

## Projektaufgabe - Analoguhr

Das Ziel ist es, eine animierte Analoguhr mit Minuten-, Sekunden- und Stundenzeiger zu programmieren, welche die aktuelle Uhrzeit anzeigt.



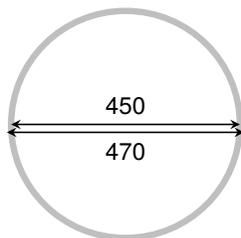
### Planung

Das Programmraster (rechts) soll dir bei der Umsetzung helfen. Die Hinweise zu den einzelnen Stellen folgen weiter unten.

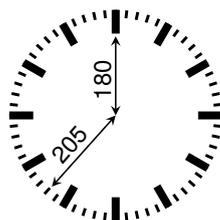
```
1 from gturtle import*
2 import datetime
3
4 # 1. Definitionen der grafischen Elemente
5 ...
6 # 2. Aktuelle Uhrzeit berechnen und speichern
7 ...
8 # 3. Winkeleinstellungen der Zeiger speichern
9 ...
10 #####
11 makeTurtle()
12 ht()
13
14 # 4. Hintergrund zeichnen und speichern
15 ...
16
17 # Animation
18 repeat:
19     # 5. Sekundenzeiger zeichnen
20     ...
21     # Minutenzeiger zeichnen
22     ...
23     # Stundenzeiger zeichnen
24     ...
25     # 6. SekZeiger dreht weiter
26     ...
27     # MinZeiger dreht weiter
28     ...
29     # StdZeiger dreht weiter
30     ...
31     delay(1000)
32     clear()
```

### 1. Definitionen der grafischen Elemente

Teile die Uhr in ihre Bestandteile auf und erstelle einzelne Befehle:



rand()  
weisser Kreis  
über grauem Kreis



zifferblatt()  
Std: Länge:30/Breite:15  
Min: Länge:10/Breite:3



sekzeiger()  
Länge:170  
Breite:5



minzeiger()  
Länge:170  
Breite:8



stdzeiger()  
Länge:100  
Breite:14

Definiere einen Befehl `uhr()`, der das Zifferblatt mit Rand zeichnet:

```
def uhr():
    rand()
    zifferblatt()
```

## 2. Aktuelle Uhrzeit ermitteln und speichern

In TigerJython stehen die Uhrzeitfunktionen zur Verfügung. Dazu muss die Bibliothek (am Anfang) eingebettet werden.

```
import datetime
```

Ermittlung und Speicherung der Zeiten für die Zeiger:

```
now = datetime.datetime.now()
mi=now.minute
sek=now.second
h=now.hour
```

## 3. Winkeleinstellungen der Zeiger speichern

Bei der Animation müssen die Zeiger nach jeder Sekunde nach der aktuellen Zeit ausgerichtet werden. Die Zeiten müssen in die entsprechenden Winkel umgerechnet werden.

### Beispiel

Die Variable `mi` liefert jeweils die aktuelle Minutenzahl. Pro Minute dreht sich der Minutenzeiger um  $360^\circ/60 = 6^\circ$ . Der Winkel `min_winkel` der Turtle, die den Minutenzeiger nach `mi` Minuten zeichnet, ist demnach

```
min_winkel=mi*6
```

### Aufgabe 7.1

Ergänze das Programm mit den Formeln zu `sek_winkel` und `std_winkel`.

## 4. Zifferblatt als Hintergrund speichern

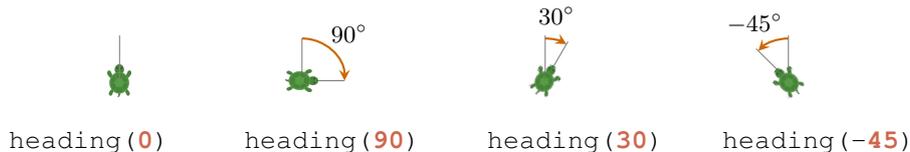
Speichere das Zifferblatt der Uhr als Hintergrund. (Damit du dich im Animationsteil auf die Zeiger konzentrieren kannst)

```
uhr()
savePlayground()
```

## 5. Zeiger zeichnen

Wie kann man den Minutenzeiger nach diesem Winkel ausrichten?

Mit dem Befehl `heading(w)` kann die Turtle **global** um `w` Grad ausgerichtet werden. Dabei wird immer von der Ausrichtung senkrecht nach oben (0 Grad) ausgegangen, unabhängig von der aktuellen Ausrichtung der Turtle.



In der Animation (im Hauptprogramm) kann man den Minutenzeiger nach `min_winkel` zeichnen:

```
heading(min_winkel)
minZeiger()
```

Für `mi=17` ist `min_winkel=mi*6=102`  
(Abbildung rechts)



## 6. Zeigerdrehungen berechnen und speichern

### Aufgabe 7.2

Vervollständige den Animationsteil mit den Drehungen der Zeiger. Um wie viel Grad muss ein Zeiger in einer Sekunde weiterdrehen? Beachte, dass es sich bei den Variablen `min_winkel` etc. um Winkel handelt und nicht um die Zeit.