

<https://python.ic.ufrj.br/aulas/aula-06/index.html?authuser=0>

Função	Para que serve	Exemplo
upper	Retorna a string passada como parâmetro em letras maiúsculas.	<code>str.upper(&lt;&lt;nome da string&gt;&gt;)</code>
lower	Retorna a string passada como parâmetro em letras minúsculas.	<code>str.lower(&lt;&lt;nome da string&gt;&gt;)</code>
count	Conta o número de ocorrências de uma substring dentro de uma string. Lembrando que os parâmetros de <code>início</code> e <code>fim</code> são opcionais.	<code>str.count(&lt;&lt;nome da string&gt;&gt;, &lt;&lt;substring&gt;&gt;, &lt;&lt;início&gt;&gt;, &lt;&lt;fim&gt;&gt;)</code>
index	Retorna o índice da primeira ocorrência de uma substring dentro de uma string. Lembrando que os parâmetros de <code>início</code> e <code>fim</code> são opcionais.	<code>str.index(&lt;&lt;nome da string&gt;&gt;, &lt;&lt;substring&gt;&gt;, &lt;&lt;início&gt;&gt;, &lt;&lt;fim&gt;&gt;)</code>
find	Retorna o índice da primeira ocorrência de uma substring em uma string. Lembrando que os parâmetros de <code>início</code> e <code>fim</code> são opcionais e que não é levantado um erro quando a substring não é encontrada, como acontece com <code>str.index()</code> .	<code>str.find(&lt;&lt;nome da string&gt;&gt;, &lt;&lt;substring&gt;&gt;, &lt;&lt;início&gt;&gt;, &lt;&lt;fim&gt;&gt;)</code>
replace	Substitui as $n$ primeiras ocorrências de uma substring por uma nova substring dentro de uma string. Lembrando que o número de ocorrências é opcional e, se não for definido, são trocadas todas as ocorrências.	<code>str.replace(&lt;&lt;nome da string&gt;&gt;, &lt;&lt;substring a ser substituída&gt;&gt;, &lt;&lt;substring de substituição&gt;&gt;, &lt;&lt;número de ocorrências&gt;&gt;)</code>

strip	Retorna uma substring de uma string passada como parâmetro sem os caracteres pertencentes à substring passada como segundo parâmetro. Se a substring <code>caractere</code> não for especificada, retira caracteres em branco.	<code>str.strip(&lt;&lt;nome da string&gt;&gt;, &lt;&lt;caractere&gt;&gt;)</code>
partition	Divide <code>string</code> em 3 partes: o que vem antes do <code>separador</code> , <code>separador</code> e o que vem depois do <code>separador</code> . Se <code>separador</code> não pertence a <code>string</code> , <code>string</code> é retornada seguida por duas strings vazias.	<code>str.partition(&lt;&lt;string&gt;&gt;, &lt;&lt;separador&gt;&gt;)</code>
split	Retorna uma lista com as substrings de <code>string</code> presentes entre cópias de uma outra string <code>separador</code> . Se <code>separador</code> não for especificado, assume-se sequências de caracteres em branco, tabs ou newlines.	<code>str.split(&lt;&lt;string&gt;&gt;, &lt;&lt;separador&gt;&gt;)</code>
join	Concatena uma sequência de strings (fornecidas em uma tupla ou lista ou string), intercalados com cópias do separador.	<code>str.join(&lt;&lt;separador&gt;&gt;, &lt;&lt;sequencia&gt;&gt;)</code>

Função	Para que serve	Exemplo
append	Insere um elemento no final da lista (modifica a lista de entrada, sem valor de retorno)	<code>list.append(&lt;&lt;nome da lista&gt;&gt;, &lt;&lt;novo elemento&gt;&gt;)</code>
extend	Insere um elemento no final da lista (modifica a lista de entrada, sem valor de retorno)	<code>list.extend(&lt;&lt;nome da lista&gt;&gt;, &lt;&lt;novo elemento&gt;&gt;)</code>
insert	Insere um elemento na lista na posição informada (modifica a lista de entrada, sem valor de retorno)	<code>list.insert(&lt;&lt;nome da lista&gt;&gt;, &lt;&lt;índice&gt;&gt;, &lt;&lt;novo elemento&gt;&gt;)</code>

Alguns exemplos:

```
>>> lista = [1, 2, 3, 4, 5]
>>> list.append(lista, 6)
>>> lista
[1, 2, 3, 4, 5, 6]
>>> list.extend(lista, [7, 8, 9, 10])
>>> lista
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>> list.insert(lista, 0, 0)
>>> lista
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Função	Para que serve	Exemplo	Observações
remove	Remove a primeira ocorrência do elemento na lista (modifica a lista de entrada, sem valor de retorno)	<code>list.remove(&lt;&lt;nome da lista&gt;&gt;, &lt;&lt;elemento&gt;&gt;)</code>	O elemento a ser removido deve estar na lista
del	Remove o elemento na posição informada (modifica a lista de entrada, sem valor de retorno)	<code>del &lt;&lt;nome da lista&gt;&gt;[&lt;&lt;índice&gt;&gt;]</code>	O índice deve estar dentro da faixa de índices da lista
pop	Remove e retorna o elemento na posição informada (modifica a lista de entrada)	<code>list.pop(&lt;&lt;nome da lista&gt;&gt;, &lt;&lt;índice&gt;&gt;)</code>	O índice deve estar dentro da faixa de índices da lista
clear	Remove todos os elementos de uma lista (sem valor de retorno)	<code>list.clear(&lt;&lt;nome da lista&gt;&gt;)</code>	

Alguns exemplos:

```
>>> lista = [1, 3, 9, 27]
>>> lista
[1, 3, 9, 27]
>>> list.remove(lista, 27)
>>> lista
[1, 3, 9]
>>> del lista[0]
>>> lista
[3, 9]
>>> list.pop(lista, -1)
9
>>> lista
[3]
>>> list.clear(lista)
>>> lista
[]
```

Função	Para que serve	Exemplo	Observações
count	Retorna quantas vezes um elemento informado aparece na lista	<code>list.count(&lt;&lt;nome da lista&gt;&gt;, &lt;&lt;elemento&gt;&gt;)</code>	
index	Retorna a posição da primeira incidência do elemento na lista	<code>list.index(&lt;&lt;nome da lista&gt;&gt;, &lt;&lt;elemento&gt;&gt;)</code>	O elemento deve estar na lista
in	Retorna True se o elemento está na lista ou False caso contrário	<code>&lt;&lt;elemento&gt;&gt; in &lt;&lt;nome da lista&gt;&gt;</code>	
reverse	Inverte a ordem dos elementos da lista de entrada (modifica a lista de entrada, sem valor de retorno)	<code>list.reverse(&lt;&lt;nome da lista&gt;&gt;)</code>	
sort	Ordena os elementos da lista de entrada (modifica a lista de entrada, sem valor de retorno)	<code>list.sort(&lt;&lt;nome da lista&gt;&gt;)</code>	Se os tipos de elementos da lista forem incompatíveis pode ocorrer erro

Mais alguns exemplos:

```
>>> lista = [0, 1, 2, 0, 1, 2, 3, 4, 5]
>>> list.count(lista, 0)
2
>>> list.count(lista, 9)
0
>>> list.index(lista, 0)
0
>>> list.index(lista, 5)
8
>>> 9 in lista
False
>>> 5 in lista
True
>>> list.reverse(lista)
>>> lista
[5, 4, 3, 2, 1, 0, 2, 1, 0]
>>> list.sort(lista)
>>> lista
[0, 0, 1, 1, 2, 2, 3, 4, 5]
```

Função	Para que serve	Exemplo	Observações
len	Retorna o número de mapeamentos do dicionário.	<code>len(&lt;&lt;nome do dicionário&gt;&gt;)</code>	
keys	Retorna o conjunto de chaves do dicionário.	<code>dict.keys(&lt;&lt;nome do dicionário&gt;&gt;)</code>	
values	Retorna o conjunto de valores do dicionário.	<code>dict.values(&lt;&lt;nome do dicionário&gt;&gt;)</code>	
items	Retorna todos os itens (chave, valor) do dicionário.	<code>dict.items(&lt;&lt;nome do dicionário&gt;&gt;)</code>	
get	Retorna o valor associado à chave passada como parâmetro. Nesta função, podemos informar um valor de retorno e, caso a chave não pertença ao dicionário, é este valor de retorno que será retornado.	<code>dict.get(&lt;&lt;nome do dicionário&gt;&gt;, &lt;&lt;chave&gt;&gt;, &lt;&lt;valor de retorno&gt;&gt;)</code>	
del	Remove um item do dicionário.	<code>del &lt;&lt;nome do dicionário&gt;&gt; [&lt;&lt;chave&gt;&gt;]</code>	A chave deve pertencer ao dicionário.

clear	Remove todos os itens do dicionário.	<code>dict.clear(&lt;&lt;nome do dicionário&gt;&gt;)</code>	
copy	Copia o dicionário para outra variável.	<code>dict.copy(&lt;&lt;nome do dicionário&gt;&gt;)</code>	Os elementos mutáveis dentro do dicionário (listas ou dicionários) do novo dicionário serão apenas referências aos elementos do dicionário original.

Exemplos:

```
>>> produtos = {'farinha': 3.00,
                'feijão': 5.00,
                'leite': 4.25,
                'açúcar': 2.49}
>>> len(produtos)
4

>>> list(dict.keys(produtos))
['farinha', 'feijão', 'leite', 'açúcar']

>>> list(dict.values(produtos))
[3.0, 5.0, 4.25, 2.49]

>>> list(dict.items(produtos))
[('farinha', 3.0), ('feijão', 5.0), ('leite', 4.25), ('açúcar', 2.49)]

>>> dict.get(produtos, 'feijão')
5.0

>>> dict.get(produtos, 'arroz')
None

>>> dict.get(produtos, 'arroz', 'Produto não encontrado')
'Produto não encontrado'

>>> del produtos['leite']
>>> produtos
{'farinha': 3.0, 'feijão': 5.0, 'açúcar': 2.49}
>>> dict.clear(produtos)
>>> produtos
{}
```