

Lists 2 (Important Methods)

یکم دنده عقب بریم ؟

داخل کارگاه اول به تابع len() اشاره داشتیم ! خب بزارید یک یادآوری کوچیک بهتون بگم :

تابع len() معمولاً برای تعیین **طول** یا **تعداد عناصر** یک شیء قابل پیمایش (iterable) استفاده می‌شود. این تابع یک داده که برای آن طول یا تعداد تعریف شده باشد دریافت می‌کند و طول تعداد عناصر آن را برمی‌گرداند. بنابراین این تابع می‌تواند برای شمردن تعداد کاراکترهای یک رشته و یا تعداد اعضا یک لیست به کار رود .

```
1 | numbers = [1, 2, 3, 4, 5]
2 | print(len(numbers))
3 |
4 | s = "Homayoun"
5 | print(len(s))
```

جست و جو کردن در لیست ؟

پایتون برای جست‌وجوی عضوی در لیست، عملگری به نام in در اختیار ما قرار می‌دهد. قبلاً نام این عملگر را در حلقه for دیده‌اید، اما اکنون استفاده‌ی متفاوتی از آن را خواهید آموخت.

این عملگر می‌تواند وجود یک عضو در یک لیست را بررسی کند و در صورت وجود آن مقدار True و در غیر این صورت مقدار False را برمی‌گرداند. به مثال زیر توجه نمایید:

```
1 | my_list = [1, 2, 4, 6]
2 | print(2 in my_list)
3 | print(3 in my_list)
```

همان‌طور که انتظار می‌رود خروجی اول True است چون عدد ۲ در لیست وجود دارد و خروجی دوم False است چون عدد ۳ در لیست وجود ندارد.

از این عملگر در شرطها نیز می‌توانیم استفاده کنیم:

```
1 x = int(input())
2 my_list = [1, 2, 4, 6]
3
4 if x in my_list:
5     print("Yes")
```

خب این ها مقدمه هایی از توابع ساده لیست در پایتون بود ، حالا بریم چنتا تابع بهتر ببینیم :

متدهای لیست در پایتون

لیست ها یکی از پرکاربردترین ساختارهای داده ای در پایتون هستند و متدهای داخلی مختلفی برای **اضافه کردن**، **حذف کردن**، **تغییر دادن و مدیریت عناصر لیست** وجود دارد. در این درسنامه، تمام متدهای لیست در پایتون را با مثال های ساده بررسی می کنیم.

♦ ۱. متد `append()` – اضافه کردن یک عنصر به انتهای لیست

این متد یک عنصر جدید را به انتهای لیست اضافه می کند.

مثال:

```
1 numbers = [1, 2, 3]
2 numbers.append(4)
3 print(numbers)
```

۲. متد `copy()` – ایجاد یک کپی سطحی از لیست

این متد یک **کپی سطحی (shallow copy)** از لیست برمی گرداند که مستقل از لیست اصلی است.

مثال:

```
1 original_list = [10, 20, 30]
2 copied_list = original_list.copy()
3 print(copied_list)
```

۳. متد clear() – حذف تمام عناصر لیست

این متد تمام عناصر لیست را پاک کرده و آن را به یک لیست خالی تبدیل می‌کند.

مثال:

```
1 | fruits = ["apple", "banana", "cherry"]
2 | fruits.clear()
3 | print(fruits)
```

۴. متد count() – شمارش تعداد تکرار یک مقدار مشخص

این متد تعداد دفعاتی که یک مقدار مشخص در لیست ظاهر شده را برمی‌گرداند.

مثال:

```
1 | numbers = [1, 2, 2, 3, 4, 2]
2 | print(numbers.count(2))
```

۵. متد extend() – افزودن چندین عنصر به انتهای لیست

این متد عناصر یک لیست دیگر را به انتهای لیست فعلی اضافه می‌کند.

مثال:

```
1 | list1 = [1, 2, 3]
2 | list2 = [4, 5, 6]
3 | list1.extend(list2)
4 | print(list1)
```

۶. متد index() – یافتن اندیس اولین رخداد یک مقدار

این متد اندیس (مکان) اولین رخداد یک مقدار خاص را در لیست برمی‌گرداند.

مثال:

```
1 | colors = ["red", "blue", "green", "blue"]
2 | print(colors.index("blue"))
```

۷. متد insert() – درج یک مقدار در موقعیت مشخص

این متد مقدار موردنظر را در یک موقعیت مشخص در لیست قرار می‌دهد.

مثال:

```
1 | letters = ["a", "b", "d"]
2 | letters.insert(2, "c")
3 | print(letters)
```

۸. متد pop() – حذف و بازگرداندن یک عنصر از لیست

این متد مقدار مشخص شده را حذف کرده و برمی‌گرداند. اگر هیچ مقداری تعیین نشود، مقدار آخر لیست حذف می‌شود.

مثال:

```
1 | numbers = [1, 2, 3, 4]
2 | removed_item = numbers.pop(2)
3 | print(numbers)
4 | print(removed_item)
```

۹. متد remove() – حذف اولین رخداد یک مقدار

این متد اولین رخداد مقدار مشخص شده را از لیست حذف می‌کند.

مثال:

```
1 | names = ["Alice", "Bob", "Charlie", "Bob"]
2 | names.remove("Bob")
3 | print(names)
```

۱۰. متد reverse() – معکوس کردن ترتیب عناصر لیست

این متد ترتیب عناصر لیست را معکوس می‌کند.

مثال:

```
1 | nums = [1, 2, 3, 4]
2 | nums.reverse()
3 | print(nums)
```

۱۱. متد sort() – مرتب‌سازی عناصر لیست

این متد لیست را به‌صورت پیش‌فرض صعودی (کوچک به بزرگ) مرتب می‌کند. می‌توان با مقدار `reverse=True` آن را نزولی کرد.

مثال:

```
1 | numbers = [5, 2, 9, 1, 5]
2 | numbers.sort()
3 | print(numbers)
4 |
5 | numbers.sort(reverse=True)
6 | print(numbers)
```

بریم برای یک جمع بندی خوب :

در این داکيومنت با مهم‌ترین متدهای داخلی لیست در پایتون آشنا شدیم که به ما کمک می‌کنند تا لیست‌ها را مدیریت، تغییر و بهینه‌سازی کنیم. هر یک از این متدها کاربرد خاصی دارند و بسته به نیاز، می‌توانیم از آن‌ها

استفاده کنیم.

Python List Methods	
Method	Description
append()	Adds an element at the end of the list
clear()	Removes all the elements from the list
copy()	Returns a copy of the list
count()	Returns the number of elements with the specified value
extend()	Add the elements of a list (or any iterable), to the end of the current list
index()	Returns the index of the first element with the specified value
insert()	Adds an element at the specified position
pop()	Removes the element at the specified position
remove()	Removes the item with the specified value
reverse()	Reverses the order of the list
sort()	Sorts the list