

Python - Rekursion und Turtle

Jan Popko
Python Grundkurs

Eingaben überprüfen

Auf ganze Zahl prüfen:

```
while True:
    zahl = input("Bitte eine Zahl eingeben:")
    if zahl.isdigit():
        break
    else:
        print("Das war keine ganze Zahl")

print("danke")
```

Eingaben überprüfen

Auf Zahl (ganze Zahl oder Kommazahl) prüfen:

```
while True:
    try:
        zahl = input("Bitte eine Zahl eingeben:")
        float(zahl)
        break

    except:
        print("Das war keine Zahl")

print("danke")
```

Eingaben überprüfen

Auf bestimmte Zeichen (z. B. j / n) prüfen:

```
while True:
    eingabe = input("j / n eingeben: ")
    if eingabe == 'j' or eingabe == 'n':
        break
    else:
        print("Bitte j oder n eingeben")

print("danke")
```

Eingaben überprüfen

Auf Zahl (ganze Zahl oder Kommazahl) mit Funktion prüfen:

```
def is_number(e):  
    try:  
        float(e)  
        return True  
    except:  
        return False  
  
while True:  
    eingabe = input("Bitte eine Zahl eingeben: ")  
    if is_number(eingabe) == True:  
        break  
    else:  
        print("Das war keine Zahl!")
```

Rekursion

Eine rekursive Funktion ist eine Funktion, die sich selbst aufruft.
Man nähert sich der Lösung immer weiter an.

Zur Standardausstattung von Python gehört ein Modul namens "turtle".

Eingebunden wird es über den Befehl `" from turtle import * "`

in der interaktiven Python-Shell können wir turtle ausprobieren:

```
>>>from turtle import *  
>>>clear()
```

Ein Fenster öffnet sich und wir sehen eine Pfeilspitze die nach rechts zeigt. Das ist Turtle.

```
>>>forward(100) # Turtle geht 100 Schritte geradeaus und erzeugt  
                dabei ein Linie  
>>>right(90) # dreht Turtle um 90 Grad nach rechts
```

Rekursion

Befehl	Bedeutung
<code>back(n)</code>	gehe n Schritte zurück
<code>circle(r)</code>	zeichne einen Kreis mit dem Radius r
<code>clear()</code>	lösche die Zeichenfläche, Turtle in die Mitte
<code>forward(n)</code>	gehe n Schritte geradeaus
<code>left(n)</code>	Drehe dich um n Grad nach links
<code>hideturtle()</code>	Turtle wird unsichtbar
<code>pendown()</code>	Stift nach unten!
<code>penup()</code>	Stift nach oben!
<code>right(n)</code>	Drehe dich um n Grad nach rechts
<code>showturtle()</code>	Turtle wird sichtbar
<code>speed(n)</code>	n ist eine Zahl von 1 bis 10

Rekursion

Eine rekursive Spirale:

```
from turtle import *
```

```
def spirale(x):  
    forward(x)  
    right(90)  
    spirale(0.9 * x)  
    return
```

```
spirale(200)
```

Turtle zeichnet eine rechteckige Spirale, die mit jedem Funktionsaufruf kleiner wird. Sie endet niemals.

Zum Abbrechen einfach das Fenster schließen.

Rekursion

Eine rekursive Spirale:

```
from turtle import *

def spirale(x):
    if x <= 10:
        return
    else:
        forward(x)
        right(90)
        spirale(0.9 * x)
        return

spirale(200)
```

Jetzt gibt es eine Abbruchbedingung im if-Zweig.

Rekursion

Binärbaum:

Erstellen Sie ein Programm, welches den folgenden Binärbaum zeichnet

