

České vysoké učení technické v Praze FIT

# Programování v Pythonu

Jiří Znamenáček

*Příprava studijního programu Informatika je podporována projektem financovaným z Evropského sociálního fondu a rozpočtu hlavního města Prahy.*

*Praha & EU: Investujeme do vaší budoucnosti*



# Čísla

1. Vypište všechna čísla mezi 2 a 100, která jsou dělitelná buď 3 nebo 7.

**[-] řešení ([typy/cisla/01.py](#))**

```
for i in range(2, 101):  
    if i % 3 == 0 or i % 7 == 0:  
        print(i)
```

2. Upravte předchozí program, aby vypisoval čísla, která jsou dělitelná 3 nebo 7, ale ne oběma najednou.

**[-] řešení ([typy/cisla/02.py](#))**

```
for i in range(2, 101):  
    if (i % 3 == 0 or i % 7 == 0) and (i % 3 != 0 or i % 7  
!= 0):  
        print(i)
```

3. Vypište všechna prvočísla mezi 2 a 100. (Použijte lepší algoritmus, než je ve slajdech :)

**[-] řešení ([typy/cisla/03a.py](#))**

```
import math  
  
for n in range(2, 101):  
    for x in range(2, int(math.sqrt(n)) + 1):  
        if n % x == 0:  
            break  
    else:  
        print(n, end=' ')
```

4. „Implementujte“ umocňování celočíselným exponentem. Zkuste to vyřešit na méně než (n-1) násobení.

**[-] řešení ([typy/cisla/04a.py](#))**

```
# pouze ukázkové zadání
mocnenec = 3.14
mocnitel = 5

# proved' „umocňování“
vysledek = mocnenec
for i in range(mocnitel-1):
    vysledek *= mocnenec

# vypiš výsledek
txt = "Výsledek: {0}^{1} = {2}".format(mocnenec, mocnitel,
vysledek)
print( txt )
```

**[-] řešení ([typy/cisla/04b.py](#))**

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>404 Not Found</title>
</head><body>
<h1>Not Found</h1>
<p>The requested URL /prikklady/typy/cisla/04b.py was not
found on this server.</p>
<hr>
<address>Apache/2.2.14 (Ubuntu) Server at vyuka.hotaru Port
80</address>
</body></html>
```

5. Spočítejte faktoriál daného čísla.

**[-] řešení ([typy/cisla/05.py](#))**

```
# vstup
n = int( input('Zadejte celé číslo: ') )

# výpočet
f = 1
for i in range(2, n+1):
    f *= i

# vypiš výsledek
print( 'Výsledek: {0}! = {1}'.format(n, f) )
```

6. Vypište daný počet členů Fibonacciho posloupnosti, tj. řady 1, 1, 2, 3, 5, 8...

**[–] řešení ([typy/cisla/06a.py](#))**

```
LIMIT = 13

a, b = 0, 1
for i in range(LIMIT):
    print(b, end=' ')
    a, b = b, a+b
```

**[–] řešení ([typy/cisla/06b.py](#))**

```
LIMIT = 13

def fib(n):
    if n <= 1:
        return n
    else:
        return fib(n-1) + fib(n-2)

for i in range(1, LIMIT+1):
    print(fib(i), end=' ')
```

7. Vyřešte iterací rovnici  $x^3 - x - 1 = 0$  s rozumnou přesností.

**[–] řešení ([typy/cisla/07.py](#))**

```
x = 2
oldx = 1 # startovní hodnota
presnost = 0.000001

while abs(oldx-x) > presnost:
    oldx = x
    x = (1+x)**(1/3.0) # iterace pro x
    print("Kořen {0} se liší od předchozího o {1}".format(x, abs(oldx-x)) )
```

8. Vstupní řetězec jedniček a nul představující číslo ve dvojkové soustavě převedte na číslo v soustavě desítkové.

**[-] řešení ([typy/cisla/08.py](#))**

```
# vstup
ns = input('Zadejte číslo v binárním tvaru: ')

# výpočet
vysledek = 0
for i in ns:
    vysledek *= 2
    vysledek += int(i)
print( '{0} binárně = {1} desítkově'.format(ns, vysledek) )
```

**9.** Doplněte předchozí skript o ověření, že vstupní řetězec je opravdu zápis čísla ve dvojkové soustavě.

**[-] řešení ([typy/cisla/09.py](#))**

```
import sys

# vstup
ns = input('Zadejte číslo v binárním tvaru: ')

# ověření podmínky
for ch in ns:
    if (ch != '0') and (ch != '1'):
        print( 'Zadaný řetězec není obrazem čísla v binárním tvaru.' )
        sys.exit()

# výpočet
vysledek = 0
for i in ns:
    vysledek *= 2
    vysledek += int(i)
print( '{0} binárně = {1} desítkově'.format(ns, vysledek) )
```

**10.** Napište opačnou funkci – vstupní číslo v desítkové soustavě převeďte do soustavy dvojkové.

**[-] řešení ([typy/cisla/10.py](#))**

```
# vstup
x = int( input('Zadejte číslo v desítkovém tvaru: ') )

# výpočet
num = x
binary = ""
while num:
    binary = str(num & 1) + binary
    num >>= 1

print( '{0} desítkově = {1} binárně'.format(x, binary) )
```

**11.** Za pomoci komplexních čísel napište tříparametrický skript, který vrátí souřadnice zadaného bodu v rovině otočené o zadaný úhel kolem počátku.

**[-] řešení ([typy/cisla/11.py](#))**

```
def rotate(x, y, fi_deg):
    """Otočí číslo [x,y] o úhel fí (zadaný ve stupních)."""

    import math

    # otáčený bod
    cislo_k_otoceni = complex(x, y)

    # přepočet úhlu na radiány
    fi_rad = uhel_v_radianech = math.radians(fi_deg)
    # zavedení rotoru
    rotor = complex( math.cos(fi_rad), math.sin(fi_rad) )

    # provedení otočení
    cislo_otocene = cislo_k_otoceni * rotor

    # převod zpět na souřadnice a zaokrouhlení na 3
    desetinná místa
    zx = round(cislo_otocene.real, 3)
    zy = round(cislo_otocene.imag, 3)

    return zx, zy

# testy
print( rotate(0, 0, 11) )
print( rotate(1, 1, 45) )
print( rotate(0, 1, 180) )
```

---

Autorem původního zadání i řešení pro Python 2.x u příkladů 1-3, 7, 8 a 10 je Bedřich Košata.