Warum ich Python mag

eEducation Praxistage

PH Oberösterreich

Martin Weissenböck

12. - 13. März 2018

Verschiedene Sprachen

APL

```
      m1←2 3⍴⍳6
      m1
1 2 3
4 5 6
      m2←100×3 4⍴⍳12
      m2
100  200  300  400
500  600  700  800
900 1000 1100 1200
      m1 +.× m2
3800 4400  5000  5600
8300 9800 11300 12800
```

LISP

```
(defun factorial (n)
  (loop for i from 1 to n
        for fac = 1 then (* fac i)
        finally (return fac)))
```

PL/I

```
Hello2: proc options(main);
     put list ('Hello, world!');
end Hello2;
```

Welche ist die beste Sprache?

Statt sachlicher Diskussionen resultieren Argumente häufig

- … aus persönliche Präferenzen
- … aus unterschiedlichem Vorwissen
- … aus bereits vorhandenen fertigen Unterlagen
- ...

Welche ist die beste Sprache?

Zu klären ist, …

- … für welche Hardware,
  (Desktop-PC, Mobiltelefon, Raspberry, µP-Steuerung)
- … für welche Aufgabe,
  (Spiel, kommerzielles Programm, Webseite erstellen)
- … und aus pädagogischer Sicht
  für den Unterricht?

Addiere zwei Zahlen

```
import java.util.*;
public class einausgabe {
  public static void main(String[] args) {
          Scanner ScIn= new Scanner(System.in);
          System.out.print("1. Zahl: ");
          int i=ScIn.nextInt();
          System.out.print("2. Zahl: ");
          int j=ScIn.nextInt();
          int s=i+j;
          System.out.println("Die Summe von "\
   +i+" und "+j+" ist "+s);
  }
}
```

Addiere zwei Zahlen

```
import java.util.*;
public class einausgabe {
  public static void main(String[] args) {
        Scanner ScIn= new Scanner(System.in);
        System.out.print("1. Zahl: ");
        int i=ScIn.nextInt();
        System.out.print("2. Zahl: ");
        int j=ScIn.nextInt();
        int s=i+j;
        System.out.println(
          "Die Summe von "+i+" und "+j+" ist "+s);
  }
}
```

Oder so?

```
i = int(input("1. Zahl: "))
j = int(input("2. Zahl: "))
print(f"Die Summe von {i} und {j} ist {i+j}")
```

Böse 7

```
public class boese7 {
  public static void main(String[] args) {
    for (int i=1; i<100; i++) {
        if (i%10==7) continue;
        if (i/10==7) continue;
        if (i%7==0) continue;
        System.out.print(i+"  ");
    }
    System.out.println();
  }
}
```

Böse 7

```
public class boese7 {
  public static void main(String[] args) {
    for (int i=1; i<100; i++) {
        if (i%10==7) continue;
        if (i/10==7) continue;
        if (i%7==0) continue;
        System.out.print(i+"  ");
    }
    System.out.println();
  }
}
```

Wie wäre es denn damit?

```
for i in range(1,100):
    if i%10==7: continue
    if i/10==7: continue
    if i%7==0:  continue
    print i,
print
```

Die Antwort ist

Der Einstieg in Python

Worauf ist zu achten?

- Wenig Theorie
- Keine Verweise auf kommende Inhalte
  Java, C#... sind didaktisch ungeeignet
- Python 2 oder Python 3?

Einstieg

- Wenig Theorie
- Keine Verweise auf kommende Inhalte
  Java, C#... sind didaktisch ungeeignet
- Python 2 oder Python 3?
- Python als Taschenrechner
- Python mit Turtlegrafik!
- Python mit micro:bit

Python als Taschenrechner

Addition, Exponentiation

```
>>> 1+2
3
>>> 123456789 * 987654321
121932631112635269
>>> 2**8
256
```

Große Zahlen

```
>>> 2**100
1267650600228229401496703205376
>>> 2**1000
10715086071862673209484250490600018105614048117055336074437503883703510511249361224931983788156958581275946729175531468251871452856923140435984577574698574803934567774824230985421074605062371141877954182153046474983581941267398767559165543946077062914571196477686542167660429831652624386837205668069376
```

Noch größere Zahlen...

```
>>> 2**10000

19950631168807583848837421626835850838234968318861924548520089498529438830221946631919961684036194597899331129423209124271556491349413781117593785932096323957855730046793794526765246551266059895520550086918193311542508608460618104685509074866089624888090489894838009253941633257850621568309473902556912388065225096643874441046759871626985453222868538161694315775629640762836880760732228535091641476183956381458969463899410840960536267821064621427333394036525565649530603142680234969400335934316651459297773279665775606172582031407994198179607378245683762280037302885487251900834464581454650557929601414833921615734588139257095379769119277800826957735674444123062018757836325502728323789270710373802866393031428133241401624195671690574061419654342324638801248856147305207431992259611796250130992860241708340807605932320161268492288496255841312844061536738951487114256315111089745514203313820202931640957596464756010405845841566072044962867016515061920631004186422275908670900574606417856951911456055068251250406007519842261898059237118054444788072906395242548339221982707404473162376760846613033778706039803413197133493654622700563169937455508241780972810983291314403571877524768509857276937926433221599399876886660808368837838027643282775172273657572744784112294389733810861607423253291974813120197604178281965697475898164531258434135959862784130128185406283476649088690521047580882615823961985770122407044330583075869039319604603404973156583208672105913300903752823415539745394397715257455290510212310947321610753474825740775273986348298498340756937955646638621874569499279016572103701364433135817214311791398222983845847334440270964182851005072927748364550578634501100852987812389473928699540834346158807043959118985815145779177143619698728131459483783202081474982171858011389071228250905826817436220577475921417653715687725614904582904992461028630081535583308130101987675856234343538955409175623400844887526162643568648833519463720377293240094456246923254350400678027273837755376406726898636241037491410966718557050759098100246789880178271925953381282421954028302759408448955014676668389697996886241636313376393903373455801407636741877711055384225739499110186468219696581651485130494222369947714763069155468217682876200362777257723781365331611196811280792669481887201298643660768551639860534602297871557517947385246369446923087894265948217008051120322365496288169035739121368338393591756418733850510970271613915439590991598154654417336311656936031122249937969999226781732358023111862644575299135758175008199839236284615249881088960232244362173771618086357015468484058622329792853875623486556440536962622018963571028812361567512543338303270029097668650568557157505516727518899194129711337690149916181315171544007728650573189557450920330185304847113818315407324053319038462084036421763703911550639789000742853672196280903477974533320468368795868580237952218629120080742819551317948157624448298518461509704888027274721574688131594750409732115080498190455803416826949787141316063210686391511681774304792596709376
```

Division, Datentypen

```
>>> 7 / 4
1.75
>>> 7.0 / 4
1.75
>>> 7 // 4
1
>>> 7.0 // 4
1.0
>>> 7 % 4
3
```

Speichern

```
>>> b = 4347826086956521739l
>>> b = 434_782_608_695_652_173_913
>>> b * 2
869565217391304347826
>>> b * 3
1304347826086956521739
>>> b * 4
1739130434782608695652
>>> b * 5
2173913043478260869565
```

Mehr als ein Taschenrechner

Zeichenketten

```
>>> s = "abcdef"
>>> print (s)
abcdef

>>> s
'abcdef'

>>> s[0]
'a'

>>> s[5]
'f'
```

Indizes von hinten gerechnet

```
>>> s[-1]
'f'
>>> s[-2]
'e'
```

Arbeiten mit Zeichenketten

```
>>> 'abc'
'abc'
>>> "def"
'def'
>>> 'abc' + "def"
'abcdef'
>>> 'o' * 20
'oooooooooooooooooooo'
```

Slices

```
>>> s[0:2]
'ab'

>>> s[2:99]
'cdef'

>>> s[2:]
'cdef'
```

Slices mit negativen Indizes

```
>>> s[-3:-1]
'de'

>>> s[-3:0]
''

>>> s[-3:]
'def'

>>> s[:]
'abcdef'
```

Slices mit Schrittweite

```
>>> s[::2]
'ace'

>>> s[::1]
'abcdef'

>>> s[::-1]
'fedcba'
```

Tuples

```
>>> a = (2,3,5,7,11,13)
>>> a[1]
3

>>> a[:4]
(2, 3, 5, 7)

>>> a[:-2]
(2, 3, 5, 7)

>>> a[-2:]
(11, 13)
```

Listen

```
>>> a = [2,3,5,7,11,13]
>>> a[1]
3

>>> a[:4]
[2, 3, 5, 7]

>>> a[:-2]
[2, 3, 5, 7]

>>> a[-2:]
[11, 13]
```

Mengen

- Mengen vereinbaren

```
>>> a = {1,2,3}
>>> a
set([1, 2, 3])
>>> b = {2,4,6}
```

- Durchschnittsmenge

```
>>> a&b
set([2])
```

- Vereinigungsmenge

```
>>> a|b
set([1, 2, 3, 4, 6])
```

- Symmetrische Differenz

```
>>> a-b
set([1, 3])
>>> b-a
set([4, 6])
```

Zahlensysteme

```
>>> 10 * 3
30
>>> 0o10 * 3
24
>>> 0x10 * 3
48
>>> 0b10 * 3
6
>>> hex(0x10 * 3)
'0x30'
```

Programmabläufe

for-Schleife

```
>>> s = "abcdef"
>>> for x in s:
        print(x)

a
b
c
d
e
f
```

for-Schleife

```
>>> a = range(1,6)
>>> for x in a:
        print(x)


1
2
3
4
5
```

Comprehension

Liste der ganzen Zahlen von 1 bis 5

- Ohne List-Comprehension

```
>>> r = []
>>> for x in range(1,6):
r.append(x)

>>> print(r)
[1, 2, 3, 4, 5]
```

- Mit List-Comprehension

```
>>> [x for x in range(1,6)]
[1, 2, 3, 4, 5]
```

Diese Notation erlaubt sehr kompakte Programme!

Comprehension

Das „ganz kleine Ein-mal-Eins“ (1 bis 5)

- Ohne List-Comprehension

```
>>> ee = []
>>> for x in range(1,6):
        zeile = []
        for y in range(1,6):
            zeile.append(x*y)
        ee.append(zeile)

>>> print(ee)
[[1, 2, 3, 4, 5], [2, 4, 6, 8, 10], [3, 6, 9, 12, 15], [4, 8, 12, 16, 20], [5, 10, 15, 20, 25]]
```

- Mit List-Comprehension

```
>>> print([[x*y for y in range(1,6)] for x in range(1,6)])
[[1, 2, 3, 4, 5], [2, 4, 6, 8, 10], [3, 6, 9, 12, 15], [4, 8, 12, 16, 20], [5, 10, 15, 20, 25]]
```

Bedingungen

- Keine Klammern um die logischen Ausdrücke
- Strukturen durch Einrücken
- Logische Konstanten: True, False

```
if 2<3:
    print ("ja")
else:
    print ("nein")
```

Funktionen

```
>>> def kub(x):
        return x**3

>>> kub(2) + kub(3)
35
>>> for x in range(1,6):
        print(kub(x))


1
8
27
64
125
```

Ein komplettes Beispiel

Römische Zahlen (1)

```
>>> def arab2roem(a):
        tab = [(1000,'M'), (500,'D'), (100,'C'),
           (50,'L'), (10,'X'), (5,'V'), (1,'I')]
        res = ""
        for key,val in tab:
            while a>=key:
                a-=key
                res+=val
        return res

>>> print(arab2roem(2018))
MMXVIII
```

Römische Zahlen (2)

```
>>> def arab2roem(a):
        tab = [(1000,'M'),
           (900,'CM'), (500,'D'), (400,'CD'),(100,'C'),
           (90,'XC'), (50,'L'), (40,'XL'), (10,'X'),
           (9, 'IX'), (5,'V'), (4, 'IV'), (1,'I')]
        res = ""
        for key,val in tab:
            while a>=key:
                a-=key
                res+=val
        return res

>>> print(arab2roem(2019))
MMXIX
```

Große römische Zahlen

```
>>> def arab2roem(a):
        tab = [(1000000,'CCCCIↃↃↃ'), (500000,'IↃↃↃ'),
           (100000,'CCCIↃↃ'), (50000,'IↃↃ'),
           (10000,'CCIↃↃ'), (5000,'IↃↃ'), (1000,'CIↃ'),
           (500,'IↃ'), (100,'C'), (50,'L'),
           (10,'X'), (5,'V'), (1,'I')]
        res = ""
        for key,val in tab:
            while a>=key:
                a-=key
                res+=val
        return res

>>> print(arab2roem(2018))
CIↃCIↃXVIII
```

Was gibt es noch in Python?

Mehr Details zur Sprache

- Komplexe Zahl
  ```
  c = 3 + 4j
  ```
- Dictionary = assoziatives Array
  ```
  {'key1':val1, 'key2':val2}
  ```
- Mehrfache Entscheidung
  ```
  if ... elif ...elif ... else
  ```
- Exception
  ```
  try ... except ... else ... finally
  ```
- lambda-Funktion
  ```
  lambda a,b: a+b
  ```

Noch mehr Details zur Sprache

- Klasse
  ```
  class MeineKlasse(object):
  ```
- Bedingte Zuweisung
  ```
  a = 10 if b<2 else 20
  ```
- Aspektorientierte Programmierung
  ```
  @login
  ```
- Funktionale Programmierung
  ```
  def f():…
  g = f
  ```

Turtlegrafik

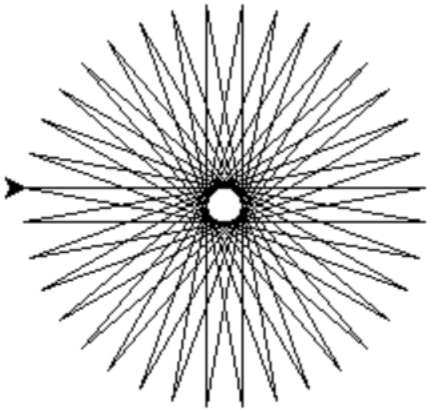
In Python 3 kann auch mit Turtles gearbeitet werden.

```
>>> from turtle import *
>>> for i in range(36):
        forward(200)
        right(170)
```

Turtlegrafik

In Python 3 kann auch mit Turtles gearbeitet werden.

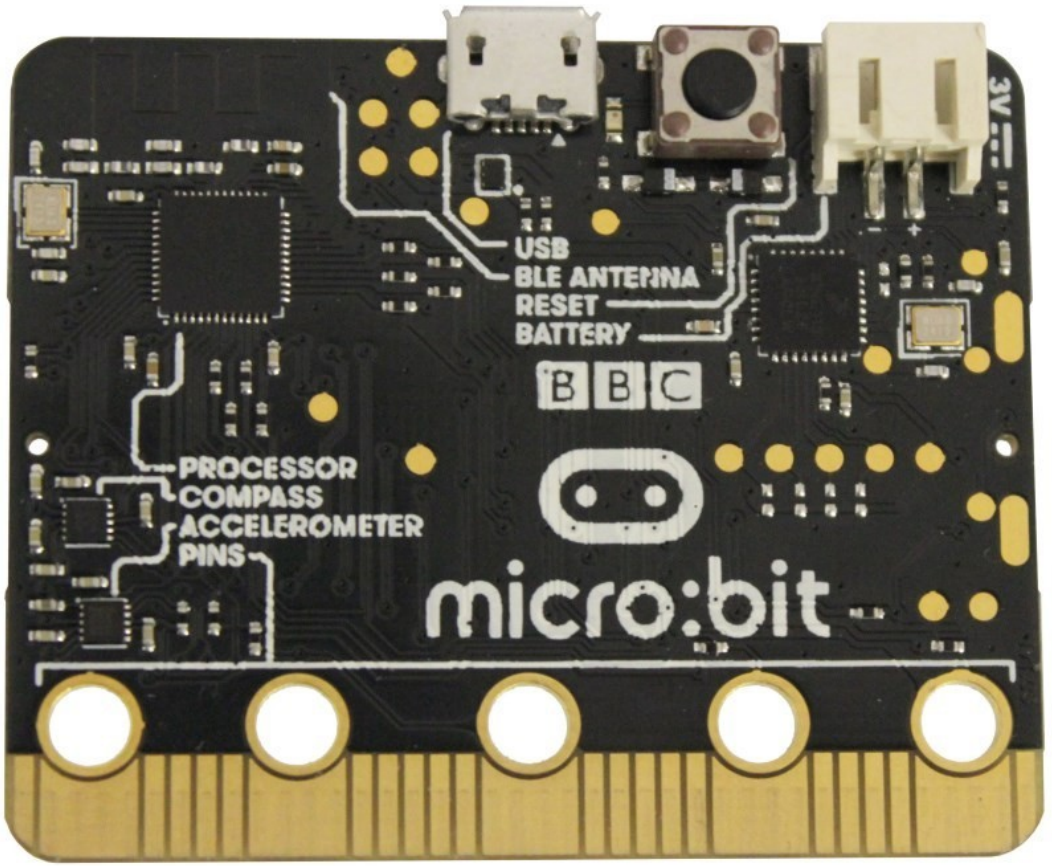
```
>>> from turtle import *
>>> for i in range(36):
        forward(200)
        right(170)
```



Einsatz von Python

Python zur Hardwaresteuerung

- BBC micro:bit
  http://microbit.org/de/
  http://python.microbit.org/



SCHUL.infosms

- Die Initiative „SCHUL.InfoSMS“ (http://www.infosms.org) stellt Applikationen für die Verbesserung der der Kommunikation zwischen Eltern und Schule zur Verfügung.
- Das gesamte Projekt ist in Python geschrieben.
- Als Webframework wird web2py verwendet.

SCHUL.infosms

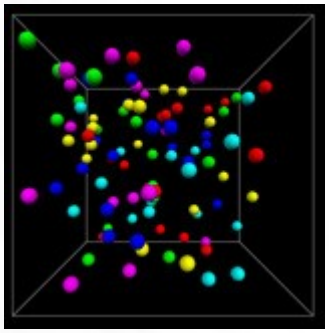
- Mitteilungen an die Eltern werden wahlweise per SMS, E-Mail oder Messenger-Programm übertragen.
- Vorübersetzte Texte erleichtern die Kommunikation mit nicht-deutschsprachigen Eltern.
- Interessenten werden gebeten, sich unter office@infosms.org zu melden!

Bibliotheken

- Umfangreiche Programmbibliotheken

```
import xx
from lib import xx
```

- Besonders nett: 3D-Bibliotheken
- Sehenswert:
  VPython (http://vpython.org)
- File-Bearbeitung



Bibliotheken

- GUI Toolkit: tkinter
  https://docs.python.org/3/library/tkinter.html#tkinter-modules
- Symbolische Mathematik mit Python: SymPy
  http://www.sympy.org/en
- C-Bibliotheken aus Python aufrufen
  https://docs.python.org/3/library/ctypes.html

Didaktik

Einige Überlegungen

- Welcher Zugang zum Programmieren ist motivierend?
- Wie sieht es mit der Fähigkeit zum abstrakten Denken aus?
- Auf bekannte Konzepte (z. B. Taschenrechner) zurück greifen

Am Beispiel 9. Schulstufe:

Der Schüler muss...

- eine meist sprachlich (und nicht mathematisch-technisch) formulierte Aufgabenstellungen verstehen (laut PISA-Studie eine Schwäche)
- daraus einen dynamischen Ablauf (eine Algorithmus) erzeugen können, nachdem er jahrelang auf geschlossene mathematische Lösungen hin trainiert wurde - er muss algorithmisch denken lernen

... und ...

- die Syntax der Programmiersprache lernen
- die Semantik der Programmiersprache lernen
- mehr oder weniger viele „notwendige“ Programmbausteine akzeptieren, die zu einem bestimmten Zeitpunkt noch gar nicht erklärt werden können
- bei JavaScript zwischen Programmiersprache und HTML-Code unterscheiden lernen
- das Testen eines Programms lernen

... und ...

- die Bedienung des Editors lernen (ist wohl das geringste Problem)
- Aufgaben als Projekte (die mehrere Files enthalten können) verstehen lernen
- mit der Entwicklungsumgebung umgehen lernen

Und das alles im Alter von 14 bis 15 Jahren, in dem die Fähigkeit zum abstrakten Denken noch nicht besonders stark ausgeprägt sind.

Warum Python?

- Google
  http://quintagroup.com/cms/python/google
- Google API
  https://cloud.google.com/ml-engine/docs/tutorials/python-guide?hl=de
- Spitzenwerte bei der Softwarequalität
  https://www.heise.de/developer/meldung/Python-erreicht-Spitzenwert-bei-Softwarequalitaet-1948541.html
- Beliebteste Programmiersprache
  https://derstandard.at/2000061701290/Python-laut-Analyse-beliebteste-Programmiersprache

Zusammenfassung

- Die Programmiersprache für den Unterricht sorgfältig wählen
- *Unterricht darf Spaß machen*

  Weigl: Ironisches schreibt man in Österreich kursiv

Weiterführendes Material

Python Quellen

- Python
  https://www.python.org/
- Diplom-Arbeit
  "Die Eignung von Python für Einführungskurse in das Programmieren im Vergleich zu anderen Programmiersprachen":

  http://repositum.tuwien.ac.at/obvutwhs/content/titleinfo/1642104?lang=de
- Bernd Klein, Einführung in Python 3
  Hanser Verlag:
  http://www.hanser-fachbuch.de/buch/Einfuehrung+in+Python+3/9783446452084

Python Quellen

- Python Basiskurs für Ein- und Umsteiger
  Basiswissen für einen Zwei-Tage-Kurs
  Herdt-Verlag

  https://shop.herdt.com/at/product/TE-PYT-G_RS
- Python – Weiterführende Themen
  Herdt-Verlag

  https://shop.herdt.com/at/product/TE-PYT-F_RS

Web Framework

- Django
  http://python.microbit.org/v/1

- Web2py
  http://www.web2py.com/

Editoren

- Thonny: Minimalistisch und sehr effizient. Besonders gut für Anfänger geeignet.
  http://thonny.org/
- Visual Studio: auch für Python verwendbar
  https://code.visualstudio.com/
- Eclipse: Gut geeignet zur Verwaltung größerer Projekte. Für Anfänger sehr unübersichtlich, viel zu überladen.
  http://www.chip.de/downloads/Eclipse_24547633.html

Python – Wikipedia (1)

Python ([ˈpaɪθn], [ˈpaɪθɑn], auf Deutsch auch [ˈpyːtɔn]) ist eine universelle, üblicherweise interpretierte höhere Programmiersprache. Python unterstützt mehrere Programmier-Paradigmen, z. B. die objektorientierte, die aspektorientierte und die funktionale Programmierung.

Ferner bietet es eine dynamische Typisierung. Wie viele dynamische Sprachen wird Python oft als Skriptsprache genutzt.

https://de.wikipedia.org/wiki/Python_(Programmiersprache)

Python – Wikipedia (2)

Ihre Entwurfsphilosophie betont Programmlesbarkeit, außerdem ist Python-Code im Vergleich mit anderssprachigem Code teilweise deutlich kürzer.

Zur besseren Lesbarkeit soll auch der Verzicht auf geschweifte Klammern zur Bildung von Code-Blöcken dienen, da die Programmstruktur durch Einrückungen gebildet wird.

Python – Wikipedia (3)

Die Sprache hat ein offenes, gemeinschafts-basiertes Entwicklungsmodell, das durch die gemeinnützige Python Software Foundation, die de facto die Definition der Sprache in der Referenzumgebung CPython pflegt, gestützt wird.

Python gilt als einfach zu erlernende Sprache, da sie über eine klare und übersichtliche Syntax verfügt. Ferner besitzt sie eine umfangreiche Standardbibliothek und zahlreiche Pakete im Python Package Index.

Warum ich Python mag

- Zum Nachlesen:
  Coding-Special 2017: Ein Sonderheft des Bildungsministeriums
  http://pubshop.bmbf.gv.at/detail.aspx?id=666

Danke für Ihr Interesse!