

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



Seznamy

1. Vraťte součet všech prvků v seznamu zadaném jako argument funkce.

[-] řešení ([typy/seznamy/secti.py](#))

```
def secti(xs):  
    ret = 0  
    for x in xs:  
        ret += x  
    return ret  
  
# ukázkové zadání  
xs = [1, 2, 3, 1, 2, 1, 2, 4, 6, 2,]  
  
print(xs)  
print( secti(xs) )
```

2. Vypište zadaný seznam v opačném pořadí bez použití metody *reverse()*.

[-] řešení ([typy/seznamy/otoc.py](#))

```
def otoc(xs):  
    ret = []  
    for x in xs:  
        ret.insert(0, x)  
    return ret  
  
# ukázkové zadání  
xs = [1, 2, 3, 1, 2, 1, 2, 4, 6, 2,]  
  
print(xs)  
print( otoc(xs) )
```

3. Napište funkci, která bere jako první argument seznam celých čísel *xs* a jako druhý jedno celé číslo *y*. Výstupem funkce je seznam *ys* obsahující ta čísla z *xs*, která jsou větší než *y*.

Ukázka: $f([1,2,3,1,2,1,2,4,6,2], 3) \Rightarrow [4,6]$

[-] řešení ([typy/seznamy/01.py](#))

```
def bigger_than(xs, y):  
    ret = []  
    for x in xs:  
        if x > y:  
            ret.append(x)  
    return ret  
  
# ukázkové zadání  
xs = [1, 2, 3, 1, 2, 1, 2, 4, 6, 2,]  
n = 3  
  
print(xs, n)  
print( bigger_than(xs, n) )
```

4. Napište funkci, která bere jako argument seznam celých čísel *xs* a vrací seznam *ys*, ve kterém jsou odstraněny všechny duplikáty prvků v seznamu *xs*.

Ukázka: $f([1,2,3,1,2,1,2,4,6,2]) \Rightarrow [1,2,3,4,6]$

[-] řešení ([typy/seznamy/02.py](#))

```
def remove_duplicates(xs):  
    ret = []  
    for x in xs:  
        if not x in ret:  
            ret.append(x)  
    return ret  
  
# ukázkové zadání  
xs = [1, 2, 3, 1, 2, 1, 2, 4, 6, 2,]  
  
print(xs)  
print( remove_duplicates(xs) )
```

5. Napište funkci, která bere jako první argument seznam celých čísel *xs* a jako druhý jedno celé číslo *y*. Výstupem funkce je seznam všech pozic prvku *y* v seznamu *xs*.

Ukázka: $f([1,2,3,1,2,1,2,4,6,2], 2) \Rightarrow [1,4,6,9]$

Ukázka: $f([1,2,3,1,2,1,2,4,6,2], 5) \Rightarrow []$

[-] řešení ([typy/seznamy/03.py](#))

```
def indexes(xs, y):
    ret = []
    i = 0
    for x in xs:
        if x == y:
            ret.append(i)
            i = i + 1
    return ret

# ukázkové zadání
xs = [1, 2, 3, 1, 2, 1, 2, 4, 6, 2,]
n = 2
print(xs, n)
print( indexes(xs, n) )

n = 5
print(xs, n)
print( indexes(xs, n) )
```

6. Napište funkci, která bere jako argument seznam celých čísel *xs* a vrátí ten prvek, který je jako první roven svému následovníkovi. Neexistuje-li takový prvek, vrátí *None*.

Ukázka: $f([0,3,4,5,5,4,2,2,4]) \Rightarrow 5$

Ukázka: $f([0,3,4,5,4,2,4]) \Rightarrow \text{None}$

[-] řešení ([typy/seznamy/04.py](#))

```
def doubles(xs):
    i = 0
    for x in xs:
        if i+1 < len(xs) and x == xs[i+1]:
            return x
        i = i + 1
    return None

# ukázkové zadání
xs = [0, 3, 4, 5, 5, 4, 2, 2, 4,]
print(xs)
print( doubles(xs) )

xs = [0, 3, 4, 5, 4, 2, 4,]
print(xs)
print( doubles(xs) )
```

7. Napište funkci, která bere jako argument seznam celých čísel *xs* a vrátí seznam takových prvků z *xs*, které jsou rovny svým následovníkům.

Ukázka: $f([0,3,4,5,5,4,2,2,4]) \Rightarrow [5,2]$

Ukázka: $f([0,3,4,5,4,2,4]) \Rightarrow []$

[-] řešení ([typy/seznamy/05.py](#))

```
def doubles(xs):
    ret = []
    i = 0
    for x in xs:
        if i+1 < len(xs) and x == xs[i+1]:
            ret.append(x)
            i = i + 1
    return ret

# ukázkové zadání
xs = [0, 3, 4, 5, 5, 4, 2, 2, 4,]
print(xs)
print( doubles(xs) )

xs = [0, 3, 4, 5, 4, 2, 4,]
print(xs)
print( doubles(xs) )
```

Autorem původního zadání i řešení pro Python 2.x u příkladů 3-7 je
Bedřich Košata.