

Strukturierte Programmierung

- ① Gib zu den folgenden Befehlen aus Python die Struktogramm-Elemente an:

Python

```
1
2 1.) forward(100)
3
4 2.) for i in range(6):
5     forward(50)
6     left(60)
7
```

Python

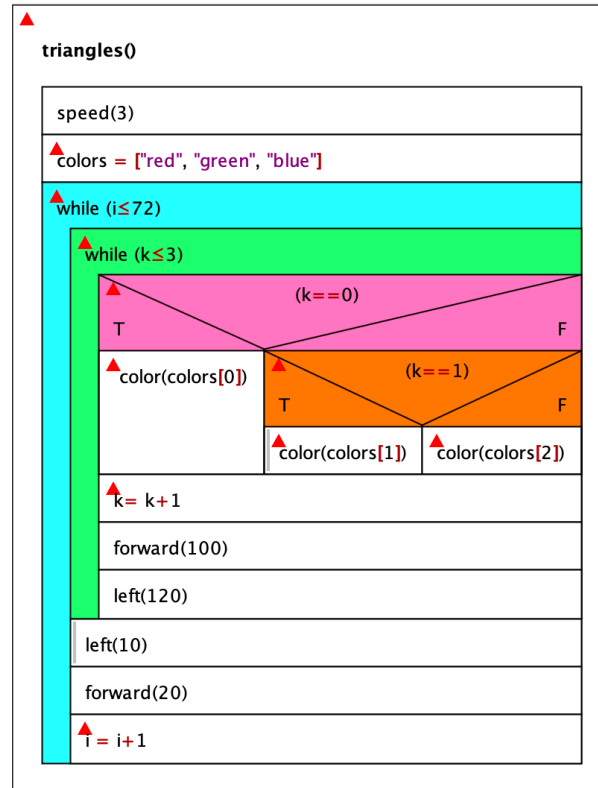
```
1
2 3.) if i == 0:
3     color("blue")
4     else:
5         color("yellow")
6
7 4.) while size < 50:
8     circle(size)
9     size = size + 10
10
```

- ② Erstelle vom Unterprogramm muster2() ein Struktogramm.

Python

```
1 from turtle import *
2
3 def muster2():
4     speed(30)
5     bgcolor("black") # Background color
6     colors = ["blue", "red"] # Alternative color choice
7     for i in range(12):
8         color(colors[i % 2]) # Alternating colors
9         circle(50)
10        right(30)
11    color("yellow")
12    penup()
13    goto(-10, -10)
14    pendown()
15    size = 20
16    for i in range(6):
17        size = 20
18        while size < 50:
19            circle(size)
20            size += 10
21        left(60)
22        penup()
23        goto(0, 0)
24        pendown()
25
26 muster2()
27 done()
28
```

- ③ Erstelle aus dem Struktogramm eine Programm. Denk daran zuerst `from turtle import *` zu schreiben, das Programm mit `def` zu beginnen und und zuletzt `triangles()` aufzurufen.



- ④ Hier gibt es noch je eine Übung.

**Achtung!**

Nur **def boxes()** soll ins Struktogramm:

```

1 from turtle import *
2
3 def boxes():
4     speed(0)
5     bgcolor("white")
6     for y in range(3):
7         for x in range(3):

```

**Achtung!**

Nur **def** soll ins Struktogramm.

```

11         if (x + y) % 2 == 0:
12             color("red")
13         else:
14             color("blue")
15         for _ in range(4):
16             forward(50)
17             right(90)
18
19 boxes()
20 done()

```

