

# Algorithmisches Zeichnen

Warm-up

 Art:  
**Warm-up**

 Level:  
**Anfänger\*innen**

 Alter:  
**Ab 10 Jahren**

 Kategorie:  
**Programmierung**

## Kurzbeschreibung

Durch diese Übung wird verdeutlicht, dass Handlungsanweisungen sehr spezifisch sein müssen, um das gleiche Ergebnis zu erzielen. Das Spiel kann frontal oder in Zweierteams gespielt werden. Eine Person entwirft zunächst eine Form und versucht dann durch genaue Anweisungen seine Teampartner\*in so anzuleiten, dass diese\*r die gleiche Form zeichnen kann.

Diese Übung eignet sich auch hervorragend als methodisches Training zur Bearbeitung von Aufgaben. Um die Aufgaben richtig lösen zu können, ist es notwendig, zunächst alle Arbeitsanweisungen zu lesen, bevor die Befehlsfolge bzw. der Algorithmus richtig umgesetzt werden kann.

## Lernziele

- Ein Verständnis für algorithmische Strukturen aufbauen
- Voraussetzungen für erfolgreiches Programmieren kennenlernen

## Benötigtes Material

- Buntstifte oder Filzstifte
- Papier
- Radiergummi
- Anspitzer
- Ausdrucke der Vorlagenblätter als Anschauungsmaterial (optional)

## Spielvorbereitung

Die Durchführung des Spiels kann sowohl eigenverantwortlich in Zweiergruppen oder durch eine übergeordnete Spielleitung gesteuert werden.

### Schritt 1: Aufteilung der Kinder

Wird die Arbeit in Kleingruppen bevorzugt, so sollten die Kinder zunächst in die jeweilige Gruppenstärke (am besten Zweier- oder Dreierteams) aufgeteilt werden.

### Schritt 2: Rollenverteilung

Die Kinder nehmen die Rollen Navigator\*innen und Zeichner\*innen ein und schauen sich ihre entsprechenden Aufgaben an.

#### Rolle Navigator\*in (1 Kind)

Deine Aufgabe ist es, den Überblick zu behalten und die Befehlsfolge vorzulesen.

#### Rolle Zeichner\*innen (min. 1 Kind)

Du bist für die praktische Umsetzung zuständig und führst die Befehlsfolge aus.

## Spielanleitung

### Ziel des Spiels:

Algorithmen sind Abfolgen von Anweisungen. In diesem Spiel sollen die Kinder anhand von Befehlsabfolgen unterschiedliche Formen malen. Danach werden die jeweiligen Befehlsfolgen und die daraus resultierenden Ergebnisse verglichen und besprochen.

### Schritt 1:

Jede\*r Navigator\*in überlegt sich eine Befehlsabfolge, liest diese zunächst in Gänze vor und überprüft, ob jede\*r Zeichner\*in die Aufgabenstellung verstanden hat. Gibt es Fragen zu den Anweisungen, werden diese geklärt. Die Anweisungen können auch noch einmal einzeln vorgelesen werden, wenn die anderen anfangen zu zeichnen. Es sollte aber in jedem Fall die ganze Befehlsfolge bereits bekannt sein.

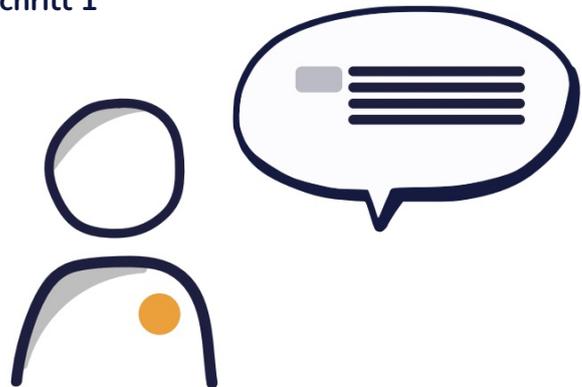
### Schritt 2:

Jede\*r Zeichner\*in führt die Befehlsfolge aus und malt das nach ihrem Verständnis richtige Bild.

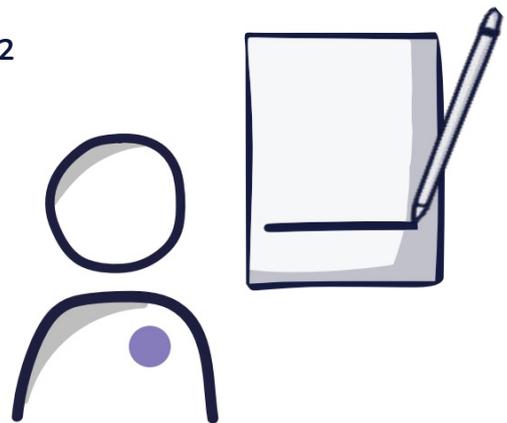
### Schritt 3:

Haben alle Teammitglieder ein Ergebnis gemalt, werden die Rollen neu verteilt. Achtet darauf, dass am Ende des Spiels jedes Kind aus der Gruppe einmal navigieren durfte.

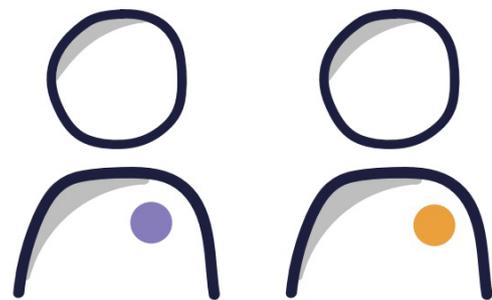
#### Schritt 1



#### Schritt 2



#### Schritt 3



So könnte eine erdachte Befehlsfolge und die daraus entstandenen Zeichnungen aussehen. Ein Bild scheint irgendwie nicht zu passen. Könnt ihr herausfinden, welches es ist?

Schritt 1: **Zeichnet ein Viereck.**

Schritt 2: **Zeichnet Linien, die sich nicht berühren.**

Schritt 3: **Benutzt für die Linien eine Farbe.**



## Beispiele für einen Mal-Algorithmus

Diese Beispiele könnt ihr nutzen, um den Einstieg in das Spiel zu erleichtern. Anhand der folgenden vier Befehlsabfolgen sind vier Formen entstanden. Welche Anweisungsabfolge passt zu welcher Form?

**Aufgabe 1:** Lest euch vier Befehlsabfolgen durch und verbinde sie mit den dazu passenden Bildern.

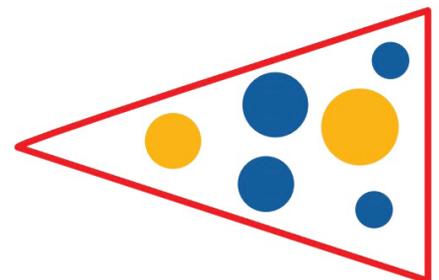
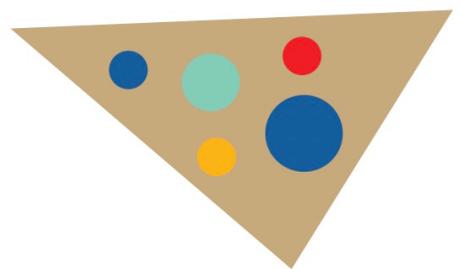
Schritt 1: **Zeichnet ein Viereck.**  
Schritt 2: **Zeichnet Linien, die sich nicht berühren.**  
Schritt 3: **Benutzt für die Linien eine Farbe.**

Schritt 1: **Zeichnet ein Dreieck.**  
Schritt 2: **Zeichnet Kreise in drei Größen hinein.**  
Schritt 3: **Benutzt für die Kreise zwei Farben.**

Schritt 1: **Zeichnet einen Kreis.**  
Schritt 2: **Zeichnet einzelne Linien hinein.**  
Schritt 3: **Benutzt für die Linien mehrere Farben.**

Schritt 1: **Zeichnet ein Dreieck.**  
Schritt 2: **Zeichnet Kreise hinein.**  
Schritt 3: **Benutzt für die Kreise vier Farben.**

**Aufgabe 2:** Erfindet eigene Befehlsabfolgen und lest sie eurer Gruppe vor. Es können auch mehr als drei Schritte sein.



## Auswertung / Reflexion

Wie bereits eingangs beschrieben, sollte der Reflexionsphase im Anschluss als wichtiges Element zur Aufarbeitung des Geschehens ein wenig Zeit eingeräumt werden.

Neben Fragen wie:

- *Wie hat es in eurer Gruppe geklappt?*
- *Warum hat es gut/schlecht funktioniert?*

die sich vor allem mit der Thematik der Funktionalität und Arbeitsweise der Gruppen befassen, sollte man vor allem folgender Frage nachgehen:

- *Was hat das Spiel mit Programmierung zu tun?*

Nun ein ganze Menge, auch wenn es zunächst vielleicht nicht so aussehen mag. Doch um dieser Frage eine Antwort geben zu können, sollten wir vielleicht zunächst noch einmal ein paar grundsätzliche Dinge klären.

### Was ist ein Algorithmus & woraus besteht er?

Wie bereits beschrieben, definiert der Begriff eine Folge von Anweisungen mit denen ein bestimmtes Problem gelöst werden kann. Dieser Definition entspricht so ziemlich jedes Kochrezept oder jede Bauanleitung, wenn man so möchte.

In unserem Fall liegt das Problem darin, dass ein Teil der jeweiligen Gruppe ein Bild malen soll. Doch wie soll das Bild gemalt werden? Es müssen also Arbeitsanweisungen bzw. Befehlsabfolgen formuliert werden, damit dieses Problem gelöst werden kann. Betrachtet man nun die Ergebnisse und bespricht diese mit den Kindern, sollten zwei grundsätzliche Erkenntnisse entdeckt werden können.

**Erkenntnis 1:** Es gibt mehr als nur eine Lösung.

**Erkenntnis 2:** Je eindeutiger die Anweisungen bzw. die Befehlsabfolgen formuliert werden, desto einheitlicher werden die Bilder. Man spricht hierbei von „wohldefinierten“ Algorithmen.



Wenn die Kinder im Vorfeld das *Farbcode* oder *Decodierungsspiel* gemacht haben, sollten sie nun die grundsätzlichen Prinzipien einer Kommunikation zwischen Mensch und Maschine verstehen können.

**Prinzip 1:** Computer und Maschinen benötigen eindeutige Befehle, um handeln zu können.

**Prinzip 2:** Jeder Befehl ist mit einer bestimmten, klar definierten Bedeutung versehen.

**Prinzip 3:** Ein Programm besteht aus aneinandergereihten Befehlen. Diese Befehlsfolgen heißen Algorithmen.

### Ihr wollt mehr über Algorithmen lernen?

Als Ergänzung zum Thema Algorithmus bietet sich auch unser *Zahnputzspiel* an. Es finden sich noch jede Menge weitere Beispiele für Algorithmen in unserem Alltag. Macht euch auf die Suche nach ihnen!

Zur weiteren Vertiefung könnt ihr auch unsere Materialien zu [Scratch Jr](#) oder [Scratch](#) auf unserer Lernplattform [digital.tueftellab.de](https://digital.tueftellab.de) anschauen.