

List 1 (Intro)

لیست چیست؟

یک شهر بزرگ و پرجمعیت را در نظر بگیرید. فرض کنید تمامی خانه‌ها در این شهر یک طبقه باشند. در این صورت، برای اسکان تمامی افراد شهر، تعداد بسیار زیادی خانه‌ی یک طبقه نیاز خواهد بود. این مسئله ممکن است مدیریت شهری را پیچیده سازد. به همین دلیل، برج‌ها و آپارتمان‌ها جایگزین خانه‌های یک طبقه شده‌اند. هر یک از این ساختمان‌ها دارای **تعداد طبقات** مشخص و **شماره پلاک** یا **نام اختصاصی** هستند. برخی از آن‌ها مسکونی و برخی دیگر اداری هستند.

اکنون به حوزه‌ی برنامه‌نویسی بازگردیم. تاکنون با مفهوم متغیر آشنا شده‌ایم و می‌دانیم که متغیرها، بخش‌هایی از حافظه هستند که اطلاعات خاصی را در خود ذخیره می‌کنند. تعریف ۱۰ متغیر از نوع عدد صحیح ممکن است ساده باشد، اما آیا تعریف ۵۰ یا ۱۰۰ متغیر نیز به همان سادگی است؟ این کار می‌تواند مشکلات متعددی نظیر دشواری در نام‌گذاری، افزایش احتمال خطا و پیچیدگی بیش از حد برنامه را ایجاد کند. برای مدیریت این شرایط، از **لیست** استفاده می‌شود.

در زبان برنامه‌نویسی پایتون، لیست یکی از چهار نوع داده‌ی داخلی محسوب می‌شود که برای ذخیره‌سازی مجموعه‌ای از داده‌ها به کار می‌رود. سایر انواع شامل **تاپل (Tuple)**، **مجموعه (Set)** و **دیکشنری (Dictionary)** هستند که هر یک کاربردها و ویژگی‌های خاص خود را دارند.

نوع داده‌های لیست

در یک لیست، اعضا می‌توانند دارای انواع داده‌ای مختلف باشند و لازم نیست که همگی از یک نوع باشند. این ویژگی انعطاف‌پذیری بالایی را در برنامه‌نویسی فراهم می‌کند.

نام‌گذاری لیست

به‌منظور استفاده‌ی آسان از لیست‌ها در برنامه، انتخاب نام مناسب ضروری است. قوانین نام‌گذاری که برای متغیرها معرفی شده‌اند، در اینجا نیز باید رعایت شوند.

تعداد عناصر لیست

هنگام تعریف اولیه‌ی یک لیست، تعداد خانه‌های آن بر اساس مقادیر اولیه مشخص می‌شود. در طول اجرای برنامه، این تعداد می‌تواند کاهش یا افزایش یابد که در ادامه‌ی این فصل به بررسی آن خواهیم پرداخت.

نحوه‌ی تعریف لیست در پایتون:

[... , عضو دوم , عضو اول] = اسم لیست

نکته‌ی مهم: اعضای لیست با علامت `,` از یکدیگر جدا می‌شوند.

نمونه‌ای از تعریف لیست:

```
names = ['Ali', 'Mina', 'Baran', 'Amin']
numbers = [1, '2', 'five', 4.5]
```

لیست اول شامل ۴ عنصر از نوع **رشته** است. لیست دوم شامل عناصری از انواع **عدد صحیح**، **رشته** و **عدد اعشاری** است.

روش‌های دسترسی به عناصر لیست

۱. دسترسی به یک عنصر خاص

برای دسترسی به یک عنصر مشخص از لیست که در خانه‌ی `i` ام قرار دارد، از روش زیر استفاده می‌شود:

```
list1[i]
```

مثال:

```
list_adad = [1, 2, 5, 8]
sum = list_adad[0] + list_adad[2]
print(sum)
```

در این مثال، لیستی شامل چهار عدد تعریف شده و مجموع عناصر خانه‌ی اول و سوم محاسبه و چاپ می‌شود.

۲. انتخاب مجموعه‌ای از عناصر

در برخی مواقع نیاز داریم که به چندین عنصر لیست به‌طور هم‌زمان دسترسی داشته باشیم. به‌عنوان مثال، اگر بخواهیم پنج عنصر اول یک لیست ده‌خانه‌ای را دریافت کنیم، می‌توان از روش زیر استفاده کرد:

```
list1[i:j + 1]
```

مثال:

```
list_adad = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
new_list = list_adad[2:6]
print(new_list)
```

نکات مهم:

- در عبارت `list[i:j]`، عنصر `j` ام در لیست جدید لحاظ نمی‌شود. برای گنجاندن آن، باید `list[i:j+1]` استفاده شود.
- خروجی حاصل همچنان یک لیست است.

ارجاعی بودن لیست‌ها

در پایتون، اشیای تغییرپذیر (مانند لیست) به‌صورت **ارجاعی (by-reference)** و اشیای تغییرناپذیر (مانند عدد صحیح و رشته) به‌صورت **مقداری (by-value)** منتقل می‌شوند.

مثال:

```
>>> main = [1, 2, 3]
>>> ref = main
>>> ref
[1, 2, 3]
>>> main[0] = 10
>>> ref
```

```
[10, 2, 3]
>>> ref[0] = 1
>>> main
[1, 2, 3]
```

در این مثال، هر دو متغیر `main` و `ref` به یک لیست ارجاع داده شده‌اند. بنابراین هر تغییری در یکی از آن‌ها، در دیگری نیز منعکس می‌شود.

تبدیل اشیا به لیست

با استفاده از تابع `list()` می‌توان انواع داده‌ای مختلف را به لیست تبدیل کرد.

مثال:

```
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> list((10, 20, 30))
[10, 20, 30]
>>> list("salam")
['s', 'a', 'l', 'a', 'm']
```

ایجاد لیست با مقادیر یکسان

فرض کنید قصد داریم لیستی به طول `n` ایجاد کنیم که تمامی عناصر آن مقدار صفر داشته باشند. استفاده از روش معمول، به‌ویژه برای مقادیر زیاد، می‌تواند دشوار باشد. روش بهینه‌تر به‌صورت زیر است:

```
zeros = [0] * 100
```