Sets

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

Sets

Os **sets** são uma coleção de itens desordenados, parcialmente imutáveis e que **não podem conter elementos duplicados**. Por ser parcialmente imutável, os sets possuem permissão de adição e remoção de elementos. Para declarar um set no Python é bem simples, seus elementos devem estar entre **chaves ({})** ou utilizando o **método set()** do próprio Python.

Sets

```
1    # declarando set de forma implicita
2    set1 = {1, 2, 3}
3    # declaração explicita
4    set2 = set([1,2,3])
5
6    print(set1)
7    print(type(set1))
```

Sets

As coleções em sets possuem algumas propriedades específicas para manipulação e outras propriedades presentes em listas, como o método de **len()**.

- add : insere um novo elemento à coleção
- update : adiciona mais de 1 elemento à coleção
- remove : remove um elemento existente (gerará um erro caso o elemento não exista)
- discard : remove um elemento, sem gerar erro que o elemento não existe

Operações com Sets

```python
set1 = { 24, 19, 43}

# inserindo um novo elemento
set1.add(9)
# inserindo + de 1 elemento
set1.update([12, 8])

# removendo um item,
# retornará um erro caso o item não existir
set1.remove(8)

# # outra opção ao remove
set1.discard(99)
```

Operações com Sets

```python
setA = {6, 9, 12, 24, 21}
setB = {9, 53, 12, 21, 30}

# União
# (união dos dois conjuntos)
print("União")
print(setA | setB)
print(setA.union(setB))

# Interseção
# (pertencem ao conjunto A e B)
print("Interseção")
print(setA & setB)
print(setA.intersection(setB))
```

Operações com Sets

```python
setA = {6, 9, 12, 24, 21}
setB = {9, 53, 12, 21, 30}

# Diferença
# (pertencem A e não pertencem a B)
print("Diferença")
print(setA - setB)
print(setA.difference(setB))

# Diferença Simétrica
# (itens que pertencem aos dois conjuntos e que não estão na interseção)
print("Diferença Simétrica")
print(setA ^ setB)
print(setA.symmetric_difference(setB))
```

Revisando

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

Conversão de Dados

- ***int()*** : conversão para inteiro
- ***float()*** : conversão para float
- ***str()*** : conversão para str
- ***list()*** : conversão para lista
- ***tuple()*** : conversão para tupla
- ***set()*** : conversão para sets

Operadores Aritméticos

Operador	Descrição	Exemplo
+	Adição	2 + 2 = 4
-	Subtração	5 - 2 = 3
*	Multiplicação	2 * 2 = 4
/	Divisão	4 / 2 = 2
%	Resto da divisão	10 % 3 = 1
**	Potência	4 ** 2 = 16

Operadores Lógicos

Operador	Descrição	Exemplo
Not	não	if not texto == "João":
And	e	(valor >= 5) and (item == 1)
Or	ou	(valor >= 25) or (item >= 5)

Estruturas de Controle

As estruturas de controle servem para definir quais blocos de construção serão executados, para o Python temos acessível a estrutura ***if/elif/else***. O conceito é similar aos de outras linguagens, como Javascript, mas com a adição do Elif que interpreta como sendo uma possível execução para a estrutura, ou seja, checa mais uma possível condição e é uma abreviatura para ***else if***.

if/elif/else (Estrutura de controle)

```python
numeroDigitado = int(input('Informe um número: '))

if numeroDigitado == 0:
    print('A entrada é 0')
elif numeroDigitado > 0 and numeroDigitado <= 5:
    print('A entrada está entre 0 e 5')
else:
    print('A entrada é maior que 5 ou abaixo de 0')
```

Estruturas de Repetição

As estruturas de repetição são utilizadas quando queremos que um bloco de código seja executado várias vezes. Em Python existem duas formas de criar um laço de repetição:

- O ***for*** (repetição contável) é usado quando se quer iterar sobre um bloco de código um número determinado de vezes. Ex: percorrer todos os itens de uma lista

- O ***while*** (repetição condicional) é usado quando queremos que o bloco seja repetido até que uma condição seja satisfeita, ou seja, é necessário que uma expressão seja declarada como ponto de parada do laço.

for (Repetição contável)

```
1   lista = [ 1, 2, 3, 4, 5 ]
2
3   for item in lista:
4       if item % 2 == 0:
5           print('{0} - Número par'.format(item))
6           continue
7       print('{0} - Número ímpar'.format(item))
```

while (Repetição condicional)

```
while True:
    palavra = input('Digite uma palavra: ')
    if palavra.lower() == 'sair':
        print('Fim')
        break
    if len(palavra) < 2:
        print('Palavra curta.')
        continue
    print('Tente digitar \"sair\"')
```

Break e Continue

Quando desejamos alterar o fluxo de execução de um laço de repetição, podemos utilizar as cláusulas break e continue.

- ***Break*** gera uma ação para interromper a execução, independente se a condição está satisfazendo ou não. Pode quebrar o laço que está ocorrendo com ***while*** ou ***for***.

- O ***continue*** é usado para prosseguir para próxima iteração, seu uso faz com o que o programa “pule” a execução posterior e volte a executar o laço a partir do bloco superior (incremento).

Classes

Classes proporcionam uma forma de organizar os dados e funcionalidades juntos, criar uma nova classe cria um novo “tipo” de objeto, permitindo que novas instâncias desse tipo sejam produzidas. Cada instância da classe pode ter atributos anexados a ela, para manter seu estado.

Classes

```
1   class Cachorro:
2       # variável da classe aplicada a todas as instâncias
3       tipo = "canino"
4
5       def __init__(self, nome):
6           # variável única para cada instância
7           self.nome = nome
8
9   c1 = Cachorro('Buzzy') # instância 1
10  c2 = Cachorro('Layka') # instância 2
11  print('{0} - {1}'.format(c1.tipo, c1.nome))
12  print('{0} - {1}'.format(c2.tipo, c2.nome))
```

Ambientes Virtuais

Quando estamos desenvolvendo em Python, é comum utilizarmos diferentes versões de uma mesma biblioteca entre diferentes projetos. Isso pode acarretar em diversos conflitos entre os pacotes dos projetos, para resolver isso criamos “ambientes virtuais” para que os projetos tenham suas dependências privadas.

Função Open File

DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

Open

A função open(), é uma função embutida no Python e ela auxilia na abertura de arquivos por meio do nosso algoritmo. Alguns dos parâmetros da função:

- **file:** caminho e nome do arquivo.
- **mode:** modo de abrir o arquivo, seu padrão é ‘r’ para apenas leitura. Mas existem outros modos disponíveis.
- **encoding:** o formato de codificação.

Open

```
1   # Abrindo um arquivo txt em Python
2
3   # passo como parametro o nome do arquivo e o modo
4   file=open('nomes.txt', 'r')
5
6   # leio todas as linhas presentes no arquivo
7   lista = file.readlines()
8   # "fecho" o arquivo
9   file.close()
10
11  print(lista)
```

Exercício 1 (30min)

Construir uma aplicação que defina sequências aleatórias para um jogo da MegaSena, crie também uma outra função para realizar o sorteio. Compare o resultado com a sequência aleatória e informe se o usuário ganhou ou não, e quantos números foram certos.

- Números a sortear: 1 a 60
- Sequência aleatória para o usuário: 6

Momento Prática

Utilizando os conceitos aprendidos ao longo da semana, desenvolvemos um Jogo de Forca em Python. Condições:

- Dar o resultado de derrota caso o usuário erre 10 letras.
- Ler um arquivo txt de nomes

Momento Prática 2

- Construir uma função para requisição de dados sobre um filme informado pelo usuário.

  - Criar um ambiente virtual com o nome “filmes_venv” (No cmd: virtualenv filmes_venv)
  - Gerar uma chave de api em https://www.themoviedb.org/settings/api
  - Ativar o ambiente por meio do comando **activate**.
  - Instalar o pacote:
    - **requests** (pip install requests)

Material Complementar

- Conjuntos em Python - https://pythonhelp.wordpress.com/2013/06/18/conjuntos-em-python/
- Função Open - https://docs.python.org/pt-br/3/library/functions.html#open
- Random - https://docs.python.org/3/library/random.html
- Como usar a biblioteca requests em Python - https://www.digitalocean.com/community/tutorials/how-to-get-started-with-the-requests-library-in-python-pt

DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!

SENAI

<LAB365>

INFORMAÇÕES DO TEMPLATE

- Título da Apresentação:
  - Fonte: Ubuntu Bold
  - Formato: Maiúsculo
  - Tamanho: 34
  - Cor: Branco
- Título do Slide:
  - Fonte: Ubuntu Bold
  - Formato: Maiúsculo
  - Tamanho: 22
  - Cor: Branco
- Parágrafos:
  - Fonte: Open Sans Normal
  - Tamanho: 14 a 18
  - Cores: Branco (Dark Mode) ou Preto (Light Mode)

- Marcadores de tópicos:
  - Formatos: Símbolos ou Alfanuméricos
  - Cor: Laranja
- Padrão de Cores:
  - Cinza - #868584
  - Preto - #1C1C19
  - Branco - #FAFAFA
  - Laranja - #F08305
  - Rosa - #c71d81
  - Azul - #0e1d8e

EXEMPLO DE TEXTO (LIGHT MODE)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO DE TEXTO (DARK MODE)

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO DE TÓPICOS (LIGHT MODE)

- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;

EXEMPLO DE TÓPICOS (DARK MODE)

- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;
- Lorem Ipsum is simply dummy text of the printing and typesetting industry;
- Lorem Ipsum has been the industry's standard dummy text ever since the 1500s;

EXEMPLO COM IMAGENS (LIGHT MODE)

Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO COM IMAGENS (DARK MODE)

Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

Título

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

EXEMPLO COM TABELAS (LIGHT MODE)

Lorem	Ipsum	Ipsum	Ipsum	Ipsum	Ipsum
	1	2	3	4	5
xxx	xxx	xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx	xxx	xxx

EXEMPLO COM TABELAS (DARK MODE)

Lorem	Ipsum	Ipsum	Ipsum	Ipsum	Ipsum
	1	2	3	4	5
xxx	xxx	xxx	xxx	xxx	xxx
xxx	xxx	xxx	xxx	xxx	xxx