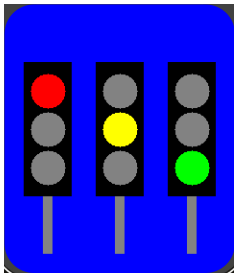
Name: _____ Klasse: _____ Datum: _____

Ordnung mit Funktionen

Ziel: Du räumst ein unübersichtliches Programm auf, indem du wiederkehrende Teile in eine eigene Funktion mit Parametern steckst. **Neu:** `def`, Parameter, Standardwerte



So funktioniert's:

```
1 def gruss(name, mal):
2     for i in range(mal):
3         print("Hallo", name)
4
5
6 gruss("Ada", 2)
7 gruss("Ben", 1)
```

`def` legt eine **Funktion** fest. Die eingerückten Zeilen sind ihr Körper; sie laufen erst beim **Aufruf** (Zeile 6 und 7). `name` und `mal` sind **Parameter**: Beim Aufruf bekommen sie die Werte in den Klammern, der Reihe nach.

Aufgabe 1 ⭐ Das Startprogramm

Öffne `ab13_start.py` (**Dateien → Vom Computer…**) und teste es. Nach `pbc.clear(pbc.BLUE)` kommt Ampel 1:

```
# Ampel 1: rot
pbc.fill_rect(20, 60, 50, 140, pbc.BLACK)
pbc.ellipse(45, 90, 18, 18, pbc.RED, fill=True)
pbc.ellipse(45, 130, 18, 18, pbc.GRAY, fill=True)
pbc.ellipse(45, 170, 18, 18, pbc.GRAY, fill=True)
pbc.fill_rect(40, 200, 10, 60, pbc.GRAY)
```

Danach folgen fast dieselben Zeilen für Ampel 2 und 3. Was ist bei den drei Ampeln **verschieden**?

_____

Aufgabe 2 ⭐ Was gibt das Programm aus?

```
1 def zeige(farbe, zahl):
2     print(zahl, "mal", farbe)
3
4
5 print("Los")
6 zeige("rot", 2)
7 zeige(3, "gelb")
```

Erst denken, dann testen! Ausgabe:

_____

_____

_____

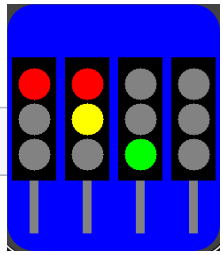
Aufgabe 3 ⭐ Eine Funktion für alle Ampeln

Ergänze die Funktion und ersetze damit die 18 Zeilen für die Ampeln:

```
def ampel(x, oben, mitte, unten):
    pbc.fill_rect(x, 60, 50, 140, pbc.BLACK)
    pbc.ellipse(x + 25, 90, 18, 18, oben, fill=True)
```

Ampel 1 zeichnest du nach `pbc.clear(pbc.BLUE)` mit `ampel(20, pbc.RED, pbc.GRAY, pbc.GRAY)`. Schreibe die Aufrufe für Ampel 2 und 3 dazu und teste. Wie viele Zeilen ist dein Programm jetzt lang? _____

Aufgabe 4 ⭐⭐ Standardwerte



Meistens ist ein Licht aus (grau). Gib den drei Farb-Parametern den Standardwert `pbc.GRAY`:

```
def ampel(x, oben=pbc.GRAY, mitte=pbc.GRAY, unten=pbc.GRAY):
```

Jetzt reicht für eine rote Ampel `ampel(5, oben=pbc.RED)`. Zeichne vier Ampeln: Rot, Rot-Gelb, Grün und eine ausgeschaltete. Nimm für x die Werte 5, 65, 125 und 185.

Wie lautet dein Aufruf für die ausgeschaltete Ampel? _____

Aufgabe 5 ⭐⭐ Die Ampel schaltet

Schreibe eine Funktion `phase(oben, mitte, unten, sekunden)`. Sie ruft `ampel(95, oben, mitte, unten)` auf und wartet dann `sekunden` Sekunden (`import time`!). Lass im Hauptprogramm in einer `while True`-Schleife die vier Phasen Rot (3 s), Rot-Gelb (1 s), Grün (3 s), Gelb (1 s) ablaufen.

Aufgabe 6 ⭐⭐⭐ Echte Lichter

Erweitere `phase` so, dass auch die drei Status-LEDs des PicoBoy mitleuchten (`pbc.led_red(True)` …). Tipp: `oben != pbc.GRAY` ist `True`, wenn das obere Licht an ist.

Das habe ich gelernt:

- ☐ Ich kann mit `def` eine Funktion mit Parametern schreiben und aufrufen.
- ☐ Ich erkenne Wiederholungen in einem Programm und ersetze sie durch Aufrufe.
- ☐ Ich weiß, dass die Argumente der Reihe nach den Parametern zugeordnet werden.
- ☐ Ich kann Parametern Standardwerte geben.

Mehr dazu im Buch: Kapitel 11 – Funktionen