



## Online Coding – Roboter „Dash“

# Agenda

- Vorstellungsrunde und Vorerfahrungen
- Workshopziele
- Coding?
- Didaktischer Sinn!?
- Warum Roboter?
- **Arbeitsphase**
- Reflexion
- Noch Fragen?

# Vorstellungsrunde



Moin!

# Vorerfahrungen



# Workshopziele

Heute werden Sie ...

- Dash und seine Funktionen kennengelernt haben.
- eine Steuerungsmöglichkeit für den Dash ausprobiert haben.
- selbst Code entwickelt haben
- Ideen kennenlernen, wie man solche Roboter in den Unterricht/ SU integrieren kann.

# Algorithmen?

# Was sind Algorithmen?

- Eine Reihe von Angaben zur Ausführung einer Aufgabe, wie ein Rezept.
- Beispiele:

Prozess	Ausführender	Algorithmus	Typische Anweisung
Kuchenbacken	Bäcker	Rezept	500g Mehl, 200 g Butter, 3 Eier, ... Vermische zuerst die...
Spielen einer Melodie	Musiker	Tonfolge	
Anruf tätigen	Anrufer	Telefonnummer	01733225677

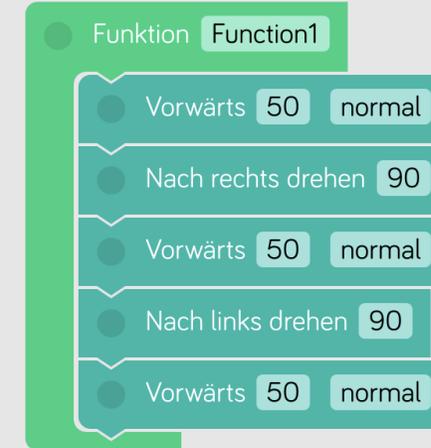
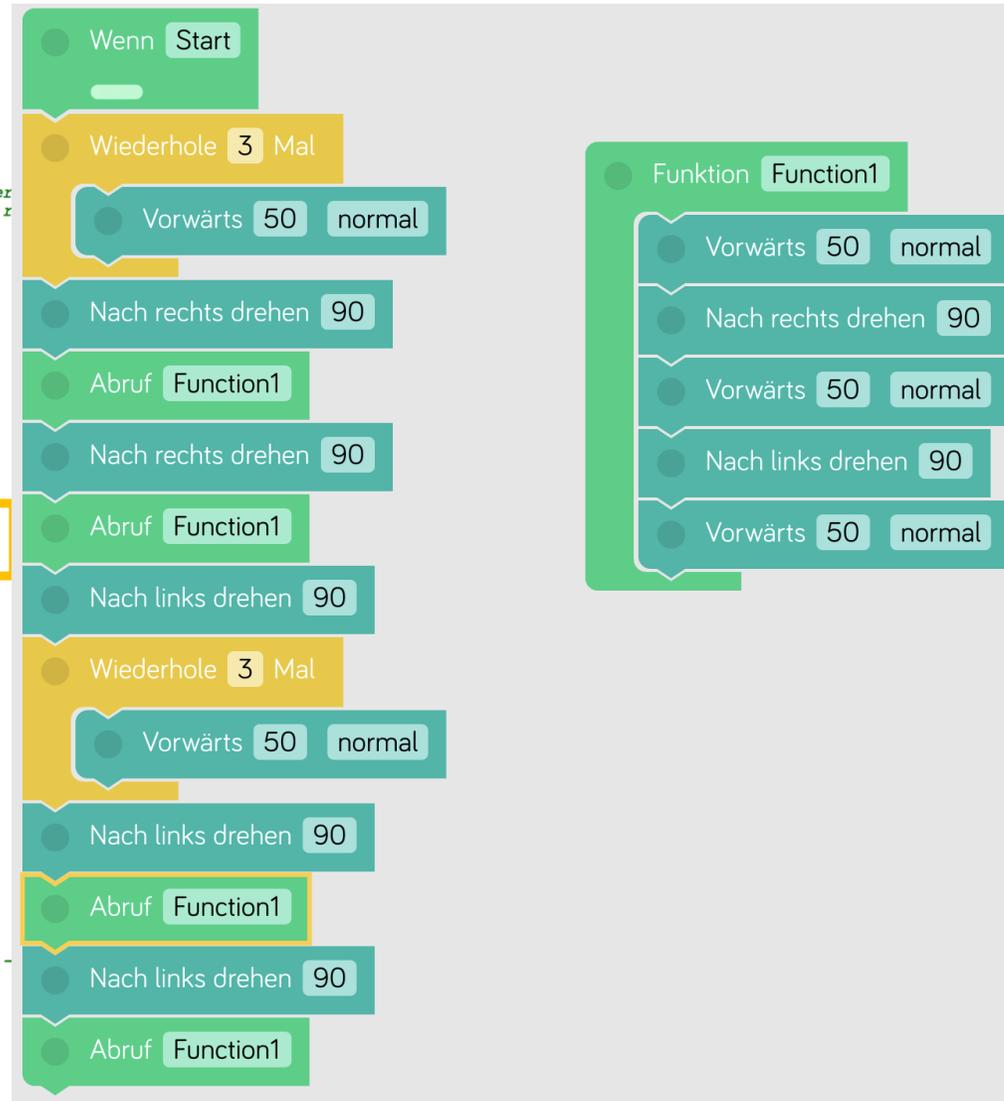
# Was ist Coding?

- Computer programming, the process of designing, writing, testing, debugging / troubleshooting, thinking, and maintaining the source code of computer programs.  
(wikipedia.com)
- Coding is what makes it possible for us to create computer software, apps and websites. Your browser, your OS, the apps on your phone, Facebook, and this website – they're all made with code. (codeconquest.com)
- Coding bzw. Programmieren bedeutet durch Eingabe eines Codes dem Computer bzw. Roboter Befehle zu geben und auf diese Weise Spiele, Programme und Webseiten zu erstellen oder Roboter zu steuern.  
(medienkindergarten.wien)

# Was ist Coding?

```
1 <!DOCTYPE html>
2 <html lang="de">
3 <head>
4
5 <meta charset="utf-8">
6 <!--
7   This website is powered by TYPO3 - inspiring people to share!
8   TYPO3 is a free open source Content Management Framework initially created by Kasper
9   TYPO3 is copyright 1998-2018 of Kasper Skaarhoj. Extensions are copyright of their r
10  Information and contribution at https://typo3.org/
11 -->
12
13 <base href="https://nawitas.uni-koeln.de/">
14
15
16 <meta name="gener:
17
18
19 <link rel="styles!
20 <link rel="styles!
21 <link rel="styles!
22 <link rel="styles!
23 <link rel="styles!
24 <link rel="styles!
25 <link rel="styles!
26 <link rel="styles!
27 <link rel="styles!
28 <link rel="styles!
29 <link rel="styles!
30 <link rel="styles!
31 <link rel="styles!
32 <link rel="styles!
33
34
35
36
37
38
39 <title> Startseite
40 <link rel="sty
41 <meta http-equiv= x-UA-Compatible content= IE=edge /
42 <meta name="viewport" content="width=device-width, initial-scale=1">
43 <meta name="robots" content="noindex, nofollow">
44
45 <link rel="shortcut icon" href="fileadmin/images/favicon.ico" type="image/x-icon">
46 <link rel="icon" href="fileadmin/images/favicon.ico" type="image/x-icon">
47
48 <!-- IE
49 ===== -->
50 <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries
51 <!--[if lt IE 9]>
52 <script src="fileadmin/templates/js/html5shiv.min.js"></script>
53 <script src="fileadmin/templates/js/respond.min.js"></script>
54 <![endif]-->
55 </head>
56 <body id="p2">
57 <div class="boxed-layout">
58
59
```

print "Hello"



Code Beispiel aus HTML



# Didaktischer Sinn!?

1. BEDIENEN UND ANWENDEN 	2. INFORMIEREN UND RECHERCHIEREN 	3. KOMMUNIZIEREN UND KOOPERIEREN 	4. PRODUZIEREN UND PRÄSENTIEREN 	5. ANALYSIEREN UND REFLEKTIEREN 	6. PROBLEMLÖSEN UND MODELLIEREN 
<b>1.1 Medienausstattung (Hardware)</b> Medienausstattung (Hardware) kennen, auswählen und reflektiert anwenden; mit dieser verantwortungsvoll umgehen	<b>2.1 Informationsrecherche</b> Informationsrecherchen zielgerichtet durchführen und dabei Suchstrategien anwenden	<b>3.1 Kommunikations- und Kooperationsprozesse</b> Kommunikations- und Kooperationsprozesse mit digitalen Werkzeugen zielgerichtet gestalten sowie mediale Produkte und Informationen teilen	<b>4.1 Medienproduktion und Präsentation</b> Medienprodukte adressatengerecht planen, gestalten und präsentieren; Möglichkeiten des Veröffentlichens und Teilens kennen und nutzen	<b>5.1 Medienanalyse</b> Die Vielfalt der Medien, ihre Entwicklung und Bedeutungen kennen, analysieren und reflektieren	<b>6.1 Prinzipien der digitalen Welt</b> Grundlegende Prinzipien und Funktionsweisen der digitalen Welt identifizieren, kennen, verstehen und bewusst nutzen
<b>1.2 Digitale Werkzeuge</b> Verschiedene digitale Werkzeuge und deren Funktionsumfang kennen, auswählen sowie diese kreativ, reflektiert und zielgerichtet einsetzen	<b>2.2 Informationsauswertung</b> Themenrelevante Informationen und Daten aus Medienangeboten filtern, strukturieren, umwandeln und aufbereiten	<b>3.2 Kommunikations- und Kooperationsregeln</b> Regeln für digitale Kommunikation und Kooperation kennen, formulieren und einhalten	<b>4.2 Gestaltungsmittel</b> Gestaltungsmittel von Medienprodukten kennen, reflektiert anwenden sowie hinsichtlich ihrer Qualität, Wirkung und Aussageabsicht beurteilen	<b>5.2 Meinungsbildung</b> Die interessengeleitete Setzung und Verbreitung von Themen in Medien erkennen sowie in Bezug auf die Meinungsbildung beurteilen	<b>6.2 Algorithmen erkennen</b> Algorithmische Muster und Strukturen in verschiedenen Kontexten erkennen, nachvollziehen und reflektieren
<b>1.3 Datenorganisation</b> Informationen und Daten sicher speichern, wiederfinden und von verschiedenen Orten abrufen; Informationen und Daten zusammenfassen, organisieren und strukturiert aufbewahren	<b>2.3 Informationsbewertung</b> Informationen, Daten und ihre Quellen sowie dahinterliegende Strategien und Absichten erkennen und kritisch bewerten	<b>3.3 Kommunikation und Kooperation in der Gesellschaft</b> Kommunikations- und Kooperationsprozesse im Sinne einer aktiven Teilhabe an der Gesellschaft gestalten und reflektieren; ethische Grundsätze sowie kulturell-gesellschaftliche Normen beachten	<b>4.3 Quelldokumentation</b> Standards der Quellenangaben beim Produzieren und Präsentieren von eigenen und fremden Inhalten kennen und anwenden	<b>5.3 Identitätsbildung</b> Chancen und Herausforderungen von Medien für die Realitätswahrnehmung erkennen und analysieren sowie für die eigene Identitätsbildung nutzen	<b>6.3 Modellieren und Programmieren</b> Probleme formalisiert beschreiben, Problemlösestrategien entwickeln und dazu eine strukturierte, algorithmische Sequenz planen, diese auch durch Programmieren umsetzen und die gefundene Lösungsstrategie beurteilen
<b>1.4 Datenschutz und Informationssicherheit</b> Verantwortungsvoll mit persönlichen und fremden Daten umgehen, Datenschutz, Privatsphäre und Informationssicherheit beachten	<b>2.4 Informationskritik</b> Unangemessene und gefährdende Medieninhalte erkennen und hinsichtlich rechtlicher Grundlagen sowie gesellschaftlicher Normen und Werte einschätzen; Jugend- und Verbraucherschutz kennen und Hilfs- und Unterstützungsstrukturen nutzen	<b>3.4 Cybergewalt und -kriminalität</b> Persönliche, gesellschaftliche und wirtschaftliche Risiken und Auswirkungen von Cybergewalt und -kriminalität erkennen sowie Ansprechpartner und Reaktionsmöglichkeiten kennen und nutzen	<b>4.4 Rechtliche Grundlagen</b> Rechtliche Grundlagen des Persönlichkeits- (u.a. des Bildrechts), Urheber- und Nutzungsrechts (u.a. Lizenzen) überprüfen, bewerten und beachten	<b>5.4 Selbstregulierte Mediennutzung</b> Medien und ihre Wirkungen beschreiben, kritisch reflektieren und deren Nutzung selbstverantwortlich regulieren; andere bei ihrer Mediennutzung unterstützen	<b>6.4 Bedeutung von Algorithmen</b> Einflüsse von Algorithmen und Auswirkung der Automatisierung von Prozessen in der digitalen Welt beschreiben und reflektieren



# Didaktischer Sinn!?

- (neuer) Lehrplan
  - Technik, digitale Technologie und Arbeit
    - Bauen und Konstruieren (Ende Klasse 4)
      - Die Schülerinnen und Schüler programmieren eine Sequenz.
      - (Die Schülerinnen und Schüler simulieren und beschreiben das EVA-Prinzip (Eingabe, Verarbeitung, Ausgabe) als Grundprinzip der Datenverarbeitung in Informatiksystemen anhand eines Beispiels)

# Warum Roboter?!

- Motivation
- Haptische Erfahrungen
- Lebensweltbezug (bewusst/unbewusst)
  - Roboter in Filmen/Serien/Games → R2D2, BB8, Wall-E, Ratchet & Clank ...
  - Staubsauger
  - Rasenmäher
  - ...
- Direkte Verknüpfung zwischen Code und Codeausführung (weniger Black Boxes)
- Nachvollziehbarer Zusammenhang zwischen Code und Codeausführung
- ...

# Warum der Dash?!

- Funktioniert „out of the box“
- robust
- Vielfältig einsetzbar
- modifizierbar
- Nutzung verschiedener Syntax möglich
  
- Kritik:
  - Sensoren



# Alternativen?

- Der Dash ist ein Platzhalter
- mBot
- Ozobot
- Lego Spike
- Welche kennen Sie?
- ...



# Coding mit Dash

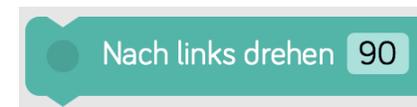
- Anweisungen („Einfache Befehle“)
- Funktionen
- Schleifen
- Variable
- (Bedingte Anweisung und Verzweigungen)
- Debugging

# Coding – Befehle (Anweisung)

- Anweisung an den Computer, die vom Computer umgesetzt werden kann
- werden vom Computer (meist) nacheinander ausgeführt
- Einfach Anweisungen (Arbeitsdefinition)
  - Befehle/Anweisungen, die direkt vom Computer verstanden und umgesetzt werden können (abhängig vom eingesetzten Bildungsmedium). Sie stellen die grundlegenden Steuerungsmöglichkeiten dar.

Beispiele für den Dash:

- „vorwärts bewegen“
- „rechts drehen“
- „links drehen“

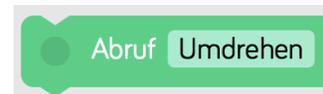
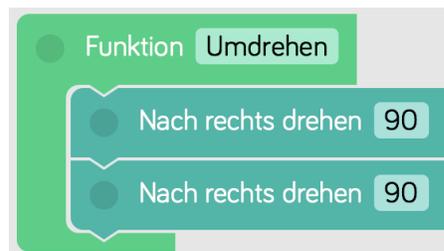


# Coding - Funktionen

- Benannte, wiederverwendbare Reihe von Programmieranweisungen, die eine bestimmte Aufgabe ausführen.
- So können Programme kürzer und übersichtlicher gestaltet werden
- Durch Kombination bekannter Befehle können eigene Befehle definiert werden

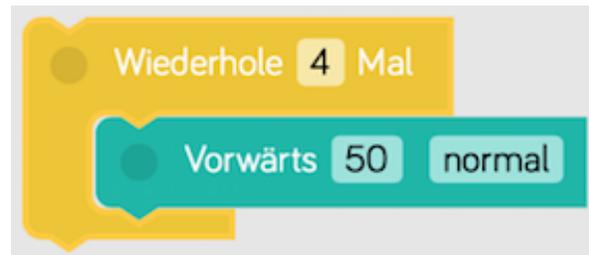
Beispiel:

- Dash soll umdrehen: „Umdrehen“ = fkt. {„drehe rechts“, „drehe rechts“}



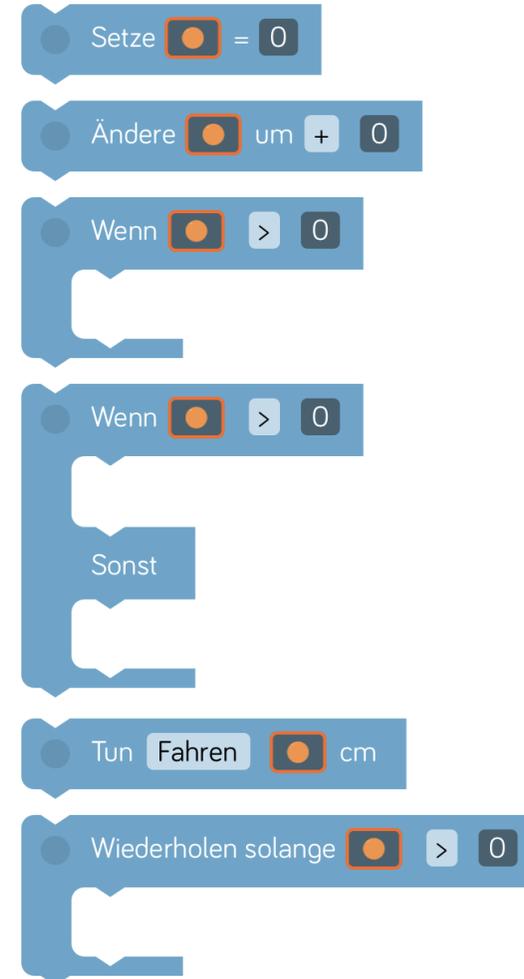
# Coding – (Für-)Schleifen

- Ein Satz von Anweisungen, die wiederholt werden, bis eine Bedingung erfüllt wird.
- So können Programme kürzer und übersichtlicher gestaltet werden
- Beispiel:
  - Wiederholungsschleife:  
„vorwärts bewegen“, „vorwärts bewegen“, „vorwärts bewegen“, „vorwärts bewegen“  
→ wiederhole: „vorwärts bewegen“ 4 mal



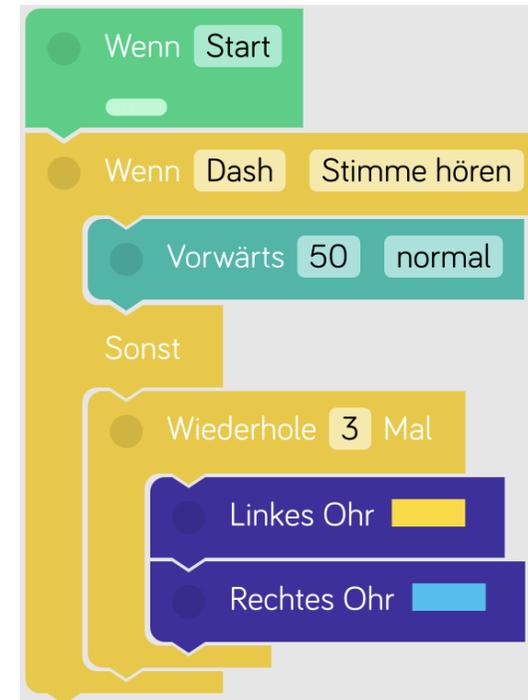
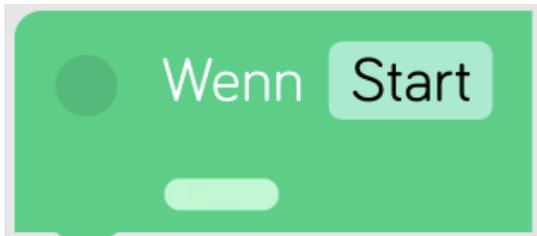
# Coding - Variablen

- In der Programmierung ist eine **Variable** ein abstrakter Behälter für einen Wert, der bei der Ausführung eines Computerprogramm auftritt.
- Der durch eine Variable repräsentierte Wert und gegebenenfalls auch die Größe kann – im Unterschied zu einer Konstante – zur Laufzeit des Rechenprozesses verändert werden.



# Coding – Bedingte Anweisung und Verzweigung

- Eine **Bedingte Anweisung** ist ein Programmabschnitt, der nur unter einer bestimmten Bedingung ausgeführt wird
- Eine **Verzweigung** legt fest, welcher von zwei (oder mehr) Programmabschnitten, abhängig von einer (oder mehreren) Bedingungen, ausgeführt wird.
- Komplexere Programme möglich.
- Beispiel:



# Coding - Debugging

- Prozess des Auffindens und Lösens von Fehlern im Programmiercode/Programmen

# Was macht ein „gutes“ Programm aus?

- Nur die notwendigsten Anweisungen nutzen
- Kein „Schnickschnack“
- Das Problem lösen, nicht mehr (möglichst effizient und übersichtlich)

# Die Aufgaben

- **Parcours** (Zahl = Fahrstrecke des Dash, in Blockly einstellbar 10 – 100)
- **Gruppenaufgabe: Choreographie**
- **Debugging**



# Die „Regeln“

- ...
- Platzproblem
- Abwechseln an den Stationen
- Screenshot des finalen Programms erstellen  
(Ausschalter+Homebutton gleichzeitig drücken)

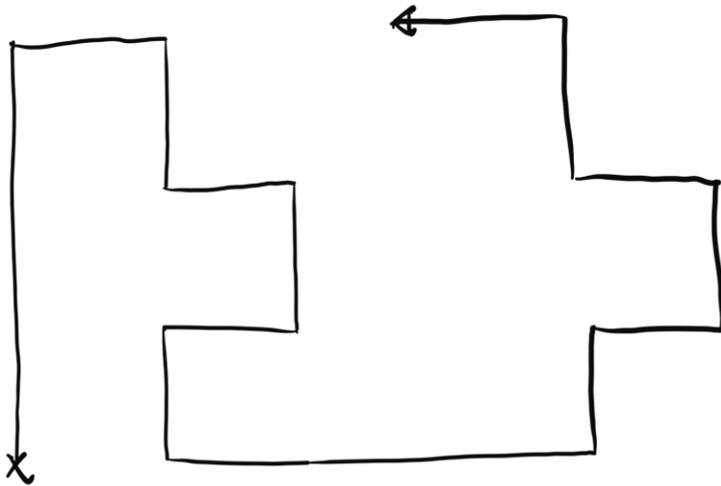
# Dash Blockly - Einführung

- Blockly App öffnen
- Mit „richtigem“ Dash verbinden (die haben Nummern)
- Loslegen  
oder
- An kleiner Blockly Einführung teilnehmen
- ...

# Gruppenaufgabe

- Entwickeln Sie gemeinsam eine Choreographie, die alle Dash mit einschließt!
- „Schnickschnack“ erlaubt!

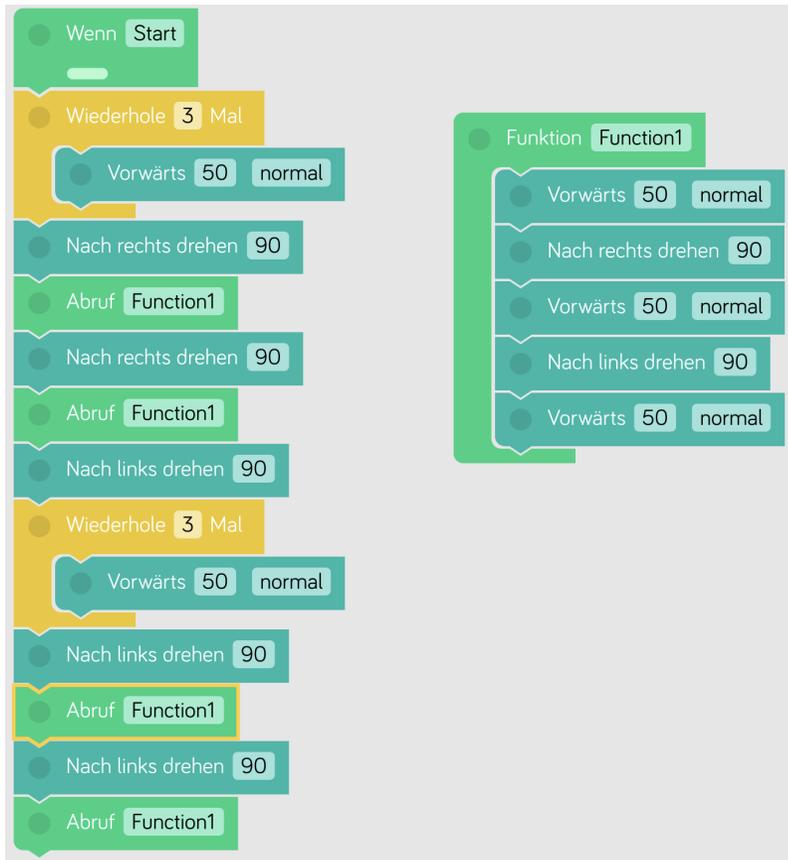
# Debugging



```
Wenn Start
  Wiederhole 3 Mal
    Vorwärts 50 normal
  Abruf Function1
  Nach rechts drehen 90
  Abruf Function1
  Nach links drehen 90
  Wiederhole 3 Mal
    Vorwärts 50 normal
  Abruf Function1
  Nach links drehen 90
  Abruf Function1

Funktion Function1
  Vorwärts 50 normal
  Nach rechts drehen 90
  Vorwärts 50 normal
  Vorwärts 50 normal
```

# Lösungen vergleichen



A Scratch script starting with a 'Wenn Start' block. It contains two 'Wiederhole 3 Mal' loops. The first loop contains 'Vorwärts 50 normal', 'Nach rechts drehen 90', and 'Abruf Function1'. The second loop contains 'Vorwärts 50 normal', 'Nach links drehen 90', 'Abruf Function1', and 'Nach links drehen 90'. A 'Funktion Function1' block is defined with 'Vorwärts 50 normal', 'Nach rechts drehen 90', 'Vorwärts 50 normal', 'Nach links drehen 90', and 'Vorwärts 50 normal'.



A Scratch script starting with a 'Wenn Start' block. It contains a 'Vorwärts 100 normal' block, followed by three 'Abruf Function3' blocks, and another 'Vorwärts 100 normal' block. This is followed by 'Nach links drehen 90', 'Abruf Function4', 'Abruf Function2', and 'Abruf Function4'. A 'Funktion Function2' block is defined with 'Vorwärts 50 normal' and 'Nach links drehen 90'. A 'Funktion Function3' block is defined with three 'Abruf' blocks: 'Abruf Function1', 'Abruf Function1', and 'Abruf Function2'. A 'Funktion Function4' block is defined with 'Abruf Function1' and 'Abruf Function2'. The script ends with 'Tanzen Dash Albern' and 'Tier Dash Löwe'.



# Weiterführendes

- Messen
- Winkel
- Raumorientierung
- Storytelling (Verkleiden, Theater)
- „Zeichnen“
- Zusätze nutzen (Katapult, Greifarm, „Lego-Modul“)
  
- Inspiration z.B. bei Pinterest