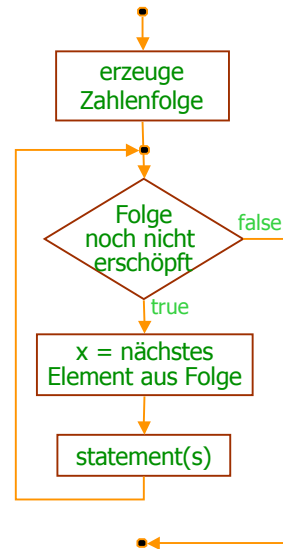




For-Schleife

- Anders als in C++
- Definition:

```
for x in list :  
    statements
```
- Oft wird *list* durch die **range**-Funktion generiert (s. später)
- Statt **range** kann jede andere Art von Listen stehen
- Erstes Beispiel eines **Iterators**!



Beispiel C++ vs. Python

- Beispiele:

```
# float q, int n  
s = 0.0  
# s = geom. Reihe 1 + q + q^2 + q^3 + ... + q^n  
qq = 1.0  
for i in range(n):  
    s += qq  
    qq *= q  
print s
```

} Loop body
(Schleifenrumpf)

"Schleifenvariable"
(loop variable)

- in C++:

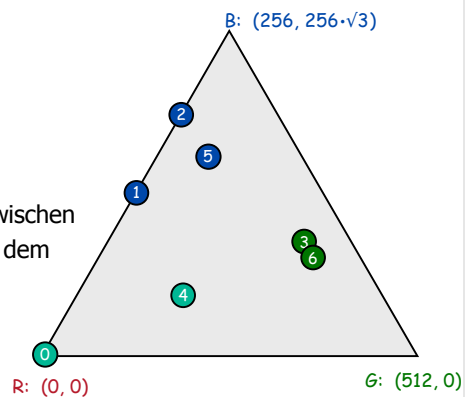
```
// float q; unsigned int n;  
float s = 0;  
// s = geom. Reihe 1 + q + q^2 + q^3 + ... + q^n  
float qq = 1;  
for ( unsigned int i = 0; i < n; i ++ )  
{  
    s += qq;  
    qq *= q;  
}
```



Beispiel: "Chaos Game"



- Spiel in einem gleichseitigem Dreieck, dessen Ecken rot, gelb und blau eingefärbt sind.
- Starte bei Punkt R
- Wiederhole:
 - Wähle zufällig einen Eckpunkt
 - Gehe die Hälfte der Strecke zwischen momentanen Standpunkt und dem ausgewählten Eckpunkt
 - Male dort einen Punkt



```
import Image
import random
import sys

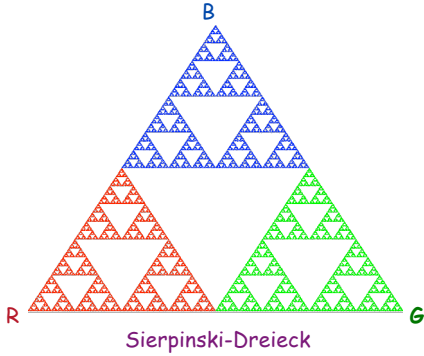
im = Image.new("RGB", (512, 512), (256, 256, 256) )
N = int( sys.argv[1] )
x = 0.0
y = 0.0

for i in range( 0, N ):
    r = random.random()
    if r < 0.333:
        x0 = 0.0
        y0 = 0.0
    elif r < 0.6667:
        x0 = 512.0
        y0 = 0.0
    else:
        x0 = 256.0
        y0 = 443.4
    x = ( x0 + x ) / 2.0
    y = ( y0 + y ) / 2.0
    im.putpixel ( (int(x), int(y)), (int(x), int(y), 0) )

im.show()
```

Hier fehlt eigtl ein Test, ob überhaupt ein Command-Line-Argument angegeben wurde!

`% ./ChaosGame.py 100000`



Sierpinski-Dreieck

- Frage: was passiert, wenn man einen zufälligen Punkt im Inneren des ursprünglichen RGB-Dreiecks als "Seed" nimmt?

G. Zachmann Informatik 1 - WS 05/06 Einführung in Python 1 69

Geschachtelte Schleifen (nested loops)

- Analog wie in anderen Sprachen
- Schleifenrumpf kann wieder Schleife enthalten:

```
for i in range(...):
    for j in range(...):
        ...
```

Andere Schleifenvariable nehmen!

- in C++:

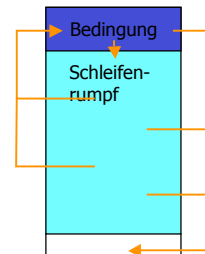
```
for ( uint i = 0; ... )
{
    for ( uint j = 0; ... )
    {
        ...
    }
}
```

G. Zachmann Informatik 1 - WS 05/06 Einführung in Python 1 70



Break und continue

- Wie in C++: zusätzliche Sprünge innerhalb der Schleife
 - **break**: springt aus Schleife heraus (nur 1 Level!)
 - **continue**: startet sofort den nächsten Schleifendurchlauf



```
while ...:
    if ...:
        break
    if ...:
        continue
    ...
    # continue jumps here
# break jumps here
```



```
while (...)
{
    if (...)
        break;
    if (...)
        continue;
    ...
    // continue jumps here
}
// break jumps here
```



Längeres Beispiel: Longest Run

```
list_size = 3
n_values = 0
longest_run = 0
length = 0
last_value = 0
while n_values < list_size:
    value = input()
    if value == last_value:
        length += 1
    else:
        length = 1
    if longest_run < length:
        longest_run = length
    last_value = value
    n_values += 1
print "Longest Run = %d\n" % (longest_run)
```

← Angenommen, wir wüssten das

```



list_size = 3
n_values = 0
longest_run = 0
length = 0
last_value = 0
while n_values < list_size:
    value = input()
    if value == last_value:
        length += 1
    else:
        length = 1
    if longest_run < length:
        longest_run = length
    last_value = value
    n_values += 1
print "Longest Run = %d\n" %(longest_run)

```

Eingabe: 1.0 5.0 5.0

list_size	3
n_values	3
longest_run	2
length	2
last_value	5.0
value	5.0

G. Zachmann Informatik 1 - WS 05/06 Einführung in Python 1 77

Korrekte Programme durch vollständige Fallunterscheidung

```

list_size = 3
n_values = 0
longest_run = 0
length = 0
last_value = 0
while n_values < list_size:
    value = input()
    if value == last_value:
        length += 1
    else:
        length = 1
    if longest_run < length:
        longest_run = length
    last_value = value
    n_values += 1
print "Longest Plateau = %d\n" % (longest_run)

```

Kennt man i.A. nicht!

G. Zachmann Informatik 1 - WS 05/06 Einführung in Python 1 78

```

import sys
lines = sys.stdin.readlines()
longest_run = 0
length = 0
last_value = 0
for line in lines:
    value = float(line)
    if value == last_value:
        length += 1
    else:
        length = 1
    if longest_run < length:
        longest_run = length
    last_value = value
print "Longest Plateau = %d\n" % (longest_run)

```

Diese Funktion liest alle Zeilen auf einmal in eine Liste

Problem: was, wenn float(lines[0]) == 0?!

NB: Eingabe auf stdin mit CTRL-D abschließen.
(CTRL-D = EOF [end-of-file] unter unix)

```

import sys
lines = sys.stdin.readlines()
length = 0
longest_run = 0
first = True
for line in lines:
    value = float(line)
    if first:
        last_value = value - 1
        first = False
    if value == last_value:
        length += 1
        if longest_run < length:
            longest_run = length
    else:
        length = 1
    last_value = value
print "Longest Plateau = %d\n" % (longest_run)

```

Noch ein Problem:
was, wenn der input stream gar nicht mehr aufhört?!



Interaktives Programm



- Lissajous-Figuren
 - Idee: zwei orthogonale Schlitten, die hin- und herfahren und gemeinsam einen Stift führen
 - Schlitten werden durch periodische Funktionen gesteuert, z.B. \sin/\cos

