

which python or which python3.4 -> pfad zu python /usr/bin/[pythonX]

http://www.michael-holzapfel.de/progs/python/python_beisp.htm

```
#das for kann mehrmals ausgeführt werden
#und der inhalt wird mehrmals ausgegeben
mylist = [x*x for x in range(3)]
for i in mylist:
    print(i) #0 1 4

#das for kann mehrmals ausgeführt werden
#jedoch wird der inhalt nur einmal ausgegeben
#da generator
mygenerator = (x*x for x in range(4))
for i in mygenerator:
    print(i) #0 1 4 9
```

$$\begin{aligned}
 ax^2 + bx + c &= 0 \\
 x^2 + b/a \cdot x + c/a &= 0 \\
 x^2 + b/a \cdot x &= -c/a \\
 x^2 + b/a \cdot x + (b/(2a))^2 &= (b/(2a))^2 - c/a \\
 (x + b/(2a))^2 &= (b/(2a))^2 - c/a \\
 x + b/(2a) &= \pm \sqrt{(b/(2a))^2 - c/a} \\
 x_{1,2} &= -b/(2a) \pm \sqrt{(b/(2a))^2 - c/a} \\
 &= -b/(2a) \pm \sqrt{b^2 - 4ac}/(2a) \\
 &= (-b \pm \sqrt{b^2 - 4ac})/(2a) \\
 x_{1,2} &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
 \end{aligned}$$

```
# Integer: Pythagoras
from math import sqrt

print("Integer:")
print("Pythagoras a**2+b**2=c**2")
a = 5
b = 6
quadratsumme = a**2 + b**2
c = float(sqrt(quadratsumme))
print("a=5, b=6, c="+str(c))
print(a,b,c)

print(100 - 36 / 10) # = 96.4
print(100 - 36 // 10) # = 97 // cuts after the .
```

```
# pi-monte_carlo.py
# pi-Bestimmung mit der Methode von Monte Carlo
from random import random
print "Monte Carlo Methode zur"
print "Naherung für pi:"
g = input("Gesamtzahl der Tropfen: ")
v = 0
x=0; y=0 # Koordinaten des Punktes P
for i in range(1,g+1):
    x = random()
    y = random()
    if x*x+y*y<= 1:
        v = v + 1
pi_naeh = 4.0*v/g
print g, "Tropfen, davon", v, "Tropfen im Viertelkreis,"
print "pi etwa", pi_naeh
```

```
# muenze.py
# muenze_Werfen
from random import randrange # Aus dem Modul random wird
n = input("Anzahl der Versuche: ") # der Befehl randrange
k = 1 # geladen
k = n
print
print "n hn(2)"
print "-----"
while n<m:
    z = 0
    n = 10*n
    k = k / 10
    for i in range(1,n-1):
        if randrange(1,3) == 1: # Erzeugen der Zufallszahl 1 oder 2
            z = z+1
    print (10-len(str(n)))*' ',n,' ',float(z)/n
    # 10 - Länge der Zeichenkette von n
    # oder: print "%10i %14f" % (n, float(z)/n)
```

```
#!/usr/bin/python3.4
import csv
import re

with open('Data1.csv', newline='') as f:
    reader = csv.reader(f, delimiter=',')
    fields = enumerate(reader, Name=1, Strasse=2, PLZ=3, Tel=4, Email=5, IP4=6, Datum=7)
    for row in reader:
        fields = 0
        for col in row:
            if fields == AdressNr:
                if not re.search("^\d+$", col):
                    print('Invalid AdressNr: '+col)
            if fields == Name: # Doppelnamen möglich
                pl = '[A-Z][a-z]*'
                if not re.search("^(?!(pl+|pl+[-]pl+|pl+[-]pl+[-]pl+|pl+[-]pl+[-]pl+)$", col):
                    print('Invalid Name: '+col)
            if fields == Strasse: # Strasse & Nr
                if not re.search("^[A-Z][a-z]*\d+\d+\d+[a-z]{1}$", col):
                    print('Invalid Street: '+col)
            if fields == PLZ: # PLZ und Ort
                if not re.search("^\d{4}\d{3}[a-z]{1}$", col):
                    print('Invalid ZIP and Place: '+col)
            if fields == Tel: # Alle gängigen Formate
                if not re.search("^\d{3}(\d{3}|\d{7}|\d{3}\d{4}|\d{2}\d{2}\d{2})\d{10}$", col):
                    print('Invalid Tel. Number: '+col)
            if fields == Email:
                if not re.search("^[a-z]+@[a-z]+\.[a-z]+$", col):
                    print('Invalid Email: '+col)
            if fields == IP4:
                octett = '\d{1-9}|\d{10}|\d{2}[0-4]|\d{25}[0-5]'
                if not re.search("^(?!(octett+|octett+[-]octett+|octett+[-]octett+[-]octett+)$", col):
                    print('Invalid IP: '+col)
            if fields == Datum:
                if not re.search("^(0[1-9]|1[12][0-9]|3[01])[.](0[1-9]|1[012])[.](19|20)[0-9]{2}$", col):
                    print('Invalid Date: '+col)
        fields += 1
```

```
# Modul csv laden
import csv
# Öffnen der Datei
with open('Data1.csv', newline='') as f:
    # Zwischen der Daten = einer Variable und festlegen des Begrenzers
    reader = csv.reader(f, delimiter=',')
    for row in reader:
        # Hier Code einfügen, welcher für jede Zeile (Datensatz) ausgeführt wird
        print('----- Neue Zeile -----')
        for col in row:
            # Hier Code einfügen, welcher für jede Zeile (Attribut) ausgeführt wird
            print('Zelle: ' + str(col))
```

int(), float(), str()

```

# pi-berechn2.py
# pi-Berechnung mit regulären 2n-Ecken
from math import sqrt, pi
n = 6 # Start mit regulärem Sechseck
s = 1 # Seitenlänge des reg. Sechsecks
print "Schrittweise Näherung von pi mit Hilfe eines 2n-Ecks"
for i in range(1,21):
    pi_naehung = 0.5*n*s
    print pi_naehung
    s = s/sqrt(2-sqrt(4-s*s))
    n = 2*n # doppelte Eckenzahl
    print "Gute Iteration!"
    print "pi =", pi

# pi-berechn1.py
# pi-Berechnung mit regulären 2n-Ecken
from math import sqrt, pi
n = 6 # Start mit regelmäßigem Sechseck
s = 1 # Seitenlänge des reg. Sechsecks
print "Schrittweise Näherung von pi mit Hilfe eines 2n-Ecks"
for i in range(1,31):
    pi_naehung = 0.5*n*s
    print pi_naehung
    s = sqrt(2-sqrt(4-s*s))
    n = 2*n # doppelte Eckenzahl
    print "Ungeeignete Iteration!"
    print "pi =", pi

```

Lösungen einer quadratischen Gleichung der Form $ax^2 - bx + c = 0$. Lösungsterm:

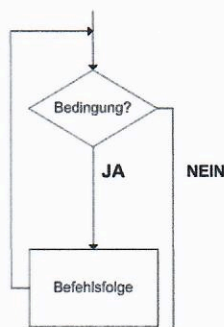
$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Programm:

```
# quadr_gleich.py
from math import sqrt # sqrt() ist die Quadratwurzelfunktion
print "Quadratische Gleichung mit den Koeffizienten a, b und c lösen."
a = input("a eingeben: ") # \n bedeutet: neue Zeile
b = input("b eingeben: ")
c = input("c eingeben: ")
d = b**2-4*a*c
if d > 0:
    x1 = (-b - sqrt(d))/(2*a)
    x2 = (-b + sqrt(d))/(2*a)
    print "Die Gleichung hat die beiden Lösungen:"
    print "x1 =", x1, "; x2 =", x2
elif d == 0:
    x1 = (-b)/(2*a)
    print "Die Gleichung hat die Lösung:"
    print "x =", x1
else:
    print "Die Gleichung hat keine Lösung!"
```

Ausgabe:

Quadratische Gleichung mit den Koeffizienten a, b und c lösen.
a eingeben: 2
b eingeben: 5
c eingeben: 3
Die Gleichung hat die beiden Lösungen:
x1 = -1.5 ; x2 = -1.0



```
# wurzel_n.py
a = float(input("Zu berechnende n.Wurzel: "))
n = float(input("Eingabe von n: "))
x = float(input("Schätzwert: "))
# float bedeutet, dass mit Kommazahlen gerechnet werden soll!
print "Iteration Näherungswert"
print "-----"
for i in range(1,6):
    x = ((n-1)*x-a/(x**(n-1)))/n
    # Bemerkung: x**(n-1) bedeutet x hoch n-1
    print ' ', i, ' ', x
```

```
# wurzel.py
a = input("Zu berechnende Quadratwurzel: ")
x = input("Startwert x1: ")
print "Iteration Näherungswert"
print "-----"
for i in range(1,6):
    x = 0.5*(x+a/x)
    # Bemerkung: 0.5 statt 1/2, damit mit Kommazahlen gerechnet wird!
    print ' ', i, ' ', x
```