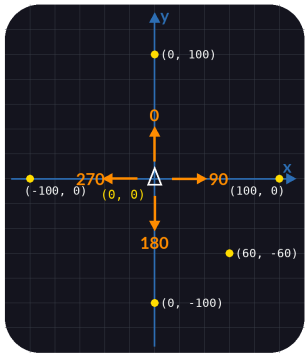
Name: _____ Klasse: _____ Datum: _____

Turtle 2: Die Turtle sieht Farben

> **Ziel:** Ein Turtle-Roboter fährt durch ein Labyrinth. Mit `on_color` erkennt er rote Wände und die grüne Ziellinie, mit `setDirection` und `setPosition` stellst du Richtung und Start ein.

So funktioniert's: Öffne `ab20_start.py` (**Dateien → Vom Computer…**). Eine versteckte Turtle `maurer` zeichnet das Labyrinth aus roten Linien. Die zweite Turtle `robo` fährt los:



ab20_start.py

```
# Der Roboter startet in der Mitte.
robo.setPenColor("yellow")
robo.speed(0)
robo.setDirection(90)
while not robo.on_color("red"):
    robo.forward(1)
```

`robo.on_color("red")` ist `True`, wenn das Pixel **3 Pixel vor** der Turtle genau rot ist. `robo.setDirection(90)` lässt die Turtle nach rechts schauen.

Aufgabe 1 ⭐ Bis zur Wand

Teste das Programm. Wo bleibt der Roboter stehen? _____

Was bedeutet Zeile 41 in Worten? „Solange vor dem Roboter _____“

Was passiert, wenn du in Zeile 42 `forward(1)` durch `forward(10)` ersetzt? Probiere es aus und erkläre. _____

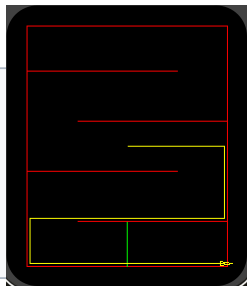
Aufgabe 2 ⭐ Abbiegen

Ergänze unten im Programm: Der Roboter dreht sich an der Wand um 90 Grad nach rechts und fährt bis zur nächsten Wand. Kopiere dazu die Schleife.

In welche Richtung schaut er nach `setDirection(180)`? _____ Und nach `setDirection(270)`? _____

Aufgabe 3 ⭐ Der Weg als Liste

Ersetze Zeile 38 bis 42 durch eine Funktion und eine Liste mit dem Weg:



```
def fahre_bis_wand(t):
    while not t.on_color("red"):
        t.forward(1)

ABBIEGEN = ["rechts", "rechts", "links", "links"]
```

Schreibe eine `for`-Schleife über `ABBIEGEN`: bis zur Wand fahren, dann je nach Eintrag `right(90)` oder `left(90)`. Nach der Schleife fährt der Roboter noch einmal bis zur Wand. Er fährt über die grüne Ziellinie.

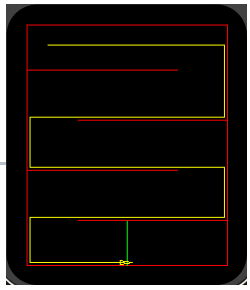
Aufgabe 4 ⭐⭐ Ziel erkannt!

Der Roboter soll auch an der **grünen** Linie anhalten. Ändere die Bedingung der `while`-Schleife in `fahre_bis_wand` mit `and`. Lass die Funktion außerdem die Schritte zählen und mit `return` zurückgeben. Gib am Ende aus: `Ziel erreicht nach … Schritten!`

Wie viele Schritte braucht dein Roboter? _____

Aufgabe 5 ⭐⭐ Neuer Start

Starte den Roboter oben links: `robo.setPosition(-80, 100)`, Blick nach rechts. Wie muss die Liste `ABBIEGEN` jetzt aussehen?



_____

Aufgabe 6 ⭐⭐⭐ Und wieder zurück

Am Ziel dreht sich der Roboter um (`right(180)`) und fährt mit einer neuen Farbe zurück in den Startgang. Schreibe eine Funktion `rueckweg(abbiegen)`, die aus der Liste den Rückweg berechnet. Überlege: In welcher Reihenfolge kommen die Abzweigungen, und was wird aus „rechts“?

Das habe ich gelernt:

- ☐ Ich kann mit `on_color` prüfen, welche Farbe vor der Turtle liegt.
- ☐ Ich kann die Richtung mit `setDirection` einstellen (0 oben, 90 rechts, 180 unten, 270 links).
- ☐ Ich kann eine Turtle mit `setPosition` an eine Stelle springen lassen.
- ☐ Ich kann einen Weg als Liste speichern und mit einer Schleife abfahren.

Mehr dazu im Buch: Kapitel 16 – Turtle-Grafik