

**Janka Marschalkova**  [9:03 AM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1679987024094249)  
joined #05-kód-z-lekce. Also, Jakub Červinka and 18 others joined.



**Janka Marschalkova**  [6:54 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680022484378499)

print("Vítejte v dlouhodobém kurzu Pythonu!") # retezec, string, str

nazev\_hry = "Romeo a Julie"

cena\_listku = 150

pocet\_listku = 5

celkova\_cena = cena\_listku \* pocet\_listku

print(f"Cena všech lístku je {celkova\_cena}")

(edited)



**Janka Marschalkova**  [7:00 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680022836920079)

pocet\_listku = int(input("Zadej počet lístků"))

(edited)



**Janka Marschalkova**  [7:06 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680023203863179)

if pocet\_listku >= 3:

celkova\_cena = celkova\_cena \* 0.9

print(f"Zlevněná cena je {celkova\_cena} z původní {celkova\_cena}")

else:

print(f"Cena všech lístků je {celkova\_cena}")

(edited)

[7:11](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680023464984519)

Zadání cvičení: :star2: Využijte příklady z výkladu. Upravte je tak, aby  
- se program zeptal uživatele i na věk, a ověříš, že má alespoň 13 let. Pokud nemá alespoň 13 let, nezeptá se už na počet lístků a skončí(tj. nákup neproběhne).  
- bonus pro netrpělivé :) viz výše + místo slevy 10% uplatní akci "při nákupu alespoň 3 lístků je jeden lístek zdarma"



**Janka Marschalkova**  [8:09 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680026997345119)

print("martin".upper())

print(' martin '.strip())

print('po ut st čt pá'.split(' '))

print('+'.join(['1', '2', '3', '4']))

print('/'.join(['dokumenty', 'dapraha', 'python', 'priklady']))

text = "Kurz vede lektor Marek"

novy\_text = text.replace("Marek", "Janka")

print(novy\_text)

[8:10](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680027027326299)

# Slicing

print("Dobrý" + " " + "den")

věnečky = [1, 2, 1, 2, 6, 3, 2]

print(věnečky[0], věnečky[3])

print(věnečky[2])

# syntaxe pythonu venečky[začátek:konec+1]

# od pondělí do pátku

print(věnečky[0:5])

# od úterý do neděle

# od úterý do konce sekvence (do neděle)

print(věnečky[1:])

# od začátku do čtvrtku

print(věnečky[:4])

[8:10](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680027039890529)

Další úkol v druhé BR bude následující: Mějme seznam pohybů na nějakém bankovním účtu:

pohyby = [1200, -250, -800, 540, 721, -613, -222]

* Vypište v pořadí třetí pohyb z uvedeného seznamu.
* Vypište všechny pohyby kromě prvních dvou.
* Vypište kolik je všech pohybů dohromady.
* Pomocí volání vhodných funkcí vypište nejvyšší a nejnižší pohyb.
* Spočítejte celkový přírůstek na účtu za dané období. Pozor, že přírůstek může vyjít i záporný.

**Thursday, March 30th**



**Janka Marschalkova**  [10:46 AM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680165986073369)

**Řešení z lekce:**  
**1. breakoutroom:**

vek = int(input("Zadej věk: "))

cena\_listku = 150

if vek >= 13:

pocet\_listku = int(input("Zadej počet lístků: "))

celkova\_cena = cena\_listku \* pocet\_listku

print(f"Celková cena je {celkova\_cena}")

else:

print(f"V případě, že jsi mladší než 13 let, nemáš bohužel povolení k nákupu lístků.")

**Bonus:**

if vek >= 13:

pocet\_listku = int(input("Zadej počet lístků: "))

celkova\_cena = cena\_listku \* pocet\_listku

zlevnena\_cena = celkova\_cena \* 0.9

print(f"Zlevněná cena je {zlevnena\_cena} z původní {celkova\_cena}")

else:

print(f"V případě, že jsi mladší než 13 let, nemáš bohužel povolení k nákupu lístků.")

[10:46](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680166014357149)

**2. breakoutroom:**

pohyby = [1200, -250, -800, 540, 721, -613, -222]

# v poradi treti

print(pohyby[2])

# vsechny krome dvou prvnich

print(pohyby[2:])

# vsechny dohromady

print(len(pohyby))

# nejvyssi a nejnizsi

print(f'nejnizsi je {min(pohyby)}, nejvyssi {max(pohyby)}')

# celkovy prirustek

print(f'zustatek = {sum(pohyby)}')

 (edited)

**Monday, April 3rd**



**Kateřina Smolová**  [9:00 AM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680505254762879)  
was added to #05-kód-z-lekce by Janka Marschalkova.

**Tuesday, April 4th**



**Janka Marschalkova**  [6:56 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680627404194059)

# Pekarna na položky a ceny v korunách

pekarna = {

"houska": 20,

"rohlik": 5,

"chleba": 40,

"loupak": 20

}

pekarna = {

("houska", "bílá"): 10,

("houska", "celozrnná"): 15,

}

print(pekarna[("houska", "bílá")])

divadlo = {

1: "Romeo a Julie",

2: "Divadle "

}

produkty = ["houska", "rohlik", "chleba", "loupak"]

print(pekarna)

print(pekarna["chleba"])

klic = "rohlik"

print(f'Klic: {klic}, hodnota: {pekarna[klic]} korun.')

print("Klic: " + klic + ", hodnota:" + str(pekarna[klic]) + "korun.")

# vypis hodnota

#polozka = input("Zadej produkt")

#print(f'{produkt} stojí {pekarna[produkt]} korun.')

#if polozka in pekarna: # Produkt je v pekárne = klic je ve slovníku

# print(f'{polozka} stojí {pekarna[polozka]} korun.')

#else: # Produkt není v pekárně = klic není ve slovníku

# print(f'Bohužel, produkt {polozka} neprodáváme')

#Přidání do slovníku

pekarna["zakusek"] = 35

print(pekarna)

#Odebírání ze slovníku

cena\_houska = pekarna.pop("houska")

print(pekarna)

print(cena\_houska)



**Janka Marschalkova**  [8:12 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680631932198559)

sales = {

"Zkus mě chytit": 4165,

"Vrah zavolá v deset": 5681,

"Zločinný steh": 2565,

}

for kniha in sales:

print(kniha)

pripadne print(f'Knihy s nazvem {kniha} se prodalo {sales[kniha]} ks.')

tombola = {

7: "Láhev kvalitního vína Château Headache",

15: "Pytel brambor z místního družstva",

23: "Čokoládový dort",

47: "Kniha o historii města",

55: "Šiška salámu",

67: "Vyhlídkový let balónem",

79: "Moderní televizor",

91: "Roční předplatné městského zpravodaje",

93: "Společenská hra Sázky a dostihy",

}

for ... in sales:

print(f'Listek {...} vyhrava {tombola[...]}.')

# Jaké jsou klíče ve slovníku?

print(sales.keys())

# Jaké jsou hodnoty ve slovníku?

print(sales.values())

#Kolik se průmerne prodalo vytisku na knihu?

print(sum(sales.values()) / len(sales))

print(sales.items())

for nazev, prodano in sales.items(): #Dvojice klic, hodnota

print(f'Knihy s nazvem {nazev} se prodalo {prodano} ks.')



**Janka Marschalkova**  [9:04 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680635076757609)

**Řešení z 2. lekce**

[9:08](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680635299935449)

**Slovníky**

1 - Vysvědčení ○○○○♦

Vytvoř slovník, který reprezentuje vysvědčení.

Klíč slovníku bude název předmětu a hodnota známka z daného předmětu.

Pro zjednodušení vlož do slovníku pouze tři předměty (například český jazyk, matematiku a dějepis).

Vypiš obsah slovníku pomocí funkce print().

Řešení

vysvedceni = {

'matematika': 1,

'cestina' : 3,

'dejepis' : 2,

}

2 - Detektivky ○○○♦♦

Vydavatel detektivek si eviduje prodané kusy u jednotlivých knih.

V následujícím slovníku najdeš tři knihy a u každé z nich je počet prodaných kusů.

prodano = {

'Zkus mě chytit' : 4165,

'Vrah zavolá v deset': 5681,

'Zločinný steh' : 2565,

}

Zkopíruj si slovník do svého programu.

Přidej do slovníku nově vydanou detektivku "Noc, která mě zabila", která zatím nebyla uvedena na trh, je tedy prodáno 0 kusů.

U knihy "Vrah zavolá v deset" zvyš počet prodaných kusů o 100.

Řešení

prodano['Noc, která mě zabila'] = 0

prodano['Vrah zavolá v deset'] += 100

3 - Tombola ○○♦♦♦

V následujícím slovníku jsou uložena čísla lístků tomboly a příslušné výhry.

tombola = {

7 : 'Láhev kvalitního vína Château Headache',

15: 'Pytel brambor z místního družstva',

23: 'Čokoládový dort',

47: 'Kniha o historii města',

55: 'Šiška salámu',

67: 'Vyhlídkový let balónem',

79: 'Moderní televizor',

91: 'Roční předplatné městského zpravodaje',

93: 'Společenská hra Sázky a dostihy',

}

Napiš program, který se nejprve zeptá uživatele na číslo jeho lístku. Vstup uživatele si převeď na int!

Zkontroluj, zda je číslo lístku ve slovníku. Pokud ne, vypiš text "Bohužel nevyhráváš nic." Pokud číslo ve slovníku je, vypiš uživateli, co vyhrál.

Odeber výhru pro daný lístek ze slovníku, abychom tam evidovali pouze nevydané ceny.

Řešení

cislo\_uzivatele = int(input('Zadej sve cislo listku: '))

if cislo\_uzivatele in tombola:

vyhra = tombola[cislo\_uzivatele]

else:

vyhra = 'Bohužel nevyhráváš nic.'

# NOTE: mnohem lepsi reseni

# vyhra = tombola.get(cislo\_uzivatele, 'Bohužel nevyhráváš nic.')

print(f'Vyhrál jsi: {vyhra}')

tombola.pop(cislo\_uzivatele)

Bonusy

4 - Paranoidní večírek ○♦♦♦♦

Pořadatel našeho večírku se stává stále více paranoidním a nyní rozhodl, že každý z hostů bude mít speciální heslo, které je platné jen pro něj. Seznam hostů a jejich hesel je níže.

Napiš program, který nejprve zkontroluje, zda je host na seznamu, a pokud tam je,zeptá se ho na heslo a zkontroluje jeho správnost.

Hostu na seznamu, který zadá správné heslo, vypíše program text: "Smíš vstoupit."

Řešení

hesla = {

'Jiří' : 'tajne-heslo',

'Natálie': 'jeste-tajnejsi-heslo',

'Klára' : 'nejtajnejsi-heslo',

}

jmeno\_hosta = input('Zadej jmeno: ')

vstup = 'Nesmíš projít.'

if jmeno\_hosta in hesla:

heslo = input('Zadej heslo: ')

if heslo == hesla[jmeno\_hosta]:

vstup = 'Smíš vstoupit.'

print(vstup)

[9:09](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680635382690619)

**Slovníky s cykly**

1 - Vysvědčení ○○○♦♦

Uvažujme vysvědčení, které máme zapsané jako slovník.

vysvedceni = {

"Český jazyk" : 1,

"Anglický jazyk" : 1,

"Matematika" : 1,

"Přírodopis" : 2,

"Dějepis" : 1,

"Fyzika" : 2,

"Hudební výchova" : 4,

"Výtvarná výchova": 2,

"Tělesná výchova" : 3,

"Chemie" : 4,

}

Napiš program, který spočte průměrnou známku ze všech předmětů.

Uprav program, aby vypsal všechny předměty, ve kterých získal student známku 1.

Řešení

prumerna\_znamka = sum(vysvedceni.values()) / len(vysvedceni)

print(f'Průměrná známka je: {prumerna\_znamka}')

print('Předměty s jedničkou:')

for predmet, znamka in vysvedceni.items():

if znamka == 1:

print('\t' + predmet)

2 - Čtenářský deník ○○○♦♦

Gustav je vášnivým čtenářem detektivek z našeho nakladatelství a navíc si zapisuje knihy, které přečetl.

Níže ve slovníku vidíme, jaké informace si eviduje - název knihy, počet stran a hodnocení od 0 do 10.

knihy = [

{'nazev': 'Vražda s příliš mnoha notami', 'pocet\_stran': 450, 'hodnoceni': 5},

{'nazev': 'Vražda podle knihy', 'pocet\_stran': 524, 'hodnoceni': 9},

{'nazev': 'Past', 'pocet\_stran': 390, 'hodnoceni': 4},

{'nazev': 'Popel popelu', 'pocet\_stran': 411, 'hodnoceni': 10},

{'nazev': 'Noc, která mě zabila', 'pocet\_stran': 159, 'hodnoceni': 7},

{'nazev': 'Vražda, kouř a stíny', 'pocet\_stran': 258, 'hodnoceni': 6},

{'nazev': 'Zločinný steh', 'pocet\_stran': 542, 'hodnoceni': 8},

{'nazev': 'Zkus mě chytit', 'pocet\_stran': 247, 'hodnoceni': 7},

{'nazev': 'Vrah zavolá v deset', 'pocet\_stran': 396, 'hodnoceni': 6},

]

Napiš program, který spočte celkový počet stran, které Gustav přečetl.

Připiš do programu výpočet počtu knih, kterým dal Gustav hodnocení alespoň 8.

Řešení

pocet\_stran = 0

for kniha in knihy:

pocet\_stran += kniha['pocet\_stran']

# NOTE: nebo lépe

# pocet\_stran = sum(kniha['pocet\_stran'] for kniha in knihy)

print(f'Přečetl {pocet\_stran} stran.')

pocet\_nad\_osm = 0

for kniha in knihy:

if kniha['hodnoceni'] >= 8:

pocet\_nad\_osm += 1

print(f'Počet knih s hodnocením >= 8: {pocet\_nad\_osm}')

Bonusy

3 - Poznávací značky ○○♦♦♦

V následujícím slovníků je evidence automobilů. Klíčem jsou státní poznávací značky (SPZ) a hodnotou je jméno majitele vozu.

spzetky = {

"4A2 3000": "František Novák",

"6P5 4747": "Jana Pilná",

"3B7 3652": "Jaroslav Sečkár",

"1P5 5269": "Marta Nováková",

"37E 1252": "Martina Matušková",

"2A5 2241": "Jan Král"

}

Napiš program, který vypíše všechny majitele, jejichž vozidlo je registrováno v Plzeňském kraji.

tj. na druhém místě (index 1) řetězce v klíči je písmeno P.

Řešení

print('Lidé z Plzeňského kraje:')

for spz, jmeno in spzetky.items():

if spz[1] == 'P':

print(f'- {jmeno}')

4 - Recepty ○○♦♦♦

Pohlédněte na následující reprezentaci receptu:

{

'nazev' : 'Batáty se šalvějí a pancettou',

'narocnost' : 'stredni',

'doba' : 30,

'ingredience': [

['batát', '1', '15 kč'],

['olivový olej', '2 lžíce', '2 kč'],

['pancetta', '4-6 plátků', '21 kč'],

['přepuštěné máslo', '2 lžíce', '5 kč'],

['mletý černý pepř', '1/2 lžičky', '0.5 kč'],

['sůl', '1/2 lžičky', '0.1 kč'],

['muškátový oříšek', 'špetka', '1 kč'],

['česnek', '2 stroužky', '1 kč'],

['šalvějové lístky', '20-25', '12 kč']

]

}

Uložte si tuto strukturu do proměnné recept na začátek nového programu. Vypište pomocí funkce print kolik bude celé jídlo stát korun zaokrouhlené na celé koruny nahoru.

Řešení

recept = {

'nazev' : 'Batáty se šalvějí a pancettou',

'narocnost' : 'stredni',

'doba' : 30,

'ingredience': [

['batát', '1', '15 kč'],

['olivový olej', '2 lžíce', '2 kč'],

['pancetta', '4-6 plátků', '21 kč'],

['přepuštěné máslo', '2 lžíce', '5 kč'],

['mletý černý pepř', '1/2 lžičky', '0.5 kč'],

['sůl', '1/2 lžičky', '0.1 kč'],

['muškátový oříšek', 'špetka', '1 kč'],

['česnek', '2 stroužky', '1 kč'],

['šalvějové lístky', '20-25', '12 kč']

]

}

celkem = 0

for ingredience in recept['ingredience']:

cena\_text = ingredience[-1]

cena = float(cena\_text.split(' ')[0])

celkem += cena

print(f'recept stojí celkem {celkem} Kč')

**Wednesday, April 5th**



**Anna Vyčítalová**  [7:02 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680714176902869)

Zdravím já si pořád nevím rady, jak pracovat s tou hodnotou, jak ji vyžádat po uživateli

8 replies

Last reply 2 months agoView thread

**Friday, April 7th**



**Anna Vyčítalová**  [12:05 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1680861950740319)

Zdravím, mám dotaz ke kódu proč to řešení u slovníků a cyklů vždycky začíná hodnota = 0, třeba pocet\_nad\_osm = 0

2 replies

Last reply 2 months agoView thread

**Tuesday, April 11th**



**Janka Marschalkova**  [6:33 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681230809911029)

Čtení ze souboru pomocí read()

import json

with open('absolventi.json') as prvni\_soubor:

data = prvni\_soubor.read()

print(prvni\_soubor.read())

print(type(data))

print(open('absolventi.json').read())

(edited)



**Janka Marschalkova**  [6:49 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681231742633129)

Otevření pomocí with + načtení json dat ze souboru

import json

with open('absolventi.json', encoding='utf-8') as prvni\_soubor:

seznam\_absolventu = json.load(prvni\_soubor)

print(seznam\_absolventu[0])

(edited)



**Janka Marschalkova**  [7:03 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681232628875629)

Zápis JSON dat

import json

hours = {'po': 8, 'ut': 7, 'st': 6, 'ct': 7, 'pa': 8}

with open('hodiny.json', mode='w', encoding='utf-8') as file:

json.dump(hours, file)

(edited)



**Janka Marschalkova**  [8:08 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681236517074289)

import requests

response = requests.get('<https://api.kodim.cz/python-data/people>')

#print(response)

#print(response.text)

# navratový kod dobrý -> httpcats.com

data = response.json()

print(data)

(edited)

[8:12](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681236725518399)

**Řešení breakoutroomu 1 - ZÁVOD**

import json

with open('zavod.json', encoding='utf-8') as file:

runners = json.load(file)

finishers = []

for runner in runners:

if runner['casy']['oficialni'] != 'DNF':

finishers.append(runner['jmeno'])

print(finishers)

**Wednesday, April 12th**



**Janka Marschalkova**  [9:46 AM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681285592433719)

**Řešení breakoutroomu 1 - Transformace**

import json

with open('words.txt') as file:

# Split lines cte soubor po radcich a vytvori

# seznam obsahujici jednotlive radky

slova = file.read().splitlines()

vystup = {}

for slovo in slova:

prvni\_pismeno = slovo[0]

# Pokud je to poprve, co jsme narazili na slovo s timto

# prvnim pismenem, musime poprve vytvorit seznam slov s

# timto pismenem. Slovo jeste nepridavame.

if prvni\_pismeno not in vystup:

vystup[prvni\_pismeno] = []

# Ted uz je jiste, ze vzdycky ten seznam existuje, takze

# slovo pridame

vystup[prvni\_pismeno].append(slovo)

with open('output.json', mode='w') as file:

json.dump(vystup, file)

:heart:1

[9:47](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681285678592799)

**2 BR - Prvni ukol z API:**

import requests

odpoved = requests.get("<https://api.kodim.cz/python-data/people>")

osoby = odpoved.json()

print(osoby)

# Pocet osob

print(len(osoby))

# Vypsat klice jako "first\_nam2"

for key in osoby[0].keys():

print(key)

for osoba in osoby:

print(osoba["first\_name"])

# Pocty muzu a zen

pocet\_muzu = 0

pocet\_zen = 0

for osoba in osoby:

if osoba["gender"] == "Male":

pocet\_muzu += 1

if osoba["gender"] == "Female":

pocet\_zen += 1

print("muzi " + str(pocet\_muzu))

print("zeny " + str(pocet\_zen))

**2 BR - Druhy ukol z API:Aktualni svatek:**

import requests

odpoved = requests.get("<https://svatky.adresa.info/json>")

json\_odpoved = odpoved.json()

print(json\_odpoved[0]["name"])

**Svatek 29. 2, kde parametr je v adrese:**

import requests

odpoved = requests.get("<https://svatky.adresa.info/json?date=2902>")

json\_odpoved = odpoved.json()

print(json\_odpoved[0]["name"])

**Svatek 29.2, kde parametr je zvlast:**

import requests

odpoved = requests.get("<https://svatky.adresa.info/json>", params={"date": "2902"})

json\_odpoved = odpoved.json()

print(json\_odpoved[0]["name"])

**API, priklad 2b:**

import requests

den = input("Vlozte den s pocatecni nulou: ")

mesic = input("Vlozte mesic s pocatecni nulou: ")

odpoved = requests.get("<https://svatky.adresa.info/json>", params={"date": den + mesic})

json\_odpoved = odpoved.json()

print(json\_odpoved[0]["name"])

**Sunday, April 16th**



**Pavla Slabihoudková**  [8:30 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681669825094039)

Ahoj Janko, chtěla bych se ještě vrátit ke cvičení se slovy (Řešení breakoutroomu 1 - Transformace). Kód v cyklu mi přijde strašně stručný, pořád úplně nechápu, jak python podle tohoto postupu ví, že klíče jsou "písmena"?? Jednotlivá slova bych do seznamů už dotlačila. Jen prostě nevím, kde říkám pythonu "klíče slovníku jsou tyto...". Nejspíš to bude zde: prvni\_pismeno = slovo[0], ale jak z toho python vyčte, že zrovna toto jsou klíče slovníku, mi hlava nebere:thinking_face: Díky za polopatické vysvětlení:wink:. Pavla

1 reply

2 months agoView thread

**Tuesday, April 18th**



**Janka Marschalkova**  [6:41 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681836113669099)

Funkce - definování pomocí def, důležité je jejich zavolání, aby se vykonaly

def secti(l, k): # parametry

return l + k

#print(l+k)

def secti(l, k): # parametry

soucet = l + k

return soucet

secti(8, 5) # argumenty

vysledek = secti(8, 5)

print(f'Vysledek je : {vysledek}')

(edited)



**Janka Marschalkova**  [6:55 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681836922942669)

# Ke kazdemu cislu v seznamu pricti 1 pomoci funkce secti

seznam\_cisel = [2, 5, 9, 8]

# Vysledek jen vypiseme

for cislo in seznam\_cisel:

print(secti(cislo, 1))

# Vysledek ulozime do noveho seznamu

soucty = []

for cislo in seznam\_cisel:

soucty.append(secti(cislo, 1))

# Přepíšeme původní seznam\_cisel na novy, kde k cislum byla prictena 1 pomocí funkce secti

seznam\_cisel = soucty



**Janka Marschalkova**  [7:03 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681837429820579)

Složitější funkce, dokumentace pomocí """ """

def eur\_to\_czk(pocet\_eur):

"""

Funkce na prevod eur do korun.

Bere jeden parametr - pocet eur. Kurz fixni 25 kc za 1 euro.

Vrati kolik stoji eura v korunach.

"""

kurz = 25 # lokalni promenna - neni videt mimo telo funkce

pocet\_czk = pocet\_eur \* kurz

return pocet\_czk

eura\_uzivatel = int(input("Kolik si prejete smenit euro?"))

print(eur\_to\_czk(eura\_uzivatel))

(edited)

[7:04](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681837448282719)

GLOBALNI PROMENNA

KURZ = 25

def eur\_to\_czk(pocet\_eur):

"""

Funkce na prevod eur do korun.

Bere jeden parametr - pocet eur. Kurz fixni 25 kc za 1 euro.

Vrati kolik stoji eura v korunach.

"""

kurz = 25 # lokalni promenna - neni videt mimo telo funkce

pocet\_czk = pocet\_eur \* KURZ

return pocet\_czk

eura\_uzivatel = int(input("Kolik si prejete smenit euro?"))

print(eur\_to\_czk(eura\_uzivatel))

print(KURZ)

[7:06](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681837616373749)

Komentáře

# jednoradkovy komentar

""" viceradkovy

komentar

"""

(edited)



**Janka Marschalkova**  [7:24 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681838654669999)

Funkce bez kódu

def code\_me\_later(a, b):

pass



**Janka Marschalkova**  [7:34 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681839246269779)

Využiti nazvu parametru pri volani funkce

print(eur\_to\_czk(pocet\_eur=eura\_uzivatel, kurz=28)) # v pořadí

print(eur\_to\_czk(kurz=28, pocet\_eur=eura\_uzivatel)) # při přehození pořadí důležité název=....

print(eur\_to\_czk(pocet\_eur=eura\_uzivatel, 28)) # Bacha, nefunguje

print(eur\_to\_czk(eura\_uzivatel, kurz=28, .... = ..., ... = ....)) # OK

(edited)



**Janka Marschalkova**  [7:52 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681840329166619)

Příklad s funkcí známky

def get\_mark(points):

if points <= 60:

mark = 5

elif points <= 70:

mark = 4

elif points <= 80:

mark = 3

elif points <= 90:

mark = 2

else:

mark = 1

return mark

points = int(input("Zadej počet bodů v testu: "))

mark = get\_mark(points)

print(f"Výsledná známka je {mark}.")

[7:52](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681840358197899)

Příklad s funkcí známky + defaultně nastavený parametr bonus

def get\_mark(points, bonus=0):

if points + bonus <= 60:

mark = 5

elif points + bonus <= 70:

mark = 4

elif points + bonus <= 80:

mark = 3

elif points + bonus <= 90:

mark = 2

else:

mark = 1

return mark

points = int(input("Zadej počet bodů v testu: "))

bonus = int(input("Zadej počet bonusových bodů: "))

mark = get\_mark(points, bonus)

print(f"Výsledná známka je {mark}.")



**Nikola Mašová**  [8:36 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681842974694499)

Janko, prosím, nasdílíš sem opět řešení pro ostatní příklady z BR? Děkuji. Hezký večer :slightly_smiling_face:

2 replies

Last reply 2 months agoView thread

**Wednesday, April 19th**



**Janka Marschalkova**  [11:19 AM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681895958576399)

**Řešení z breakoutroomu - 4.lekce**  
**1 - Násobení**:diamonds::diamonds:  
Napiš funkci mult, která bude mít dva číselné parametry. Funkce oba parametry vynásobí a vrátí výsledek.Řešení

def mult(a, b):

return a \* b

vysledek = mult(5, 2)

print(f'5 \* 2 = {vysledek}')

(edited)

[11:19](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681895991777399)

**2 - Hotel**:diamonds::diamonds:  
Napiš funkci total\_price, která vypočte cenu noci v hotelu. Funkce bude mít dva parametry - people a breakfast.  
Cena za noc za osobu je 850 Kč a cena za snídani za osobu je 125 Kč.  
Funkce vrátí výslednou cenu. Parametr breakfast je nepovinný a výchozí hodnota je False.  
Funkci vyzkoušej se zadáním dvou i jedné hodnoty, např. total\_price(3), total\_price(2, True).Řešení

def total\_price(people, breakfast=False):

cena\_za\_noc = 850

cena\_za\_snidani = 125

celkem = cena\_za\_noc \* people

if breakfast:

celkem += cena\_za\_snidani \* people

return celkem

print(f'Cena za tři lidi: {total\_price(3)}')

print(f'Cena za dva lidi se snidani: {total\_price(2, True)}')

(edited)

[11:20](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681896044168569)

**3 - Rámeček**

:diamonds::diamonds::diamonds:

Napiš funkci, která jako parametr převezme řetězec a vytiskne jej obalen hvězdičkami.

Zadej slovo: ahoj

\*\*\*\*\*\*\*\*

\* ahoj \*

\*\*\*\*\*\*\*\*

Řešení

def ramecek(slovo):

delka = len(slovo)

sirka\_ramecku = delka + 4 # chceme trochu mista kolem slova

print('\*' \* sirka\_ramecku)

print(f'\* {slovo} \*')

print('\*' \* sirka\_ramecku)

s = input('Zadej slovo: ')

ramecek(s)

(edited)

[11:20](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681896053181799)

**BONUSY**

[11:21](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681896068435489)

**3 - Měsíc narození**

:diamonds::diamonds::diamonds:

Napiš funkci month\_of\_birth, která bude mít jeden parametr - rodne\_cislo.  
Funkce ze zadaného rodného čísla určí měsíc narození, které vrátí jako výstup. Nezapomeň, že pro ženy je k měsíci připočtena hodnota 50.  
Příklad:

* Pro hodnotu 9207054439 vrátí 7.
* Pro hodnotu 9555125899 vrátí 5.

Řešení

def month\_of\_birth(rodne\_cislo):

cislo\_mesice = int(str(rodne\_cislo)[2:4])

if cislo\_mesice <= 12:

return cislo\_mesice

else:

return cislo\_mesice - 50

print(f'9207054439 -> {month\_of\_birth(9207054439)}')

print(f'9555125899 -> {month\_of\_birth(9555125899)}')

(edited)

[11:21](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681896084538809)

**5 - Zarovnání výpisu**

:diamonds::diamonds::diamonds::diamonds:

Vypište seznam čísel každé na nový řádek zarovnané vpravo na délku nejdelšího čísla.

numbers = [7728, 88, 958621, 5941, 959847272, 3944, 80, 521, 57035, 3967894]

Návod: Zjistěte kolik znaků zabírá nejdelší číslo ze seznamu Napište funkci, která dostane řetězec a délku, na kterou má text vyplnit zleva mezerami  
Bonus: funkce bude mít volitelný parametr, jakým znakem má text vyplňovat  
Výstup:

7728

88

958621

5941

959847272

3944

80

521

57035

3967894

Výstup bonusu:

.....7728

.......88

...958621

.....5941

959847272

.....3944

.......80

......521

....57035

..3967894

Řešení

def zarovnej(retezec, delka, znak=' '):

zleva = delka - len(retezec)

print(znak \* zleva + retezec)

nejdelsi = 0

for n in numbers:

aktualni\_delka = len(str(n))

if aktualni\_delka > nejdelsi:

nejdelsi = aktualni\_delka

# ted uz jen pomoci funkce vypiseme

for n in numbers:

zarovnej(str(n), nejdelsi)

# a jako bonus

for n in numbers:

zarovnej(str(n), nejdelsi, '.')

(edited)

[11:21](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681896113027799)

**6 - Ruleta**

:diamonds::diamonds::diamonds::diamonds::diamonds:

Napiš funkci, která bude jednoduchou simulací rulety. Budeme uvažovat pouze možnost sázení na řady. Uživatel si může vybrat jednu ze tří řad:

* první řada (hodnoty 1, 4, 7 atd.),
* druhá řada (hodnoty 2, 5, 8 atd.),
* třetí řada (hodnoty 3, 6, 9 atd.).
* Zeptej se uživatele, na kterou ze tří řad sází a na výši sázky.
* Vytvoř funkci roulette, která bude mít parametry cislo\_rady a sazka. Pomocí funkce randint z modulu random vygeneruj náhodné číslo v rozsahu od 0 (včetně) do 36. Vyhodnoť, do které řady číslo náleží. Nezapomeň, že 0 nepatří do žádné z řad a pokud padne, uživatel vždy prohrává.
* Funkce roulette vrací hodnotu výhry. Pokud uživatel vsadil na správnou řadu, vyhrává dvojnásobek sázky, v opačném případě nevyhrává nic jeho sázka propadá.

Řešení

import random

cislo\_rady = int(input('Na kterou řadu sázíš? (1-3): '))

sazka = int(input('Kolik sázíš?: '))

def roulette(cislo\_rady, sazka):

# pouzijeme funkci range na vytvoreni seznamu cisel

rady = [

range(1, 37, 3), # 1, 4, 7, ...

range(2, 37, 3), # 2, 5, 8, ...

range(3, 37, 3), # 3, 6, 9, ...

]

hozeno = random.randint(0, 36)

print(f'hozeno: {hozeno}')

return 2 \* sazka if hozeno in rady[cislo\_rady - 1] else 0 # "- 1" kvůli indexování od nuly

[11:23](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681896186749509)

**7 - Zpřeházená písmena**

:diamonds::diamonds::diamonds::diamonds:

Slovo je stále možné pohodlně přečíst, když jsou pomíchaná písmena. Stačí, když první a poslední písmeno je na své pozici zachováno. Napiš funkci, která bude mít jako vstupní parametr slovo a vrátí slovo, kde zpřehází všechny znaky kromě prvního a posledního.  
Nápověda: random.shuffle()  
Super bonus: Napiš program, který takovou funkci využije na delší text více slov.Řešení

import random

def zamichej\_slovo(slovo):

pismena\_jako\_list = list(slovo)

stred = pismena\_jako\_list[1:-1]

random.shuffle(stred)

return slovo[0] + ''.join(stred) + slovo[-1]

print(zamichej\_slovo('lokomotiva'))

text = '''Slovo je stále možné pohodlně přečíst, když jsou pomíchaná písmena.

Stačí, když první a poslední písmeno je na své pozici zachováno. Napiš funkci,

která bude mít jako vstupní parametr slovo a vrátí slovo, kde zpřehází všechny

znaky kromě prvního a posledního.

'''

zamichana\_slova = []

for slovo in text.split():

zamichana\_slova.append(zamichej\_slovo(slovo))

vysledny\_text = ' '.join(zamichana\_slova)

print(vysledny\_text)

[11:23](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1681896201712539)

**8 - Nápravy**

:diamonds::diamonds::diamonds::diamonds:

Náprava je část vozidla, která spojuje kola s karosérií vozidla. U nákladních vozidel ho můžeme chápat jako počet " dvojic kol". Počet náprav je důležitý napříkad kvůli maximální povolené hmotnosti vozidla.  
Uvažuj limity pro maximální hmotnost nákladního vozidla, které jsou v tabulce níže.  
Počet nápravMaximální dovolená hmotnost v tunách218325432548  
Pokud je limit překročen, platí provozovatel pokutu 1000 Kč za každou tunu, o které je vozidlo těžší. Například pokud má vozidlo 4 nápravy a hmotnost 34 tun, platí provozovatel pokutu 2000 Kč.  
Napiš funkci spocitej\_pokutu(), která bude mít dva parametry - pocet\_naprav (počet náprav vozidla) a hmotnost ( hmotnost vozidla v tunách).  
Funkce na základě těchto parametrů vypočte výpiš pokuty a vráti ji jako celé číslo.

pokuta = spocitej\_pokutu(4, 34)

print(pokuta) # 2000

Dále uvažuj následující dvourozměrný seznam, kde na prvním místě vnořeného seznamu je počet náprav vozidla a na druhém místě je zjištěná hmotnost.

vazeni = [

[4, 33],

[2, 19],

[3, 29],

[3, 27],

[5, 53],

[5, 51],

[2, 20],

]

Projdi seznam pomocí cyklu a pro každé vážení urči (s využitím funkce spocitel\_pokutu()) výši pokuty. Spočítej celkovou výši pokut za všechna vážení.Řešení

def spocitej\_pokutu(pocet\_naprav, hmotnost):

# rozdeleni pomoci podminek

if pocet\_naprav == 2:

max\_hmotnost = 18

if pocet\_naprav == 3:

max\_hmotnost = 25

if pocet\_naprav == 4:

max\_hmotnost = 32

if pocet\_naprav == 5:

max\_hmotnost = 48

# musime omezit nulou aby nebyly zaporne pokuty

prekrocena\_hmotnost = min([hmotnost - max\_hmotnost, 0])

return 1000 \* prekrocena\_hmotnost

for data in vazeni:

pokuta = spocitej\_pokutu(data[0], data[1])

print(f'dostavas pokutu {pokuta}')

(edited)

**Monday, April 24th**



**Anna Vyčítalová**  [2:05 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682337905190339)

Zdravím mám úplný základní dotaz k práci s řetězci, proč pro vyjádření rodného čísla použiji [2:4] a ne [2:3] proč je tam vždycky o číslo víc, už jsme to brali, ale já jsem to úspěšně zapomněla, ale hledala jsem na různých stránkách ale nemůžu to najít

[2:08](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682338123175149)

A ještě jeden dotaz co znamená print(f'9207054439 -> {month\_of\_birth([9207054439](tel:9207054439))}') ta šipka v tom kódu? Díky



**Jakub Červinka**  [2:43 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682340196725599)

No, do hranatých závorek při dělání "slice" (výseku) se jako kocová hodnota uvádí vždycky o jedna víc. To tak prostě je, to tak vymysleli.  
No a ta šipka v tom textu nic nedělá, to je jen text. Až v těch složených závorkách se provede ten výpočet.  
Je to ten "f" string (má f před uvozovkama) v tom se dá mezi složenými závorkami vkládat proměnné nebo udělat výpočet. V tomhle případě zavolání funkce.

1 reply

2 months agoView thread

**Tuesday, April 25th**



**Janka Marschalkova**  [6:34 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682440481878749)

**Typování**

# def eur\_to\_czk(pocet\_eur, kurz=25):

def eur\_to\_czk(pocet\_eur: int, kurz: int=25) -> int:

"""

Funkce na prevod eur do korun.

Bere dva parametry - pocet eur a kurz.

Vrati kolik stoji eura v korunach.

"""

pocet\_czk = pocet\_eur \* kurz

return pocet\_czk

eura\_uzivatel = input("Kolik si prejete smenit euro?")

print(eur\_to\_czk(eura\_uzivatel))

1 reply

1 month agoView thread



**Janka Marschalkova**  [6:43 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682441004194369)

import random

# cyklus pomocí číselné řady

""" jmena = ["Anna", "Adela", "Alena", "Adam"]

for jmeno in jmena:

print(jmeno)

"""

pocet\_hracu = 4

hod\_1 = random.randint(1, 6)

hod\_2 = random.randint(1, 6)

hod\_3 = random.randint(1, 6)

hod\_4 = random.randint(1, 6)

#print(hod\_1, hod\_2, hod\_3, hod\_4)

#print(list(range(pocet\_hracu)))

#Chceme tolikrat "hodit kostkou" kolik mame hracu

for hrac in range(pocet\_hracu): # Opakuj tolikrat, kolik zadame parametr funkci range (zde pocet\_hracu)

print(random.randint(1,6))

[6:45](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682441137706649)

# zbytecne bychom museli vypisovat, velke cislo by bylo narocne

for hrac in [0, 1, 2, 3, 4, 5]:

print(random.randint(1,6))

[6:48](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682441300793829)

# Nepojmenovana promenna podtrzitko "\_"

for \_ in range(pocet\_hracu):

print(f'Hrac hodil {random.randint(1,6)}')

[6:52](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682441567950289)

jmena = ["Anna", "Adela", "Alena", "Adam"]

for hrac in range(pocet\_hracu): # Opakuj tolikrat, kolik zadame parametr funkci range (zde pocet\_hracu)

print(f'Hrac {jmena[hrac]} hodil {random.randint(1,6)}')



**Janka Marschalkova**  [7:29 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682443780381669)

for hrac in range(0, 4): # range (od, do + 1)

print(f'Hrac {hrac} hodil {random.randint(1,6)}')



**Janka Marschalkova**  [7:43 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682444583797839)

# Chroustání seznamů = List comprehension

# Priklad 1

znamky = [0, 2, 0, 1, 1, 3] # znamkovani programatoru

opravene\_znamky = [] # bezne znamkovani (o 1 vetsi)

for znamka in znamky:

opravene\_znamky.append(znamka + 1)

# opravene\_znamky = opravene\_znamky + [znamka + 1]

print(opravene\_znamky)

# Priklad 2

kilometry = [2.4, 2.6, 0, 3.5, 1.8]

zaokrouhlene\_kilometry = []

for zaznam in kilometry:

zaokrouhlene\_kilometry.append(round(zaznam))

# zaokrouhlene\_kilometry = zaokrouhlene\_kilometry + [round(zaznam)]

print(zaokrouhlene\_kilometry)

# Priklad 1 - list comprehension

znamky = [0, 2, 0, 1, 1, 3] # znamkovani programatoru

opravene\_znamky = [znamka + 1 for znamka in znamky] # modifikace x for x in puvodni seznam

print(opravene\_znamky)

# Priklad 2 - list comprehension

kilometry = [2.4, 2.6, 0, 3.5, 1.8]

zaokrouhlene\_kilometry = [round(zaznam) for zaznam in kilometry]

print(zaokrouhlene\_kilometry)



**Janka Marschalkova**  [7:51 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682445112271499)

# místo if znamka < 3:

# opravene\_znamky.append(znamka + 1)

# else:

# opravene\_znamky.append(znamka)

# Znamky - opravime jen znamky, které jsou < 3

znamky = [0, 2, 0, 1, 1, 3] # znamkovani programatoru

opravene\_znamky = [znamka + 1 if znamka < 3 else znamka for znamka in znamky] # modifikace x for x in puvodni seznam

print(opravene\_znamky)

# Znamky - opravime jen znamky, které jsou < 3

znamky = [0, 2, 0, 1, 1, 3] # znamkovani programatoru

def oprav\_znamku(znamka: int) -> int:

if znamka < 3:

return znamka + 1

else:

return znamka

#opravene\_znamky = [oprav\_znamku(znamka) for znamka in znamky]

opravene\_znamky = [oprav\_znamku(x) for x in znamky]

print(opravene\_znamky)

print(opravene\_znamky)

(edited)



**Janka Marschalkova**  [8:54 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682448881217389)

**ŘEŠENÍ - BR 1 - CYKLY 2** (edited)

[8:54](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682448892609899)

1 - Tombola ○○○♦♦

V tombole bylo prodáno celkem 1000 lístků. Naším úkolem je vylosovat náhodně tři výherce. Napište program, který vygeneruje a vypíše tři čísla mezi 1 a 1000.

Využijte funkci randint, nezapomeňte ale, že si ji musíte importovat z modulu random.

Neřešte, že jedno číslo může být vygenerováno dvakrát.

Řešení

import random

for \_ in range(3):

print(random.randint(1, 1000))

2 - Dělitelnost více čísly ○○○♦♦

Vypišme si čísla z nějakého rozsahu na základě jejich dělitelnosti dvěma čísly.

Zkuste z nějakého rozsahu čísel vypsat čísla, která jsou dělitelná 3 i 4 současně.

Zkuste z nějakého rozsahu čísel vypsat čísla, která jsou dělitelná 5 nebo 6.

Stačí vypsat text: "Číslo je dělitelné 5 nebo 6."

Řešení

for n in range(30):

delitelne\_3 = n % 3 == 0

delitelne\_4 = n % 4 == 0

if delitelne\_3 and delitelne\_4:

print(f'{n} je dělitelné 3 i 4')

delitelne\_5 = n % 5 == 0

delitelne\_6 = n % 6 == 0

if delitelne\_5 or delitelne\_6:

print(f'{n} je dělitelné 5 nebo 6')

3 - Seznam hostů na párty ○○○♦♦

Vraťte se k příkladu se zadáváním seznamu hostu na párty a zkopírujte si kód k sobě do editoru.

Upravte program tak, že pokud uživatel v průběhu zadávání jmen napíše "konec", cyklus na zadávání jmen se ukončí.

Řešení

number\_of\_guests = int(input("Zadej počet hostů: "))

guest\_list = []

for i in range(number\_of\_guests):

new\_guest = input("Zadej jméno hosta: ")

if new\_guest == 'konec':

break

guest\_list.append(new\_guest)

print(guest\_list)

Bonusy

4 - Prvočíslo ○♦♦♦♦

Požádej uživatele o zadání celého čísla. Následně urči, zda je toto číslo prvočíslo.

Prvočíslo je číslo, které je dělitelné beze zbytku pouze 2 čísly - 1 a sebou samotným.

Například 5 je prvočíslo, protože je dělitelná pouze 1 a 5.

Naopak 4 není prvočíslo, protože je dělitelná 1, 2 a 4.

Řešení

pozn.: Níže je velmi neefektivní způsob. Samozřejmě existují specializované algorytmy.

testovane = int(input('Zadej číslo na otestování: '))

prvocislo = True

for n in range(2, testovane):

if testovane % n == 0:

prvocislo = False

break

print(f'cislo {testovane} {"je" if prvocislo else "není"} prvocislo')

[8:55](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682448932948129)

**ŘEŠENÍ BR 2 - LIST COMPREHENSION**

[8:56](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1682449010309129)

1 - Seznamy čísel ○○○○♦

Mějme zadaný následující seznam

cisla = [1.12, 4.51, 2.64, 13.1, 0.1]

Vytvořte seznam, který obsahuje

každé z čísel ze seznamu cisla vynásobené dvěma,

každé z čísel ze seznamu cisla s opačným znaménkem,

každé z čísel ze seznamu cisla umocněné na druhou,

každé z čísel ze seznamu cisla jako řetězec.

Řešení

krat\_dva = [n \* 2 for n in cisla]

minus = [-n for n in cisla]

na\_druhou = [n \*\* 2 for n in cisla]

jako\_str = [str(n) for n in cisla]

2 - Seznamy řetězců ○○○♦♦

Mějme zadaný následující seznam

jmena = ['Roman', 'Jan', 'Miroslav', 'Petr', 'Gabriel']

Vytvořte seznam, který obsahuje

počty písmen ve všech jménech,

všechna jména napsaná velkými písmeny,

všechna jména plus písmeno 'a' na konci (stanou se z nich tedy ženská jména),

všechna jména převedená na malá písmena s koncovkou '@email.cz'.

Řešení

pocty = [len(jmeno) for jmeno in jmena]

velkymi = [jmeno.upper() for jmeno in jmena]

s\_ackem = [jmeno + 'a' for jmeno in jmena]

emaily = [f'{jmeno.lower()}@email.cz' for jmeno in jmena]

3 - Teploty - stejné zadání jako povinný úkol z lekce 5 - řešení nasdílím po deadlinu

(edited)

**Tuesday, May 2nd**



**Janka Marschalkova**  [6:38 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683045536142629)

Vítejte ve Směnárně na Růžku!  
Kurzy měn pro 19. 12. 2020 jsou:  
1 €   = 26.35 Kč  
1 $   = 21.76 Kc  
1 £   = 28.78 Kč  
1 DKK = 3.54 kc  
Neúčtujeme žádné poplatky.



**Janka Marschalkova**  [6:50 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683046243889099)

\d?\d\.\d\d [Kk][cč]

[6:53](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683046390398589)

\d+\.\d\d [Kk][cč]

[6:55](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683046504077239)

+ znak alespoň jednou  
\*nulakrat az nekonecno

[6:55](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683046533329349)

9. led. 2021 sobota 9:30 - 16:30 Úvod do programování 1 - Java  
16. led. 2021 sobota 7:30 - 15:30 Úvod do programování 1 - Javascript  
16. led. 2021 úterý 9:30 - 16:30 Úvod do programování 1 - Python  
18. led. 2021 sobota 8:30 - 15:30 Úvod do programování 1 - Java  
23. led. 2021 pondělí 9:30 - 16:30 Úvod do programování 1 - Javascript  
27. led. 2021 neděle 8:30 - 17:30 Úvod do programování 1 - Python  
ONLINE  
14. úno. 2021 neděle 8:30 - 17:30 Úvod do programování 1 - Python  
ONLINE  
20. úno. 2021 sobota 9:30 - 16:30 Úvod do programování 1 - Tes (edited)

[6:57](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683046633323669)

\d?\d:\d\d - \d?\d:\d\d

[7:00](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683046810835049)

\d?\d\. (led|úno)\. \d{4}



**Janka Marschalkova**  [8:02 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683050568322689)

import re

# Vytvoreni regularni vyrazu

regularni\_vyraz = re.compile(r"(\+420)? ?\d{3} ?\d{3} ?\d{3}")

vstup = input("Zadej tel. číslo: ")

# Vstup musi presne odpovidat regularnimu vyrazu

vstup\_ok = regularni\_vyraz.fullmatch(vstup)

print(vstup\_ok)

if vstup\_ok:

print("Tel. číslo je v pořádku")

else:

print("Nesprávné tel. číslo!")

**Wednesday, May 3rd**



**Janka Marschalkova**  [8:34 AM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683095649342949)

**Otázky Kahoot - opakování z 5. lekce**  
Který zápis volání funkce mesic\_narozeni je chybný? (edited)

1.otazka.jpg

[8:34](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683095667701189)

Správná odpověď**:** 4 (edited)

[8:35](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683095725135539)

Program je v pořádku (True = Ano, False= Ne)

2.otazka.jpg

[8:35](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683095746029889)

Správná odpověď: Ne (definice funkce musí být před voláním funkce)

[8:36](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683095762963719)

Funkce overeni\_veku\_1 a overeni\_veku\_2 jsou ekvivalentní:

3.otazka.jpg

[8:36](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683095776230739)

Správná odpověď: Ano

[8:36](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683095788002799)

Co vypíše program?

4.otazka.jpg

[8:36](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683095804575209)

Správná odpověď: ['A. N.', 'B. N.', 'C. D.', 'D. S.']



**Anna Vyčítalová**  [12:03 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683108217864259)

Zdravím, jak zajisím , aby se výraz vyhledával jenom na jedné řádce?

1 reply

1 month agoView thread



**Janka Marschalkova**  [11:01 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683147712141049)

**Řešení k BR - Regulární výrazy (regex 101)**

[11:02](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683147748012069)

Předčíslí

\d{0,6}-?\d{6,10}/\d{4}

<https://regex101.com/r/n7x9Wg/1>  
Není to dokonalé, ale jako řešení stačíČíslo účtu podruhé

[12][1-3]\d{6,8}/2100

<https://regex101.com/r/VoUKri/3>Registrační značka

\d[A-Z]\w \d{4}

<https://regex101.com/r/wXlJDo/1>

\d(A|B|C|E|H|J|K|L|M|P|S|T|U|Z)\w \d{4}

<https://regex101.com/r/ZrPyTs/1>Telefonní číslo

(\+420)? ?\d{3} ?\d{3} ?\d{3}

<https://regex101.com/r/O50NSs/1>Ministerstva

Ministerstvo[\w ]\*

<https://regex101.com/r/wrXVhD/1>Nápravy

\w{1}\,\w{1} m

<https://regex101.com/r/RydgSW/1>

\w{2}\,\w{2} t

<https://regex101.com/r/hbwv8Z/1>Slavný soude

\d{2} [A-Z]{1,3} \d{1,4}/\d{4}

<https://regex101.com/r/zy8n4X/1>Ave, Caesar!

I?(V|X)?I{0,3}

(V|X|I)+

<https://regex101.com/r/D31W3G/2>

regex101**regex101**

[**regex101: build, test, and debug regex**](https://regex101.com/r/n7x9Wg/1)

Regular expression tester with syntax highlighting, explanation, cheat sheet for PHP/PCRE, Python, GO, JavaScript, Java, C#/.NET, Rust. (14 kB)

<https://regex101.com/r/n7x9Wg/1>

regex101**regex101**

[**regex101: build, test, and debug regex**](https://regex101.com/r/VoUKri/3)

Regular expression tester with syntax highlighting, explanation, cheat sheet for PHP/PCRE, Python, GO, JavaScript, Java, C#/.NET, Rust. (14 kB)

<https://regex101.com/r/VoUKri/3>

regex101**regex101**

[**regex101: build, test, and debug regex**](https://regex101.com/r/wXlJDo/1)

Regular expression tester with syntax highlighting, explanation, cheat sheet for PHP/PCRE, Python, GO, JavaScript, Java, C#/.NET, Rust. (14 kB)

<https://regex101.com/r/wXlJDo/1>

regex101**regex101**

[**regex101: build, test, and debug regex**](https://regex101.com/r/ZrPyTs/1)

Regular expression tester with syntax highlighting, explanation, cheat sheet for PHP/PCRE, Python, GO, JavaScript, Java, C#/.NET, Rust. (14 kB)

<https://regex101.com/r/ZrPyTs/1>

regex101**regex101**

[**regex101: build, test, and debug regex**](https://regex101.com/r/O50NSs/1)

Regular expression tester with syntax highlighting, explanation, cheat sheet for PHP/PCRE, Python, GO, JavaScript, Java, C#/.NET, Rust. (14 kB)

<https://regex101.com/r/O50NSs/1>

[11:02](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683147769775299)

**Řešení BR - Regulární výrazy v Pythonu**

[11:03](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683147798063509)

Uživatelské jméno

import re

regularni\_vyraz = re.compile(r"[a-z]{1,8}")

uzivatelske\_jmeno = input("Zadej uživatelského jména: ")

vysledek = regularni\_vyraz.fullmatch(uzivatelske\_jmeno)

if vysledek:

print("Uživatelské jméno je v pořádku.")

else:

print("Uživatelské jméno nesplňuje požadavky")

E-mail s tečkou

import re

regularni\_vyraz = re.compile(r"\w+\.\w+@\w+\.cz")

email = "jiri.pesik@python.cz"

vysledek = regularni\_vyraz.fullmatch(email)

if vysledek:

print("E-mail jméno je v pořádku.")

Záznamy

zaznamy = """

searchNumber: pavca.czechitas action: search phone number of user dita

user: pavca action: send sms to phone number +420728123456

user: jirka: action: send 2 sms to phone number +420734123456

"""

import re

regularniVyraz = re.compile(r"[+\d]{13}")

vysledky = regularniVyraz.findall(zaznamy)

for vysledek in vysledky:

print(vysledek)

anonymni\_zaznamy = regularniVyraz.sub("X" \* 9, zaznamy)

print(anonymni\_zaznamy)

Adresy stránek

email\_s\_radami = """

Ahoj,

posílám ti pár tipů, kam se podívat. [https://realpython.com](https://realpython.com/) nabízí spoustu článků i kurzů. [http://docs.python.org](http://docs.python.org/) nabízí tutoriál i rozsáhlou dokumentaci. [http://www.learnpython.org](http://www.learnpython.org/) nabízí hezky strukturovaný kurz pro začátečníky, rozebírá ale i nějaká pokročilejší témata. [https://www.pluralsight.com](https://www.pluralsight.com/) je placený web, který ale kvalitou kurzů víceméně nemá konkurenci. Určitě ale sleduj i web [https://www.czechitas.cz](https://www.czechitas.cz/) a přihlašuj se na naše kurzy!

"""

import re

regularni\_vyraz = re.compile(r"https?://[\w\.]\*")

vysledky = regularni\_vyraz.findall(email\_s\_radami)

for vysledek in vysledky:

print(vysledek)

IP adresy

import re

regularni\_vyraz = re.compile(r"[12]?\d{1,2}\.[12]?\d{1,2}\.[12]?\d{1,2}\.[12]?\d{1,2}")

uzivatelske\_jmeno = input("Zadej adresu serveru: ")

vysledek = regularni\_vyraz.fullmatch(uzivatelske\_jmeno)

if vysledek:

print("Odesílám zprávu.")

else:

print("Adresa není platná.")

Práce s kódem

kod = """

sender\_field\_title = "Příjemce"

copy\_field\_title = "Kopie"

if blind\_copy:

blind\_copy\_title = "Skrytá kopie"

if action == "send":

button\_title = "Odeslat"

else:

button\_title = "Uložit koncept"

"""

import re

regularni\_vyraz = re.compile(r"[\w\_]\* \= \"[\w ]\*\"")

vysledky = regularni\_vyraz.findall(kod)

for vysledek in vysledky:

regularni\_vyraz\_vnitrni = re.compile(r"\"[\w ]\*\"")

vysledky\_vnitrni = regularni\_vyraz\_vnitrni.findall(vysledek)

print(vysledky\_vnitrni[0])

**Tuesday, May 9th**



**Janka Marschalkova**  [6:14 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683648885810109)

class Zamestnanec: # definici tridy

jmeno = ""

pozice = ""

frantisek = Zamestnanec() # objekt

[6:16](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683649009531639)

frantisek = Zamestnanec() # objekt

frantisek.jmeno = "Frantisek Novak"

frantisek.pozice = "ucetni"

print(frantisek.pozice)



**Janka Marschalkova**  [6:34 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683650067397269)

class Zamestnanec: # definici tridy

def \_\_init\_\_(self, jmeno, pozice): # konstruktor tridy - specialni metoda

self.jmeno = jmeno

self.pozice = pozice

def info(self): # metoda tridy

return f"Zamestnanec {self.jmeno} pracuje na pozici {self.pozice}"

frantisek = Zamestnanec("Frantisek Novotny", "ucetni") # objekt - pri vytvareni se nam zavola \_\_init\_\_

alena = Zamestnanec("Alena Drobna", "sefova")

print(frantisek.info())



**Janka Marschalkova**  [6:39 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683650384713479)

class Zamestnanec: # definici tridy

def \_\_init\_\_(self, jmeno, pozice): # konstruktor tridy - specialni metoda

self.jmeno = jmeno

self.pozice = pozice

def \_\_str\_\_(self): # specialni metoda

return f"Zamestnanec {self.jmeno} pracuje na pozici {self.pozice}"

def info(self): # metoda tridy

return f"Zamestnanec {self.jmeno} pracuje na pozici {self.pozice}"

frantisek = Zamestnanec("Frantisek Novotny", "ucetni") # objekt - pri vytvareni se nam zavola \_\_init\_\_

alena = Zamestnanec("Alena Drobna", "sefova")

print(frantisek) # print(str(frantisek))



**Janka Marschalkova**  [6:48 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683650921495929)

class Zamestnanec: # definici tridy

def \_\_init\_\_(self, jmeno, pozice): # konstruktor tridy - specialni metoda

self.jmeno = jmeno

self.pozice = pozice

self.pocet\_dni\_dovolene = 25

def \_\_str\_\_(self): # specialni metoda

return f"Zamestnanec {self.jmeno} pracuje na pozici {self.pozice}"

def info(self): # metoda tridy

return f"Zamestnanec {self.jmeno} pracuje na pozici {self.pozice}"

def cerpani\_dovolene(self, days):

if self.pocet\_dni\_dovolene >= days:

self.pocet\_dni\_dovolene -= days

return f"Užij si to."

else:

return f"Bohužel už máš nárok jen na {self.pocet\_dni\_dovolene} dní."

frantisek = Zamestnanec("Frantisek Novotny", "ucetni") # objekt - pri vytvareni se nam zavola \_\_init\_\_

alena = Zamestnanec("Alena Drobna", "sefova")

print(frantisek.cerpani\_dovolene(5))

print(frantisek.pocet\_dni\_dovolene)

print(frantisek.cerpani\_dovolene(15))

print(frantisek.pocet\_dni\_dovolene)

print(frantisek.cerpani\_dovolene(15))

print(alena.pocet\_dni\_dovolene)



**Janka Marschalkova**  [7:40 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683654019433059)

class Zamestnanec: # definici tridy

def \_\_init\_\_(self, jmeno, pozice): # konstruktor tridy - specialni metoda

self.jmeno = jmeno

self.pozice = pozice

def \_\_str\_\_(self): # specialni metoda

return f"Zamestnanec {self.jmeno} pracuje na pozici {self.pozice}"

class Manazer():

def \_\_init\_\_(self, jmeno, pozice, pocet\_podrizenych):

self.jmeno = jmeno

self.pozice = pozice

self.pocet\_podrizenych = pocet\_podrizenych

def \_\_str\_\_(self):

return f"Manazer {self.jmeno} pracuje na pozici {self.pozice} a má počet podřízených {self.pocet\_podrizenych}"

alena = Manazer(jmeno="Alena", pozice="Sefova vsech ucetni", pocet\_podrizenych=3)

print(alena)



**Janka Marschalkova**  [7:50 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683654640790809)

DĚDIČNOST

class Zamestnanec(): # definici tridy

def \_\_init\_\_(self, jmeno, pozice): # konstruktor tridy - specialni metoda

self.jmeno = jmeno

self.pozice = pozice

def \_\_str\_\_(self): # specialni metoda

return f"Zamestnanec {self.jmeno} pracuje na pozici {self.pozice}"

class Manazer(Zamestnanec): # Manazer dedí od Zamestnance - Manazer prebira atributy a metody Zamestnance

def \_\_init\_\_(self, jmeno, pozice, pocet\_podrizenych):

#Zamestnanec.\_\_init\_\_(self, jmeno, pozice)

super().\_\_init\_\_(jmeno, pozice)

self.pocet\_podrizenych = pocet\_podrizenych

alena = Manazer(jmeno="Alena", pozice="Sefova vsech ucetni", pocet\_podrizenych=3)

print(alena.jmeno)

print(alena.pozice)

print(alena.pocet\_podrizenych)

[7:53](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683654822242639)

class Manazer(Zamestnanec): # Manazer dedí od Zamestnance - Manazer prebira atributy a metody Zamestnance

def \_\_init\_\_(self, jmeno, pozice, pocet\_podrizenych):

#Zamestnanec.\_\_init\_\_(self, jmeno, pozice)

super().\_\_init\_\_(jmeno, pozice)

self.pocet\_podrizenych = pocet\_podrizenych

def \_\_str\_\_(self):

return f"{super().\_\_str\_\_()} a má tým o velikosti {self.pocet\_podrizenych}"

alena = Manazer(jmeno="Alena", pozice="Sefova vsech ucetni", pocet\_podrizenych=3)

print(alena)



**Janka Marschalkova**  [8:02 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683655358409749)

class Manazer(Zamestnanec): # Manazer dedí od Zamestnance - Manazer prebira atributy a metody Zamestnance

def \_\_init\_\_(self, jmeno, pocet\_podrizenych):

super().\_\_init\_\_(jmeno, pozice="manazer")

self.pocet\_podrizenych = pocet\_podrizenych

def \_\_str\_\_(self):

return f"{super().\_\_str\_\_()} a má tým o velikosti {self.pocet\_podrizenych}"

alena = Manazer(jmeno="Alena", pocet\_podrizenych=3)

milan = Manazer("Milan Velký", pocet\_podrizenych=5)

print(alena)

print(milan)



**Janka Marschalkova**  [8:09 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683655765803559)

class Manazer(Zamestnanec): # Manazer dedí od Zamestnance - Manazer prebira atributy a metody Zamestnance

def \_\_init\_\_(self, jmeno, pocet\_podrizenych = 3):

super().\_\_init\_\_(jmeno, pozice="manazer")

self.podrizeni = []

def \_\_str\_\_(self):

return f"{super().\_\_str\_\_()} a má tým o velikosti {self.pocet\_podrizenych}"

def pridej\_podrizeneho(self, novy\_podrizeny):

self.podrizeni.append(novy\_podrizeny)

def vypis\_podrizene(self):

podrizeni = ""

for person in self.podrizeni:

podrizeni += person.jmeno + ", "

return podrizeni

frantisek = Zamestnanec("Frantisek", "ucetni")

alena = Manazer(jmeno="Alena", pocet\_podrizenych=3)

alena.pridej\_podrizeneho(frantisek)

print(alena.vypis\_podrizene())



**Janka Marschalkova**  [8:18 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683656335566929)

**Řešení z BR - Třídy**

1. Balík

class Package:

def \_\_init\_\_(self, address, weightInKilos):

self.address = address

self.weightInKilos = weightInKilos

self.delivered = False

def deliver(self):

self.delivered = True

def \_\_str\_\_(self):

if self.delivered:

deliveredText = "Balík byl doručen"

else:

deliveredText = "Balík zatím nebyl doručen."

return "Balík je na adresu {self.address} a váží {self.weightInKilos}. {deliveredText}"

my\_package = Package("Brno Vojtova 5", 20)

2. Kniha

class Book:

def \_\_init\_\_(self, title, pages, price):

self.title = title

self.pages = pages

self.price = price

def discount(self, discountInPercents):

self.price \*= (1 - discountInPercents/100)

def \_\_str\_\_(self):

print(f"Název knihy: {self.title}. Počet stran: {self.pages}. Cena: {self.price}")

kniha = Book("Noc, která mě zabila", 590, 499)

kniha.getInfo()

kniha.discount(10)

kniha.getInfo()

3. Zaměstnanec

class Zamestnanec:

def \_\_init\_\_(self, jmeno, pozice, zkusebni\_doba):

self.jmeno = jmeno

self.pozice = pozice

self.zkusebni\_doba = zkusebni\_doba

self.pocet\_dni\_dovolene = 25

def \_\_str\_\_(self):

text = f"Jméno: {self.jmeno}, pozice: {self.pozice}."

if self.zkusebni\_doba:

text = text + " Je ve zkušební době."

return text

def cerpani\_dovolene(self, dny):

if self.pocet\_dni\_dovolene >= dny:

self.pocet\_dni\_dovolene = self.pocet\_dni\_dovolene - dny

return "Dovolená schválena"

else:

return f"Neschváleno, zbývá pouze {self.pocet\_dni\_dovolene} dní."

(edited)



**Janka Marschalkova**  [8:56 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1683658572461829)

**Řešení BR - Dědičnost**

1. Částečný úvazek

class Zamestnanec:

def \_\_init\_\_(self, name, position):

self.name = name

self.position = position

self.remainingHolidayDays = 25

def takeHoliday(self, days):

if self.remainingHolidayDays >= days:

self.remainingHolidayDays -= days

return f"Užij si to."

else:

return f"Bohužel už máš nárok jen na {self.remainingHolidayDays} dní."

def getInfo(self):

return f"{self.name} pracuje na pozici {self.position}."

class Brigadnik(Zamestnanec):

def getInfo(self):

return f"{super().getInfo()} Má úvazek {self.workload}."

def \_\_init\_\_(self, name, position, workload):

super().\_\_init\_\_(name, position)

self.workload = workload

brigadnik = Brigadnik("Jan Dvořák", "asistent", 0.5)

print(brigadnik.getInfo())

2. Balík

class Balik:

def \_\_init\_\_(self, address, weightInKilos):

self.address = address

self.weightInKilos = weightInKilos

self.delivered = False

def deliver(self):

self.delivered = True

def getInfo(self):

if self.delivered:

deliveredText = "Balík byl doručen"

else:

deliveredText = "Balík zatím nebyl doručen."

print(f"Balík je na adresu {self.address} a váží {self.weightInKilos}. {deliveredText}")

class CennyBalik(Balik):

def \_\_init\_\_(self, address, weightInKilos, value):

super().\_\_init\_\_(address, weightInKilos)

self.value = value

cennyBalik = CennyBalik("Plzeňská 123, Praha", 2, 10000)

cennyBalik.getInfo()

cennyBalik.deliver()

cennyBalik.getInfo()

(edited)

**Tuesday, May 16th**



**Janka Marschalkova**  [8:00 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684260046114259)

nakupy.iloc[5:10] *# rozsah [od: do+1]*

[8:02](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684260125435449)

nakupy.iloc[5:] *# od radky 5 do konce*  
nakupy.iloc[:10] *# od zacatku do radky 10*

**Tuesday, May 23rd**



**Janka Marschalkova**  [6:28 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684859315422609)

staty\_jizni\_ameriky = staty[staty["subregion"] == "South America"]

staty\_jizni\_ameriky["name"]

[6:30](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684859409632829)

staty[staty["gini"] < 30]

[6:32](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684859569370129)

# zresetování indexu + odstraneni sloupce puvodniho indexu

staty[staty["gini"] < 30].reset\_index(drop=True)

[6:36](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684859779859559)

staty[staty["population"] < 1\_000\_000][staty["area"] > 100\_000]



**Janka Marschalkova**  [6:43 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684860192134529)

staty[staty["region"].isin(["Europe", "Africa"])]

[6:47](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684860452754389)

staty[staty["population"].between(1\_000\_000, 5\_000\_000, inclusive='both')]



**Janka Marschalkova**  [7:29 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684862948930109)

# vraci unikatni hodnoty

staty["region"].unique()

# název regionu a počet statu

staty["region"].value\_counts()

[7:30](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684863029829279)

# nazev regionu a procento

staty["region"].value\_counts(normalize=True)



**Janka Marschalkova**  [7:40 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684863614961459)

u202["znamka"].isna() #.isnull()

[7:41](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684863716725219)

*Kterým studentům chybí známka ve tříde 202?*  
*u202[u202["znamka"].isna()] #.isnull()*



**Janka Marschalkova**  [7:47 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684864059946519)

u202.dropna(subset=["znamka", "predmet"])  
*# do subset zadávám ty sloupce, ktere chci aby se zkontrolovali NaN hodnoty*  
*# bez subset se zkontroluji vsechny sloupce*

[7:52](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684864339933249)

pandas.concat([u202, u203, u302], ignore\_index=True)

[7:55](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684864512479409)

u202["mistnost"] = "u202"  
u203["mistnost"] = "u203"  
u302["mistnost"] = "u302"

[7:58](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684864690758029)

[maturita.to](http://maturita.to/)\_csv('maturita.csv', index=False)

[8:02](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1684864937547089)

maturita = pandas.concat([u202, u203, u302], ignore\_index=True)  
maturita\_nova = maturita.dropna(subset=["znamka"])  
maturita\_nova

**Tuesday, May 30th**



**Pavla Moravkova**  [6:27 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1685464067942419)

typicky obrazek, jak znazornit typy joinu, aneb onen parametr how

image.png

:smile:3



**Janka Marschalkova**  [6:39 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1685464776943699)

predsedajici["den"] = predsedajici["den"].str.strip()



**Janka Marschalkova**  [6:57 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1685465833654459)

staty = pandas.read\_json("staty.json")  
staty = staty.set\_index("name")  
staty["population\_density"] = staty["population"] / staty["area"]  
staty = staty.sort\_values("population\_density")  
staty

[7:01](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1685466105358449)

*# abych nemusela zbycne vypisovat dotaz, muzu si nastavit vlastni identifikator konkretniho radku pomoci set\_index*  
staty = staty.set\_index("name")print(staty.loc["Macao"])  
macao = staty[staty["name"] == "Macao"]  
macao (edited)



**Janka Marschalkova**  [7:18 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1685467116591619)

Vkládám opravené linky na count a size na konci 05 notebooku - <https://pandas.pydata.org/docs/reference/api/pandas.core.groupby.DataFrameGroupBy.count.html>, <https://pandas.pydata.org/docs/reference/api/pandas.core.groupby.DataFrameGroupBy.size.html>



**Janka Marschalkova**  [8:33 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1685471608972449)

Řešení z BR Propojeni dat, agregace :

import pandas

studenti1 = pandas.read\_csv("studenti1.csv")

studenti2 = pandas.read\_csv("studenti2.csv")

# Načti dva datové sety studentů do oddělených pandas DataFrame a pomocí funkce `concat` je spoj do jednoho setu.

studenti = pandas.concat([studenti1, studenti2], ignore\_index=True)

print(studenti)

# Pokud studentovi chybí ročník, znamená to, že již nestuduje. Pokud mu chybí číslo skupiny, znamená to, že jde o dálkového studenta. Kolik studentů v datovém setu již nestuduje a kolik jsou dálkoví studenti?

print(studenti[studenti["ročník"].isnull()].shape)

print(studenti[studenti["kruh"].isnull()].shape)

# Vyčisti data od studentů, kteří nestudují nebo studují jen dálkově. Nadále budeme pracovat pouze s prezenčními studenty.

studenti = studenti.dropna()

studenti.shape

# Zjisti průměrný prospěch studentů v každém oboru.

studenti.groupby("obor").count()

studenti.groupby("obor")["prospěch"].mean()

# Načti datový set s křestními jmény. Proveď join s tabulkou studentů tak, abychom věděli pohlaví jednotlivých studentů.

jmena = pandas.read\_csv("jmena.csv")

pandas.merge(studenti, jmena)

studenti\_se\_jmeny = pandas.merge(studenti, jmena)

studenti\_se\_jmeny.columns

# Zjisti, zda na naší fakultě studují IT spíše ženy nebo spíše muži.

studenti\_se\_jmeny.groupby("pohlaví").count()

# studenti\_se\_jmeny.groupby("pohlaví").size()

(edited)

[8:36](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1685471816795819)

Řešení z BR Vizualizace:

import matplotlib.pyplot as plt

import pandas

import requests

# 1

url = "<https://kodim.cz/czechitas/progr2-python/python-pro-data-1/vizualizace/excs/hazeni-kostkami/assets/kostky.csv>"

r = requests.get(url, allow\_redirects=True, verify=False)

open('kostky.txt', 'wb').write(r.content)

# Načtěte tato data do tabulky a zobrazte histogram hodů. Zvolte vhodné rozložení přihrádek a zodpovězte následující dotazy:

# Jaká je nejčastější hodnota, která na dvou kostkách padne?

# Je větší šance, že padne hodnota 12 než že padne hodnota 2?

kostky = pandas.read\_csv("kostky.txt")

kostky.hist(bins=[2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12])

# kostky.hist(bins=11)

plt.show()

url = "<https://kodim.cz/czechitas/progr2-python/python-pro-data-1/vizualizace/excs/call-centrum/assets/callcentrum.csv>"

r = requests.get(url, allow\_redirects=True, verify=False)

open('callcentrum.txt', 'wb').write(r.content)

# 2

callcentrum = pandas.read\_csv("callcentrum.txt")

callcentrum = callcentrum["hodnota"].str.split(':', expand=True).astype(int)

callcentrum["vteriny"] = callcentrum[0] \* 60 + callcentrum[1]

callcentrum["vteriny"].hist()

plt.show()

callcentrum["vteriny"].plot(kind='box', whis=[0, 100])

plt.show()

# 3

snih = [

[1968, 480, 351],

[1969, 462, 663],

[1970, 443, 490],

[1971, 518, 444],

[1972, 537, 420],

[1973, 446, 941],

[1974, 446, 691],

[1975, 450, 477],

[1976, 356, 395],

[1977, 381, 652],

[1978, 345, 525],

[1979, 430, 762],

[1980, 266, 316],

[1981, 533, 781],

[1982, 471, 769],

[1983, 407, 801],

[1984, 526, 633],

[1985, 391, 488],

[1986, 361, 624],

[1987, 470, 471],

[1988, 506, 514],

[1989, 333, 208],

[1990, 462, 909],

[1991, 438, 443],

[1992, 364, 488],

[1993, 452, 579],

[1994, 484, 519],

[1995, 460, 809],

[1996, 465, 682],

[1997, 431, 814],

[1998, 463, 595],

[1999, 460, 512],

[2000, 503, 750],

[2001, 462, 951],

[2002, 429, 413],

[2003, 405, 738],

[2004, 477, 777],

[2005, 385, 316],

[2006, 368, 417],

[2007, 513, 635],

[2008, 448, 689],

[2009, 525, 443],

[2010, 427, 225],

[2011, 460, 618],

[2012, 417, 742],

[2013, 517, 247],

[2014, 466, 552],

[2015, 523, 441],

[2016, 422, 690],

[2017, 420, 699]

]

snihdf = pandas.DataFrame(snih, columns=['rok', 'hora', 'udoli'])

snihdf = snihdf.set\_index('rok')

snihdf.plot(kind='box', whis=[0, 100])

plt.show()

**Tuesday, June 6th**



**Janka Marschalkova**  [6:44 PM](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1686069859932319)

html.find('h1, h2')  
*#html.find('h1', 'h2') #takhle ne*

white_check_markeyesraised_hands

[6:47](https://python-1-talk.slack.com/archives/C0504NC0WJ3/p1686070055250499)

#varianta 1: vypis pomoci printu v cyklu

for odstavec in html.find('p'):

print(odstavec.text)

#varianta 2: list comprehension - vytvoreni seznamu s textem odstavcu

[odstavec.text for odstavec in html.find('p')]

Message 05-kód-z-lekce