

## Arbeitsauftrag

1. Öffnen Sie die Webseite: <http://inf-schule.de/6.1.3.1>
2. Lesen und Bearbeiten Sie die **Lernstrecke 3: Fallunterscheidung**
3. Bearbeiten Sie zu den jeweiligen Teilabschnitten, die Aufgaben auf dem Arbeitsblatt.



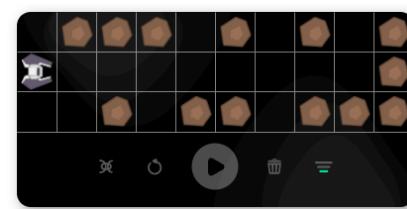
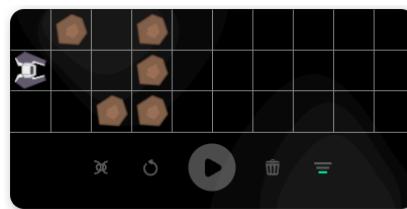
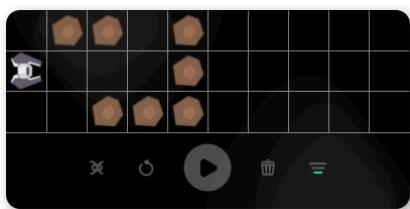
**Hinweis:** Das Dokument „Wissensspeicher“ auf inf-schule.de soll nicht bearbeitet werden.  
Es sind ausschließlich die Aufgaben auf diesem Arbeitsblatt zu bearbeiten.

### Aufgabe a) (Erkundung)

Schreiben Sie ein Programm, dass die Anforderung aus **Aufgabe 3: Das Hauptproblem lösen (Erkundung)** löst.

→ Alle Löcher im Korridor sollen mit PowerUp's verschlossen werden, egal wo sie sich befinden.  
Das Programm soll wieder für jeden beliebigen Korridor funktionieren.

```
while not rockFront():
    move()
    if not rockRight():
        turnRight()
        move()
        putPow()
        turnLeft()
        turnLeft()
        move()
        turnRight()
    if not rockLeft():
        turnLeft()
        move()
        putPow()
        turnRight()
        turnRight()
        move()
        turnLeft()
```



 HELENE-WEBER-SCHULE BUCHEN	<b>Grundlagen der Programmierung</b> <b>Fallunterscheidung (wenn, dann, sonst)</b>	<b>Name:</b> LÖSUNG <b>Datum:</b> <b>Fach:</b> Informatik
--	---	---

## Aufgabe b) (Strukturierung)

Formulieren Sie in eigenen Worten, wozu das Konzept „Fallunterscheidungen“ in der Programmierung genutzt wird.

*Das Konzept der Fallunterscheidungen in der Programmierung wird genutzt, um Entscheidungen basierend auf bestimmten Bedingungen zu treffen. Es ermöglicht dem Programm, unterschiedliche Aktionen auszuführen, abhängig davon, welche Bedingung erfüllt ist. Dadurch kann der Programmfluss gesteuert werden, sodass verschiedene Szenarien oder Fälle unterschiedlich behandelt werden.*

## Aufgabe c) Aufbau einer Fallunterscheidung (Strukturierung)

- I. Lesen Sie sich die Seite „**Strukturierung**“ aufmerksam durch und bearbeiten Sie die dortigen Aufgaben zunächst "im Kopf".
- II. Erklären Sie den Unterschied zwischen **einseitigen Fallunterscheidung** und einer **zweiseitigen Fallunterscheidung**.

### 1. Einseitige Fallunterscheidung:

*Bei einer **einseitigen** Fallunterscheidung wird eine Bedingung geprüft, und nur wenn diese Bedingung erfüllt ist, wird eine Aktion ausgeführt. Falls die Bedingung nicht zutrifft, wird der Code einfach fortgesetzt, ohne dass etwas anderes passiert.*

#### *Beispiel (Pseudocode):*

```
if (Bedingung) {
    // Aktion, wenn die Bedingung wahr ist
}
Wenn die Bedingung nicht wahr ist, passiert nichts.
```

### 2. Zweiseitige Fallunterscheidung:

*Bei einer **zweiseitigen** Fallunterscheidung gibt es zwei mögliche Wege: Wenn die Bedingung erfüllt ist, wird eine Aktion ausgeführt, und wenn die Bedingung nicht erfüllt ist, wird eine alternative Aktion ausgeführt.*

#### *Beispiel (Pseudocode):*

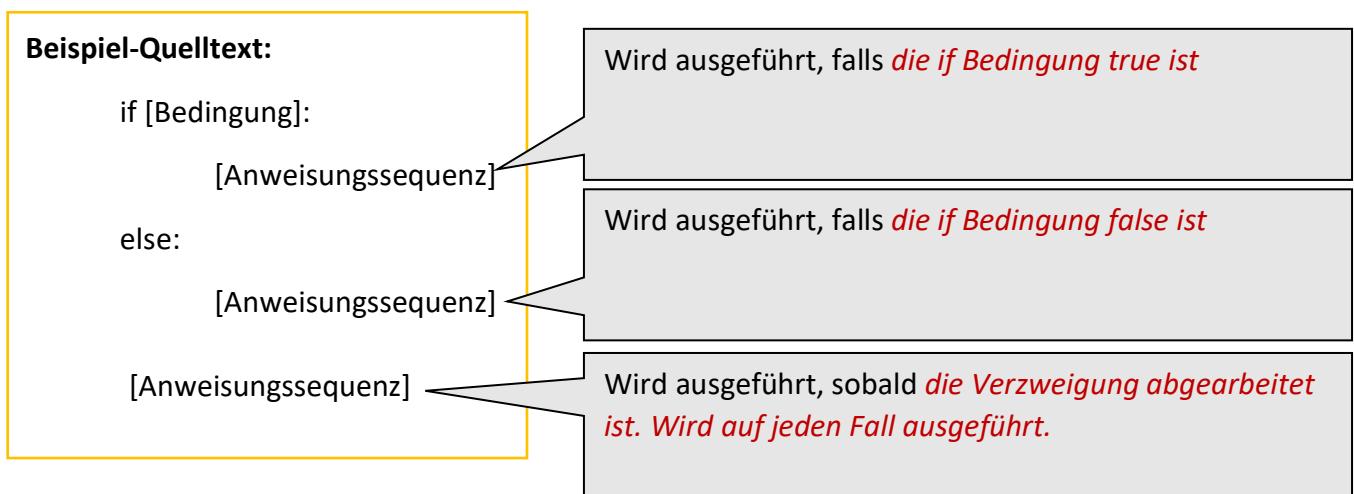
```
if (Bedingung) {
    // Aktion, wenn die Bedingung wahr ist
} else {
    // Alternative Aktion, wenn die Bedingung falsch ist
}
Hier wird also in jedem Fall eine der beiden Aktionen ausgeführt, unabhängig davon, ob die Bedingung wahr oder falsch ist.
```

- III.** Markieren Sie im nachfolgenden Beispiel-Quelltext in Farbe, worauf man unbedingt bei Fallunterscheidungen in Python achten muss, und halten Sie diese als Syntaxregeln fest.

**Syntaxregeln für Fallunterscheidungen in Python:**

1. *Verzweigung beginnt mit dem if-Schlüsselwort, gefolgt von einer Bedingung und einem Doppelpunkt*
2. *Die Anweisungssequenz muss eingerückt werden*
3. *Der else-Fall wird ohne Bedingung formuliert, es folgt auch ein Doppelpunkt sowie die Einrückung der Anweisungssequenz*

- IV.** Beschreibe rechts in den grauen Bereichen, was wann ausgeführt wird.



**Aufgabe d) Logische Operatoren und Wahrheitswerte (Fachkonzept)**

- I.** Lesen Sie sich die Seite „**Fachkonzept**“ aufmerksam durch.

- II. Erklären Sie den Begriff "**Wahrheitswerte**" und wozu **logische Operatoren** genutzt werden.
- Ein Wahrheitswert kann nur einen von zwei möglichen Zuständen haben: **True** (wahr) / **False** (falsch)
  - Logische Operatoren werden genutzt, um Bedingungen oder Ausdrücke zu verknüpfen und ihre Wahrheitswerte zu beeinflussen. In der Programmierung gibt es drei grundlegende logische Operatoren: **AND**, **OR**, **NOT**
- III. Vervollständigen Sie die Tabelle, indem Sie die Werte für **not a**, **a and b** und **a or b** ergänzen.
- IV. Ergänzen Sie außerdem wichtige Erklärungen, um das Verständnis für das Thema weiter zu festigen.

Die drei logischen Grundoperatoren				
a		not a		Erklärung
False		True		Not = Negierung → dreht die Wahrheitswerte um
True		False		
a	b	a and b	a or b	
False	False	False	False	Bei AND müssen immer beide Bedingungen True sein, damit die Bedingung True ergibt.
False	True	False	True	Bei OR muss nur eine Bedingung True ergeben (oder beide) dass die Auswertung True ergibt.
True	False	False	True	
True	True	True	True	

### Aufgabe f) (Übung)

Programmieren Sie die Steuerung der Argo, wie in **Aufgabe 1 bis 4** gefordert.



*Hinweis: Eine Ergebnissicherung auf dem Arbeitsblatt ist nicht notwendig.*

---



**Für die Schnellen!**

### Vertiefungsaufgabe: (**Vertiefung und Vernetzung**)

Bearbeiten Sie auf **inf-schule.de** die Aufgaben 1 bis 4 auf der Seite „**Vertiefung und Vernetzung**“



*Hinweis: Eine Ergebnissicherung auf dem Arbeitsblatt ist nicht notwendig.*