rom math import sqrt rint("Integer:")

orint("Pythagoras a**2+b**2=c**2")

print(100 - 36 / 18) # = 96.4 print(100 - 36 // 10) # = 97 (// cuts after the

x=0; y=0 # Koordinaten des Punktes P

pi-monte_carlo.py
pi-Bestimmung mit der Methode von Monte Carlo
from random import random
print "Monte Carlo Methode zur"
print "Näherung für pi:"
g = input("Gesamtzahl der Tropfen: ")
v = 0

muenze.py
muenze.py
muenze werfen
ine random import randrange # Aus dem Modul random wird
ine input("Anzahl der Versuche: ") # der Befehl randrange
= 1

hn(2)'

27

quadratsumme = a**2 + b**2 c = float(sqrt(quadratsumme))

print("a=5, b=5, c="+str(c)) print(a,b,c)

= 1 = m

erint erint'

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

und der inhalt wird mehrmals ausgegeben

#das for kann mehrmals ausgeführt werden

= -c/a

 $= \pm \sqrt{((b/(2a))^2}$

 $x^2 + b/a \cdot x + (b/(2a))^2 = (b/(2a))^2 - c/a$ $(x + b/(2a))^2 = (b/(2a))^2 - c/a$

 $x_{1/2} = -b/(2a) \pm \sqrt{((b/(2a))^2 - c/a)}$ = $-b/(2a) \pm \sqrt{(b^2 - 4ac)/(2a)}$ = $(-b \pm \sqrt{(b^2 - 4ac))/(2a)}$

mygenerator = (x*x for x in range(4))

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

awi + hw + c

xº + b/a x

x + b/(2a)

for kann mehrmals

for i in mylist: print(i) #0 1

for i in mygenerator: print(i) # 0 1 4 9

mylist = [x*x for x in range(3)]

```
Mirio Eggmann
                                                                                                                                                                                                                                 http://www.michael-holzapfel.de/progs/python/python_beisp.htm
                                                                                                                                                           with open('Datal.csv', newline='') as f:
    reader = csv.reader(f, delimiter=";")
    fields = enum(AdressNr=0, Name=1, Strasse=2, PLZ=3, Tel=4, Email=5, IP4=6, Datum=7)
                                                                                                                                                              Tields = enum(adressNr=0, Name=1, Strass

for row in reader:

fields = 0

for col in row:

   if fields == AdressNr:

   if not re.search('d$', col):

       print('Invalid AdressNr: '+col)
#jedoch wird der inhalt nur einmal ausgegeben
                                                                                                                                                                       print('Invalid AdressNr: +col)
if fields == Name: # Doppelnamen moglich
pl='[A-2](1][a-2]+'
   if not re.search('^('epl+'|'+pl+'[-]'+pl+')\s('+pl+'|'+pl+'[-]'+pl+')\s', col);
   print('Invalid Name: '+col)
if fields == Strasse: # Strasse & Nr
   if not re.search('^[A-2]{1}[a-2]+\s(\d+|\d+[a-2]{1})\s', col);
   print('Invalid Street: '+col)
if fields == PLZ: # PLZ und Ort
                                                                                                                                                                      print('Invalid street: '*cou;
if fields == PlZ: # PtZ und Ort
if not re.search('^\d(4)\s[A-Z]{1}[a-z]+$', col):
    print('Invalid ZIP and Place: '*col)
if fields == Tel: # Alle Ganglogn Formate
if not re.search('^\d(3)\s[\/]\d(3)\s(\d(4)\d(2)\s\d(2))))(\d(10)\s', col):
    print('Invalid Tel. Number: '*col)
if fields == Email:
if not re.search('[*@]+@[*@]+\.[*@]+', col):
    print('Invalid Email: '*col)
if fields == IP4:
    octett='(\d[1.9]\d[1\d]\d[2[0-4]\d[25[0-5])'
if not re.search('^*+octett+'[.]'+octett+'[.]'+octett+'[.]'+octett+'s', col):
    print('Invalid IP: '*col)
if fields == Datum:
                                                                                                                                                                        if fields == Datum:
if net re.search('^(0[1-9]|[12][0-9]|3[01])[.](0[1-9]|1[012])[.](19|20)[0-9]{2}$', col):
    print('Invalid Date: '+col)
fields += 1
                                                                                                                                                                                                                       csy
                                                                                                                                                                                                        with open('Datal.csv', newline='') as f:
                                                                                                                                                                                                                  reader = csv.reader(f, delimiter=":")
for row in reader:
                                                                                                                                                                                                                                                                                                 ür jede Zeile (Datensatz) ausgeführt wird
                                                                                                                                                                                                                         print('---- Neue Zeile -----')
for col in row:
                                                                                                                                                                                                                                  # Haer Code einfügen, welcher für jede Zelle (Attribut) ausgeführt wird print('Zelle:' + str(col))
                                                                                                                                                                                                                                                                                                                                               Int(), float(), str()
                                                                                                                                                                                                                                                                                                                                # potenze 1 3 py
# potenzen von 1/3 u
n = input ("Bis zu we
potenz = 1.0/3
print
sum potenz = 1.0/3
print - 1/2
print - 1/3
print - 1/3
print + 3d + 1/3
potenz = 1/1
potenz = 1/1
potenz = 1/1
potenz = 1/1
                                                                                                                                            sum_potenz+poten;
                                                                                                                                                                                                                                                                                                                                                                                      welcher Potenz v
                                                                                                                                                                                                                                                                                                                                                 600
Hh
                                                                                                                                                                                                                                                                                                                                                 100
                                                                                                                                            elp
                                                                                                                                                                                                                                                                                                                                                                                        Non
                                                                                                                                                                                                                                                                                                                                                 potenz,
                                                                                                                                                                                                                                                                                                                                                                                        37
                                                                                                                                                                                                                                                                                                                                                                                        :
                                                                                                                                                                                                                                                                                                                                                 sum potenz
                                                                                                                                                                                                                       x_{1/2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{}
                                                                                                                                                                                             # wurzel_n.py
a = float(input("Zu berechnende n.Wurzel: "))
n = float(input("Eingabe von n: "))
x = float(input("Schatzwert: "))
# float bedeutet, dass mit Kommazahlen gerechnet werden soll!
print "Iteration Naherungswert"
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
```

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

Programm:

s quadr_gleich.py
from math import sqrt # sqrt() ist die Quadratwurzelfunktion
print "Quadratsische Gleichung mit den \nKceffizienten a, b und c losen."

a = input("a eingeben: ")
b = input("b eingeben: ")
c = input("c eingeben: ")
d = b*fb-4*a*c
if d > 0:
x1 = (-b - sqrt(d))/(2*a)
x2 = (-b + sqrt(d))/(2*a)
print "Die Gleichung hat die beiden Losungen:"
print "x1 = ',x1,'; x2 = ',x2
elif d == 0:
x1 = (-b)/(2*a);
print "Die Gleichung hat die Lösung:"
print "x = ',x1
else:

Bedingung?

Bedingung?

Bedingung?

Bedingung? print "Die Gleichung hat keine Lösung!" JA Quadratische Gleichung mit den Koeffizienten a, b und c lösen. a a eingeben: 2 b eingeben: 5 c eingeben: 5 c eingeben: 3 lie Gleichung hat die beiden Lösungen: x1 = -1.5; x2 = -1.0 Befehlsfolge

NFIN

```
# wurzel.py
a = input("Zu berechnende Quadratwurzel: ")
x = input("Startwert x1: ")
print "Iteration Naherungswert"
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
```

```
Mirio Eggmann
                                                                                                                                                                   calculateGgt(a, b):
                                                             cpuinfo = open('/proc/cpuinfo','r')
                                                                                                                                                              if(b > 0)
                                                                                                                                                                    return calculateGgt(b, a % b)
                                                                                                                                                               else:
                         wohl
                                                              rint(cpuinfo)
                                                                                                                                                                    return a
                                                             for line in cpuinfo:
print(line, end='
                                                                                                                                                           def main():
                                                                                                                                                              a = int(input("Bitte geben Sie die erste Zahl ein: "))
b = int(input("Bitte geben Sie die zweite Zahl ein: "))
print("Gerechnet wird ggt von " + str(a) + " und " + str(b))
print("Der gerechnete ggt(" + str(a) + "," + str(b) + ") ist: " + str(calculateGgt(a, b)))
                                                              rint(cpuinfo.read())
                -5 < 0:
("Das s
                                                             print(cpuinfo.readline())
                 age int
                                                             print(cpuinfo.readline())
                 if a
                                                              rint(couinfo.readlines())
                                           ausgabe = "Python lernen!"
            HIEREIFUEGEN
                                            print(ausgabe)
                                                                                                                                    from array import array regex = []
                                          print(ausgabe[0] + " " + ausgabe[7])
print(ausgabe[5] + " " + ausgabe[12])
                                                                                                                                     nameRegex = '^[A-ZĀŌŪ]{1}[a-zāōū]+\s[A-ZĀŌŪ]{1}[a-zāōū]+$'
                                                                                                                                  nameRegex = '^[A-ZA0U]{1}{a-zāoū]+\s[A-ZāōŪ]{1}{a-zāoū]+$'
regex.append(nameRegex)
streetRegex = '^[A-Z]{1}{a-z]+\s[0.9]+{a-zA-Z]*'
regex.append(streetRegex)
placeRegex = '^[0.9]{4}\s[A-Z]{1}{a-z]+$'
regex.append(placeRegex)
emailRegex = '^[a-zA-Z-]+[0]{1}{a-zA-Z]+[.]{1}{a-zA-Z]{2,4}$'
regex.append(emailRegex)
birthDateRegex = '^[a-zA-Z-]+[.]{1}{a-zA-Z]+[.]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4}\s[0.9]{4
  , 'test'
test:
, end='
                                         # Get the last character of a stri
letztes zeichen = len(ausgabe) -1
                                               int(ausgabe[letztes_zeichen])
  element in te
int(element,
                        = [ 'Katze'
kittycat:
<, len(x))
 t = ['hall
element i
                         ycat = x in ki
                                          while i <= int(letztes zeichen):
                                             s = s + ausgabe[i]
i=i+l
 test
for
                                              print(s)
                                                                                                                                    for element in regex:
                                                                                                                                      or element in regex:

valid = False

while (not valid):

inputUser = input(questions[i]+": ")

if re.search(element, inputUser):

valid = True

i = i + i
      nt("Unendlicher (25 Mal) Fibonacci-Generator")
fibonacci():
   a, b = 0, 1
while True:
                                                                                                                                           i = i + 1
userData.append(inputUser)
else:
                 ("before"+str(a))
       yield a
print("after"+str(a))
                                                                                                                                       print('Falsche Eingabe! ')
rint('Sie wurden erfolgreich registriert!')
       a, b = b, a + b
fibonacci()
 counter
for x in f:
    print("fib(" + str(counter) + ") = " + str(x))
                                                                                                                                                                   Start of string
                                                                                                                                                                                                                                                   0 or more
                                                                                                                                                                                                                                                                                                                             Any character except
  counter += 1
if (counter > 1): break
                                                                                                                                     \A
                                                                                                                                                                   Start of string
                                                                                                                                                                                                                                                   1 or more
                                                                                                                                     $
                                                                                                                                                                   End of string
                                                                                                                                                                                                                                                   0 or 1
                                                                                                                                                                                                                                                                                                    falbi
                                                                                                                                                                                                                                                                                                                             a or h
                                                                                                                                      \Z
\b
                                                                                                                                                                   End of string
                                                                                                                                                                                                                     {3}
{3,}
                                                                                                                                                                                                                                                  Exactly 3
                                                                                                                                                                                                                                                                                                    (...)
                                                                                                                                                                                                                                                                                                                             Group
     def getmemoryinfo():
    memoryinfo = open('/proc/meminfo', '
    memoryinfos = memoryinfo.readlines()
    return memoryinfos
                                                                                                              for
                                                                                                                                                                   Word boundary
                                                                                                                                                                                                                                                   3 or more
                                                                                                                                                                                                                                                                                                                             Passive Group
                                                                                                                                      B
                                                                                                                                                                                                                                                                                                    [abc]
                                                                                                                                                                   Not word boundary
                                                                                                                                                                                                                     {3,5}
                                                                                                                                                                                                                                                   3.4 or 5
                                                                                                                                                                                                                                                                                                                             Range (a or b or c)
                                                                                                          arg
                                                                                                                                      1<
                                                                                                                                                                   Start of word
                                                                                                                                                                                                                                                                                                                              Not a or b or c
                                                                                                                     Sys
                                                                                                         in
                                                                                                                                      13
                                                                                                                                                                  End of word
                                                                                                                                                                                                                                                                                                    [a-o]
                                                                                                                                                                                                                                                                                                                             Letter between a and o
     def getMemoryArguments(argument):
                                                                                                                                                                                                                                                                                                                             Upper case letter
                                                                                                                                                                                                                                                                                                    [A-Q]
                                                                                                             sys.argv:
        switcher = {
  "totalmemory": 0,
  "freememory": 1,
  "buffers": 2,
                                                                                                                                                                                                                                                                                                                                etween A and Q
                                                                                                                                                                                                                      "x" below represents a quantifier
                                                                                                                                                                                                                                                                                                    [0-7]
                                                                                                                                                                                                                                                                                                                             Digit between 0 and 7
                                                                                                                                                                                                                                                  Ungreedy version of "x"
                                                                                                                                                                                                                                                                                                                             nth group/subpattern
                                                                                                                                                                   Control character
                                                                                                                                                                                                                                                                                                    Note: Ranges are inclusive.
                                                                                                                                                                   White space
                                                                                                                                      15
        return switcher.get(argument, "Not defined")
                                                                                                                                                                   Not white space
                                                                                                                                      \d
\D
                                                                                                                                                                   Digit
                                                                                                                                                                                                                     ١
                                                                                                                                                                                                                                                  Escape Character
                                                                                                                                                                   Not digit
                                                                                                                                                                                                                                                                                                                            Global match
                                                                                                                                                                                                                                                                                                    g
     def getcpuinfo():
    cpuinfo = open('/proc/cpuinfo', 'r')
    cpuinfos = cpuinfo.readlines()
    return cpuinfos
                                                                                                                                      \w
                                                                                                                                                                   Word
                                                                                                                                                                                                                                                                                                                           Case-insensitive
                                                                                                                                                                   Not word
                                                                                                                                                                                                                                                                                                    m
                                                                                                                                                                                                                                                                                                                             Multiple lines
                                                                                                                                      1x
                                                                                                                                                                   Hexadecimal digit
                                                                                                                                                                                                                                                                                                                             Treat string as single line
                                                                                                                                                                   Octal digit
                                                                                                                                                                                                                                                                                                                             Allow comments and
     def getCpuArguments(argument):
             getCpuArguments(
switcher = {
    "vendor": 1,
    "model": 4,
    "mhz": 7,
    "cache": 8
                                                                                                                                                                                                                                                                                                                             white space in pattern
                                                                                                                                                                                                                                                                                                                            Evaluate replacement
                                                                                                                                                                                                                                                                                                    H
                                                                                                                                                                                                                                                                                                                             Ungreedy pattern
                                                                                                                                      [:upper:]
                                                                                                                                                                   Upper case letters
                                                                                                                                                                   Lower case letters
                                                                                                                                      [:alpha:]
                                                                                                                                                                   All letters
              return switcher.get(argument, "Not defined")
                                                                                                                                       (:alnum:)
                                                                                                                                                                   Digits and letters
                                                                                                                                                                                                                     In
                                                                                                                                                                                                                                                  New line
                                                                                                                                       [:digit:]
                                                                                                                                                                   Digits
                                                                                                                                                                                                                                                                                                    $n
                                                                                                                                                                                                                                                                                                                           nth non-passive group
                                                                                                                                                                                                                                                  Carriage return
                                                                                                                                                                                                                                                                                                    $2
                                                                                                                                                                                                                                                                                                                             'xvz" in /^(abc(xvz))$/
                                                                                                                                       [:xdigit:]
                                                                                                                                                                   Heyaderimal dinits
                                                                                                                                                                                                                     10
                                                                                                                                                                                                                                                   Tab
                                                                                                                                                                                                                                                                                                    $1
                                                                                                                                                                                                                                                                                                                           "xyz" in /^(?:abc)(xyz)$/
                                                                                                                                       [:punct:]
                                                                                                                                                                   Punctuation
           listcpuinfo
listmemoryinfo
                                                                                                                                                                   Space and tab
Blank characters
                                                                                                                                                                                                                                                  Vertical tab
                                                                                                                                                                                                                                                                                                                           Before matched string
                                                                                                                                       (:blank:)
                                                                                                                                                                                                                     \v
                                                                                                                                                                                                                                                                                                    $
                                                                                                                                                                                                                                                   Form feed
                                                                                                                                                                                                                                                                                                                           After matched string
                                                                                                                                      [:space:]
                                                                                                                                                                                                                                                                                                                           Last matched string
                                                                                                                                                                                                                     book
                                                                                                                                                                                                                                                  Octal character xxx
                                                                                                                                                                                                                                                                                                    54
                                                                                                                                       [:cntrl:]
                                                                                                                                                                   Control characters
                                                                                                                                                                                                                                                                                                                           Entire matched string
                                                                                                                                      [:graph:]
                                                                                                                                                                   Printed characters
                                                                                                                                                                                                                                                   Hex character hh
                                                                                                                                                                                                                                                                                                    $8
rvariables

cpuArguments = []

memoryArguments = []

cpu = listcpuinfo.getcpuinfo()

memory = listmemoryinfo.getmemoryinfo()
                                                                                                                                                                   Printed characters and
                                                                                                                                       [:print:]
                                                                                                                                                                   spaces
Digits, letters and
                                                                                                                                      [:word:]
                                                                                                                                                                                                                      Pattern
                                                                                                                                                                                                                                                                                                     Will Match
                                                                                                                                                                   underscore
                                                                                                                                                                                                                     (FA-Za-z0-9-1+)
                                                                                                                                                                                                                                                                                                     Letters, numbers and hyphens
 or arg in sys.argv:
   if (listcpuinfo.getCpuArguments(arg) != "Not defined"):
        cpuArguments.append(listcpuinfo.getCpuArguments(arg))
                                                                                                                                                                                                                      (\d{1,2}\/\d{1,2}\/\d{4})
                                                                                                                                                                                                                                                                                                     Date (e.g. 21/3/2006)
                                                                                                                                                                                                                      ([^\s]+(?=\.(jpg|gif|png))\\2)
(^[1-9]{1}$|^[1-4]{1}{0-9]{1}$|^50$)
                                                                                                                                                                                                                                                                                                     jpg, gif or png image
                                                                                                                                                                                                                                                                                                     Any number from 1 to 50 inclusive
                                                                                                                                                                   Lookahead assertion
      arg in sys.argv:
f (listmemory.nfo.getMemoryArguments(arg) != "Not defined"):
memoryArguments.append(listmemoryinfo.getMemoryArguments(arg)
                                                                                                                                                                                                                     (#?([A-Fa-f0-9])(3)(([A-Fa-f0-9])(3))?)
((?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,15})
                                                                                                                                                                                                                                                                                                     Valid hexadecimal colour code
                                                                                                                                                                   Negative lookahead
                                                                                                                                                                                                                                                                                                     String with at least one upper case
                                                                                                                                                                   Lookbehind assertion
                                                                                                                                       ?!= 01
                                                                                                                                                                   Negative lookbehind
                                                                                                                                                                                                                                                                                                     letter, one lower case letter, and one
                                                                                                                                                                                                                                                                                                     digit (useful for passwords).
                                                                                                                                                                   Once-only Subexpression
```

(\w+@[a-zA-Z_]+?\.[a-zA-Z](2,6}) (\<(/?[^\>]+)\>) Email addresses

Note: These patterns are intended for reference purposes and have not been extensively tested. Please use with caution and test thoroughly before use.

rint('BASIC CPU INFORMATION:')

print(".....)

for element in memoryArguments:
 print(memory[element])

7()

?()|

7#

Condition [if then]

Comment

Available free from

AddedBytes.com

Condition [if then else]