

SSC0800 - Introdução à Ciência de Computação I

Strings, Tuplas, Dicionários, Conjuntos

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

Python List Methods

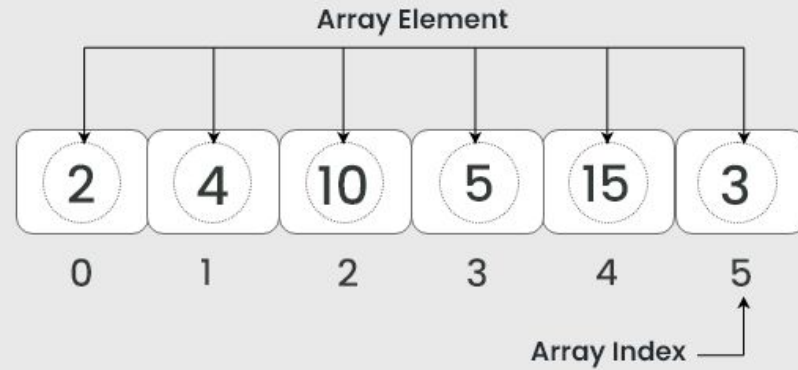
Input	Method	Output
	<code>.append(▲)</code>	
	<code>.insert(1, ▲)</code>	
	<code>.pop(1)</code>	
	<code>.remove(■)</code>	
	<code>.reverse()</code>	
	<code>.sort()</code>	
	<code>.index(▲)</code>	2
	<code>.count(■)</code>	2

Métodos de Lista. Fonte: <https://br.pinterest.com/pin/516506651025771855/>



Array

Data Structure



Fonte: <https://www.geeksforgeeks.org/array-data-structure/>

Memory Location

200	201	202	203	204	205	206	■	■	■
U	B	F	D	A	E	C	■	■	■
0	1	2	3	4	5	6	■	■	■

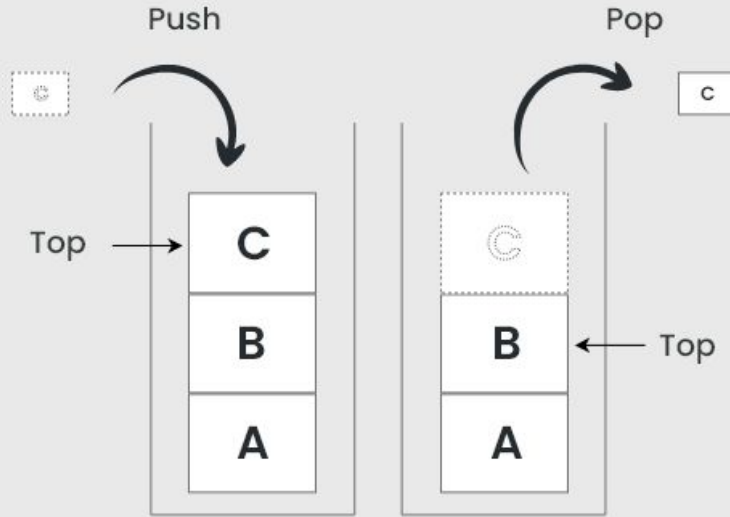
Index

Fonte: <https://www.geeksforgeeks.org/what-is-array/>

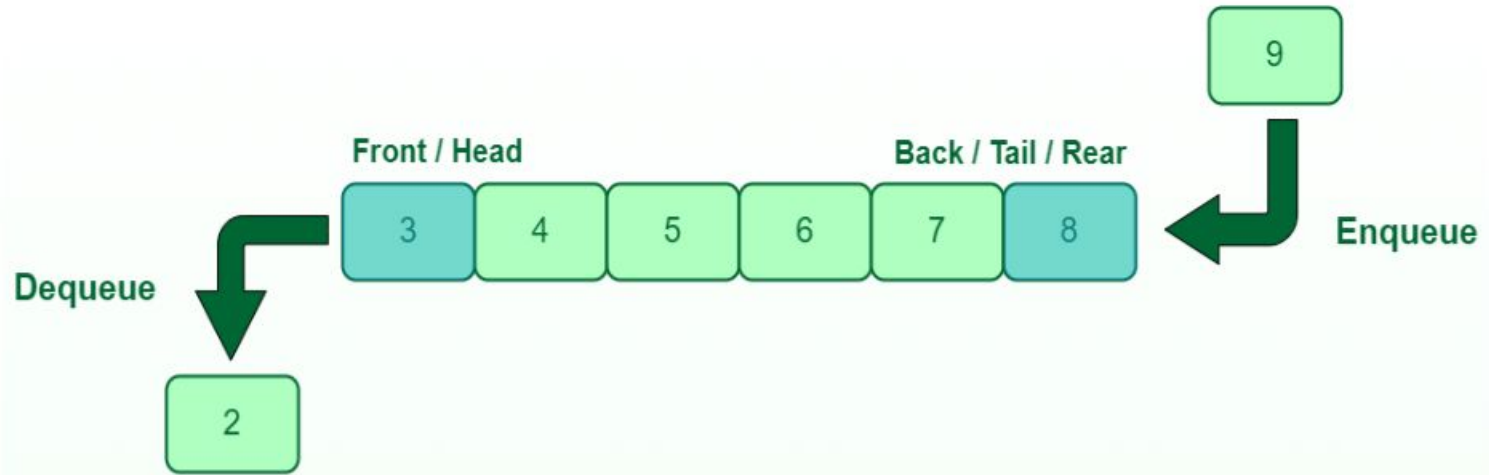


Stack

Data Structure



Pilha. Fonte: <https://www.geeksforgeeks.org/stack-data-structure/>



Queue Data Structure

Fila. Fonte: <https://www.geeksforgeeks.org/queue-data-structure/>

Para Refrescar

<https://www.hackerrank.com/challenges/nested-list>

O que vamos aprender hoje?



Objetivos

- Aprender um pouco mais sobre strings
- Conhecer os tipos de dados de coleção:
 - ◆ Tuplas
 - ◆ Dicionários
 - ◆ Conjuntos

Tópicos da Aula

- Strings
- Tuplas
- Dicionários
- Conjuntos

Strings

Strings

- Vetores de bytes que representam caracteres Unicode
- Coleção de 1 ou mais caracteres
 - ◆ Dentro de uma, duas ou três aspas
- Python não tem char
 - ◆ É uma string de tamanho 1
- Vem da classe "str"

Strings

```
string_single = 'Hello single world'
```

```
string_double = "Hello double world"
```

```
string_triple = '''Hello triple world'''
```

```
string_multi_line = '''Hello  
Multi-line  
World'''
```

Strings

- Para acessar os caracteres de uma string, usa-se “[]”
 - ◆ O valor do índice é passado
 - ◆ Aceita valores negativos
 - Começa do final

Strings

```
string_single = 'Hello single world'  
string_single[0]  
string_single[-1]
```

Strings

- Strings implementam todas as operações comuns de sequência (e.g., vetores)
- E outras específicas!
- Também permitem formatação de duas maneiras
 - ◆ [str.format\(\)](#)
 - ◆ [Formatação do printf \(do C\)](#)
- Strings são arrays com mais funções

Algumas funções de strings

Função	O que faz?
<code>str.capitalize()</code>	retorna cópia da <i>str</i> com primeira letra maiúscula e resto minúscula
<code>str.casefold()</code>	retorna cópia com tudo em minúscula, inclusive chars como 'ß'
<code>str.count(substr[,start[,end]])</code>	retorna número de ocorrências não sobrepostas da <i>substr</i> no <i>range</i>
<code>str.find(substr[,start[,end]])</code>	retorna menor índice da <i>str</i> em que a <i>substr</i> foi encontrada no <i>range</i>
<code>str.isalnum()</code>	True se todos os caracteres são alfanuméricos (e tem pelo menos 1)
<code>str.isalpha()</code>	True se todos os caracteres são alfabéticos (e tem pelo menos 1)
<code>str.isdecimal()</code>	True se todos os caracteres são decimais (e tem pelo menos 1)
<code>str.isdigit()</code>	True se todos os caracteres são dígitos (e tem pelo menos 1)

Exemplos

Tuplas

Tuplas

- Valores, ordenados, separados por vírgula
- Sempre envoltas por parênteses na saída
- Tuplas pode conter listas e outros objetos
- São imutáveis

Tuplas

- Geralmente tem sequência heterogênea de elementos
 - ◆ Acessado ao desempacotar ou indexar
- Listas são mutáveis e geralmente acessadas iterativamente
- Tuplas podem ser concatenadas

Similarities	Differences
Functions that can be used for both lists and tuples: <code>len()</code> , <code>max()</code> , <code>min()</code> , <code>sum()</code> , <code>any()</code> , <code>all()</code> , <code>sorted()</code>	Methods that cannot be used for tuples: <code>append()</code> , <code>insert()</code> , <code>remove()</code> , <code>pop()</code> , <code>clear()</code> , <code>sort()</code> , <code>reverse()</code>
Methods that can be used for both lists and tuples: <code>count()</code> , <code>Index()</code>	we generally use 'tuples' for heterogeneous (different) data types and 'lists' for homogeneous (similar) data types.
Tuples can be stored in lists.	Iterating through a 'tuple' is faster than in a 'list'.
Lists can be stored in tuples.	'Lists' are mutable whereas 'tuples' are immutable.
Both 'tuples' and 'lists' can be nested.	Tuples that contain immutable elements can be used as a key for a dictionary.

Tuplas vs Listas. Fonte: <https://www.geeksforgeeks.org/python-tuples/>

Exemplos

Dicionários

Dicionários

→ Indexados por chaves

◆ Qualquer tipo imutável

- Strings, números, etc.
- Tuplas que contenham tipos imutáveis

Dicionários

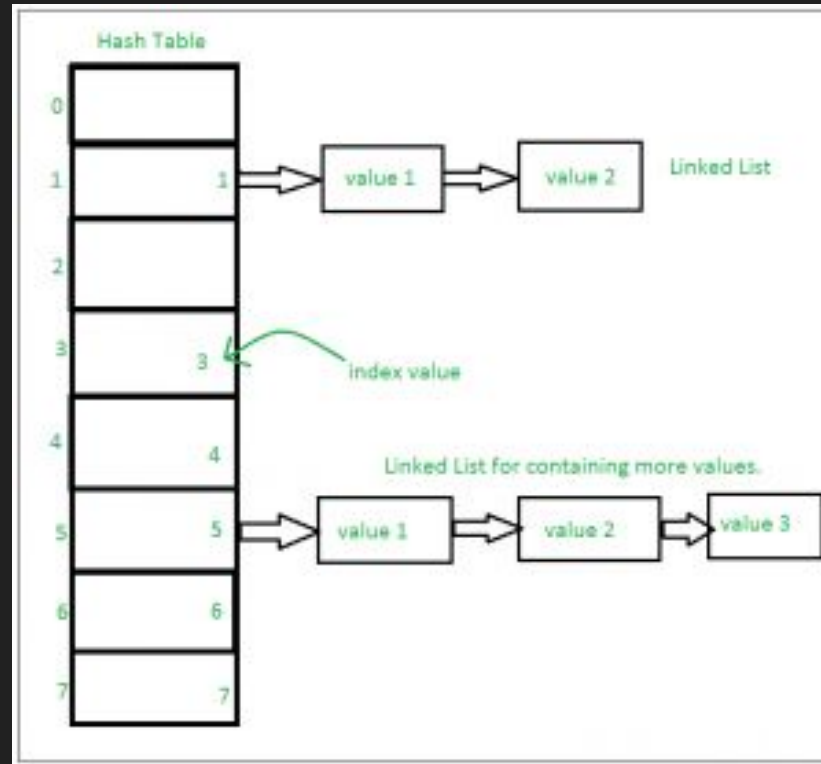
- Conjunto de pares chave-valor
 - ◆ Chaves são únicas
- Não permite duplicatas
- Pode ser alterado
- Ordenado (a partir do Python 3.7)

Exemplos

Conjuntos

Conjuntos

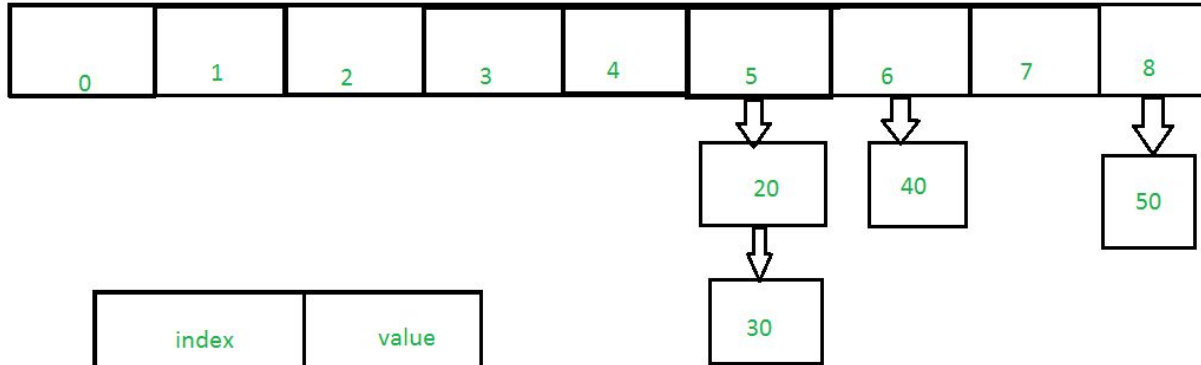
- Coleção não ordenada sem elementos duplicados
- Ex:
 - ◆ Teste de pertencimento
 - ◆ Eliminar duplicatas
- Permite união, interseção, diferença, diferença simétrica



Como funciona um conjunto. Fonte:

<https://www.geeksforgeeks.org/sets-in-python/>

Index-value pair in HashTable



index	value
5	20
5	30
6	40
8	50

similar index values

For Performing :-
Traversal
Insertion
Deletion

This table is implementable for all set operations.

Como funciona um conjunto. Fonte:

<https://www.geeksforgeeks.org/sets-in-python/>

Operation	Average case	Worst Case	notes
$x \text{ in } s$	$O(1)$	$O(n)$	
Union $s t$	$O(\text{len}(s)+\text{len}(t))$		
Intersection $s\&t$	$O(\min(\text{len}(s), \text{len}(t)))$	$O(\text{len}(s) * \text{len}(t))$	replace "min" with "max" if t is not a set
Multiple intersection $s1\&s2\&\dots\&sn$		$(n-1)*O(l)$ where l is $\max(\text{len}(s1), \dots, \text{len}(sn))$	
Difference $s-t$	$O(\text{len}(s))$		

<https://www.geeksforgeeks.org/sets-in-python/>
[https://www.geeksforg
eeks.org/sets-in-python/](https://www.geeksforg
eeks.org/sets-in-python/)

Operators	Notes
key in s	containment check
key not in s	non-containment check
s1 == s2	s1 is equivalent to s2
s1 != s2	s1 is not equivalent to s2
s1 <= s2	s1 is subset of s2
s1 < s2	s1 is proper subset of s2

s1 >= s2	s1 is superset of s2
s1 > s2	s1 is proper superset of s2
s1 s2	the union of s1 and s2
s1 & s2	the intersection of s1 and s2
s1 - s2	the set of elements in s1 but not s2
s1 ^ s2	the set of elements in precisely one of s1 or s2

Complexidade de conjuntos. Fonte:

<https://www.geeksforgeeks.org/sets-in-python/>
[https://www.geeksforg
eeks.org/sets-in-python/](https://www.geeksforg
eeks.org/sets-in-python/)

Exemplos

Iterando sobre coleções

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://www.geeksforgeeks.org/python-data-types/#string>
6. <https://docs.python.org/3/library/stdtypes.html#string-methods>
7. <https://docs.python.org/3/tutorial/datastructures.html>
8. https://www.w3schools.com/python/python_tuples.asp
9. <https://www.geeksforgeeks.org/python-tuples/>
10. <https://www.geeksforgeeks.org/python-dictionary/>
11. https://www.w3schools.com/python/python_dictionaries.asp
12. https://www.w3schools.com/python/python_sets.asp
13. <https://www.geeksforgeeks.org/sets-in-python>