



# Cvičení 4

**Jan Caha**

jan.caha@upol.cz

KGI/PRG1 - Programování 1

**Katedra Geoinformatiky  
Univerzita Palackého v Olomouci**



# Python 3.4.0

- nová verze
- <http://docs.python.org/3.4/>
- co se změnilo
- pozor při čtení dokumentace!



# Obsah hodiny

---

- domácí úkol
- práce s řetězci
- práce se seznamy
- funkce



# Rěťezce

- reprezentace textu v programu
- nejrůznější využití
- `http://docs.python.org/3/library/stdtypes.html`
- ukázka dostupných funkcí
- příklady

# Rěťezce - funkce

- `str.find(sub[, start[, end]])`, podobně operator *in*
- `str.isdecimal()`
- `str.isnumeric()`
- `str.replace(old, new[, count])`
- `str.rfind(sub[, start[, end]])`
- `str.split(sep=None, maxsplit=-1)`
- `str.splitlines([keepends])`
- `str.strip([chars])`
- zjištění délky řetězce - funkce `len(str)`



# Seznam

- datová struktura
- příklad - `a = [66.25, 333, 333, 1, 1234.5]`
- dokumentace `http://docs.python.org/3/tutorial/datastructures.html`
- skrze jeho prvky lze iterovat
- jednotlivé prvky lze odkazovat jako `a[i]`



# Seznam pro cyklus

---

```
for i in [1, 2, 3]:  
    print(i)
```



# Seznam pro cyklus

```
names = ["Bob", "Alice", "John"]  
print(names[2])  
  
for name in names:  
    print(name)
```





# Seznam - důležité funkce

- `list.append(x)`
- `list.insert(i, x)`
- `list.remove(x)`
- `list.pop([i])`
- `list.index(x)`
- `list.count(x)`
- `list.sort()`
- příkaz - `del a[2:4]`
- `len(list)` - počet prvků v seznamu

# Seznam v seznamu

- často používané konstrukce
- v Pythonu neexistují pole jako takové - vyžívají se seznamy
- dvojrozměrné pole - seznam v seznamu
- příklad -  $a = [66.25, 333, 333, 1, 1234.5]$ ,  $b = [1, 2, 5, 8]$
- $c = [a, b]$
- pro iterování se používají vnořené cykly
- ukázka

# Funkce

- bloky kódu, které se vícekrát opakují
- definují se názvem a seznamem argumentů
- klíčová slova def a return
- dva typy funkcí
  - funkce, které vrací hodnotu
  - funkce bez návratové hodnoty (často pouze tisknou výstup)



# Funkce - ukázka

```
def sectiUmocni (a,b):  
    result = a + b  
    result = result ** 2  
    return(result)
```



# Funkce - ukázka

```
def sectiUmocni (a,b):  
    result = a + b  
    result = result ** 2  
    return(result)  
  
print(sectiUmocni(1,2))  
print(sectiUmocni(5,4))  
print(sectiUmocni(15,8))  
print(sectiUmocni(214,519))
```

# Definice funkce v externím souboru

- funkce nemusí být zapsána přímo v souboru, kde ji využíváme
- naopak je žádoucí, aby byla uložena samostatně
- často využíváme jednu funkci napříč mnoha programy
- import buď celého souboru (víc funkcí) nebo přímo vybraných funkcí - rozdílné volání funkcí

# Fibonacci.py

```
def fib(n):    # write Fibonacci series up to n
    """Print a Fibonacci series up to n."""
    a = 0
    b = 1
    while a < n:
        print(a, end=' ')
        temp = a + b;
        a = b;
        b = temp;
    print()
```



# FibonacciRun.py

```
import Fibonacci

print(Fibonacci.fib(5))
print(Fibonacci.fib(50))
print(Fibonacci.fib(100))
print(Fibonacci.fib(1000))
```





# FibonacciRun2.py

```
from Fibonacci import fib
```

```
print(fib(5))
```

```
print(fib(50))
```

```
print(fib(100))
```

```
print(fib(1000))
```

# Poznámka!!!

- v rámci úkolů budeme zatím zanedbávat kontrolu, zda-li jsou vstupy funkcí opravdu tím, čím mají být
- jedná se pouze o dočasné zjednodušení, které v praxi nemá místo - je vždy nutné ověřovat vstupy
- pokud se ověření neudělá, může se funkce chovat neočekávaně



# Úkol 1

Napište funkci, která u vstupní hodnoty zkontroluje zda-li je toto číslo celočíselně beze zbytku dělitelné čísly od 3 do 9 (použijte cyklus) a pro každou hodnotu vypíše že hodnota buď je nebo není celočíselně dělitelná číslem i.

Funkci otestuje na vstupu od uživatele.



# Úkol 1 - řešení

```
def delitelnost(cislo):  
    for x in range(3,10):  
        if cislo%x == 0:  
            print ("Cislo",cislo, "JE  
bezezbytku delitelne",x)  
        else:  
            print ("Cislo",cislo, "NENI  
bezezbytku delitelne",x)  
  
cislo = int(input("Zadej vstupni hodnotu: "))  
delitelnost(cislo)
```