



Relatório

Processamento de Linguagens

Aluno/os: Pedro Vieira Simões / Duarte Melo

Professor/es: Alberto Simões

Licenciatura em Engenharia de Sistemas Informáticos

Barcelos, novembro, 2021

Resumo

Trabalho prático a pares com o objetivo de escolher um dos 3 temas indicados pelo professor, neste caso, o grupo escolheu o tema B: Comma Separated Values.

Consiste num programa Python + Ply capaz de ler um file CSV e reproduzir outputs em HTML e LaTeX.

Conteúdo

1. Introdução	5
2. Expressões Regulares	6
3. Outputs	8

1. Introdução

Tal como mencionado anteriormente, este projeto visa reproduzir resultados perante o input do utilizador na consola, tendo como dados a informação registada em uma tabela CSV, tendo esta informação a cerca de vários países, indicando suas capitais, línguas e moeda.

A seleção deste tema por parte do grupo teve como fundamento a ideia de que este seria o mais fácil de ser realizado e interpretado.

O grande desafio aqui foi conseguir separar tudo por vírgulas e detetar valores dentro de aspas.

Foi criada uma classe Reader, um menu de interação e algumas funções de auxílio algorítmico.

Os dados são lidos primeiramente pelo lexer e guardados numa dictionary, onde mais tarde será usada para fazer o output de resultados.

```
def read(self):
    my_dict = {}
    self.lexer.input(slurp(self.filename))
    i=0
    for token in iter(self.lexer.token, None):
        if i<4:
            my_dict[token.type] = []
            i+=1
        else:
            my_dict[token.type].append(token.value)
    return my_dict
```

2. Expressões Regulares

Decidimos usar 5 tokens diferentes, cada uma para as respectivas colunas da tabela csv e a NEWLINE para ler os “\n”

```
# Tokens
tokens = ("COUNTRY", "CAPITAL", "CURRENCY", "LANGUAGE", "COMMENTARY", "NEWLINE")
```

Utilizamos também 3 estados diferentes, sendo que o “country” seria o estado inicial (“INITIAL”):

```
# States
states = (
    ("capital", "exclusive"),
    ("currency", "exclusive"),
    ("language", "exclusive"),
)
```

Utilizamos estes para perante cada leitura para cada token, avançar para o estado seguinte.

Leitura de COUNTRY

Para ler a coluna do país foi usada uma expressão simples `r"[^,]+"`, pois estando este em primeiro lugar da tabela e sendo que o mesmo é único, todo o texto até a vírgula será um país.

```
r"[^,]+"
```

Após a leitura de cada palavra, é indicado o type da palavra lida e para qual estado o lexer deve saltar.

```
def t_COUNTRY(self, t):
    r"[^,]+"
    t.type = "COUNTRY"
    t.lexer.begin("capital")
    return t
```

Leitura de CAPITAL E CURRENCY

Para ler a coluna da capital e para a coluna da moeda foi usada uma expressão mais complexa, pois neste caso é possível que haja valores entre aspas, algo que temos de ter em atenção nesta entrega.

```
r'"([A-Z][a-z]*,?\s?)*"|((([A-Z]|[a-z])[^,]+))'
```

Assim é capaz de ler qualquer palavra dentro de aspas (vírgulas incluídas) ou ler qualquer palavra até à próxima vírgula (caso detete que o primeiro carácter não é ' ' ').

Leitura de LANGUAGE

Para esta expressão não foi preciso criar uma expressão regular tão complexa, pois apenas poderia ler tudo dentro de aspas até ao fim, ou qualquer palavra sem aspas até ao "\n".

```
r'"(.+)"|((([A-Z]|[a-z])[^\\n]+))'
```

Leitura de COMMENT

Expressão regular que lê o comentário.

Lê uma linha que comece num # e termine num \n, dá pass dado que não é necessário armazenar comentários.

```
COMMENTARY  
r"#\[^\n]+"  
pass
```

Leitura de NEWLINE

Expressão regular que lê o \n e dá pass.

```
r"\n"  
pass
```

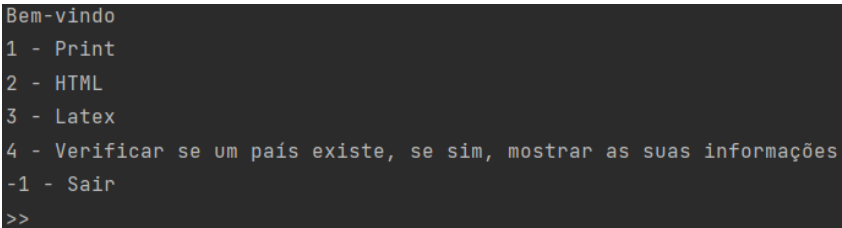
Em relação às abordagens das expressões regulares, o grupo procurou ser o mais objetivo possível de forma a arranjar a solução mais completa.

3. Outputs

De forma ao programa ser o mais eficiente possível, a leitura da tabela CSV é feita uma vez apenas e guardada numa dictionary, esta foi usada para fazer os respetivos outputs

Em relação ao input do utilizador:

- Utilizador pode escolher entre 5 Opções no menu:

- 

- Nas opções 1/2/3:
 - Poderá escolher de que tema pretende ver a informação.
 - Poderá também introduzir entre vírgulas 2 ou mais temas para visualizar, por exemplo: (country,capital) -> printa 2 colunas, uma dos países e outra das capitais
 - No caso de não introduzir nada, ou de um valor que não esteja incluído no cabeçalho da tabela, irá produzir o documento CSV inteiro no formato escolhido.

Consola

Inicialmente, para teste, criamos a funcionalidade de
printar todos os resultados em consola. Achamos por bem,
mais tarde, manter essa funcionalidade de forma que dê
mais funcionalidades ao programa.

```
Caso contrário insira um token: country, capital
COUNTRY CAPITAL
Afghanistan Kabul
Albania Tirane
Algeria Algiers
Andorra Andorra la Vella
Angola Luanda
Antigua & Barbuda Saint John's
Argentina Buenos Aires
Armenia Yerevan
Australia Canberra
Austria Vienna
Azerbaijan Baku
The Bahamas Nassau
Bahrain Manama
Bangladesh Dhaka
Barbados Bridgetown
Belarus Minsk
Belgium Brussels
Belize Belmopan
Benin Porto-Novo
Bhutan Thimphu
Bolivia La Paz, Sucre
Bosnia and Herzegovina Sarajevo
Botswana Gaborone
Brazil Brasilia
Brunei Bandar Seri Begawan
Bulgaria Sofia
Burkina Faso Ouagadougou
Burundi Bujumbura
Cambodia Phnom Penh
```

HTML

Reproduz uma tabela em HTML, foi acrescentado também um bocado de CSS, de forma a
facilitar a leitura dos resultados.

COUNTRY	CAPITAL	CURRENCY	LANGUAGE
Afghanistan	Kabul	Afghani	Dari Persian, Pashto
Albania	Tirane	Lek	Albanian
Algeria	Algiers	Algerian Dinar	Arabic, Tamazight, French
Andorra	Andorra la Vella	Euro	Catalan
Angola	Luanda	Kwanza	Portuguese
Antigua & Barbuda	Saint John's	East Caribbean Dollar	English
Argentina	Buenos Aires	Argentine Peso	Spanish
Armenia	Yerevan	Dram	Armenian
Australia	Canberra	Australian Dollar	English
Austria	Vienna	Euro	German
Azerbaijan	Baku	Manat	Azerbaijani
The Bahamas	Nassau	Bahamian Dollar	English
Bahrain	Manama	Bahraini Dinar	Arabic
Bangladesh	Dhaka	Taka	Bangla
Barbados	Bridgetown	Barbadian Dollar	English
Belarus	Minsk	Belarusian Ruble	Belarusian, Russian
Belgium	Brussels	Euro	Dutch, French, German
Belize	Belmopan	Belize Dollar	English
Benin	Porto-Novo	West African CFA Franc	French
Bhutan	Thimphu	Ngultrum	Dzongkha
Bolivia	La Paz, Sucre	Boliviano	Spanish, Quechua, Aymara
Bosnia and Herzegovina	Sarajevo	Convertible Mark	Bosnian, Croatian, Serbian
Botswana	Gaborone	Pula	English, Tswana
Brazil	Brasilia	Real	Portuguese
Brunei	Bandar Seri Begawan	Brunei Dollar	Malay
Bulgaria	Sofia	Lev	Bulgarian

\documentclass{article}\begin{document}\begin{center}\begin{tabular}{|c|} \hline CURRENCY\{0.5ex\} \\ \hline Afghani \\ \hline Lek \\ \hline Algerian Dinar \\ \hline Euro \\ \hline Kwanza \\ \hline East C. al African CFA Franc \\ \hline Colon \\ \hline West African CFA Franc \\ \hline Croatian \\ \hline Cuban Peso \\ \hline Euro \\ \hline Czech Koruna \\ \hline Danish Krone \\ \hline Djiboutian Franc \\ \hline Ea line Yen \\ \hline Jordanian Dinar \\ \hline Tenge \\ \hline Kenyan Shilling \\ \hline Kiribati Dollar \\ \hline North Korean Won \\ \hline South Korean Won \\ \hline Euro \\ \hline Kuwaiti Dollar \\ \hline Som \\ \hline Norwegian Krone \\ \hline Omani Rial \\ \hline Pakistani Rupee \\ \hline United States Dollar \\ \hline Balboa \\ \hline Papa New Guinean Kina \\ \hline Guarani \\ \hline Nuevo Sol \\ \hline Phillipine