Aula 02

Definindo funções

Funções sao definidas com a palavra chave def;

Podem receber dois tipos de atributos, nomeados (**kwargs) ou posicionais (*args);

São High Order Functions, ou seja, aceitam outras funções como parâmetro, podem retornar outras funções e podem ser passadas como parâmetros;

Estruturas de dados no Python e suas diferenças

Estruturas de dados no Python e suas diferenças

- Lists
  - Filas (queue - deque)
  - Pilhas (stack)
- Tuples
- Sets
- Dictionaries

Mais sobre Listas

- Usando listas como "pilhas"
  - Você pode utilizar listas como pilhas utilizando o método pop()
- Usando listas como "filas"
  - Para utilizar listas como filas, utilize o tipo *deque* da biblioteca *collections*
    - *deque.append()*
    - *deque.popleft()*
- Desempacotamento de sequências
- Outros métodos
  - append()
  - extend(l)
  - insert(i, x)
  - remove(x)
  - pop([i])
  - index(i)
  - count(x)
  - reverse()
  - O método del

Tuplas

Tuplas, assim como strings são estruturas imutáveis e não aceitam atribuição de valores depois de instanciadas;

Servem como coleção de valores tão bem e de forma mais sucinta que os dicionários;

Aceitam estruturas mutáveis como valor;

Uma tupla também possui *unpacking (ela é uma sequência);*

Tuplas podem ser aninhadas;

Tuplas aceitam *packing;*

Sets

Sets são *coleções desordenadas* com elementos únicos';

Sets também suportam operações matemáticas como união, interseção, diferença e diferença simétrica;

São indicados para testes de diferenças e remoção de itens duplicados;

Dicionários (Dictionaries / Dicts)

Dicionários são conjuntos de pares *chave -> valor* sem ordem definida;

São muito úteis para agrupar valores;

A função principal de um dicionário é servir como etiquetas para armazenagem e recuperação de valores

Apenas estruturas imutáveis podem ser *chaves* dentro de um dicionário, tuplas podem ser chaves desde que não possuam nenhum valor mutável;

Compreensão de listas e expressões geradoras

(list comprehension / genexp)

List compreheension

Uma compreensão de lista é uma construção sintática disponível em algumas linguagens de programação para criação de uma lista baseada em listas existentes. Ela segue a forma da notação de definição de conjunto matemática (compreensão de conjunto) como forma distinta para uso de funções de mapa e filtro. (*wikipedia*)

A LC, substitui o uso das funções MAP, FILTER e REDUCE no Python que apesar de disponíveis, têm seu uso desencorajado pelo *core team* da linguagem por serem menos concisos e legíveis;

Expressões e funções geradoras

Um gerador é uma expressão/função que gera um iterável que pode ser consumido por um iterador

Funções geradora é qualquer classe que possua implementado o *magicmethod* __iter__ chamando a palavra reservada *yield* OU função que *yield algum valor*

Por definição, iteradores produz valor iterando sobre uma coleção;

Geradores são objetos que não instanciam valores diretamente na memória, consumindo um valor de cada vez até esgotar a *pilha*

list(), sum(), any(), all() todos consome expressões geradoras, ou seja, iteradores;

Manipulação de arquivos
(context manager "with")

Leitura e escrita de arquivos

A função open() é a responsável por abrir e manipular os arquivos no sistema;

Um arquivo deve ser sempre aberto para uso e fechado após seu uso;

A biblioteca padrão do já trás a possibilidade de manipular arquivos CSV;

Character	Meaning
'r'	abrir par leitura (padrão)
'w'	abrir para escrita, o arquivo é truncado primeiro
'x'	abrir para criação exclusiva, falhando se o arquivo existe
'a'	abrir para escrita, anexando o conteúdo para o fim do arquivo caso ele exista
'b'	modo binário (pode ser usado em conjunto com os de abertura)
't'	modo texto (padrão)
'+'	abrir um arquivo do disco para atualização (funciona para escrita e leitura)
'U'	modo quebra de linhas universal (depreciado)

Gerenciador de contexto (with)

O with() serve para facilitar a manipulação de arquivos;

Ele pode ser usado com outras coisas no Python, como Locks() de processos paralelos;

Você não precisa se preocupar em fechar o arquivo após o uso;

Estruturas de dados avançadas

from collections import defaultdict, namedtuple

defaultdict

Muito útil para contagem de valores;

Se uma chave não é encontrada, ele chama a função passada como argumento ao invés de levantar uma *KeyError Exception*

namedtuples

namedtuples são estruturas idênticas as tuplas comuns, porém seus atributos podem ser acessados através da notação de ponto;

Elas são perfeitas para manter a legibilidade do código, pois retornam parâmetros nomeados;

Quaisquer valores podem servir como *fieldnames*, desde que não sejam *keywords* ou comecem com *underscore;*

São excelentes para atribuir valores a resultados em leitura de arquivos ou bancos de dados;

Mini projeto - Lista de convidados Pythonica

Objetivo: Controlar todos os convidados do meu evento

- Funcionalidades:
  - Posso adicionar convidados;
  - Posso remover convidados;
  - Posso listar convidados;
  - Posso consultar se um convidado está na minha lista;
  - Posso dizer quantos convidados possue minha lista;
  - Minha lista fica salva em um arquivo local, então eu posso recuperá-la a qualquer momento