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/ULA - Úvod do lin. algebry a diskrétní mat. (2022)
MTI/ALG1 - Algoritmizace a programování 1 (2022)
Samostatné úlohy z předmětu Algoritmizace a programování 1
MTI/ALG2 - Algoritmizace a programování 2 (2022)
MTI/CIP - Číslicové počítače (2023)
MTI/DBS - Databázové systémy (2023)
MTI/PRJC - Programování v jazyce C/C++ (2023)
MTI/STIN - Softwarové inženýrství (2023)
MTI/UDI - Úvod do inženýrství (2022)
MTI/VAPW - Vývoj aplikací pro Windows (2023)
NTI/ALD - Algoritmizace a datové struktury (2023)
NTI/OPS - Operační systémy (2023)
NTI/PJPA - Programovací jazyk Python (2023)
Počítačové sítě
NTI/PST - Počítačové sítě (2022)
NTI/SH - Úvod do Shellu (2022)
NTI/TWS - Tvorba WWW stránek (2023)
NTI/USA - Úvod do statistické analýzy (2023)

Navigation
Dashboard
Site home
Site pages
Courses enrollment (STAG)
Courses unenrolment
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/ULA - Úvod do lin. algebry a diskrétní mat. (2...
MTI/ALG1 - Algoritmizace a programování 1 (2022)
Samostatné úlohy z předmětu Algoritmizace a progra...
MTI/ALG2 - 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/AFI-P - Automaty a formální jazyky (2023)
NTI/ALD - Algoritmizace a datové struktury (2023)
NTI/AMP - Alternativní metody programování (2023)
NTI/APMO - Aplikace počítačových modelů (2023)
NTI/ARP - 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/KAS - Kybernetická bezpečnost a šifrování (2023)
NTI/PJPA - Programovací jazyk Python (2023)
Participants
Competencies
Grades
Programovací jazyk Python - PJPA LS 2024
Úkoly a cvičení
úkol 0 - přihlaste se na gitlab.tul.cz
úkol 1. - první program
úkol 2. - konvexní 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 zkouškova otázka)
úkol 9. - Binární Vyhledávací Strom
úkol 10. - Cenzor (starší zkouškova otázka)
Účast na přednáškách
1. Ponořme se do Pythonu - úvod do předmětu
2. proměnné a konstanty
3. struktuované 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...
Fraktální geometrie
Hodnocení kvality výuky (BS-IT 2021/22)
Virtuální setkání akademické obce FM
Příprava na přijímačky z informatiky
Příprava na přijímačky z matematiky
Studentská konference Fakulty mechatroniky
MTI/CSHARP - T1
Podnikový informační systém SAP
NÁVODY, MANUÁLY
Další podpůrné materiály
Kurzy pro zaměstnance TUL
Externí kurzy
FA
FE
FP

úkol 7. - zpracování JSON a HTML dat

Opened: Wednesday, 26 October 2022, 1:00 AM
Due: Tuesday, 16 April 2024, 11:59 PM

V tomto úkolu si procvičíte zpracování známých souborových formátů.

Na vstupu programu jsou dány dva soubory. První soubor, ve formátu HTML, obsahuje **výsledky několika závodů** - jména a časy závodníků. Druhý pak obsahuje **databázi závodníků uloženou jako JSON.** V databázi je krom jiných údajů, také id každého ze závodníků, které budete potřebovat najít podle jména a příjmení.

Cílem je vytvořit program, který **data z těchto dvou zdrojů propojí.** Konkrétně nás bude zajímat závod štafet. **Cílem je tedy ke každému závodníkovi ve štafetě pomocí programu najít jeho id.**

Pozor na to, že data jsou reálná a tudíž nejsou ideální, takže nastane i případ, že id z výsledků najít nejde. I tuto situaci ale musí program korektně ošetřit, tedy musí pokračovat dál a tato špatná data uložit pro další zpracování. Výsledky musí program zapsat do výstupních souborů.

Vstupní data

V repozitáři předmětu v adresáři **cv07** najdete soubory result.html a competitors.json.

Soubor **result.html** obsahuje výsledky několika závodů, ale pro vás jsou důležité **pouze výsledky závodu štafet** (relay) - je to poslední ze závodů na stránce. Data jsou uložena poněkud nešťastně - jako text uvnitř jednoho odstavce v html. Jde ale o naprosto reálný příklad uložení historických dat na webu mezinárodní sportovní federace.

Nejprve tedy musíte tato data ze souboru vyparsovat a následně rozdělit konkrétní štafetu na jednotlivé závodníky. Pozor - při rozdělení musíte zachovat také informaci o umístění štafety v závodě. Pořadí jednotlivých členů štafety totiž budete potřebovat pro výsledek.

Soubor **competitors.json** obsahuje informace o sportovcích. Jedná se vlastně o databázi, uloženou ve formátu JSON. Data jsou uložena jako list objektů. Každý objekt reprezentuje jednoho závodníka pomocí následujících informací - id, jméno, příjmení, státní příslušnost, rok narození, pohlaví. Například může vypadat takto:

{
"id": 10816,
"firstname": "Jiri",
"lastname": "Hradil",
"nationality": "CZE",
"birth": "1986",
"gender": "M"
}

Doplňující informace k datům:

- Jména souborů zapište v programu jako konstanty (není potřeba zadávat parametrem)
- Můžete předpokládat, že data budou vždy validní dle specifikace daného formátu - jak json tak html.
- Můžete předpokládat, že jména závodníků obsahují pouze velká a malá ASCII písmena, mezeru a pomlčku

Výstup z programu

Jak už bylo řečeno, je vaším úkolem **přiřadit jednotlivým závodníkům ve štafetě jejich id.** Je tedy rozumné vyhledávat nejprve příjmení, v případě že nestačí pak i křestní jméno. Můžete ale hledat i celý řetězec najednou.

Složitější situace, jako že závodník má ve jméně více slov než dvě a není jasné co z toho je jméno a co příjmení, můžete ignorovat. Respektive je v pořádku, když program takového závodníka označní za nedohledatelného. Typicky jde o dánská či španělská jména, ale může se jednat i o závodníky z jiných zemí.

Program musí **zapsat výsledky do třech souborů**. První z nich bude obsahovat kompletní výsledek, tedy všechny záznamy (pozitivní i negativní match), druhý stručný přehled nalezených a třetí ty závodníky, které se nepodařilo dohledat.

První soubor - tedy kompletní výsledek je třeba zapsat ve formátu JSON:

{
"id": 10816,
"result": 2
"time": "2:05:26"
}

Při zpracování dat narazíte také na situaci, že závodníka v souboru competitors nenajdete. Ať už proto že tam jeho jméno zapsané není, a nebo proto, že v souboru s výsledky je zaspsané s překlepem. A nebo to bude již zmíněné jméno z více slov než dvou.

V tomto případě uložte jako id hodnotu False a jméno závodníka zapište pod klíč "no_match". Závodníka zapište jak do souboru kompletních výsledků, tak do extra souboru errors.txt (viz dále) . Výsledek by tedy měl vypadat například následovně:

{
"id": "False",
"result": 1,
"time": "2:25:18",
"no_match": "Elisabeth Hohenwarter"
}

Kompletní data ve výše uvedeném formátu zapište souboru output.json jako list slovníků. Aby soubory s řešením bylo možné porovnat, použijte pro metodu json.dumps následující nastavení: json.dumps(results, indent=4, sort_keys=True). Ve vzorovém souboru relay.py je metoda output_json, kterou můžete využít pro uložení. Nebo pro inspiraci jak má vypadat formátování.

Druhý soubor compare.txt bude jednoduchý txt formát. Na každý řádek v souboru zapište jedno nalezené id mezeru a pořadí závodníka. Id v tomto případě musí být seřazená vzestupně. Závodníky, kteří mají id = False v tomto výstupu ignorujte.

Třetí soubor bude opět jednoduchý txt soubor. Soubor se bude jmenovat errors.txt. Na každý řádek zapište, jméno závodníka, kterého se nepodařilo dohledat. Tedy závodníky, které jste ve druhém souboru vynechali.

Další požadavky:

- Zdrojová data nesmíte nijak upravovat (nahrazovat znaky apod.)
- Váš modul musí být v adresáři **cv07** a musí se jmenovat **relay.py**.
- Do repozitáře neukládejte soubory s výsledky - nechte výsledný program aby je vytvořil. Zdrojové soubory v repozitáři být mohou.
- Jako obvykle musíte pro řešení využít **pouze built-in balíčky** ze standardní instalace Pythonu 3.x. Navíc ale můžete využít parser BeautifilSoup (BS4) pokud chcete. A samozřejmě vaše vlastní balíčky.
- Výsledný kód musí při testu programem PyLint se standardním nastavením získat alespoň 8 bodů. Za každý bod dolů, máte bod dolů i vy

Nápověda na závěr:

- Snadno si spočítáte, že štafet bylo v závodě celkem 24 - 9 ženských a 15 mužských. To dává celkem 72 jmen či id.
- Problematických jmen je ve výsledcích štafet 10 či 9. Tady může být váš výsledek jiný, záleží na algoritmu jméno+příjmení.
- Soubor compare.txt bude mít 62(63) řádků. První řádek - tedy nejmenší id je 182 6, poslední řádek pak 18442 1.

cv07-template.zip 31 January 2024, 6:31 PM

Submission status

Submission status	This assignment does not require you to submit anything online
Grading status	Graded
Time remaining	The due date for this assignment has now passed
Last modified	-
Submission comments	Comments (0)

Feedback

Grade	10.00 / 10.00
Graded on	Tuesday, 21 May 2024, 11:30 AM
Graded by	LM Lukáš Mázl
Feedback comments	V pořádku