1 Ausgaben im Terminal

Zum Warmwerden: es soll ein Python-Programm geschrieben werden, das mit print()-Kommandos in einer while-Schleife folgende Ausgaben erzeugt:

1.1 eine einzelne Zeile

Das Programm soll zunächst Folgendes ausgeben:

```
.....|.....
```

Hinweis: mit

```
print(".",end="")
```

kann man dafür sorgen, dass am Ende der `print()`-Funktion **kein** Zeilenvorschub ausgegeben wird.

1.2 ganze Terminalbreite ausnutzen

Nun soll die gesamte Breite vom Terminal ausgenutzt werden. Hierzu folgende Import-Anweisung an den Anfang des Programms einfügen:

```
import os
```

Mit dieser Import-Anweisung bekommt man innerhalb des Programms die Terminal-Abmessungen mit den folgenden beiden Ausdrücken:

```
os.get_terminal_size().lines
os.get_terminal_size().columns
```

Damit man nicht jedesmal mit diesen langen Ausdrücken hantieren muss, kann man den Inhalt der beiden Variablen, in zwei Variablen mit kürzeren Namen kopieren. Z.B. so:

```
zeilen_max = os.get_terminal_size().lines
spalten_max = os.get_terminal_size().columns
```

Das Zeichen `|` soll genau in der Mitte, also bei der Hälfte von `spalten_max` ausgeben werden.
Jetzt kann es aber sein, dass `spalten_max` eine ungerade Zahl ist, dann ist `spalten_max/2` eine Zahl mit einer Nachkommastelle. Mit folgendem Ausdruck kann man die Nachkommastelle abschneiden:

```
int(spalten_max/2)
```

Welche Funktion hat man mit folgendem Ausdruck?

```
int(spalten_max/2+0.5)
```

1.3 Feld ausgeben

Nun sollen soviele von den `.....|.....` Zeilen ausgeben werden, dass das Terminal komplett gefüllt wird.
Es sollen dabei keine Zeilen oben aus dem Bildrand verschwinden.

1.4 Eine Gerade zeichnen

Nun soll eine Diagonale in das Feld eingezeichnet werden. Das soll so zunächst so aussehen:

```
o....................|....................
.o...................|....................
..o..................|....................
...o.................|....................
....o................|....................
.....o...............|....................
......o..............|....................
```

Für die Entscheidung, ob man ein `o` oder ein `|` oder ein `.` Zeichen ausgeben muss, kann man eine `if-elif-else`-Fallunterscheidung verwenden. Das Struktogramm von `elif` sieht so aus, hier am Beispiel für 5 Fälle gezeigt (wir brauchen nur 3 Fälle):

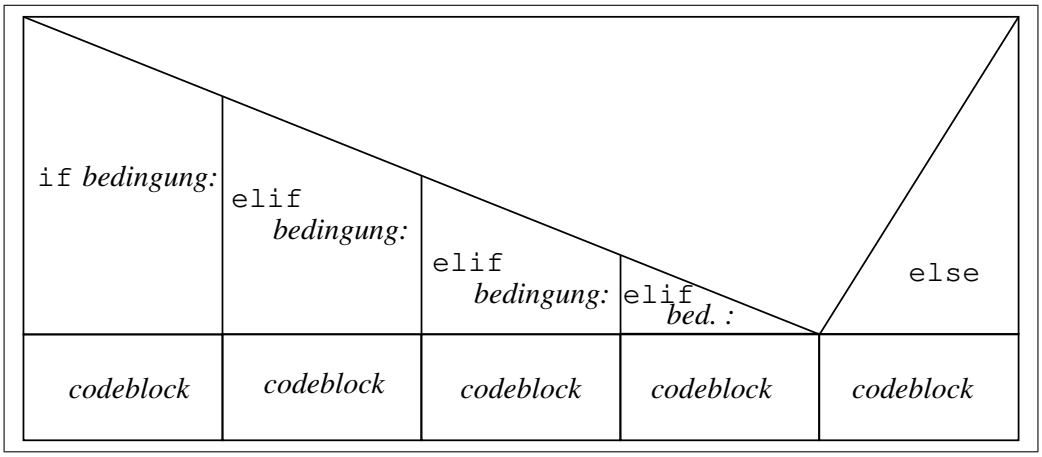

Abbildung 1: if-elif-else

1.5 Die Gerad über die ganze Terminalbreite ausdehnen

Nun soll die Gerade über die ganze Terminalbreite verlaufen. D.h. man muss ihre Steigung ändern: