

03 Bedingte Verzweigungen

00 Problemstellung

- Die Gutenbergschule liegt am Hang → Ausrutschgefahr!
- Ziel: Programm, das mich vor Glatteis warnt.
- Gegeben: Aktuelle Temperatur.

Arbeitsauftrag

1. Überlege *zunächst alleine*, wie das Programm deiner Meinung nach ablaufen soll, wenn es die Temperatur des aktuellen Tags als Eingabe erhält.
2. Tauscht euch anschließend zu zweit aus und erklärt euch eure Ideen.



03 Verzweigungen

01 Erarbeitung

Arbeitsauftrag (ca. 15 Minuten)

Lies dir die folgenden Informationen zu *bedingten Anweisungen* durch. Besprecht anschließend jeweils zu zweit den Aufabenteil a (02_aufgaben).

Bedingte Anweisungen

In sogut wie jedem realen Programm kommt es dazu, dass das Programm "entscheiden" soll, wie es weiterarbeiten wird. Wenn beispielsweise die rechte Maustaste gedrückt wird, soll der Computer etwas anderes machen als wenn die linke Maustaste gedrückt wird.

In einer Programmiersprache nennt man das Konzept, dass diese Art von Entscheidungsmöglichkeit bereitstellt eine **Verzweigung**.

Eine Verzweigung legt fest, welcher Programmteil in Abhängigkeit von einer Bedingung ausgeführt wird.

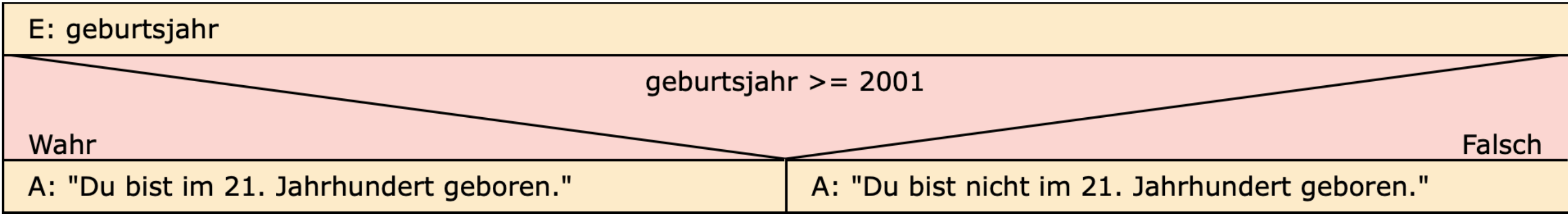
Verzweigungen gehören zu den sogenannten **Kontrollstrukturen**, von denen ihr im Laufe der Reihe noch mehr kennenlernen werdet.



Verzweigungen und Struktogramme

Angenommen, du willst ein Programm schreiben, das dich nach deinem Geburtsjahr fragt und ausgeben soll, ob du im 21. Jahrhundert geboren bist oder nicht. In diesem Fall benötigst du eine **Verzweigung**, die je nach Geburtsjahr unterschiedliche Nachrichten ausgibt.

Ein solches Programm lässt sich gut mithilfe eines Struktogramms beschreiben, das von oben nach unten gelesen wird (**E:** ist eine Eingabe, **A:** ist eine Ausgabe):



Die If-Anweisung

Mithilfe der **if -Anweisung** wird der Programmablauf in Python (und den allermeisten Programmiersprachen) gesteuert.

Die **if** -Anweisung in einem Python-Programm sieht folgendermaßen aus:

```
if bedingung:
    anweisung
    anweisung
    #...weitere Anweisungen, falls nötig
```

Nach dem Schlüsselwort **if** folgt eine Bedingung, gefolgt von einem Doppelpunkt. Nur, wenn die Bedingung wahr ist (also den Wahrheitswert **True** besitzt), werden die eingerückten Anweisungen ausgeführt. Die Anweisungen selbst können auch wieder Kontrollstrukturen, also auch weitere **if** -Anweisungen, sein.

Hier ein Beispiel:

```
In [ ]: # Abfrage des Geburtsjahres
geburtsjahr = int(input("In welchem Jahr bist du geboren?"))

# Ausgabe des Geburtsjahres zur Überprüfung der anschließenden Anweisung
print(f"Geburtsjahr: {geburtsjahr}")

# Überprüfung, ob das Geburtsjahr im 21. Jahrhundert liegt
if geburtsjahr >= 2001:
    print("Du bist im 21. Jahrhundert geboren.")
```

Geburtsjahr: 2005
Du bist im 21. Jahrhundert geboren.

Die if-else-Anweisung

Das Programm soll nun auch eine Nachricht ausgeben, wenn das Geburtsjahr nicht im 21. Jahrhundert liegt. Dies kann mithilfe der Anweisung **else** erreicht werden. Die **else** -Anweisung wird genau dann ausgeführt, wenn die vorherige Bedingung den Wert **False** geliefert hat.

```
In [ ]: # Abfrage des Geburtsjahres
geburtsjahr = int(input("Wann bist du geboren?"))

# Ausgabe des Geburtsjahres zur Überprüfung der anschließenden Anweisung
print(f"Geburtsjahr: {geburtsjahr}")

# Überprüfung, ob das Geburtsjahr im 21. Jahrhundert liegt oder nicht
if geburtsjahr >= 2001:
    print("Du bist im 21. Jahrhundert geboren.")
else:
    print("Du bist nicht im 21. Jahrhundert geboren.")
```

Geburtsjahr: 1978
Du bist nicht im 21. Jahrhundert geboren.

Insgesamt

- **muss** eine bedingte Verzweigung immer genau einen **if** -Zweig zu Beginn und
- **kann** eine bedingte Verzweigung maximal einen **else** -Zweig am Ende enthalten.

03 Bedingte Verzweigungen


02 Aufgaben

Aufgabe

1. Erklärt euch in Partnerarbeit gegenseitig in eigenen Worten die Funktionsweise von Verzweigungen in der Programmierung sowie deren Darstellung im Struktogramm und in Python.

Lösung:

```
<ol type="a", start="2">
```

- Nun soll Programm erstellt werden, das eine Glatteiswarnung ausgibt, wenn es zu kalt ist, und ansonsten angibt, dass keine Gefahr besteht. Stelle den Programmverlauf mithilfe eines Struktogramms grafisch dar. Nutze dafür die Internetseite <https://dditools.inf.tu-dresden.de/dev/struktog/> der TU Dresden. (Du musst zuerst auf den gewünschten Block klicken und dann auf das  -Symbol an der gewünschten Stelle.)

► Tipp 1

► Tipp 2

```
</ol>
```

```
<ol type="a", start="3">
```

- Implementiere ein Programm, das eine Glatteiswarnung ausgibt, wenn es zu kalt ist und ansonsten ausgibt, dass keine Gefahr besteht. </br> (sollte es beim Ausführen von Code in dieser Datei zu Problemen kommen, verwende die Datei *02_wetterwarnung.py* für deine Lösung)

► Tipp 1

► Tipp 2

```
</li> </ol>
```

```
In [ ]: """Aufgabe c"""
```

Zusatzaufgaben für Schnelle

1. Bestimme ohne das Erstellen von Code, was das folgende Programm für die Werte `a = 0` , `a = 5` und `a = 10` ausgibt. Begründe kurz deine Antwort.

```
if a < 5:
    print("Hallo")
else:
    if a < 10:
        print("Bin da")
    else:
        print("Tschau")
```

Lösung:

2. (Für Schnelle) If -Anweisungen können auch innerhalb anderer If -Anweisungen stehen. Dies nennen wir auch **verschachtelte** Verzweigungen. Gib ohne das Erstellen von Code an, was das folgende Programm für die Werte `a = 1` , `a = 4` und `a = 10` ausgibt. Begründe kurz deine Antwort.

```
if (a < 5):
    if (a <= 3):
        print("Ausgabe 1")
    else:
        print("Ausgabe 2")
else:
    if (a < 4):
        print("Ausgabe 3")
    else:
        print("Ausgabe 4")
```

Gib begründet an, für welche `a` "Ausgabe 3" ausgegeben wird.

Lösung:

```
<ol type="a", start="3">
```

- Erweitere das Programm der Haupt-Aufgabe so, dass der User zusätzlich vor zu großer Hitze (ab 34 Grad) gewarnt wird.

► Tipp 1

```
In [ ]: """Aufgabe c"""
```

```
<ol type="a", start="4">
```

- Zusätzlich erhält das Programm nun auch noch die Eingabe, ob es an diesem Tag regnet oder nicht. Implementiere ein Programm, das zusätzlich zum letzten Punkt bei niedrigen Temperaturen und Regen vor Eisregen warnt.

```
</ol>
```

```
In [ ]: """Aufgabe d"""
```

```
<ol type="a", start="5">
```

- Überlege dir weitere Warnungen und/oder Eingaben, die in der Ausgabe berücksichtigt werden und implementiere diese.

```
</ol>
```

```
In [ ]: """Aufgabe e"""
```

03 Bedingte Verzweigungen

03 elif

Die elif-Anweisung

Zurück zum Geburtsjahres-Programm:

In []:

```
# Abfrage des Geburtsjahres
geburtsjahr = int(input("Wann bist du geboren?"))

# Ausgabe des Geburtsjahres zur Überprüfung der anschließenden Anweisung
print(f"Geburtsjahr: {geburtsjahr}")

# Überprüfung, ob das Geburtsjahr im 21. Jahrhundert liegt oder nicht
if geburtsjahr >= 2001:
    print("Du bist im 21. Jahrhundert geboren.")
else:
    print("Du bist nicht im 21. Jahrhundert geboren.")
```

Geburtsjahr: 2001
Du bist im 21. Jahrhundert geboren.

Angenommen, du möchtest dieses Programm um ein Ausgaben für das 20. Jahrhundert erweitern. Dies ist mit mithilfe verschachtelter Verzweigungen möglich:

In []:

```
# Abfrage des Geburtsjahres
geburtsjahr = int(input("Wann bist du geboren?"))

# Ausgabe des Geburtsjahres zur Überprüfung der anschließenden Anweisung
print(f"Geburtsjahr: {geburtsjahr}")

# Überprüfung, ob das Geburtsjahr im 20. oder 21. Jahrhundert liegt oder in keinem von beiden
if geburtsjahr >= 2001:
    print("Du bist im 21. Jahrhundert geboren.")
else:
    if geburtsjahr >= 1901:
        print("Du bist im 20. Jahrhundert geboren.")
    else:
        print("Du bist in keinem der überprüften Jahrhunderte geboren.")
```

Geburtsjahr: 1978
Du bist im 20. Jahrhundert geboren.

Das sieht schon echt kompliziert aus. Wenn noch das 19. Jahrhundert hinzukommt, verliert man schnell den Überblick:

In []:

```
# Abfrage des Geburtsjahres
geburtsjahr = int(input("Wann bist du geboren?"))

# Ausgabe des Geburtsjahres zur Überprüfung der anschließenden Anweisung
print(f"Geburtsjahr: {geburtsjahr}")

# Überprüfung, ob das Geburtsjahr im 19., 20. oder 21. Jahrhundert liegt oder in keinem der genannten
if geburtsjahr >= 2001:
    print("Du bist im 21. Jahrhundert geboren.")
else:
    if geburtsjahr >= 1901:
        print("Du bist im 20. Jahrhundert geboren.")
    else:
        if geburtsjahr >= 1801:
            print("Du bist im 19. Jahrhundert geboren. Ganz schön alt!")
        else:
            print("Du bist in keinem der überprüften Jahrhunderte geboren.")
```

Geburtsjahr: 1899
Du bist im 19. Jahrhundert geboren. Ganz schön alt!

Einfacher ist das mit der `elif`-Verzweigung, die beliebig oft hinter einer `if`-Verzweigung und der `else`-Verzweigung eingesetzt werden kann. Das obige Code kann dann so vereinfacht werden:

In []:

```
# Abfrage des Geburtsjahres
geburtsjahr = int(input("Wann bist du geboren?"))

# Ausgabe des Geburtsjahres zur Überprüfung der anschließenden Anweisung
print(f"Geburtsjahr: {geburtsjahr}")

# Überprüfung, ob das Geburtsjahr im 19., 20. oder 21. Jahrhundert liegt oder in keinem der genannten
if geburtsjahr >= 2001:
    print("Du bist im 21. Jahrhundert geboren.")
elif geburtsjahr >= 1901:
    print("Du bist im 20. Jahrhundert geboren.")
elif geburtsjahr >= 1801:
    print("Du bist im 19. Jahrhundert geboren. Ganz schön alt!")
else:
    print("Du bist in keinem der überprüften Jahrhunderte geboren.")
```

Geburtsjahr: 1899
Du bist im 19. Jahrhundert geboren. Ganz schön alt!

Verwende in den Übungen gerne, wenn benötigt `elif`-Verzweigungen.

03 Bedingte Verzweigungen

04 Übung

Arbeitsauftrag: Sammle mindestens 5 ★ durch die Bearbeitung der folgenden Aufgaben.

Aufgabe 1 (★)

Gib für folgenden Code an, was das Programm für die Eingaben 3, 5, 9 und 10 von a jeweils ausgibt.

```
if (a > 5 and a <= 10):
    print("Hallo")
else:
    print("Tschüss")
```

Lösung:

Aufgabe 2 (★)

Implementiere ein Programm, das dich nach deinem Alter fragt und ausgibt, ob du schon den Führerschein machen darfst oder nicht.

In []:

Aufgabe 3 (★)

Implementiere ein Programm, das die Wassertemperatur als Eingabe erhält und angibt, ob das Wasser fest, flüssig oder gasförmig ist.

In []:

Aufgabe 4 (★ ★)

Ein Jahr ist genau dann ein Schaltjahr, wenn es durch 4 teilbar ist, aber nicht durch 100, es sei denn, es ist durch 400 teilbar. Implementiere ein Programm, das ein Jahr als Eingabe erhält und ausgibt, ob es sich bei diesem Jahr um ein Schaltjahr handelt oder nicht.

- Tipp 1
- Tipp 2

In []:

jahr = int(input("Um welches Jahr handelt es sich?"))

Aufgabe 5 (★ ★)

Schreibe ein Programm, das angibt, ob eine von User eingegebene Zahl durch 2, durch 3 oder durch beide Zahlen teilbar ist.

In []:

zahl = int(input("Gib eine Zahl ein."))

Aufgabe 6 (★ ★)

Implementiere ein Programm, das als Eingabe erhält, wie viel Prozent der Aufgaben ein/e Schüler*in in einer Klausur richtig beantwortet hat und anschließend die Note zurückgibt nach dem folgenden Schlüssel:

- sehr gut: ab 85%
- gut: ab 70%
- befriedigend: ab 55%
- ausreichend: ab 45%
- mangelhaft: ab 20%
- ungenügend: ab 0%

In []:

Aufgabe 7 (★ ★)

Eine Firma hat entschieden, ihren Mitarbeiter*innen einen Bonus auszuzahlen, je nachdem, wie lange sie schon bei der Firma arbeiten.

- Mehr als 10 Jahre: 10% Bonus,
- mindestens 6 Jahre, aber maximal 10 Jahre: 8% Bonus,
- weniger als 6 Jahre, aber mehr als 2 Jahre: 5% Bonus,
- maximal 2 Jahre: kein Bonus.

Implementiere ein Programm, das einen spezifischen Lohn und die Firmenzugehörigkeit erhält und die Höhe des Bonus ausgibt.

In []:

Aufgabe 8 (★ ★ ★)

Schreibe ein Programm, das 3 Zahlen als Eingabe erhält und ausgibt, welche dieser Zahlen die größte ist.

In []:

Aufgabe 9 (★ ★ ★)

Mithilfe des Schlüsselworts `in` lässt sich in Python überprüfen, ob ein Buchstabe in einem String vorkommt oder nicht.

Schreibe ein Programm, das einen Buchstaben als Eingabe erhält und ausgibt, ob es sich dabei um einen Vokal handelt oder nicht (Umlaute (ä, ö, ü, Ä, Ö, Ü) könnt ihr ignorieren).

- Tipp 1
- Tipp 2
- Tipp 3

In []:

Aufgabe 10 (★ ★ ★)

Ein Dreieck ist gleichseitig, wenn alle seine Seitenlängen gleich lang sind und gleichschenkl, wenn genau zwei seiner Seiten gleich lang sind.

Implementiere ein Programm, das die drei Seitenlängen eines Dreiecks als Eingabe erhält und ausgibt, ob das Dreieck gleichseitig, gleichschenkl, keins von beidem ist oder gar kein Dreieck sein kann.

- Tipp

In []:

Aufgabe 11 (★ ★ ★)

Implementiere ein Programm, das zwei Zahlen und einen mathematischen Operator als Eingabe erhält und das Ergebnis der Rechenoperation berechnet.

Beispiel:

```
Erste Zahl: 6
Zweite Zahl: 8
Operator: +
Ausgabe: 14
```

In []:

03 Bedingte Verzweigungen

04 Pbung (Lösung)

Aufgabe 1 (★)

Gib für folgenden Code an, was das Programm für die Eingaben 3, 5, 9 und 10 von a jeweils ausgibt.

```
if (a > 5 and a <= 10):
    print("Hallo")
else:
    print("Tschüss")
```

Lösung:

Fall	Ausgabe
a = 3	Tschüss
a = 5	Tschüss
a = 9	Hallo
a = 10	Hallo

Aufgabe 2 (★)

Implementiere ein Programm, das dich nach deinem Alter fragt und ausgibt, ob du schon den Führerschein machen darfst oder nicht.

```
In [ ]: alter = int(input("Wie alt bist du?"))

if alter >= 18:
    print("Du darfst den Führerschein machen.")
else:
    print("Du bist noch zu jung.")
```

Aufgabe 3 (★)

Implementiere ein Programm, das die Wassertemperatur als Eingabe erhält und angibt, ob das Wasser fest, flüssig oder gasförmig ist.

```
In [ ]: temperatur = float(input("Welche Temperatur hat das Wasser?"))

if temperatur < 0:
    print("Das Wasser ist fest.")
elif temperatur >= 100:
    print("Das Wasser ist gasförmig.")
else:
    print("Das Wasser ist flüssig.")
```

Aufgabe 4 (★ ★)

Ein Jahr ist genau dann ein Schaltjahr, wenn es durch 4 teilbar ist, aber nicht durch 100, es sei denn, es ist durch 400 teilbar. Implementiere ein Programm, das ein Jahr als Eingabe erhält und ausgibt, ob es sich bei diesem Jahr um ein Schaltjahr handelt oder nicht.

- Tipp 1
- Tipp 2

```
In [ ]: jahr = int(input("Um welches Jahr handelt es sich?"))

if jahr % 4 == 0 and (not jahr % 100 == 0 or jahr % 400 == 0):
    print(f"{jahr} ist ein Schaltjahr.")
else:
    print(f"{jahr} ist kein Schaltjahr.")
```

Aufgabe 5 (★ ★)

Schreibe ein Programm, das angibt, ob eine von User eingegebene Zahl durch 2, durch 3 oder durch beide Zahlen teilbar ist.

```
In [ ]: zahl = int(input("Gib eine Zahl ein. "))

if zahl % 2 and zahl % 3:
    print(f"{zahl} ist durch 2 und 3 teilbar.")
elif zahl % 2:
    print(f"{zahl} ist durch 2 aber nicht durch 3 teilbar.")
elif zahl % 3:
    print(f"{zahl} ist durch 3 aber nicht durch 2 teilbar.")
else:
    print(f"{zahl} ist weder durch 2 noch durch 3 teilbar.")
```

Aufgabe 6 (★ ★)

Implementiere ein Programm, das als Eingabe erhält, wie viel Prozent der Aufgaben ein/e Schüler*in in einer Klausur richtig beantwortet hat und anschließend die Note zurückgibt nach dem folgenden Schlüssel:

- sehr gut: ab 85%
- gut: ab 70%
- befriedigend: ab 55%
- ausreichend: ab 45%
- mangelhaft: ab 20%
- ungenügend: ab 0%

```
In [ ]: prozent = int(input("Wie viel Prozent der Aufgaben hast du in der Klausur richtig beantwortet?"))

if prozent >= 85:
    print("sehr gut")
elif prozent >= 70:
    print("gut")
elif prozent >= 55:
    print("befriedigend")
elif prozent >= 45:
    print("ausreichend")
elif prozent >= 20:
    print("mangelhaft")
else:
    print("ungenügend")
```

Aufgabe 7 (★ ★)

Eine Firma hat entschieden, ihren Mitarbeiter*innen einen Bonus auszuzahlen, je nachdem, wie lange sie schon bei der Firma arbeiten.

- Mehr als 10 Jahre: 10% Bonus,
- mindestens 6 Jahre, aber maximal 10 Jahre: 8% Bonus,
- weniger als 6 Jahre, aber mehr als 2 Jahre: 5% Bonus,
- maximal 2 Jahre: kein Bonus.

Implementiere ein Programm, das einen spezifischen Lohn und die Firmenzugehörigkeit erhält und die Höhe des Bonus ausgibt.

```
In [ ]: lohn = float(input("Welchen Lohn erhältst du?"))
jahre_firma = int(input("Wie lange arbeitest du schon bei der Firma?"))
bonus = 0 # Bonus wird in der Verzweigung berechnet

if jahre_firma > 10:
    bonus = lohn * 0.1
elif jahre_firma >= 6:
    bonus = lohn * 0.08
elif jahre_firma > 2:
    bonus = lohn * 0.05

print(f"Dein Bonus beträgt {bonus} Euro.")
```

Aufgabe 8 (★ ★ ★)

Schreibe ein Programm, das 3 Zahlen als Eingabe erhält und ausgibt, welche dieser Zahlen die größte ist.

```
In [ ]: zahl_1 = float(input("Wie lautet die erste Zahl?"))
zahl_2 = float(input("Wie lautet die zweite Zahl?"))
zahl_3 = float(input("Wie lautet die dritte Zahl?"))

if zahl_1 >= zahl_2 and zahl_1 >= zahl_3:
    print(f"{zahl_1} ist die größte der drei Zahlen.")
elif zahl_2 >= zahl_1 and zahl_2 >= zahl_3:
    print(f"{zahl_2} ist die größte der drei Zahlen.")
else:
    print(f"{zahl_3} ist die größte der drei Zahlen.")
```

Aufgabe 9 (★ ★ ★)

Mithilfe des Schlüsselworts `in` lässt sich in Python überprüfen, ob ein Buchstabe in einem String vorkommt oder nicht.

Schreibe ein Programm, das einen Buchstaben als Eingabe erhält und ausgibt, ob es sich dabei um einen Vokal handelt oder nicht (Umlaute (ä, ö, ü, Ä, Ö, Ü) könnt ihr ignorieren).

- Tipp 1
- Tipp 2
- Tipp 3

```
In [ ]: buchstabe = input("Gib einen Buchstaben ein.")
vokale = "aeiouAEIOU"

if buchstabe in vokale:
    print(f"{buchstabe} ist ein Vokal.")
else:
    print(f"{buchstabe} ist kein Vokal.")
```

Aufgabe 10 (★ ★ ★)

Ein Dreieck ist gleichseitig, wenn alle seine Seitenlängen gleich lang sind und gleichschenkl, wenn genau zwei seiner Seiten gleich lang sind.

Implementiere ein Programm, das die drei Seitenlängen eines Dreiecks als Eingabe erhält und ausgibt, ob das Dreieck gleichseitig, gleichschenkl, keins von beidem ist oder gar kein Dreieck sein kann.

- Tipp

```
In [ ]: seite_1 = float(input("Wie lang ist die erste Seite des Dreiecks?"))
seite_2 = float(input("Wie lang ist die zweite Seite des Dreiecks?"))
seite_3 = float(input("Wie lang ist die dritte Seite des Dreiecks?"))

# Ist eine Seite länger als die Summe der beiden anderen Seiten oder gleich lang, kann kein Dreieck gebildet werden.
if (seite_1 + seite_2 <= seite_3) or (seite_1 + seite_3 <= seite_2) or (seite_2 + seite_3 <= seite_1):
    print("Kein Dreieck!")
# Prüfe zuerst auf Gleichseitigkeit, damit die Prüfung auf Gleichschenkligkeit einfacher ist
elif seite_1 == seite_2 == seite_3:
    print("Das Dreieck ist gleichseitig.")
# Da bereits auf Gleichseitigkeit geprüft wurde, reicht der Vergleich zweier Seiten
elif (seite_1 == seite_2) or (seite_1 == seite_3) or (seite_2 == seite_3):
    print("Das Dreieck ist gleichschenkl.")
else:
    print("Das Dreieck ist weder gleichseitig noch gleichschenkl.")
```

Aufgabe 11 (★ ★ ★)

Implementiere ein Programm, das zwei Zahlen und einen mathematischen Operator als Eingabe erhält und das Ergebnis der Rechenoperation berechnet.

Beispiel:

```
Erste Zahl: 6
Zweite Zahl: 8
Operator: +
Ausgabe: 14
```

```
In [ ]: zahl_1 = float(input("Wie lautet die erste Zahl?"))
zahl_2 = float(input("Wie lautet die zweite Zahl?"))
operator = input("Welche Rechenoperation möchtest du anwenden?")

if operator == "+":
    ergebnis = zahl_1 + zahl_2
elif operator == "-":
    ergebnis = zahl_1 - zahl_2
elif operator == "*":
    ergebnis = zahl_1 * zahl_2
elif operator == "/":
    ergebnis = zahl_1 / zahl_2
elif operator == "//":
    ergebnis = zahl_1 // zahl_2
elif operator == "%":
    ergebnis = zahl_1 % zahl_2
elif operator == "**":
    ergebnis = zahl_1 ** zahl_2

print(f"Es gilt {zahl_1} {operator} {zahl_2} = {ergebnis}.")
```