

SPLN

MI-EI

Teste, 7 de Junho de 2021

Questão 1. Escreva da forma mais simples e elegante que conseguir:

- (a) Considere que uma lista homogênea em PT é uma lista de palavras separadas por ‘,’ e pela conjunção ‘e’ na última separação. Escreva uma função que dado um texto extraia e imprima as listas homogêneas.

- Ele gosta de **bicicletas e skates**. →
[bicicletas, skates]
- Ela, porém **dia e noite**, só pensa em **viajar, ler e sonhar**. →
[dia, noite]
[viajar, ler, sonhar]

- (b) Uma função Python que receba como argumento um dicionário e devolva o dicionário substituindo nas chaves de modo a ficarem em camelCase:

```
d = {'chave um': 'valor um', 'chave dois': 'v2', 'e chave três': 'v3'}

repl_spaces(d) # {'chaveUm': 'valor um', 'chaveDois': 'v1', 'eChaveTrês': 'v3'}
```

Questão 2. Pretendemos criar uma tabela de polaridade que a cada palavra associe um valor $\in [-1..1]$.

Considere que teve acesso a um site Travel.com que tem milhões de avaliações de estabelecimentos. Seguem-se dois exemplo de avaliação:

```
*      O Hotel Camões em Vigo tem péssimo serviço de bar, cama cheia de pulgas. Não recomento.
***** O Hotel Colombo em Braga tem um bom de bacalhau, tem boa vista para o mar. Recomendo.
```

Note que:

- Cada avaliação ocupa uma única linha.
 - As estrelas iniciais indicam a polaridade (1 a 5 estrelas) = (-1, -0.5, 0, 0.5, 1)
- (a) Escreva uma script que dado o corpus de avaliações, crie uma tabela de pares (palavra, polaridade). Propõem-se que para cada avaliação:
- Sejam retirado as entidade mencionadas – não queremos que associe polaridades às cidades ou a nomes específicos de hotéis;
 - a cada palavra dessa avaliação, lhe seja somada a polaridade da frase;
 - quando uma palavra é precedida por “não”, inverta a polaridade a somar;
 - no final as polaridades obtidas, deverão ser normalizadas de modo que fiquem entre -1 e 1.

A tabela deverá ser guardada para uso posterior. Que modo escolheria para guardar essa informação?

- (b) Escreva uma script que dada uma nova frase, e carregando a tabela anteriormente calculada, determine a sua polaridade

Questão 3. Pretende-se implementar uma função toXML() que recebe 2 argumentos, tag (String) e conteúdo, e devolve a representação em XML, de modo que:

```
toXML("a", "b")           → <a>b</a>
toXML("a", [1, 2, 3] )    → <a_s> <a>1</a> <a>2</a> <a>3</a> </a_s>
toXML("a", { b1: c, b2 : d} ) → <a> <b1>c</b1> <b2>d</b2> </a>
toXML("a", [1, {b1:c, b2:d} ] ) → <a_s> <a>1</a> <a> <b1>c</b1> <b2>d</b2> </a> </a_s>
```

Questão 4. Construa em Python um tradutor PT-EN baseado em regras do tipo:

```
caso a caso=each case
caso=case
(\d+) de Outubro=October \1
um=a
em=in
```

- Arranje uma representação em Python para este tipo de regras (apresente o exemplo nessa representação).
- Construa um programa que traduza as regras na sua representação Python
- Construa uma função **trad** que faça a tradução de uma frase usando as regras. Sugere-se que:
 - No início, junte uma marca “@” no início da frase a traduzir, indicativa da posição actual;
 - Em cada iteração veja se há regras aplicáveis na posição actual, em cujo caso deve fazer a respectiva substituição, avançando a marca;
 - Se nenhuma regra for aplicável, avance a palavra actual, prefixando-a com “= ”;
 - Quando a marca chegar ao fim da frase, remova a marca e devolva a frase EN

Exemplo : em 23 de Outubro um estranho caso ... → in October 23 a =estranho case ...

Se precisar, a função **re.subn: RE, str, str → (str, int)** faz substituição e retorna o número de substituições feitas.

Questão 5. No BB de cada disciplina da U.Minho há um ficheiro PDF com os dados e fotografia dos alunos. Seja esse hipotético ficheiro <https://elearning.uminho.pt/8401N3/Alunos.pdf>, sendo 8401N3 o código da disciplina. Pretendemos criar uma script que dado um código de disciplina:

- Vá buscar o PDF respetivo
- Usando o comando **pdftohtml -xml Alunos.pdf** crie um ficheiro Alunos.xml com o seguinte aspeto:

```
<pdf2xml producer="poppler" version="0.86.1">
<page number="1" position="absolute" top="0" height="1263" width="892">
<image top="252" width="90" height="90" src="DUC_Alunos_8401N5-1_1.png"/>
<image top="375" width="90" height="90" src="DUC_Alunos_8401N5-1_2.png"/>
<image top="498" width="90" height="90" src="DUC_Alunos_8401N5-1_3.png"/>
<image top="621" width="90" height="90" src="DUC_Alunos_8401N5-1_4.png"/>
<image top="1239" width="30" height="15" src="DUC_Alunos_8401N5-1_5.png"/>
<text top="62" width="273" height="18" ><b>Dossiê da Unidade Curricular</b></text>
<text top="119" width="227" height="18" >Terminologia e Tecnologias da Informação</text>
<text top="207" width="56" height="15" ><b>Alunos</b></text>
<text top="255" width="43" height="11" >Número 95837</text>
<text top="273" width="32" height="11" >Nome Adriana Filipa de Oliveira Gomes</text>
<text top="87" width="43" height="11" >Número 95635</text>
<text top="105" width="32" height="11" >Nome Ângela Filipa Ferreira Baltazar</text>
...
<text top="1245" width="21" height="8" >Pg. 1</text>
</page>
<page number="2" position="absolute" top="0" height="1263" width="892">
<image top="84" width="90" height="90" src="DUC_Alunos_8401N5-2_1.png"/>
...
<text top="87" width="43" height="11" >Número 95995</text>
<text top="105" width="32" height="11" >Nome Filipa ... </text>
...
</page>
...
</pdf2xml>
```

- Extraia a lista das fotografias (image com width=90) e a lista dos nomes dos alunos;
- Renomeie as imagens para o nome de aluno respetivo
(DUC_Alunos_8401N5-1_1.png → Adriana_Filipa_de_Oliveira_Gomes.png)