

SSC0304 - Introdução à Programação para Engenharias

Fatiamento, Compreensão, Collections

Prof.: Leonardo Tórtoro Pereira

leonardop@usp.br

Baseado no material dos profs Fernando S. Osório e Claudio F.M. Toledo

Na aula passada...



List in Python

```
L = [ 20, 'Jessa', 35.75, [30, 60, 90] ]
```

↑
L[0]

↑
L[1]

↑
L[2]

↑
L[3]

- ✓ **Ordered:** Maintain the order of the data insertion.
- ✓ **Changeable:** List is mutable and we can modify items.
- ✓ **Heterogeneous:** List can contain data of different types
- ✓ **Contains duplicate:** Allows duplicates data

Tuples in Python

T = (20, 'Jessa', 35.75, [30, 60, 90])

↑
T[0]

↑
T[1]

↑
T[2]

↑
T[3]

- ✓ **Ordered:** Maintain the order of the data insertion.
- ✓ **Unchangeable:** Tuples are immutable and we can't modify items.
- ✓ **Heterogeneous:** Tuples can contains data of types
- ✓ **Contains duplicate:** Allows duplicates data

Set in Python

```
S = { 20, 'Jessa', 35.75 }
```

- ✓ **Unordered:** Set doesn't maintain the order of the data insertion.
- ✓ **Unchangeable:** Set are immutable and we can't modify items.
- ✓ **Heterogeneous:** Set can contains data of all types
- ✓ **Unique:** Set doesn't allows duplicates items

Dictionary in Python PYnative.com

Unordered collections of unique values stored in (Key-Value) pairs.

```
d = {'a': 10, 'b': 20, 'c': 30}
```

↑ ↑ ↑
d['a'] d['b'] d['c']

- ✓ **Unordered:** The items in dict are stored without any index value
- ✓ **Unique:** Keys in dictionaries should be Unique
- ✓ **Mutable:** We can add/Modify/Remove key-value after the creation

Para Refrescar

<https://www.hackerrank.com/challenges/sock-merchant>

O que vamos aprender hoje?



Objetivos

- Estudar fatiamento de listas e tuplas
- Praticar compreensão de listas, tuplas, conjuntos, e dicionários
- Conhecer o módulo Collections

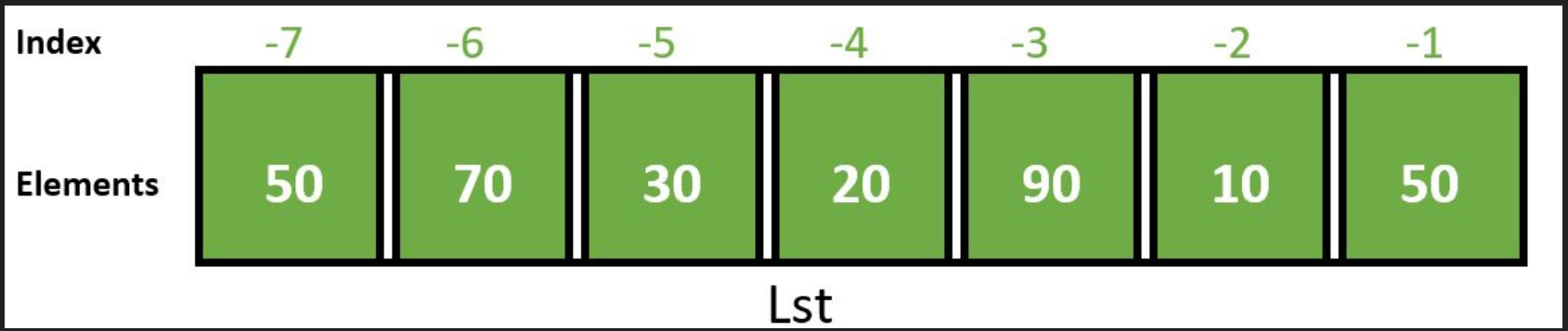
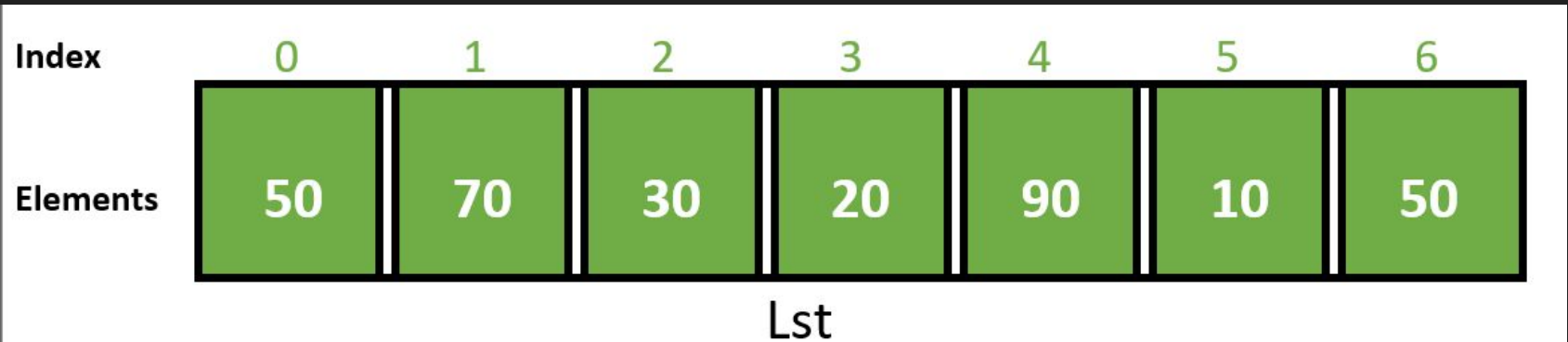
Tópicos da Aula

- Fatiamento
- Compreensão
- Collections

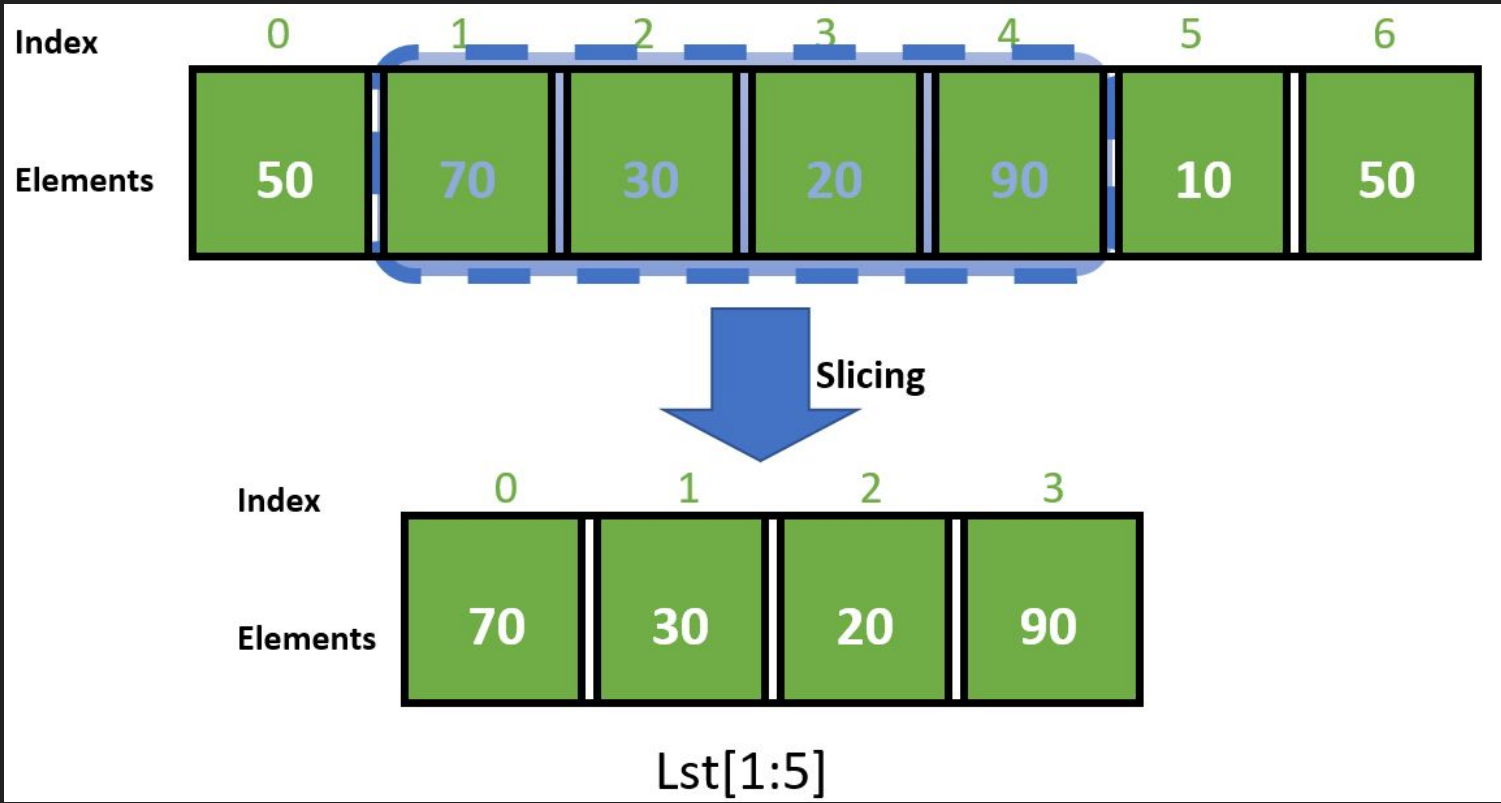
Fatiamento

Fatiamento

- Acesso prático a uma faixa de dados de listas e tuplas
- `listname[start_index : end_index : step]`
 - ◆ Serve também para tuplas
- Retorna uma nova lista
- Índices podem ser positivos ou negativos (começa do fim)



Fonte: <https://www.geeksforgeeks.org/python-list-slicing/>



Fonte: <https://www.geeksforgeeks.org/python-list-slicing/>

Fatiamento

- Pode ser usado para reverter lista
 - ◆ `_list[::-1]`
- Se expressão não faz sentido ou não pode ser computada
 - ◆ Retorna lista vazia
 - ◆ `List[-1:-1:-1]`
 - ◆ `List[:0:]`
 - ◆ ...

Fatiamento

- Pode modificar vários valores de lista de uma vez
 - ◆ `List[1:5] = [10, 30, 20, 40, 50]`
- Ou deletar elementos em grupo
 - ◆ `List[:3] = []`
- Também pode alterar dados via concatenação
 - ◆ `newList = List[:3]+List[7:]`

Exemplos

Compreensão

Compreensão

→ Maneira curta e concisa de construir novas sequências usando sequências existentes

- ◆ Listas
- ◆ Dicionários
- ◆ Conjuntos
- ◆ *Generator*

Sintaxe

→ Lista

```
output_list = [output_exp for var in input_list if (var satisfies this condition)]
```

→ Dicionário

```
output_dict = {key:value for (key, value) in iterable if (key, value satisfy condition)}
```

→ Conjunto

```
output_set = {output_exp for var in input_set if (var satisfies this condition)}
```

→ Generator

```
output_gen = (output_exp for var in input_list if (var satisfies this condition))
```

Compreensão

→ Generator

◆ Igual a lista, tirando o uso de parênteses ao invés de colchetes

→ Generator não aloca memória para a lista toda

◆ Cria cada valor 1 a 1

◆ Mais eficiente em memória

Exemplos

Coleção

Collections

- Módulo com tipos de dados especializados
 - ◆ Alternativas para listas, conjuntos, tuplas, e dicionários

→

<code>namedtuple()</code>	factory function for creating tuple subclasses with named fields
<code>deque</code>	list-like container with fast appends and pops on either end
<code>ChainMap</code>	dict-like class for creating a single view of multiple mappings
<code>Counter</code>	dict subclass for counting <code>hashable</code> objects
<code>OrderedDict</code>	dict subclass that remembers the order entries were added
<code>defaultdict</code>	dict subclass that calls a factory function to supply missing values
<code>UserDict</code>	wrapper around dictionary objects for easier dict subclassing
<code>UserList</code>	wrapper around list objects for easier list subclassing
<code>UserString</code>	wrapper around string objects for easier string subclassing

Fonte: <https://docs.python.org/3/library/collections.html>

Counter

- Subclasse de dicionário
- Conta elementos em um *iterable* na forma de dicionário não ordenado
- ◆ Chave é o elemento no *iterable* e valor é a contagem
- Equivalente a *bag* ou *multiset* em outras linguagens

OrderedDict

- Subclasse de dicionário
- Lembra da ordem de inserção
- Focado em reordenação
 - ◆ Eficiência de espaço, velocidade de iteração, performance de atualização é secundária

DefaultDict

- Subclasse de dicionário
- Provê valores padrões para chaves que não existem
 - ◆ Não levanta *KeyError*

ChainMap

- Encapsula vários dicionários em uma unidade
- ◆ Retorna lista de dicionários
- Costuma ser mais rápido que criar um novo dicionário e fazer várias chamadas a `update()`

NamedTuple

- Retorna uma tupla com nomes para cada posição
- Pode chamar itens pelo argumento à esquerda ao invés de índice

Deque

- Double Ended Queue (Fila com dois finais)
- ◆ Lista otimizada para append/pop mais rápidos
 - Ambas as extremidades
- $O(1)$ para append e pop
- ◆ Lista é $O(n)$

UserDict

- Container similar a dicionário
- ◆ Envolve objetos de dicionário
- Para quando você quer criar seu próprio dicionário com alguma funcionalidade diferente

UserList

- Container similar a lista
- ◆ Envolve objetos de listas
- Para quando você quer criar sua própria lista com alguma funcionalidade diferente

UserString

- Container similar a string
 - ◆ Envolve objetos de strings
- Para quando você quer criar sua própria string com alguma funcionalidade diferente

Exemplos

Referências

Referências

1. <https://www.learnpython.org/>
2. <https://www.w3schools.com/python/>
3. <https://panda.ime.usp.br/cc110/static/cc110/index.html>
4. https://www.youtube.com/playlist?list=PLcoJJSvnDgcKpOi_UeneTNTIV0igRQwcn
5. <https://pynative.com/python-lists/>
6. <https://pynative.com/python-tuples/>
7. <https://pynative.com/python-sets/>
8. <https://pynative.com/python-dictionaries/>
9. <https://www.geeksforgeeks.org/python-list-slicing/>
10. <https://www.geeksforgeeks.org/comprehensions-in-python/>
11. <https://docs.python.org/3/library/collections.html>
12. <https://www.geeksforgeeks.org/python-collections-module/>