

Gruppe A • Hilfe 1

hilft bei
Aufgabe 1

verwendet

Bisher lief jedes Programm von oben nach unten durch, Zeile für Zeile. Eine Verzweigung ändert das: Vor dem eingerückten Teil steht eine Bedingung, und nur wenn sie zutrifft, wird er ausgeführt. Trifft sie nicht zu, überspringt das Programm ihn und macht darunter weiter.

Tipp:

Lest die `if`-Zeile laut vor, mit dem Wert eingesetzt: "Wenn 18.0 größer als 20 ist, dann ..." Stimmt das? Und was soll das Programm tun, wenn es nicht stimmt?

Gruppe A • Hilfe 2

hilft bei
Aufgabe 2 und 4

verwendet

So ist eine Verzweigung gebaut:

```
if <Bedingung>:  
    <if-Rumpf>  
elif <Bedingung>:  
    <elif-Rumpf>  
else:  
    <else-Rumpf>
```

`elif` und `else` sind freiwillig, Doppelpunkt und Einrückung nicht: ohne Doppelpunkt ein `SyntaxError`, ohne Einrückung ein `IndentationError`.

ID
G2A1



Gruppe A • Hilfe 1

Gruppenpuzzle • Workshoptag 2

ID
G2A2



Gruppe A • Hilfe 2

Gruppenpuzzle • Workshoptag 2

Gruppe B • Hilfe 1

hilft bei
Aufgabe 1 und 3

verwendet

Geht die Schleife von Hand durch und schreibt auf, was in `zaehler` steht:

Durchlauf	<code>zaehler</code> vorher	Bedingung erfüllt?
-----------	-----------------------------	--------------------

- 1.
- 2.
- 3.

Wann wird die Bedingung zum ersten Mal falsch?

Gruppe B • Hilfe 2

hilft bei
Aufgabe 2 und 3

verwendet

Eine `while`-Schleife hält nur an, wenn drei Dinge zusammenpassen:

Wo	Was
vor der Schleife	Startwert setzen
in der <code>while</code> -Zeile	Bedingung prüfen
im Rumpf	den Wert verändern

Fehlt das Verändern, wird die Bedingung nie falsch, die Schleife läuft endlos $\Rightarrow$ Dauerschleife. Stoppen mit dem roten **Stop**-Knopf.

Gruppe B • Hilfe 3

hilft bei
Aufgabe 1

verwendet

```
zaehler = 1

while zaehler <= 5:
    print("Messung Nummer", zaehler)
    zaehler = zaehler + 1
```

`zaehler = zaehler + 1` liest sich von rechts nach links: alten Wert plus eins ausrechnen, Ergebnis wieder unter demselben Namen ablegen.

ID
G2B1



Gruppe B • Hilfe 1

Gruppenpuzzle • Workshoptag 2

ID
G2B2



Gruppe B • Hilfe 2

Gruppenpuzzle • Workshoptag 2

ID
G2B3



Gruppe B • Hilfe 3

Gruppenpuzzle • Workshoptag 2

Gruppe C • Hilfe 1

hilft bei
Aufgabe 1 und 2

verwendet

So ist eine `for`-Schleife gebaut:

```
for <Name> in <Liste>:  
    <Rumpf>
```

`for`, `in` und der Doppelpunkt stehen fest. Den **Namen** sucht ihr euch aus; er gilt nur in der Schleife und trägt bei jedem Durchlauf den nächsten Wert aus der **Liste**. Der eingerückte **Rumpf** läuft einmal je Wert – die Liste bestimmt also, wie oft.

Gruppe C • Hilfe 2

hilft bei
Aufgabe 5

verwendet

So wird die LED angesteuert:

```
import machine, neopixel, time  
  
led = neopixel.NeoPixel(machine.Pin(48), 1)  
  
led[0] = (0, 30, 0)  
led.write()  
time.sleep(0.5)
```

Die drei Zahlen sind Rot, Grün und Blau.

ID
G2C1



Gruppe C • Hilfe 1

Gruppenpuzzle • Workshoptag 2

ID
G2C2



Gruppe C • Hilfe 2

Gruppenpuzzle • Workshoptag 2