

# Processamento de Linguagens

## Ficha 1 sobre ER em Python

### Uso básico de ERs para pesquisa de um padrão

O objetivo deste primeiro exercício é duplo.

- Por um lado, pretende-se que explore o conceito de Expressão Regular (RE) e seus operadores (básicos ou estendidos / notação matemática e notação no computador) para descrever padrões de texto (sequências de caracteres);
- Por outro lado, pretende-se que recorra ao módulo `'re'` do Python para pesquisar a ocorrência de strings dentro de uma linha de texto respeitando diferentes opções de pesquisa (conforme se pede nas alíneas abaixo).

Para facilitar o trabalho que se pede nas alíneas seguintes, explore os programas anexos `'basic0.py'` a `'basic5a.py'`, considerando o seguinte pedido.

Dado um texto de entrada terminado por NewLine (uma linha como se costuma dizer), pesquise:

1. a palavra `'hello'` no início da linha (`basic0`)
2. a palavra `'hello'` em qualquer posição da linha (`basic1`)
3. todas as ocorrências da palavra `'hello'` dentro da linha, admitindo que a palavra seja escrita com maiúsculas ou minúsculas (`basic2`)
4. todas as ocorrências da palavra `'hello'` dentro da linha substituindo cada uma por `'*YEP*'` (`basic3`, `basic4`)
5. todas as ocorrências do caráter vírgula, separando cada parte da linha por esse caráter (`basic5`, `basic5a`)

### Pesquisa com ERs para Cálculo da Nota Média

Dada uma base textual com as notas dos alunos de uma turma, em que cada linha o Número, Nome e lista de Notas de cada aluno, sendo os vários campos separados por `','` Explore o programa anexo `'basic6.py'` para calcular a Nota Média de cada aluno.

### Uso mais generalizado de ERs

Para consolidar os conhecimentos anteriores e treinar outras ER, realize as várias pesquisas de um padrão de texto dentro de cada linha do ficheiro de entrada, recorrendo à função genérica `'busca(ER)'` proposta no ficheiro anexo `'search-geral.py'`.

Pesquise:

1. Linhas que contém algures a palavra `'padrão'`
2. Linhas que começam por `'PRH'` ou `'JCR'`
3. Linhas que começam por um Número
4. Linhas que terminam com `'PMoura'`

5. Linhas que terminam com: uma Letra (maiúscula ou minúscula), ou com um ponto de interrogação, ou ponto final
6. Linhas que só contém dígitos, hífen ou pontos
7. Linhas que começam por um ou mais Separadores de palavras (white spaces)
8. Linhas que contém um número de telefone português (9 dígitos); depois permita que comece pelo indicativo do país
9. Linhas que contém uma STRING não nula entre aspas; depois repita o exercício anterior permitindo Strings vazias; repita o exercício anterior escrevendo apenas o texto entre aspas.

## Conversão de datas

Escreva um filtro de texto que converta cada data encontrada no texto fonte no formato "dd/mm/aaaa" para o atual e desejável formato europeu "aaaa/mm/dd", conforme sugerido no ficheiro anexo 'datas-sub.py'.

## Conversão de Nomes

Escreva um filtro de texto que converta cada nome completo de uma pessoa encontrado no texto fonte no formato "Nproprio OutroN.... UltimoN" (cada nome começado obrigatoriamente por 1 maiúscula e separado do seguinte por 1 ou mais espaços) para o formato "UltimoN, LNproprio." em que aparece o ultimo apelido e depois a primeira letra do nome próprio seguida de ponto e separados por vírgula, conforme sugerido no ficheiro anexo 'nomes-sub.py'..

## Expansão de Abreviaturas

Escreva um filtro de texto que expanda as abreviaturas que encontrar no texto fonte no formato "/abrev". Crie inicialmente uma tabela de abreviaturas com que queira trabalhar. Veja o que está sugerido no ficheiro anexo 'abrev-sub.py'.

## Remoção de Repetidos

Escreva um filtro de texto que sempre que encontrar no texto fonte uma palavra repetida elimine as repetições, ou seja, substitua a lista de palavras por 1 só palavra. Veja o que está sugerido no ficheiro anexo 'repetidos.py'.