

لیست‌ها (Lists) در پایتون

لیست‌ها یکی از پرکاربردترین ساختارهای داده‌ای (Data Structures) در پایتون هستند که برای ذخیره مجموعه‌ای از داده‌ها استفاده می‌شوند. آن‌ها قابل تغییر (Mutable) و ترتیب‌دار (Ordered) هستند، به این معنی که می‌توان مقدار عناصر آن‌ها را تغییر داد و ترتیب عناصر حفظ می‌شود.

۱. ایجاد و دسترسی به لیست‌ها

ایجاد لیست

لیست‌ها با استفاده از `[]` (براکت مربع) تعریف می‌شوند و می‌توانند شامل انواع داده‌ای مختلف باشند.

لیست عددی

```
numbers = [1, 2, 3, 4, 5]
```

لیست ترکیبی از انواع داده‌ای مختلف

```
mixed_list = [1, "hello", 3.5, True]
```

لیست خالی

```
empty_list = []
```

دسترسی به عناصر لیست

برای دسترسی به یک عنصر خاص از لیست از ایندکس (Index) استفاده می‌کنیم. ایندکس‌ها از ۰ (صفر) شروع می‌شوند.

```
fruits = ["سیب", "موز", "گیلاس"]
```

```
print(fruits[0]) # خروجی: سیب
```

```
print(fruits[1]) # خروجی: موز
```

```
print(fruits[-1]) # خروجی: گیلاس (ایندکس منفی: شمارش از انتها)
```

برش (Slicing) لیست

برای دریافت بخش خاصی از لیست می‌توان از برش (`[start:end:step]`) استفاده کرد.

```
numbers = [10, 20, 30, 40, 50]
```

```
print(numbers[1:4]) # خروجی: [20, 30, 40] (اندیس 1 تا 3)
```

```
print(numbers[:3]) # خروجی: [10, 20, 30] (از ابتدا تا اندیس 2)
```

```
print(numbers[2:]) # خروجی: [30, 40, 50] (از اندیس 2 تا انتها)
```

```
print(numbers[::-1]) # خروجی: [50, 40, 30, 20, 10] (معکوس لیست)
```

۲. عملیات مختلف روی لیست‌ها

افزودن عناصر به لیست

→ اضافه کردن یک مقدار به انتهای لیست `append()` ✓

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

→ اضافه کردن چند مقدار به انتهای لیست `extend()` ✓

```
numbers = [1, 2, 3]
numbers.extend([4, 5, 6])
print(numbers) # خروجی: [6, 5, 4, 3, 2, 1]
```

→ افزودن مقدار در یک موقعیت خاص `insert(index, value)` ✓

```
numbers = [1, 2, 4]
numbers.insert(2, 3) # عدد ۳ را در ایندکس ۲ قرار می‌دهد
print(numbers) # خروجی: [4, 3, 2, 1]
```

حذف عناصر از لیست

→ حذف اولین مقدار مشخص‌شده `remove(value)` ✓

```
numbers = [1, 2, 3, 4, 3]
numbers.remove(3) # فقط اولین عدد ۳ حذف می‌شود
print(numbers) # خروجی: [3, 4, 2, 1]
```

→ حذف مقدار بر اساس ایندکس (و بازگشت مقدار) `pop(index)` ✓

```
numbers = [10, 20, 30]
removed_item = numbers.pop(1) # حذف مقدار ایندکس ۱ (عدد 20)
print(numbers) # خروجی: [30, 10]
print(removed_item) # خروجی: 20
```

→ حذف یک مقدار یا کل لیست `del` ✓

```
numbers = [10, 20, 30, 40]
del numbers[1] # حذف ایندکس ۱ (عدد 20)
print(numbers) # خروجی: [40, 30, 10]

del numbers # حذف کل لیست
```

تغییر مقدار یک عنصر

لیست‌ها قابل تغییر (Mutable) هستند، بنابراین می‌توان مقدار یک عنصر خاص را تغییر داد.

```
numbers = [10, 20, 30]
numbers[1] = 50
print(numbers) # خروجی: [10, 50, 30]
```

۳. لیست‌های تو در تو (Nested Lists)

لیست‌های تو در تو لیست‌هایی هستند که حاوی لیست‌های دیگر هستند.

```
matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]

# دسترسی به مقدار خاص
print(matrix[1][2]) # خروجی: 6 (ردیف دوم، ستون سوم)
```

۴. توابع کاربردی لیست‌ها

→ مرتب‌سازی لیست (به صورت افزایشی) 

```
numbers = [5, 2, 9, 1]
numbers.sort()
print(numbers) # خروجی: [1, 2, 5, 9]
```

→ مرتب‌سازی به صورت نزولی 

```
numbers = [5, 2, 9, 1]
numbers.sort(reverse=True)
print(numbers) # خروجی: [9, 5, 2, 1]
```

→ مرتب‌سازی بدون تغییر لیست اصلی 

```
numbers = [3, 1, 4, 2]
sorted_numbers = sorted(numbers)
print(sorted_numbers) # خروجی: [1, 2, 3, 4]
print(numbers) # لیست اصلی تغییر نمی‌کند
```

تعداد عناصر لیست → `len()` 


```
numbers = [10, 20, 30]
print(len(numbers)) # خروجی: 3
```

پیدا کردن ایندکس یک مقدار خاص → `index(value)` 

```
numbers = [10, 20, 30]
print(numbers.index(20)) # خروجی: 1
```

شمارش تعداد تکرار یک مقدار → `count(value)` 

```
numbers = [1, 2, 3, 2, 2, 4]
print(numbers.count(2)) # خروجی: 3
```

معکوس کردن لیست → `reverse()` 

```
numbers = [1, 2, 3, 4]
numbers.reverse()
print(numbers) # [4, 3, 2, 1] خروجی:
```

نتیجه‌گیری

- لیست‌ها (Lists) از پرکاربردترین ساختارهای داده‌ای در پایتون هستند.
- امکان افزودن، حذف و تغییر مقدار عناصر لیست وجود دارد.
- از توابع داخلی مانند `append()`، `sort()`، `reverse` برای مدیریت لیست استفاده می‌شود.
- لیست‌های تو در تو برای نمایش داده‌های چندبعدی مانند ماتریس‌ها کاربرد دارند.

🚀 با تسلط بر لیست‌ها، می‌توانید بسیاری از مسائل برنامه‌نویسی را به‌سادگی حل کنید. 📌