

# NTI/PJPA - Programovací jazyk Python (2023)

Dashboard / Courses / FM / NTI / 2023/24 / NTI/PJPA - Programovací jazyk Python (2023) / Úkoly a cvičení / Úkol 10. - Cenzor (starší zkoušková otázka)

## My courses

- 🔗 ITE/CITE - Číslicová technika (2022)
- 🔗 ITE/EDK - Elektronická dokumentace (2022)
- 🔗 ITE/MTLB - Výpočty, simulace a vizualizace Matlab (2022)
- 🔗 ITE/SGI - Signály a informace (2023)
- 🔗 ITE/ZKO - Základy konstruování (2023)
- 🔗 KAP/JULA - Úvod do lin. algebry a diskrétní mat. (2022)
- 🔗 MTI/AUG1 - Algoritmizace a programování 1 (2022)
- 🔗 Samostatné úlohy z předmětu Algoritmizace a programování 1
- 🔗 MTI/AUG2 - Algoritmizace a programování 2 (2022)
- 🔗 MTI/CIP - Číslicové počítače (2023)
- 🔗 MTI/IOS - Datové systémy (2023)
- 🔗 MTI/PJJC - Programování v jazyce C/C++ (2023)
- 🔗 MTI/USTIN - Softwareové inženýrství (2023)
- 🔗 MTI/UOI - Úvod do inženýrství (2022)
- 🔗 MTI/NAIPW - Vývoj aplikací pro Windows (2023)
- 🔗 MTI/ALD - Algoritmizace a datové struktury (2023)
- 🔗 MTI/OPS - Operační systémy (2023)
- 🔗 NTI/PJPA - Programovací jazyk Python (2023)
- 🔗 Počítačové sítě
- 🔗 NTI/PST - Počítačové sítě (2022)
- 🔗 NTI/ISH - Úvod do Shellu (2022)
- 🔗 NTI/TWIS - Trůnba WWW stránek (2023)
- 🔗 NTI/USA - Úvod do statistické analýzy (2023)

## Navigation

- ▼ Dashboard
- 🏠 Site home
- Site pages
- 📅 Courses enrollment (STAG)
- 📅 Courses unenrollment
- 🔗 propojení se STAGem
- ▼ My courses
- ITE/CITE - Číslicová technika (2022)
- ITE/EDK - Elektronická dokumentace (2022)
- ITE/MTLB - Výpočty, simulace a vizualizace Matlab ...
- ITE/SGI - Signály a informace (2023)
- ITE/ZKO - Základy konstruování (2023)
- KAP/JULA - Úvod do lin. algebry a diskrétní mat. (2...
- MTI/AUG1 - Algoritmizace a programování 1 (2022)
- Samostatné úlohy z předmětu Algoritmizace a progra...
- MTI/AUG2 - Algoritmizace a programování 2 (2022)
- MTI/CIP - Číslicové počítače (2023)
- More...
- ▼ Courses
- ▼ FM
- DFM
- ITE
- MTI
- ▼ NTI

- 2023/24
- 🔗 NTI/ADA - Algoritmy a datové struktury (2023)
- 🔗 NTI/ATP - Automaty a formální jazyky (2023)
- NTI/ALD - Algoritmizace a datové struktury (2023)
- 🔗 NTI/LAMP - Alternativní metody programování (2023)
- 🔗 NTI/ARMO - Aplikace počítačových modelů (2023)
- 🔗 NTI/IARP - Architektura počítačů (2023)
- 🔗 NTI/CFD - Výpočetní mechanika tekutin (2023)
- 🔗 NTI/DPG - Distribuované programování (2023)
- 🔗 NTI/EMM - Experimentální metody v mechanice (2023)
- 🔗 NTI/JPD - Jazyky pro popis dat (2023)
- 🔗 NTI/XAS - Kybernetická bezpečnost a šifrování (2023)
- ▼ NTI/PJPA - Programovací jazyk Python (2023)
- Participants
- 📋 Competencies
- 📋 Grades
- Programovací jazyk Python - PIPA LS 2024
- ▼ Úkoly a cvičení
- 🔗 Úkol 0 - přihlaste se na github/tulcz
- 🔗 Úkol 1 - první program
- 🔗 Úkol 2 - kmenování čtyřúhelník
- 🔗 Úkol 3 - transformace dat
- 🔗 Úkol 4 - Caesarova šifra
- 🔗 Úkol 5 - algoritmizace problému
- 🔗 Úkol 6 - regulární výrazy
- 🔗 Úkol 7 - zpracování JSON a HTML dat
- 🔗 Úkol 8 - Poker (starší zkoušková otázka)
- 🔗 Úkol 9 - Binární Vyhledávací Strom
- 🔗 Úkol 10. - Cenzor (starší zkoušková otázka)
- 🔗 Účast na přednáškách
- 1. Ponoříme se do Pythonu - Úvod do předmětu
- 2. proměnné a konstanty
- 3. strukturované datové typy - kolekce a sekvence
- 4. další vlastnosti jazyka
- 5. testování kódu
- 6. standardní textové formáty a jejich zpracování
- 7. funkce a jejich pokročilé využití
- 8. tvorba vlastních typů, principy OOP
- 9. tvorba aplikací s CLI (command line interface)
- 10. další moduly standardní knihovny jazyka Python
- 11. Výkonnost Python programů
- Topic 14
- 2022/23
- 2021/22
- 2020/21
- 2019/20
- 2018/19
- 🔗 Aplikace GIS
- 🔗 Diplomové a bakalářské práce 2021/22
- 🔗 Geografické informační systémy
- 🔗 Kopetschke DP, BP, PRO, PRJ 2016/17
- Počítačové sítě
- RSS
- Kurzy mimo STAG
- 🔗 Bezpečnost práce na elektrickém zařízení v laborat...
- 🔗 Admission Test Mechatronics 2024
- 🔗 Samostatný elektrotechnik pro elektromagnetickou k...
- 🔗 Měření geometrie
- 🔗 Hodnocení kvality výuky (BS-IT 2021/22)
- 🔗 Virtuální setkání akademické obce FM
- 🔗 Příprava na přijímací zkoušky z informatiky
- 🔗 Příprava na přijímací zkoušky z matematiky
- 🔗 Studentská konference Fakulty mechatroniky
- 🔗 MTI/CSHARP - TI
- 🔗 Podnikový informační systém SAP
- NÁVODY, MANUÁLY
- Další podpůrné materiály
- Kurzy pro zaměstnance TUL
- Externí kurzy
- FA
- FE
- FP

## 🔑 Úkol 10. - Cenzor (starší zkoušková otázka)

Opened: Sunday, 27 November 2022, 12:00 AM  
Due: Tuesday, 21 May 2024, 11:55 PM

Také v tomto úkolu si můžete vyzkoušet starší zadání ze zkoušky a udelat si tak lepší představu o její náročnosti.

U zkoušky máte na vyřešení podobného problému **120 minut**. Můžete použít libovolnou literaturu, včetně hledání na webu. Programuje se na počítači. **Úlohu musíte řešit samostatně a řešení pak následně obhájit u ústní zkoušky.**

A teď již to samotné zadání dnešního úkolu.

Cílem je vytvořit jednoduchý cenzorovací systém. Svoboda slova je rozhodně důležitá, ale jsou situace, kdy na webu nechcete zobrazit cokoliv - například stránky určené dětem apod. Budete tedy tvořit program, který určitá předem daná slova dokáže z textu odfiltrovat.

Vstupem do programu jsou dva soubory. První z nich je HTML stránka obsahující text, který je potřeba vyčistit. Druhý soubor obsahuje seznam zakázaných slov v jednoduchém formátu 1 slovo = 1 řádek.

Program musí být možné ovládat argumenty z příkazové řádky.

Ovládání programu musí zvládnout následující parametry:

- i, --input : soubor který má být upraven (běžný text)
- l, --list : soubor se seznamem zakázaných slov - jedno slovo jeden řádek
- c, --clean : přepínač vyčištění souboru od html - viz. dále
- o, --output: výstupní soubor, pokud není volba použita, tak vypsat data na obrazovku
- h, --help : nápověda - o čem program je a jak se ovládá

### Požadované vlastnosti:

- Nahradiť všechna zakázaná slova v textu sekvencí znaků #. Délka nahrazené sekvence musí být rovná délce původního (nahrazeného) slova.
- Pokud je zadán parameter -c (--clean) musí program odstranit z textu všechny HTML značky - to znamená na výstup pouze text.
- Pokud bude program spuštěn s parametrem -o *jmeno*, musí vypsat výstup do souboru *jmeno*. Jinak se provede výpis na standardní výstup (na obrazovku).

**Slova v textu:** pro dělení slov můžete použít metodu split u řetězce, nebo u regulárního výrazu, což umožňuje dělit rovnou i přes složitéjší kombinace interpunkčních znamének, nejen přes jeden konkrétní řetězec (mezera, čárka apod.)

**HTML značky:** jsou formátovací sekvence HTML. Od běžného textu jsou odděleny znaky < a >.

Další požadavky pro odezdání:

- musí jít o testovatelný kód - vše tedy musí být buď ve funkcích, nebo uzavřeno pod `__name__ == "__main__"`
- pro každou funkci, nebo metodu, kterou vytvoříte musí být v souboru `unif_test_censor.py` alespoň jeden test. Zde by mělo jít minimálně o testy na odfiltrování slova z řetězce a odstranění HTML značek. Testy budou spouštěny pomocí PyTest. Jako obvykle nemusíte testovat build-in funkce, v případě této úlohy načítání souboru, nebo argparse.
- modul musí být v adresáři `cv10` a musí se jmenovat `censor.py`. Šablona je jako vždy k dispozici ve vzorovém repozitáři.
- výsledný kód musí při testu programem PyLint se standardním nastavením získat alespoň 8 bodů. Za každý bod dolů, máte bod dolů i vy.
- Pro řešení můžete použít pouze moduly ze standardní instalace Pythonu 3.x. Dvě výjimky: pro práci s argumenty příkazové řádky je povoleno použít modul Click, pro zpracování html BS4.

## Submission status

Submission status	This assignment does not require you to submit anything online
Grading status	Not graded
Time remaining	10 hours 24 mins remaining
Last modified	-
Submission comments	🔗 Comments (0)

→ Úkol 9. - Binární Vyhledávací Strom

Jump to...

Účast na přednáškách →

- > FS
- > FT
- > FZS
- > Rektorát
- > UKN
- > UZS
- > Velejné kurzy
- > \_ARCHIV

You are logged in as Martin Šimon (Log out)  
NTU/PIPA (2023)  
[Get the mobile app](#)