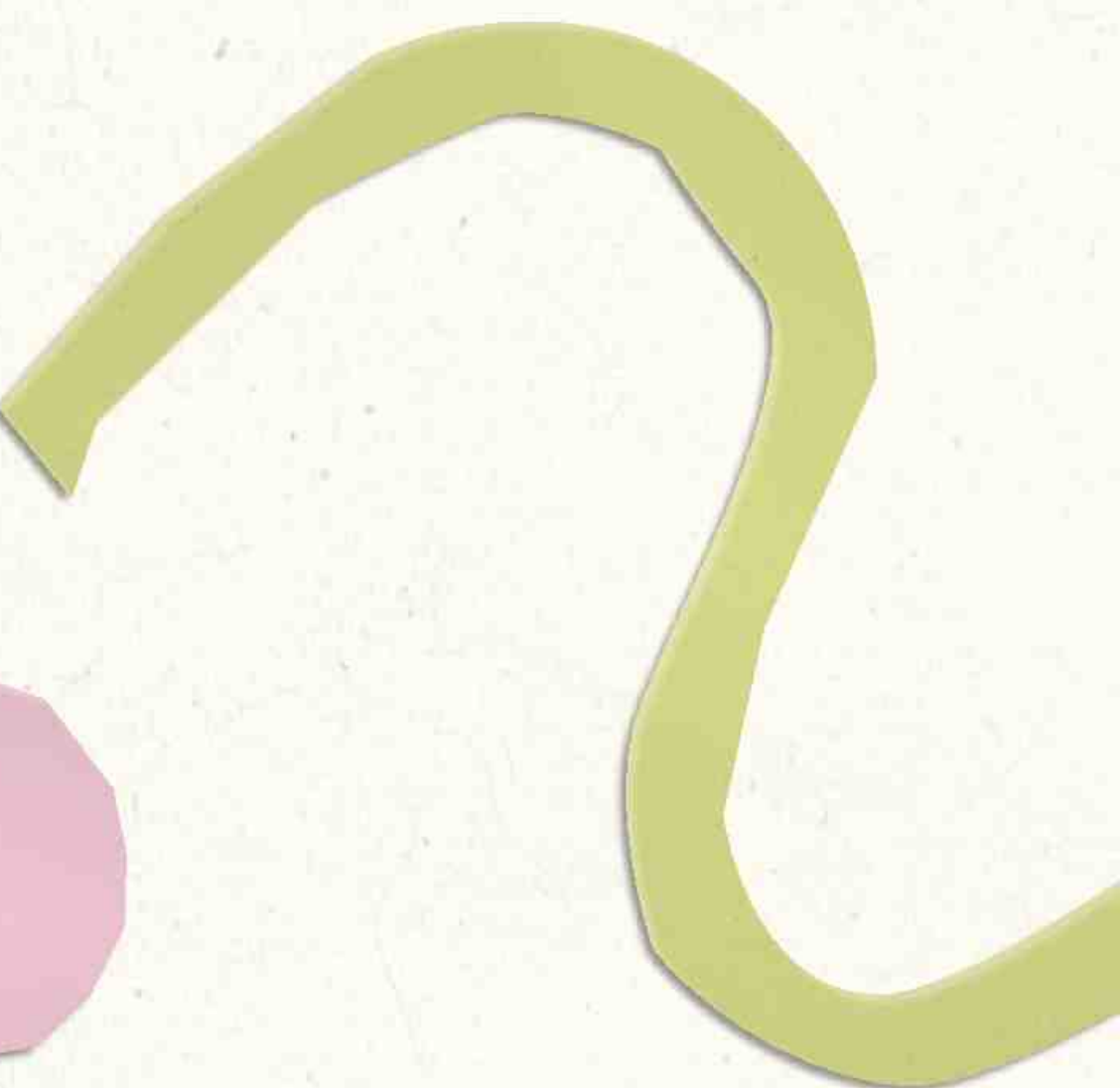
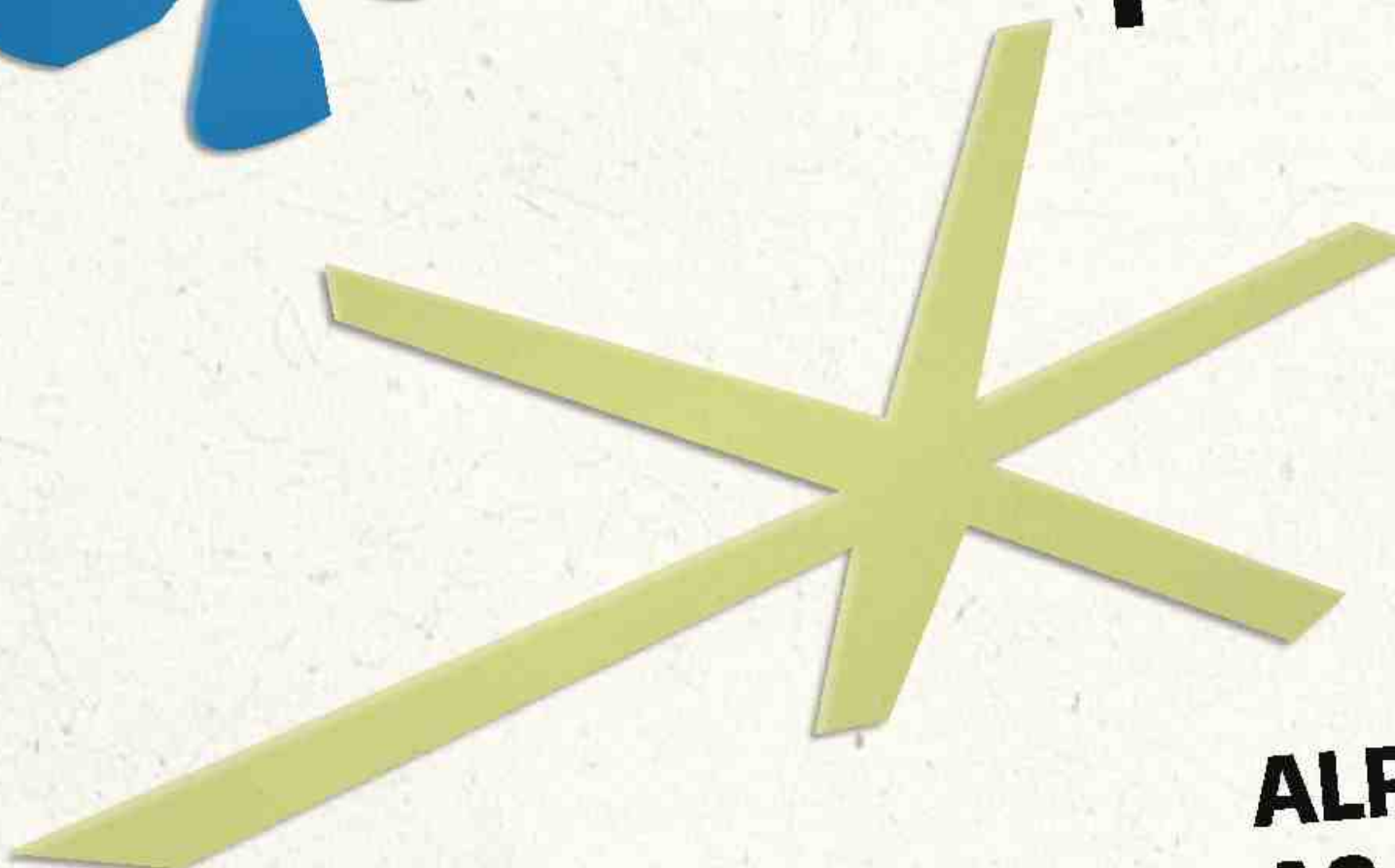
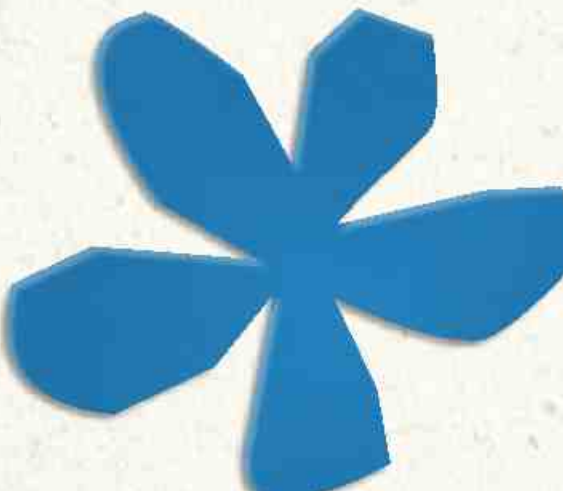
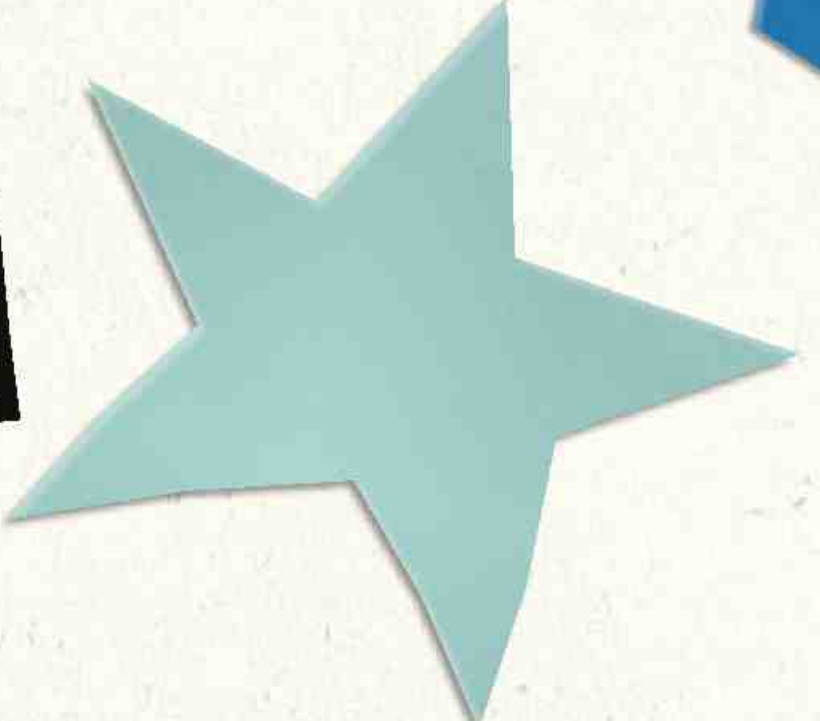
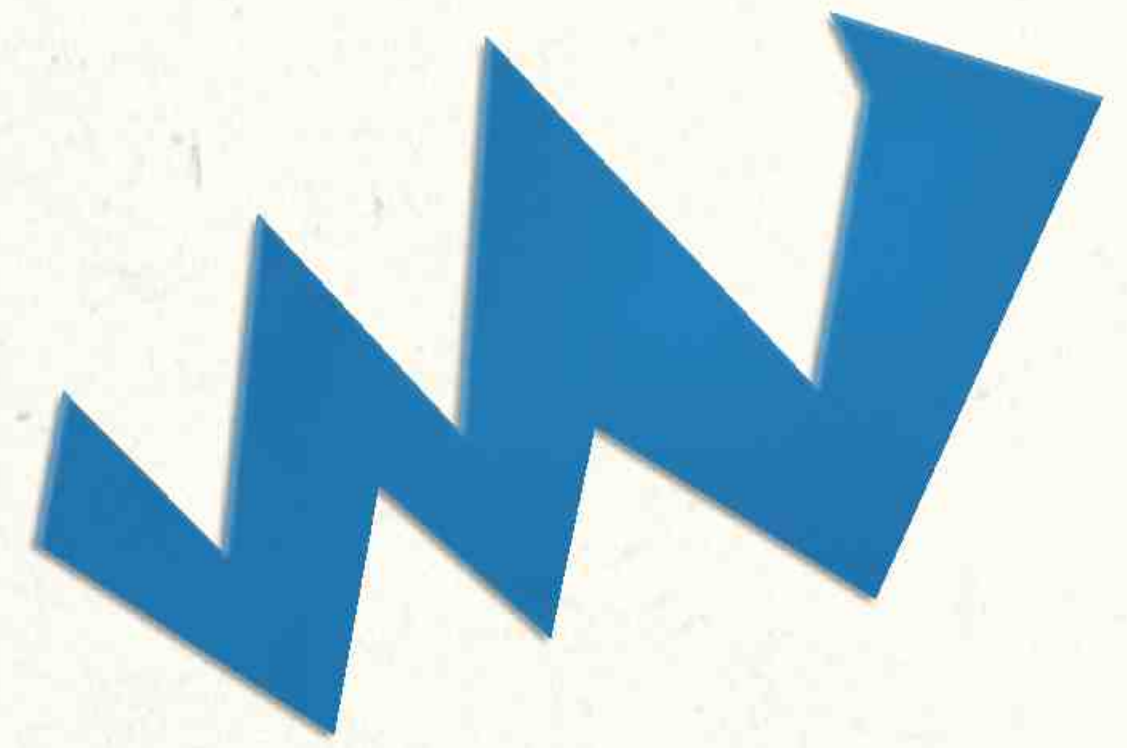




# python class

**ALPAN ACADEMY  
ASAL RASOULI**



شرط چیست؟

در برنامه نویسی، همیشه قرار نیست برنامه از ابتدا تا انتها دستورات را به یک شکل اجرا کند. گاهی لازم است برنامه بر اساس یک وضعیت یا یک شرط، تصمیم بگیرد که چه کاری انجام دهد. در پایتون، برای تصمیم گیری از دستورات **if**، **elif** و **else** استفاده می کنیم.

به زبان ساده، شرط مثل یک سؤال درست یا غلط است:

اگر این شرط درست بود، این کار را انجام بده؛  
در غیر این صورت، کار دیگری انجام بده.



# یک مثال ساده

```
score = 75
```

```
if score >= 10:
```

```
    print("قبول شدی")
```

```
else:
```

```
    print("مردود شدی")
```

فرض کنید می‌خواهیم بررسی کنیم آیا نمره  
یک دانش‌آموز برای قبولی کافی است یا نه:

در این مثال **score >= 10** یک شرط است

اگر شرط **True** باشد، دستور داخل **if** اجرا  
می‌شود.

اگر شرط **False** باشد، دستور داخل **else** اجرا  
می‌شود.

## چرا شرط‌ها مهم هستند؟

شرط‌ها یکی از مهم‌ترین ابزارهای تصمیم‌گیری در برنامه‌نویسی هستند و به ما اجازه می‌دهند:

- بر اساس اطلاعات مختلف تصمیم بگیریم.
- رفتار برنامه را تغییر دهیم.
- ورودی‌های کاربر را بررسی کنیم.
- داده‌ها را دسته‌بندی کنیم.
- از اجرای دستورات نامناسب جلوگیری کنیم.



# ساختار if

ساده‌ترین نوع شرط در پایتون با if ساخته می‌شود.

ساختار کلی:

**if condition:  
statement**

در این مثال condition همان شرطی است که باید بررسی شود.

اگر نتیجه شرط True باشد، دستورات داخل if اجرا می‌شوند.

```
age = 20  
if age >= 18:  
    print("you are adult")
```

چون مقدار age برابر ۲۰ است،  
شرط  $age \geq 18$  درست است  
و پیام نمایش داده می شود.



# تورفتگی یا Indentation

در پایتون، تورفتگی اهمیت بسیار زیادی دارد.

مثال نادرست

```
age = 20  
if age >= 18:  
print("you are adult")
```

مثال صحیح

```
age = 20  
if age >= 18:  
    print("you are adult")
```



دستور else

**if condition:**

**statement 1**

**else:**

**statement 2**

گاهی می‌خواهیم اگر شرط if درست نبود،  
یک کار دیگر انجام شود

در این حالت از **else** استفاده می‌کنیم.

ساختار:



# یک مثال ساده

```
age = 15
```

```
if age >= 18:
```

```
    print("you're adult!")
```

```
else:
```

```
    print("you aren't adult!")
```

فرض کنید می‌خواهیم بررسی کنیم آیا نمره  
یک دانش‌آموز برای قبولی کافی است یا نه:

در این مثال

چون شرط  $age \geq 18$  نادرست است،  
بخش `else` اجرا می‌شود.



# دستور elif

گاهی بیش از دو حالت داریم.

برای مثال، می‌خواهیم نمره دانش‌آموز را بررسی کنیم:

- نمره ۱۸ تا ۲۰ → عالی
- نمره ۱۵ تا کمتر از ۱۸ → خوب
- نمره ۱۰ تا کمتر از ۱۵ → قابل قبول
- کمتر از ۱۰ → مردود
- 

در این شرایط از elif استفاده می‌کنیم.

elif

## ساختار:

```
if condition1:  
    statement1  
elif condition2:  
    statement2  
elif condition3:  
    statement3  
else:  
    statement4
```

## مثال:

```
if score >= 18:  
    print("عالی")  
elif score >= 15:  
    print("خوب")  
elif score >= 10:  
    print("قابل قبول")  
else:  
    print("مردود")
```

پایتون شرط‌ها را از بالا به پایین بررسی می‌کند.

به محض اینکه یک شرط درست پیدا شود، همان بخش اجرا می‌شود و شرط‌های بعدی دیگر بررسی نمی‌شوند.

به همین دلیل ترتیب شرط‌ها اهمیت دارد.

**نکته مهم درباره elif**



# تفاوت elif، if و else

## دستور کاربرد

|                                                           |      |
|-----------------------------------------------------------|------|
| بررسی اولین شرط                                           | if   |
| بررسی شرط‌های بعدی                                        | elif |
| زمانی اجرا می‌شود که هیچ‌کدام از شرط‌های قبلی درست نباشند | else |



# استفاده از عملگرهای مقایسه‌ای در شرط‌ها

**مثال :**

```
age = 20
```

```
print(age == 20)
```

```
print(age > 18)
```

```
print(age < 18)
```



استفاده از and

# مثال :

```
age = 20
```

```
score = 17
```

```
if age >= 18 and score >= 15:
```

```
    print("ok!")
```

```
else:
```

```
    print("not ok!")
```

گاهی لازم است چند شرط به صورت همزمان درست باشند.

برای این کار از and استفاده می‌کنیم.

مثلاً برای ثبت نام یک دوره، فرد باید:

- حداقل ۱۸ سال داشته باشد.
- نمره اش حداقل ۱۵ باشد



استفاده از or

# مثال :

```
day = "جمعه"  
if day == "پنجشنبه" or day == "جمعه":  
    print("امروز تعطیل است")  
else:  
    print("امروز تعطیل نیست")
```

گاهی کافی است حداقل یکی از  
شرطها درست باشد.

در این حالت از or استفاده می‌کنیم.

چون یکی از شرطها درست است،  
پیام «امروز تعطیل است» چاپ  
می‌شود.



استفاده از not

# مثال :

عملگر not نتیجه یک شرط را برعکس می‌کند.

اگر نتیجه یک شرط True باشد، not آن را False می‌کند و برعکس.

چون is\_student برابر False است، عبارت not is\_student برابر True می‌شود.

```
is_student = False
```

```
if not is_student:
```

```
    print("شما دانشجو نیستید")
```



# ترکیب چند عملگر منطقی

# مثال :

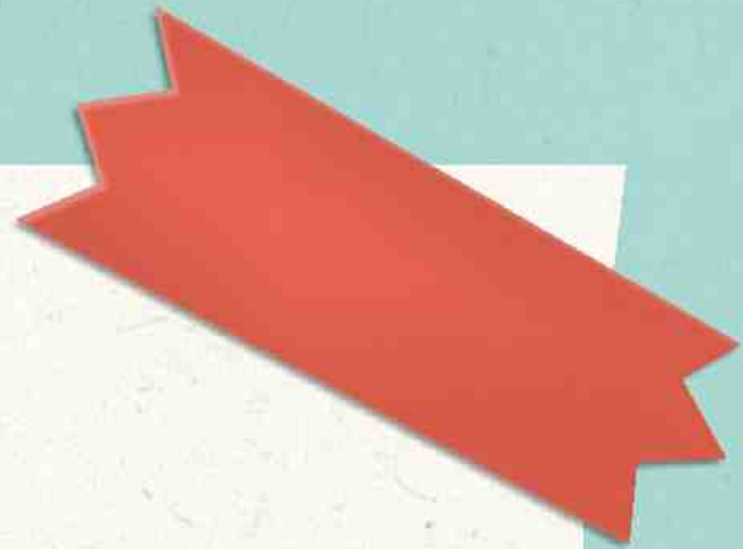
می‌توانیم and و or را در یک شرط ترکیب کنیم.

```
age = 20
score = 17
has_permission = True
if age >= 18 and score >= 15 and has_permission:
    print("شرایط کامل است")
else:
    print("شرایط کامل نیست")
```

در این مثال هر سه شرط باید درست باشند.

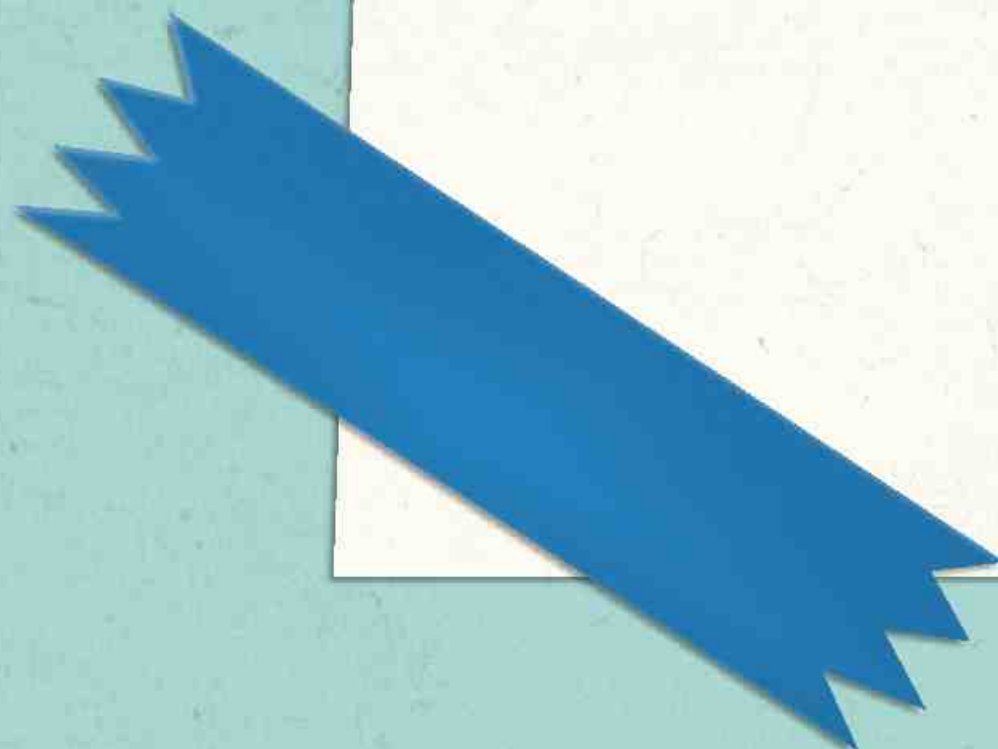


# شرط‌های تو در تو (Nested Conditions)



گاهی لازم است بعد از درست بودن یک  
شرط، شرط دیگری را بررسی کنیم.

به این نوع شرط، شرط تو در تو یا  
**Nested if** می‌گوییم.



**score = 18**

**attendance = 95**

**if score >= 10:**

**print("دانش آموز قبول شده است")**

**if attendance >= 90:**

**print("حضور بسیار خوبی داشته است")**

**else:**

**print("حضور دانش آموز می تواند بهتر باشد")**

**else:**

**print("دانش آموز مردود شده است")**

در این برنامه، شرط attendance فقط زمانی بررسی می شود که شرط اول یعنی  $score \geq 10$  درست باشد.

ساختار کلی:

**if condition1:**

**if condition2:**

**statement**

**else:**

**statement**

**else:**

**statement**



# استفاده از شرط با رشته‌ها

```
name = "Ali"
```

```
if name == "Ali":
```

```
    print("سلام علی")
```

```
else:
```

```
    print("نام شما علی نیست")
```

```
text = "I love Python"
```

```
if "Python" in text:
```

```
    print("در متن وجود دارد Python کلمه")
```



# استفاده از شرط با لیست

می‌توانیم وجود یک مقدار در لیست را بررسی کنیم.

```
fruits = ["apple", "banana", "orange"]
```

```
if "apple" in fruits:
```

```
    print("سیب در لیست وجود دارد")
```

```
else:
```

```
    print("سیب در لیست وجود ندارد")
```

همچنین می‌توانیم از توابعی مانند `min` و `len` ، `sum` ، `max` در شرط‌ها استفاده کنیم.

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

```
if sum(numbers) > 50:
```

```
    print("مجموع اعداد بیشتر از ۵۰ است")
```



# استفاده از شرط با دیکشنری

در دیکشنری می‌توانیم مقدار مربوط به یک کلید را بررسی کنیم.

```
student = { "name": "Sara", "age": 17, "grade": 18 }
```

```
if student["grade"] >= 10:
```

```
    print(student["name"], "قبول شده است.")
```

```
else:
```

```
    print(student["name"], "مردود شده است.")
```



# استفاده از شرط با Set

در مجموعه‌ها نیز می‌توانیم وجود یک عنصر را بررسی کنیم.

```
items = {"book", "pen", "notebook"}
```

```
if "pen" in items:
```

```
    print("قلم در مجموعه وجود دارد")
```

همچنین می‌توانیم تعداد عناصر را بررسی کنیم:

```
if len(items) >= 3:
```

```
    print("حداقل سه وسیله داریم")
```

می‌توانیم چند شرط را نیز ترکیب کنیم:

```
if len(items) >= 3 and "pen" in items:
```

```
    print("مجموعه حداقل سه عنصر دارد و شامل قلم است")
```



# تفاوت چند if با if و elif

## چند if مستقل

```
score = 85
```

```
if score >= 50:
```

```
    print("قبول")
```

```
if score >= 80:
```

```
    print("نمره خوب")
```

در این حالت هر دو شرط بررسی می‌شوند و هر دو پیام چاپ می‌شوند.

## استفاده از elif

**score = 85**

**if score >= 80:**

**print("نمره خوب")**

**elif score >= 50:**

**print("قبول")**

در این حالت، وقتی شرط اول درست شود، شرط elif دیگر بررسی نمی‌شود.

بنابراین:

استفاده از if‌های جداگانه برای شرایط مستقل مناسب هستند.

استفاده از if / elif / else برای انتخاب یکی از چند حالت مناسب است.

در این بخش یاد گرفتیم:

- ✓ استفاده از if برای بررسی یک شرط
- ✓ استفاده از elif برای بررسی حالت‌های بعدی
- ✓ استفاده از else برای زمانی که هیچ‌کدام از شرط‌ها درست نباشند
- ✓ عملگرهای مقایسه‌ای برای ساخت شرط‌ها
- ✓ استفاده از and برای زمانی که همه شرط‌ها باید درست باشند
- ✓ استفاده از or برای زمانی که حداقل یکی از شرط‌ها باید درست باشد
- ✓ استفاده از not برای برعکس کردن نتیجه یک شرط
- ✓ شرط‌های تو در تو (Nested if)
- ✓ تفاوت if‌های مستقل با ساختار if / elif / else



# تمرین کلاسی

66

یک برنامه بنویسید که یک عدد از کاربر دریافت کند و با استفاده از if، elif و else مشخص کند که عدد:

- مثبت است.
- منفی است.
- صفر است.

66

۶۶

برنامه‌ای بنویسید که سن کاربر را دریافت کند.  
اگر سن:

- کمتر از ۱۳ بود → «کودک»
  - بین ۱۳ تا ۱۷ بود → «نوجوان»
  - ۱۸ یا بیشتر بود → «بزرگسال»
- را چاپ کند.

۶۷

۶۶

برنامه‌ای بنویسید که نمره دانش‌آموز را دریافت کند و وضعیت او را مشخص کند:

- ۱۸ تا ۲۰ → عالی
- ۱۵ تا کمتر از ۱۸ → خوب
- ۱۰ تا کمتر از ۱۵ → قابل قبول
- کمتر از ۱۰ → مردود

۶۷

66

برنامه‌ای بنویسید که سن و نمره دانش‌آموز را دریافت کند.

دانش‌آموز فقط در صورتی شرایط ثبت‌نام را دارد که:

- سن حداقل ۱۸ باشد.
  - نمره حداقل ۱۵ باشد.
- از and استفاده کنید.

67

66

لیستی شامل چند نام تعریف کنید.  
بررسی کنید آیا نام "Ali" در لیست وجود دارد یا  
خیر.  
از in استفاده کنید.

66



# تمرین‌های خانگی

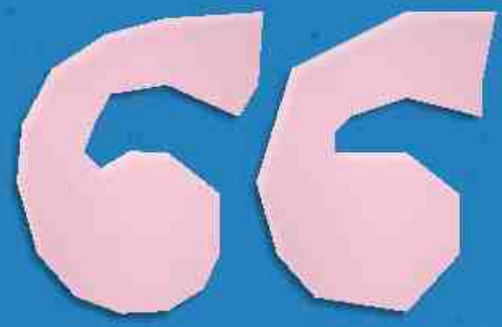
66

برنامه‌ای بنویسید که یک عدد از کاربر دریافت کند  
و مشخص کند:

- عدد مثبت و زوج است.
- عدد مثبت و فرد است.
- عدد منفی است.
- یا صفر است.

راهنمایی: از آ‌های تو در تو و عملگر % استفاده  
کنید.

99

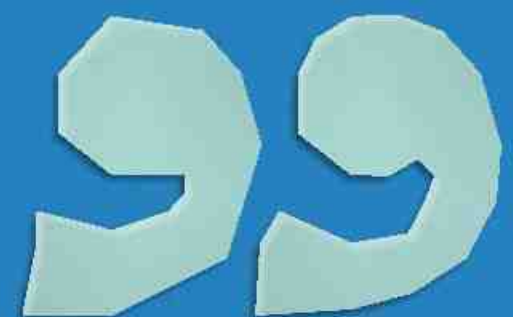


یک دیکشنری مانند زیر ایجاد کنید:

```
student = {  
    "name": "Ali",  
    "age": 17,  
    "grade": 18  
}
```

با استفاده از شرط‌ها بررسی کنید:

1. آیا نمره حداقل ۱۰ است؟
2. اگر قبول شده، آیا سن او حداقل ۱۸ است؟
3. اگر زیر ۱۸ سال بود، پیام مناسب چاپ کنید.



۶۶

یک لیست شامل ۴ عدد تعریف کنید.

با استفاده از شرط‌ها بررسی کنید:

- آیا مجموع اعداد بیشتر از ۵۰ است؟

- آیا بزرگ‌ترین عدد بیشتر از ۲۰ است؟

در صورت برقرار بودن شرایط، پیام مناسب چاپ کنید.

۶۷

THANKS!

**ALPAN ACADEMY**  
**ASAL RASOULI**