

## Die zwei Inseln: Paradies und Hölle

### Formel für Kaninchen Population:

$$\begin{array}{rclcl} \text{NeueAnzahlKaninchen} & = & \text{anzahlKaninchen} & * & \text{zuwachsrateKaninchen} \\ 1080 & = & 1000 & * & 1.08 \end{array}$$

### Formel für Fuchs Population:

$$\begin{array}{rclclcl} \text{NeueAnzahlFuechse} & = & \text{anzahlFuechse} & - & \text{abnahmerateFuechse} & * & \text{anzahlFuechse} \\ 32 & = & 40 & - & 0,02 & * & 40 \end{array}$$

## Die Insel der Füchse und Kaninchen

### Gegebene Werte:

anzahlKaninchen = 1000  
anzahlFuechse = 40  
zuwachsrateKaninchen = 0.08  
abnahmerateFuechse = 0.2  
abnahmerateKaninchenFuchs = 0.002  
zunahmerateFuchsKaninchen = 0.0004

### Formel für Kaninchen Population:

$$\begin{aligned} \text{neueAnzahlKaninchen} &= \text{anzahlKaninchen} + (\text{zuwachsrateKaninchen} * \text{anzahlKaninchen}) \\ &\quad - (\text{abnahmerateKaninchenFuchs} * \text{anzahlKaninchen} * \text{anzahlFuechse}) \end{aligned}$$

### Formel für Fuchs Population:

$$\begin{aligned} \text{neueAnzahlFuechse} &= \text{anzahlFuechse} - (\text{abnahmerateFuechse} * \text{anzahlFuechse}) \\ &\quad + (\text{zunahmerateFuchsKaninchen} * \text{anzahlFuechse} * \text{anzahlKaninchen}) \end{aligned}$$

# Vergleich von zwei Funktionen in Python

## zur Populationsberechnung der Kaninchen

**zKa und zFu sind die Zuwachsraten**

**AnzKa und anzKa sowie AnzFu und anzFu stehen für die Anzahl der Kaninchen- bzw. Fuchspopulation**

### Funktion 1:

```
def popKa3Schritte(anzKa, zKa):
    neueAnzKa = anzKa
    neueAnzKa = neueAnzKa + zKa * neueAnzKa
    neueAnzKa = neueAnzKa + zKa * neueAnzKa
    neueAnzKa = neueAnzKa + zKa * neueAnzKa
    return neueAnzKa
```

### Funktion 2:

```
def popKa3Schritte(anzKa, zKa):
    schritte = 3
    zaehler = 0
    neueAnzKa = anzKa
    while zaehler < schritte:
        neueAnzKa = neueAnzKa + zKa * neueAnzKa
        zaehler = zaehler + 1
    return neueAnzKa
```

### Warum ist Funktion 2 schöner als 1?

Um bei Funktion 1 einen weiteren Schritt hinzuzufügen, muss man eine komplette Zeile zu dieser hinzufügen.

Um bei Funktion 2 einen weiteren Schritt hinzuzufügen, muss man lediglich die Zahl der Schritte um 1 erhöhen.

### Gleiches Ergebnis?

War die Frage ernst gemeint? Es werden natürlich die gleichen Ergebnisse ausgegeben.

### Gibt es Verbesserungen? Ja

```
def popKaSchritte(anzKa, zKa, schritte):
    neueAnzKa = anzKa
    for i in range(schritte):
        neueAnzKa = neueAnzKa + zKa * neueAnzKa

    neueAnzKa = round(neueAnzKa)
    return neueAnzKa
```

### Was habe ich getan? (Nummerierung nach Zeilen der "Funktion 2")

1. Ich habe die Variable "schritte" als Funktionsvariable hinzugefügt.
2. |Variable "schritte" aus der Funktion entfernt|
3. |Variable "zaehler" aus der Funktion entfernt|
4. <Zeile beibehalten (Zeile nun in Zeile 2 von FV zu finden)>
5. ~while-Schleife wurde durch eine for-Schleife (Zähler-Schleife) mit der Bedingung \*i in range(schritte)\* ersetzt~
6. <Zeile beibehalten (Zeile nun in Zeile 4 von FV zu finden)>
7. |Zeile gelöscht|
8. <Zeile eine Zeile nach unten verschoben (Zeile nun in Zeile 7 von FV zu finden)>

Zusätzlich hinzugefügte Zeile: round(neueAnzKa) (Zu finden in Zeile 6 von FV)

### Liefern diese Funktionen unsere Ergebnisse aus den vorherigen Aufgaben?

Nein. Ok komplett verneinen kann ich es nicht. Der Grund dafür ist, dass die verwendete Formel nur auf die Insel der Kaninchen der zwei Inseln Welt zutrifft.

### Wie sähe die Funktion mit unseren Formeln aus der eine Insel Welt aus?

#### Funktionen der Kaninchen

```
def popKaSchritte(anzKa, zKa, AnzFu, schritte):
    neueAnzKa = anzKa
    for i in range(schritte):
        neueAnzKa = neueAnzKa+zKa*neueAnzKa-abnahmrtaKaFu*neueAnzKa*AnzFu

    neueAnzKa = round(neueAnzKa)
    return neueAnzKa
```

#### Funktion der Füchse

```
def popFuSchritte(anzFu, zFu, AnzKa, schritte):
    neueAnzFu = anzFu
    for i in range(schritte):
        neueAnzFu = neueAnzFu+zKa*neueAnzFu+zunahmrtaFuKa*neueAnzFu*AnzKa

    neueAnzKa = round(neueAnzFu)
    return neueAnzFu
```