



# Differenzierung Informatik – Übersicht

## Ziele?

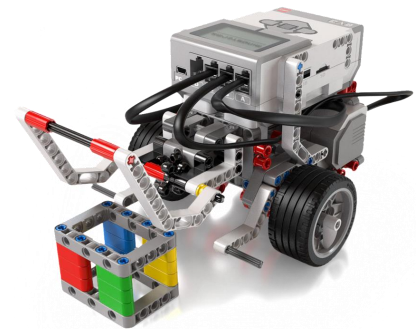
- Vertiefung der Kompetenzen aus der Unterstufe
- Spaß an informatischen Themen

## Voraussetzungen?

- quasi keine
- gewisse Fähigkeit und Bereitschaft zu logisch-abstraktem Denken

## Unterrichtspraxis?

- abwechslungsreicher Mix an Sozial- und Arbeitsformen
- Schwerpunkt: selbständige, projektartige Arbeit, häufig im Team



# Scratch-Spieleprogrammierung

The image displays a Scratch project titled "Super Mario 2 (nutzen)". The game stage shows Mario on a platform with two Paratroopa enemies and several coins. The interface includes a score display for two players (Münzen Spieler 1: 0, Münzen Spieler 2: 0) and a timer (Zeit: 27). The Sprites list contains various assets like Mario, levels, enemies (Anleitung, Paratroopa, Goomba), and coins (Münze).

The Scripts area contains numerous event-driven scripts:

- Initial Setup:** Scripts for moving Mario to specific coordinates (e.g., x: -201, y: 94) and switching costumes to "MarioIr".
- Level Transitions:** Scripts triggered by "when I receive" messages (mz3.2 to mz4.5) that change the backdrop to a new level and reposition Mario.
- Gameplay Logic:** Scripts for handling key presses (right arrow, left arrow, up arrow, down arrow) to move Mario. A "touching platform?" condition is used to trigger a "forever" loop that continuously updates Mario's position based on the platform he is on.
- Game Over:** A script triggered by "when I receive" message "gameover" that broadcasts "gameover" and sets the backdrop to "gameover".
- Collision Detection:** Scripts for "when I receive" messages (bb?) that use a "repeat until" loop to check for collisions with various colors (red, blue, green, yellow, brown) and respond by changing y-position or triggering level transitions.

# Kryptologie von Caesar bis zur Enigma: Sichere Kommunikation



# Digitaler Fußabdruck: Umgang mit eigenen Daten im Netz



**Tik Tok**



# Algorithmen: Kara

The image shows three windows from the Kara programming environment:

- Kara programmieren:** A state machine editor with a diagram showing a 'start' state, a 'middle' state, and a 'stop' state. Below the diagram is a table for defining transitions between states based on sensor inputs.
- Kara, der programmierbare Marienkäfer:** A simulation window showing a 10x10 grid world. The world contains a red ant (Kara), several green trees, and various mushrooms (red and green). The interface includes a toolbar with movement and action icons, a 'Größe' (size) control, a 'Zoom' control, a 'Geschwindigkeit' (speed) slider, and an 'Ausführen' (execute) button.
- Aufgaben:** A task window titled 'Kleeblattsuche im Wald IV (mittel)'. It contains a 10x10 grid representing the world layout with trees, mushrooms, and a green clover. Below the grid is a text description of the task and its origin.

**Kara macht: Nächster Zustand:**

	start	middle	stop	start	middle	stop
yes						
or						
no						
yes						
or						
no						
yes						
or						
no						
yes						
or						
no						
yes						
or						
no						
yes						
or						
no						

**Kleeblattsuche im Wald IV (mittel)**

**Aufgabe** **Weiten** **Lösung**

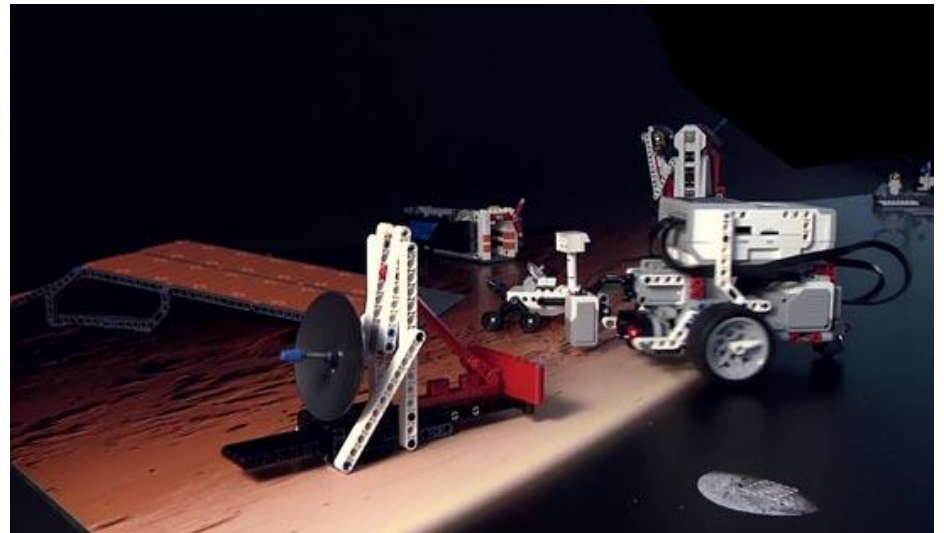
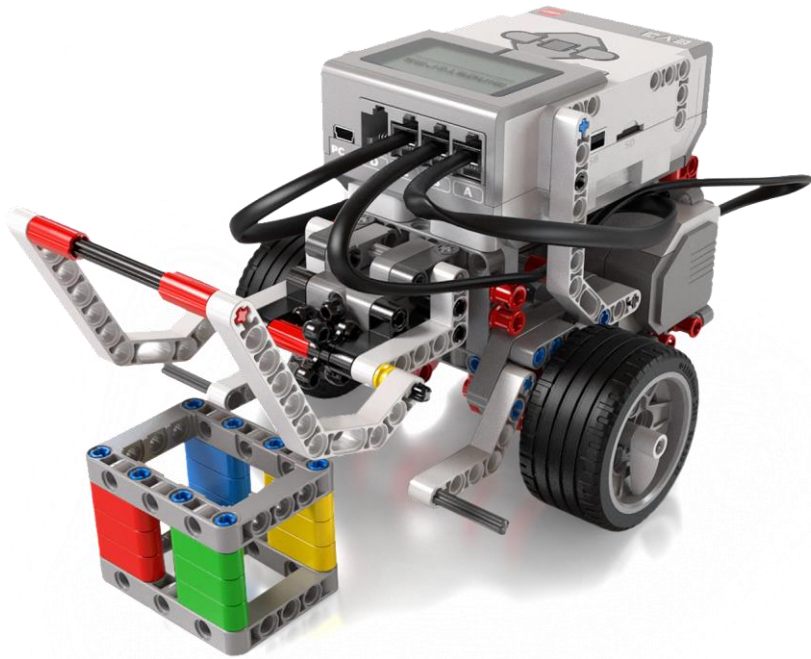
**Kleeblattsuche im Wald IV**

In Karas Welt gibt es Wälder mit Rundgängen, in denen Kara auf Kleeblättersuche geht. Die meisten Felder in einem Rundgang haben genau zwei freie benachbarte Felder. Einige Felder sind durch Pilze blockiert, aber es um jeden Pilz genügend freie Felder, so dass Kara die Pilze problemlos beiseite schieben und um sie herum laufen kann. Zudem sind die Pilze nie so angeordnet, dass Kara neben ihnen zu stehen kommen würde (da Kara keinen Sensor hat, mit dem eine solche Situation entdeckt werden könnte).

Programmieren Sie Kara so, dass es das Kleeblatt findet. Wenn er das Blatt gefunden hat, soll er es aufnehmen und dann stoppen.

Diese Aufgabe stammt von Zhang Hui, National University of Singapore.

# Lego Mindstorms EV3



# Python

Tigerlython

Datei Bearbeiten Ausführen Tools Hilfe

untitled 1\*

```
1 from turtle import*
2
3 def spirale(pSeite, pLinien):
4     repeat pLinien:
5         forward(pSeite)
6         left(90)
7         pSeite += 5
8
9 makeTurtle()
10 spirale(10, 40)
11
```

Tigerlython Turtle Playground