

Python

- skriptovací jazyk
- objektově orientovaný
- open source
- přenositelný
- dá se lepit s C a Fortranem
- dobře se učí
- je výkonný
 - dynamické typování
 - šikovné standartní typy a mocné nástroje pro práci s nimi
 - množství standartních knihoven a spousta dalších (mj. NumPy, SciPy, ObsPy)
 - automatická správa paměti
 - elegantní ošetření chyb

Datové typy: čísla a řetězce

```
In [ ]: i=2
        print type(i)
        f=3.14
        print type(f)
        print i*f
        c=2+3j
        print c**2
        print abs(-42), 2**8, 2**0.5
```

```
In [ ]: s='cokolada'
        print s+'!'
        print "Ne!"*4
        print len(s), s[:], s[0:4], s[:-1]
        print 'kol' in s

        block=""
        A
        B
        C
        print len(block)
```

Moduly

Vytvořme soubor 'mymodule.py' s následujícím řádkem:

```
nadpis = 'Seminar o seismologickem softwaru'
```

```
In [ ]: #print nadpis # chyba
import mymodule
#print nadpis # opet chyba
print mymodule.nadpis
import mymodule as m
print m.nadpis
from mymodule import nadpis
print nadpis
```

```
In [ ]: s='cokolada'
print s.upper(), s.find('kol')
import string
print string.upper(s), string.find(s, 'kol')
print string.__doc__ # Pokud jste to jeste nepouzili, funguje a
```

Seznamy, slovníky a n-tice (lists, dictionaries and tuples)

```
In [ ]: # seznamy - usporadane posloupnosti objektu
L1 = [1, 2, 3, 3.14, 4, 1E3, -4]
L1.append(5) # pridame prvek na konec
print L1[1:3] # cislovani jako u stringu,, t.j. vytiskne [2, 3]
L1.sort() # seradime
print L1
del L1[0]
print L1
print 1 in L1, 0 in L1

SELMY = ['medved', 'rys', 'vlk'] # pozor, Python je case-senzit.
print len(SELMY) #
SELMY.reverse()
print SELMY
print SELMY.index('rys')
print SELMY*2
```

```
In [ ]: # slovníky - neusporadane dvojice klic-hodnota (asociativni pol.
d = {'jmeno' : 'Jan', 'prijmeni' : 'Novak'}
d['vek']=0
print d
print d.keys()
print d.values()
```

```
In [ ]: # n-tice (tuple) - konstantni seznamy
t = (0, 1, 2, (-1j, 'a'))
```

```
print t, len(t)
```

Výrazy a operátory

```
In [ ]: a,b = 1,2
print a, b
print a<b, a<=b, a==b, a!=b
print a<b and a>0, a<b or a>b, a<b and not a>b
a,b=b,a
print a,b
c=b=a
print a,b,c
```

Řídící struktury

```
In [ ]: # if
# logické bloky začínají dvojtečkou a jsou určeny odsazením
ls = ['a', 2, 3.1]
if not ls:
    print 'empty list'
elif len(ls) == 1:
    print 'list with one item!'
else:
    print 'this list has more than one item!'
```

```
In [ ]: # while
y = 259 # testované číslo - je prvočíslo?
x = y / 2 # celocíselné dělení
while x > 1:
    if y%x == 0: # zbytek po celocíselném dělení
        print y, 'je dělitelné', x
        break
    x -= 1 # x = x-1
else:
    print y, 'je prvočíslo'
```

```
In [ ]: # for
mylist = ['abcdef', 10, 'hijk', 100.1, 1+1j]

for item in mylist:
    print item, type(item)
```

```
# vypsát lichá čísla do 30 s výjimkou 27
for i in range(1, 30, 2):
    if i == 27:
        continue
    print i
```

Funkce

```
In [ ]: def even_odd(number):
        if number % 2 == 0:
            return str(number)+' je sude'
        else:
            return str(number)+' je liche'

print even_odd(5)
print even_odd(4)
```

Ošetření chyb

```
In [ ]: def deleni(x, y):
        try:
            print x/y
        except:
            print "deleni nulou"

deleni(10,2)
deleni(10,0)
```