

USJT - 2018 - Linguagens e Paradigmas de Programação - Prática

Professor: Bossini

Aula: 02 - Python

Assunto: lists, dictionaries, tuples, sets, booleans, files

Introdução**Listas**

São sequências de elementos em Python, separadas por vírgulas, dentro de colchetes. As listas são indexadas e mutáveis. Podem crescer e diminuir de tamanho. E os elementos podem ser de diferentes tipos.

Exemplos no notebook `a02_Lists.ipynb`**Dicionários**

Dicionários são estruturas do tipo chave/valor semelhantes a hashtables. São construídos entre chaves, e a chave sempre é uma string. A separação entre a chave e o valor são dois pontos `:`. Os valores podem ser qualquer coisa.

Os dicionários não são indexados mas são mutáveis.

Exemplos no notebook `a02_2Dictionary.ipynb`**Tuplas**

São semelhantes às listas, mas são imutáveis. Vem escritas dentro de parênteses.

Exemplos no notebook `a02_3Tuples.ipynb`**Sets**

Os sets são conjuntos de elementos ordenados, não indexados e sem repetição. Se você tentar adicionar um elemento já existente a um set ele simplesmente não é adicionado. Os sets são construídos dentro de chaves, como os dicionários, mas não são dicionários.

Booleans

São valores **True** ou **False**.

Exemplos no notebook a02_4Sets e Booleans.ipynb

Arquivos

Para abrir arquivos, use `open(nome_do_arquivo, modo)`. O modo pode ser `w` para abrir somente para escrita, zerando o conteúdo do arquivo, ou `w+` para abrir para leitura e escrita, também apagando tudo; `a`, para abrir para escrita mas desta vez fazendo `append`, isto é, acrescentando conteúdo no final, mantendo o conteúdo anterior; `a+` abre para escrita e leitura; caso não seja passado nada em modo, o arquivo é aberto somente para leitura.

Para ler o arquivo use a função `read()`, que lê o arquivo todo. Para ler linha a linha, use `readlines()`. Após a leitura usando `read()` é necessário usar a função `seek(0)` para reposicionar no início do arquivo. Para escrever, use a função `write()`. Adicione um `\n` no final da string escrita para pular linha.

Exemplos no notebook a02_5Files.ipynb

Operadores Relacionais e Lógicos

Os operadores relacionais são idênticos ao do Java: `==`, `>`, `>=`, `<`, `<=`. Os operadores lógicos são o `and`, o `or` e o `not`.

Exemplos no notebook a02_5Files.ipynb

Exercício Prático 2

Forma de entrega:

- Crie um notebook chamado Exercicio02
- Para cada exercício: copie o enunciado em uma célula do tipo Markdown e resolva em uma célula do tipo Code.
- Crie um repositório no GitHub chamado LingProg
- Suba o notebook como arquivo avulso para este repositório.

Listas

1 Crie um programa que recebe uma lista de números e

- retorne o maior elemento
- retorne a soma dos elementos
- retorne o número de ocorrências do primeiro elemento da lista
- retorne a média dos elementos
- retorne o valor mais próximo da média dos elementos
- retorne a soma dos elementos com valor negativo
- retorne a quantidade de vizinhos iguais

2 Faça um programa que receba duas listas e retorne True se são iguais ou False caso contrário.

Duas listas são iguais se possuem os mesmos valores e na mesma ordem.

3 Faça um programa que receba duas listas e retorne True se têm os mesmos elementos ou False caso contrário

Duas listas possuem os mesmos elementos quando são compostas pelos mesmos valores, mas não obrigatoriamente na mesma ordem.

4 Faça um programa que percorre uma lista com o seguinte formato: [['Brasil', 'Italia', [10, 9]], ['Brasil', 'Espanha', [5, 7]], ['Italia', 'Espanha', [7,8]]]. Essa lista indica o número de faltas que cada time fez em cada jogo. Na lista acima, no jogo entre Brasil e Itália, o Brasil fez 10 faltas e a Itália fez 9.

O programa deve imprimir na tela:

- o total de faltas do campeonato
- o time que fez mais faltas
- o time que fez menos faltas

Dicionários

5 Escreva um programa que conta a quantidade de vogais em uma string e armazena tal quantidade em um dicionário, onde a chave é a vogal considerada.

6 Escreva um programa que lê duas notas de vários alunos e armazena tais notas em um dicionário, onde a chave é o nome do aluno. A entrada de dados deve terminar quando for lida uma string vazia como nome. Escreva uma função que retorna a média do aluno, dado seu nome.

7 Uma pista de Kart permite 10 voltas para cada um de 6 corredores. Escreva um programa que leia todos os tempos em segundos e os guarde em um dicionário, onde a chave é o nome do corredor. Ao final diga de quem foi a melhor volta da prova e em que volta; e ainda a classificação final em ordem (1º o campeão). O campeão é o que tem a menor média de tempos.

8 Escreva um programa para armazenar uma agenda de telefones em um dicionário. Cada pessoa pode ter um ou mais telefones e a chave do dicionário é o nome da pessoa. Seu programa deve ter as seguintes funções:

incluirNovoNome – essa função acrescenta um novo nome na agenda, com um ou mais telefones. Ela deve receber como argumentos o nome e os telefones.

incluirTelefone – essa função acrescenta um telefone em um nome existente na agenda. Caso o nome não exista na agenda, você deve

perguntar se a pessoa deseja inclui-lo. Caso a resposta seja afirmativa, use a função anterior para incluir o novo nome.
excluirTelefone - essa função exclui um telefone de uma pessoa que já está na agenda. Se a pessoa tiver apenas um telefone, ela deve ser excluída da agenda.
excluirNome - essa função exclui uma pessoa da agenda.
consultarTelefone - essa função retorna os telefones de uma pessoa na agenda.

Arquivos

9 Faça um programa que leia um arquivo texto contendo uma lista de endereços IP e gere um outro arquivo, contendo um relatório dos endereços IP válidos e inválidos.

O arquivo de entrada possui o seguinte formato:

```
200.135.80.9
192.168.1.1
8.35.67.74
257.32.4.5
85.345.1.2
1.2.3.4
9.8.234.5
192.168.0.256
```

O arquivo de saída possui o seguinte formato:

[Endereços válidos:]

```
200.135.80.9
192.168.1.1
8.35.67.74
1.2.3.4
```

[Endereços inválidos:]

```
257.32.4.5
85.345.1.2
9.8.234.5
192.168.0.256
```

10 A ACME Inc., uma empresa de 500 funcionários, está tendo problemas de espaço em disco no seu servidor de arquivos. Para tentar resolver este problema, o Administrador de Rede precisa saber qual o espaço ocupado pelos usuários, e identificar os usuários com maior espaço ocupado. Através de um programa, baixado da Internet, ele conseguiu gerar o seguinte arquivo, chamado "usuarios.txt":

alexandre	456123789
anderson	1245698456
antonio	123456456
carlos	91257581
cesar	987458
rosemary	789456125

Neste arquivo, o nome do usuário possui 15 caracteres. A partir deste arquivo, você deve criar um programa que gere um relatório, chamado "relatório.txt", no seguinte formato:

ACME Inc.	Uso do espaço em disco pelos usuários
-----------	---------------------------------------

Nr.	Usuário	Espaço utilizado	% do uso
1	alexandre	434,99 MB	16,85%
2	anderson	1187,99 MB	46,02%
3	antonio	117,73 MB	4,56%
4	carlos	87,03 MB	3,37%
5	cesar	0,94 MB	0,04%
6	rosemary	752,88 MB	29,16%

Espaço total ocupado: 2581,57 MB
Espaço médio ocupado: 430,26 MB

O arquivo de entrada deve ser lido uma única vez, e os dados armazenados em memória, caso sejam necessários, de forma a agilizar a execução do programa. A conversão da espaço ocupado em disco, de bytes para megabytes deverá ser feita através de uma função separada, que será chamada pelo programa principal. O cálculo do percentual de uso também deverá ser feito através de uma função, que será chamada pelo programa principal.