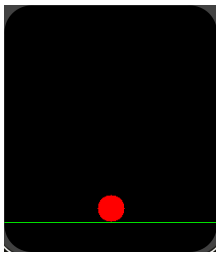
Name: _____ Klasse: _____ Datum: _____

Der springende Ball

Ziel: Du bringst einen Ball zum Springen: Er fällt, prallt auf und steigt wieder, endlos. Dafür brauchst du Schleifen, Pausen und das Übermalen der alten Stelle.

So funktioniert's: Öffne `ab07_start.py` (**Dateien → Vom Computer…**).



```
1 import pbc
2 import time
3
4 pbc.clear()
5 pbc.hline(0, 246, 240, pbc.GREEN)
6 # Der Ball faellt von oben bis auf den Boden
7 for y in range(30, 231, 5):
8     pbc.ellipse(120, y, 15, 15, pbc.RED, fill=True)
9     time.sleep_ms(20)
10    pbc.ellipse(120, y, 15, 15, pbc.BLACK, fill=True)
11 pbc.ellipse(120, 230, 15, 15, pbc.RED, fill=True)
```

Für jedes `y` gilt: **zeichnen – warten – übermalen**. `time.sleep_ms(20)` wartet 20 Millisekunden. Ohne `import time` gibt es einen `NameError`.

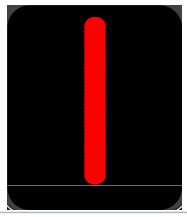
Aufgabe 1 ⭐ Schneller, langsamer

Teste das Programm. Ändere dann jeweils nur eine Zahl und teste wieder.

- `time.sleep_ms(60)` statt `20`: _____
- Schrittweite `10` statt `5` in Zeile 7: _____
- Wie viele Bilder zeigt die Schleife mit Schrittweite 5? _____

Aufgabe 2 ⭐ Ohne Übermalen

Was passiert, wenn du Zeile 10 löschst? Vermute zuerst, teste dann.



_____

Warum malt Zeile 10 den Ball **schwarz**?

_____

Aufgabe 3 ⭐ Hoch und runter, immer wieder

Der Ball soll nach dem Aufprall wieder nach oben steigen und dann endlos weiterspringen. Baue das Programm so um (Zeile 11 brauchst du nicht mehr; achte auf die Einrückung):

```
1 while True:
2     # Fallen
3     for y in range(30, 231, 5):
4         ...    # die drei Zeilen von oben
5     # Steigen
6     for y in range(230, 29, -5):
7         ...    # dieselben drei Zeilen
```

Warum steht im zweiten `range` die 29 und nicht 30? _____ Wie hältst du das Programm an? _____

Aufgabe 4 ⭐⭐ Platsch!

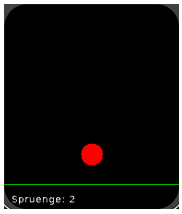
Ein echter Ball wird beim Aufprall kurz flach gedrückt. Zeichne zwischen den beiden `for`-Schleifen eine breite, flache rote Ellipse auf den Boden, z. B. `pbc.ellipse(120, 237, 22, 8, pbc.RED, fill=True)`, warte 80 ms und übermale sie wieder schwarz.

Aufgabe 5 ⭐⭐ Sprünge zählen

Lege vor `while True` eine Variable `spruenge = 0` an. Erhöhe sie bei jedem Aufprall um 1 und zeige sie unten an: erst ein schwarzes Rechteck über die alte Zahl, dann `pbc.text12(f"Spruenge: {spruenge}", 10, 258, pbc.WHITE)`

In welche Zeile deines Programms gehört `spruenge += 1`?

_____



Aufgabe 6 ⭐⭐⭐ Der Ball hüpft weiter

Lass den Ball beim Springen nach rechts wandern. Ersetze `120` durch eine Variable `x` und erhöhe sie in **beiden** Schleifen nach jedem Bild: `x = (x + 2) % pbc.WIDTH`. Was bewirkt `% pbc.WIDTH`?

_____

> Der Ball flackert ein wenig, weil jedes Zeichnen sofort auf dem Display erscheint. Ganz ohne Flackern geht es mit einem Canvas (Buch, Kapitel 17). Einen Ball, der an allen Rändern abprallt, baust du mit `if` (Kapitel 7).

Das habe ich gelernt:

- ☐ Ich kann ein Programm mit `time.sleep_ms` warten lassen.
- ☐ Ich kann eine Figur bewegen: zeichnen – warten – übermalen.
- ☐ Ich kann mit `range` vorwärts und rückwärts zählen.
- ☐ Ich kann mit `while True` etwas endlos wiederholen und es mit Stop anhalten.

Mehr dazu im Buch: Kapitel 6 – Schleifen und Zeit