sende«. Wenn du auf eine Nachricht, die gesendet wurde, reagieren möchtest, verwendest du den Befehl »Wenn ich ... empfange«. Du kannst die Ausklappbox öffnen, um neue Nachrichten zu definieren.

So, jetzt ist es aber genug mit den Vorbereitungen. Du möchtest sicherlich dein erstes Programm entwickeln. Was du sonst noch über Scratch in PictoBlox wissen musst, erkläre ich dir in den weiteren Kapiteln.

STICHWORTVERZEICHNIS

A

Abbruchbedingung 100
abs() 108
Advanced 37
Ampelsteuerung 83, 88
And 101
Anzeigefenster 16
Arduino 83
Array 110
Audioklassifizierung 64
Auto 137
Axon 24

B

Batch Size 37, 48
Bedingung 103
Befehl
 löschen 17
Bibliothek 108
Bild
 klassifizieren 35
Binär 107
Blüte erkennen 77
break 107
Bühne 19
 Bilder laden 40

C

Chatbot 161
ChatGPT 25, 39

D

Daten 135
 aufbereiten 79
Datenstruktur 110
Deep Learning 48

def 114

Dendrit 24
Dezimalzahlen 107
Dictionary 110, 113
Differenzmenge 113
Drag and Drop 17

E

Editor 15, 97
Element 24
elif 103
else 103
Endlosschleife 42, 102, 103, 106
Epochs 37
Eratosthenes 111
Exponentialfunktion 108

F

FaceDetection 120
Fakultät 108
Falsch 101
False 101
Fehler 38, 49, 105
Fehlerkurve 49, 147
Fibonacci-Zahlen 101
Figur
 eigene verwenden 17
 erstellen 17
 löschen 16
 Programm retten 17
 wählen 16
Figurenfenster 16
for-Schleife 99
for-Statement 99
Funktion 109, 114
 definieren 114

G

Gedicht 27
Gehirn 23, 24
Generalisieren 51
Geräusch 61
 aufnehmen 62
Gesichtserkennung 119
Gesten 71
 lernen 71
Gewicht 25
 finales 133
Gleich-Bedingung 101
Gleitkommadarstellung 107
GND 84
Granularität 37

H

Handgeste 53, 155
Hexadezimal 107
Hidden-Schicht 48
Hintergrundgeräusch 62

I

if-Statement 103
Index 105, 110
Integrierte Entwicklungsumgebung 15
Intelligenz 9, 161
 künstliche 9, 23
Internet 15
Iris 77
Iteration 48
Iterativ 108
Iterative Lösung 109

K

Kamera 123
Kategorisieren 23
KI 9, 30, 39, 83, 161
 Gesichtserkennung 119
 Klänge 19

Klasse 23, 45
 anlegen 36
 erzeugen 123
 Name 46
Klassifizierer 29
Klassifizierung 30, 78
Kommazahlen 98
Kommentarzeile 116
Kostüme 18
Künstliche Intelligenz 9, 23, 161

L

Label List 46
Labyrinth 71
Layer 48
Learning Rate 37
LED 84
Lernen 23
list 105
Liste 105, 110
Logische Operatoren 101
Lösung
 iterative 109
 rekursive 109

M

Machine Learning 28
Masse 84
math 108, 114
max() 108
Menge 105, 113
Microcontroller 84
min() 108
Morsecode 86

N

Nachricht 20
 empfangen 42
Nervenzelle 23
Netz 24

Neuronales Netz 9, 23, 161

Aufbau 25

exportieren nach Python 127

Fehler 26

Hidden-Schicht 24

Input-Schicht 24

Output-Schicht 24

Schicht 24

Sicherheit 26

trainieren 25, 31

Nimm-Spiel 115

Not 101

O

Objekterkennung 45

Objektklassifizierung 143

Oktal 107

Operatoren

logische 101

Or 101

Output

festlegen 133

P

Palindrom 106

PictoBlox 12, 15

installieren 15

Pixabay 35

Platine 84

Primzahlen 111

Print-Statement 98

Problemlösung 9

Programm

beenden 18

starten 18

Programmschleife 99

Python 19, 97, 162

als Rechner 98

Python-Referenz 98

Q

Qualität

Daten 80

R

range 100

Rekursiv 109

Rekursive Lösung 109

return 109, 114

Roboter 143

round() 108

S

Schaltung 84

Schere Stein Papier 155

Schleife

abbrechen 107

Schlüssel 113

Schnittmenge 113

Schrittweite 37

Scratch 9, 15, 162

Scratch-Befehle 16

set 105

Set 110, 113

Setosa 78

Sicherheit 40

Signal 24

Spiel 53, 61, 71, 155

Sprite 120, 159

Statistik 131

String 98, 104

Synapse 24

T

Terminal 97

Test 32

Testen 38

Text

umwandeln in Zahlen 133

Tobi 16

- Training 31, 36, 37, 54, 64, 73, 79, 125,
133, 139, 147, 157
- Advanced 37
- Epoche 37
- gute Werte 38
- Zufall 38
- Trainingsdaten 23
- Trainingsdurchgänge 37
- True 101
- Tupel 110, 112

- U**
- Überanpassung 51
- Unsicherheit 161

- V**
- Variable 19, 98
 - anlegen 19, 40, 56
 - für alle Figuren 20
 - globale 115
 - kontrollieren 20
 - nur für diese Figur 20
- Vereinigungsmenge 113

- Verlässlichkeit 43
- Versicolor 78
- Versuch und Irrtum 161
- Vertrauen 40
- Virginica 78

- W**
- Wahr 101
- Wahrheitswert 101
- Webcam 54, 59
- while-Schleife 100
- while-Statement 100
- Wurzelfunktion 108

- X**
- XOR 113

- Z**
- Zahlen 107
 - Typ 107
- Zeichenkette 98, 104
- Zufallszahl 117
- Zuverlässigkeit 23