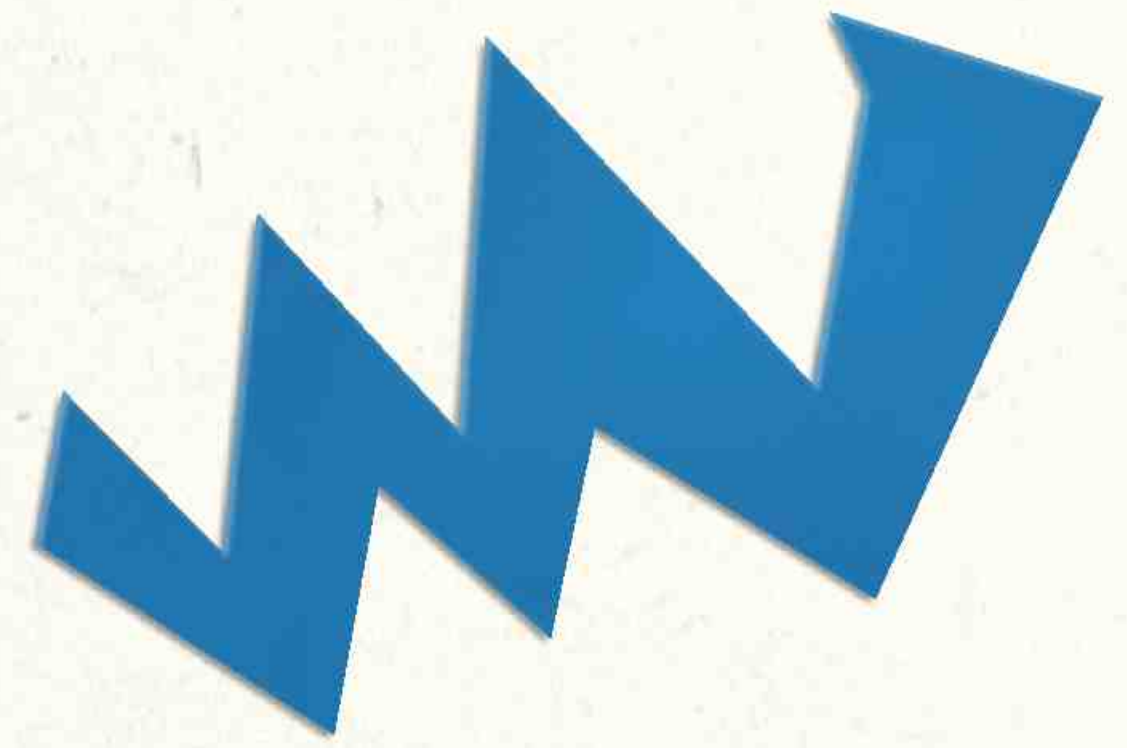
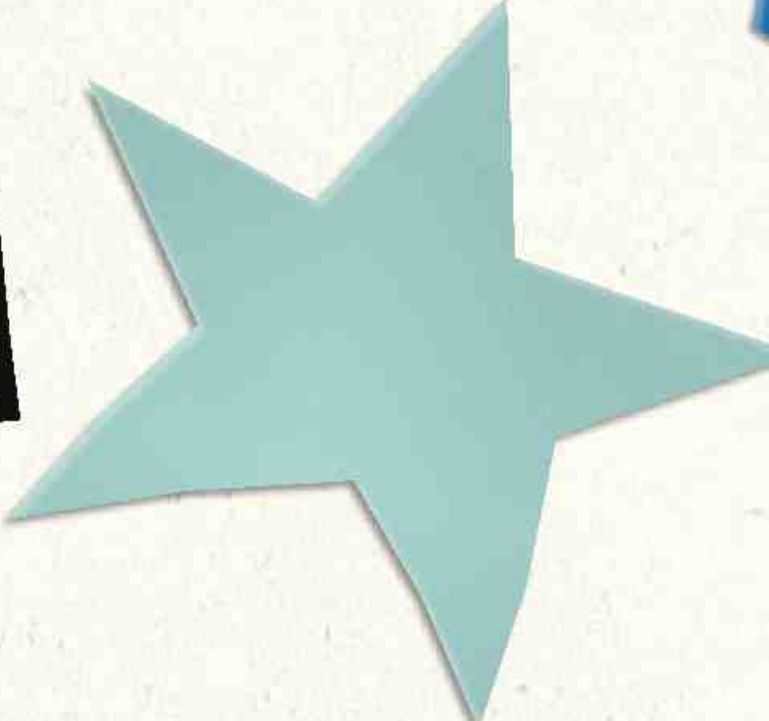
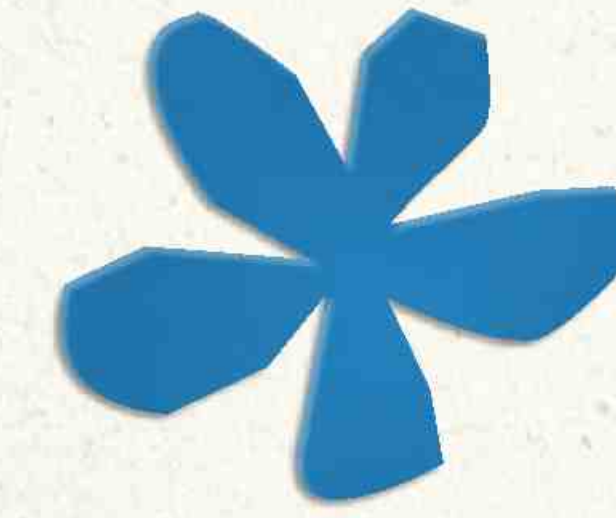
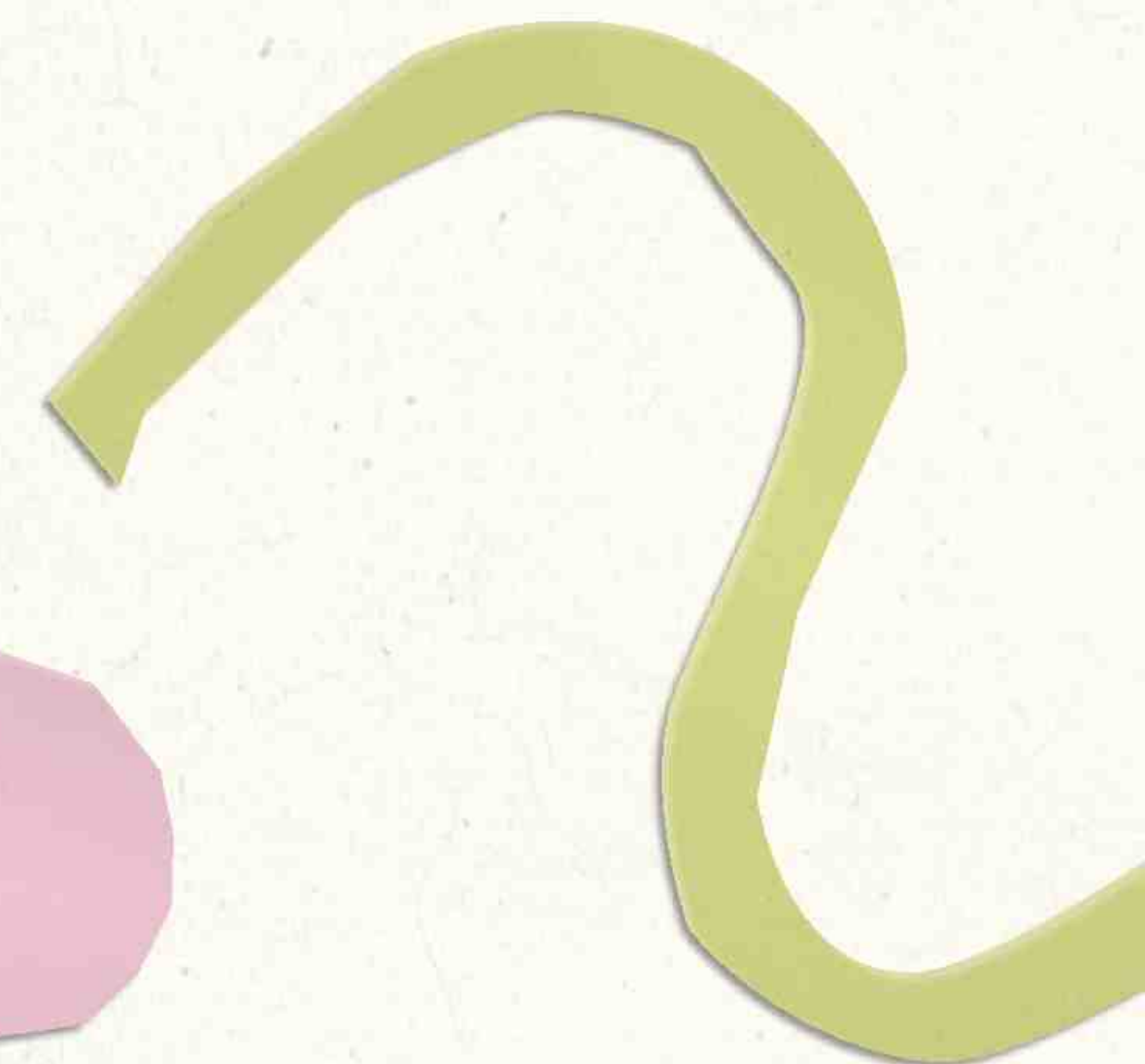
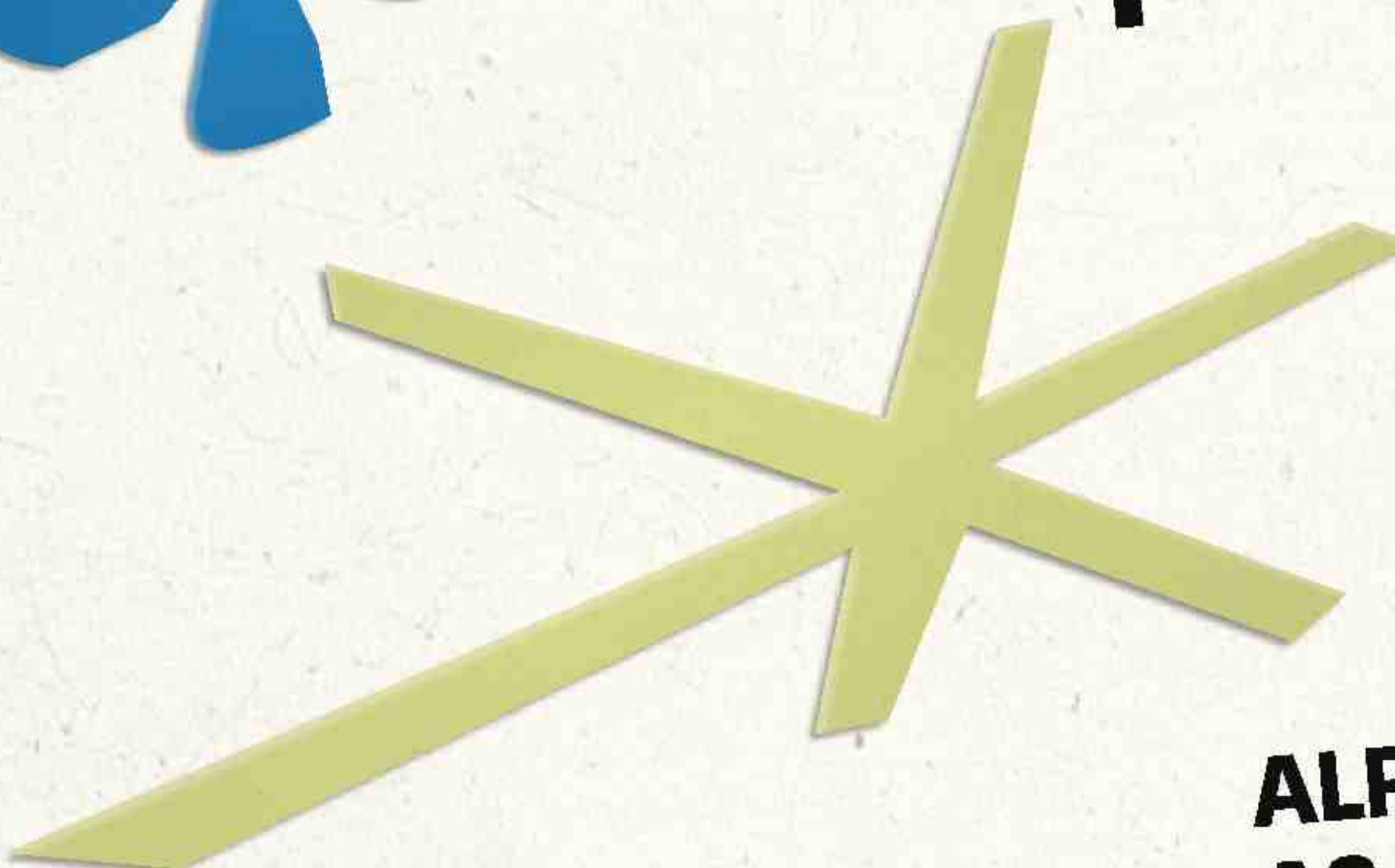




python

class

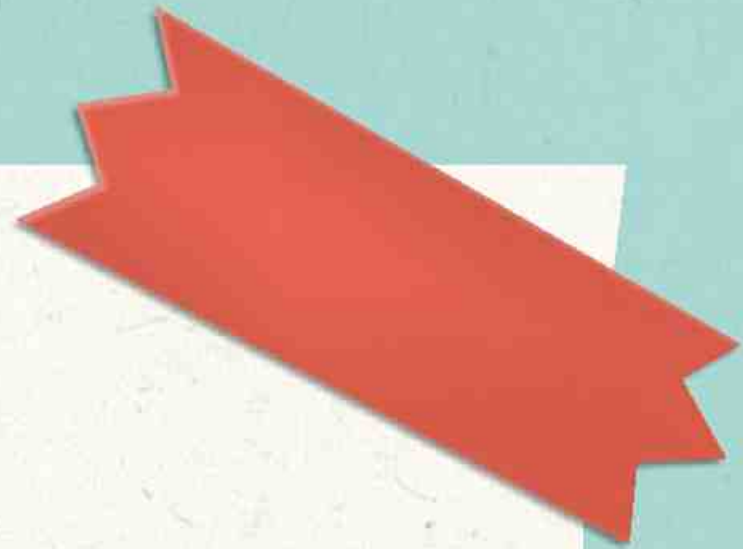


ALPAN ACADEMY
ASAL RASOULI





عملیات روی رشته‌ها چیست؟



همان طور که روی اعداد می توان عملیات مختلفی مانند جمع و ضرب انجام داد، روی رشته ها نیز می توان عملیات خاصی انجام داد.
به عنوان مثال:

- چسباندن دو رشته به یکدیگر
- تکرار یک رشته
- بررسی وجود یک کلمه در متن
- مقایسه دو رشته

این عملیات باعث می شوند بتوانیم متن ها را پردازش کنیم و برنامه های کاربردی تری بنویسیم.



الحاق رشته‌ها (Concatenation)

```
first_name = "Ali"  
last_name = "Ahmadi"  
  
full_name = first_name + last_name  
  
print(full_name)
```

گاهی لازم است دو یا چند رشته را به هم متصل کنیم.

به این کار الحاق (Concatenation) گفته می شود.

برای الحاق رشته ها از عملگر + استفاده می کنیم.

اضافه کردن فاصله

```
first_name = "Ali"  
last_name = "Ahmadi"  
  
full_name = first_name + " " + last_name  
print(full_name)
```

برای قرار دادن فاصله بین دو رشته، باید خودمان یک رشته شامل فاصله اضافه کنیم.

نکته مهم

```
age = 20
```

```
print("سن: " + age)
```

```
age = 20
```

```
print("سن: " + str(age))
```

عملگر + فقط رشته‌ها را به هم متصل می‌کند.

بنابراین کد زیر خطا ایجاد می‌کند:

زیرا age یک عدد است.

روش صحیح:



تکرار رشته‌ها (Repetition)

اگر بخواهیم یک رشته چند بار پشت سر هم تکرار شود، از عملگر * استفاده می‌کنیم.

```
word = "ها"
```

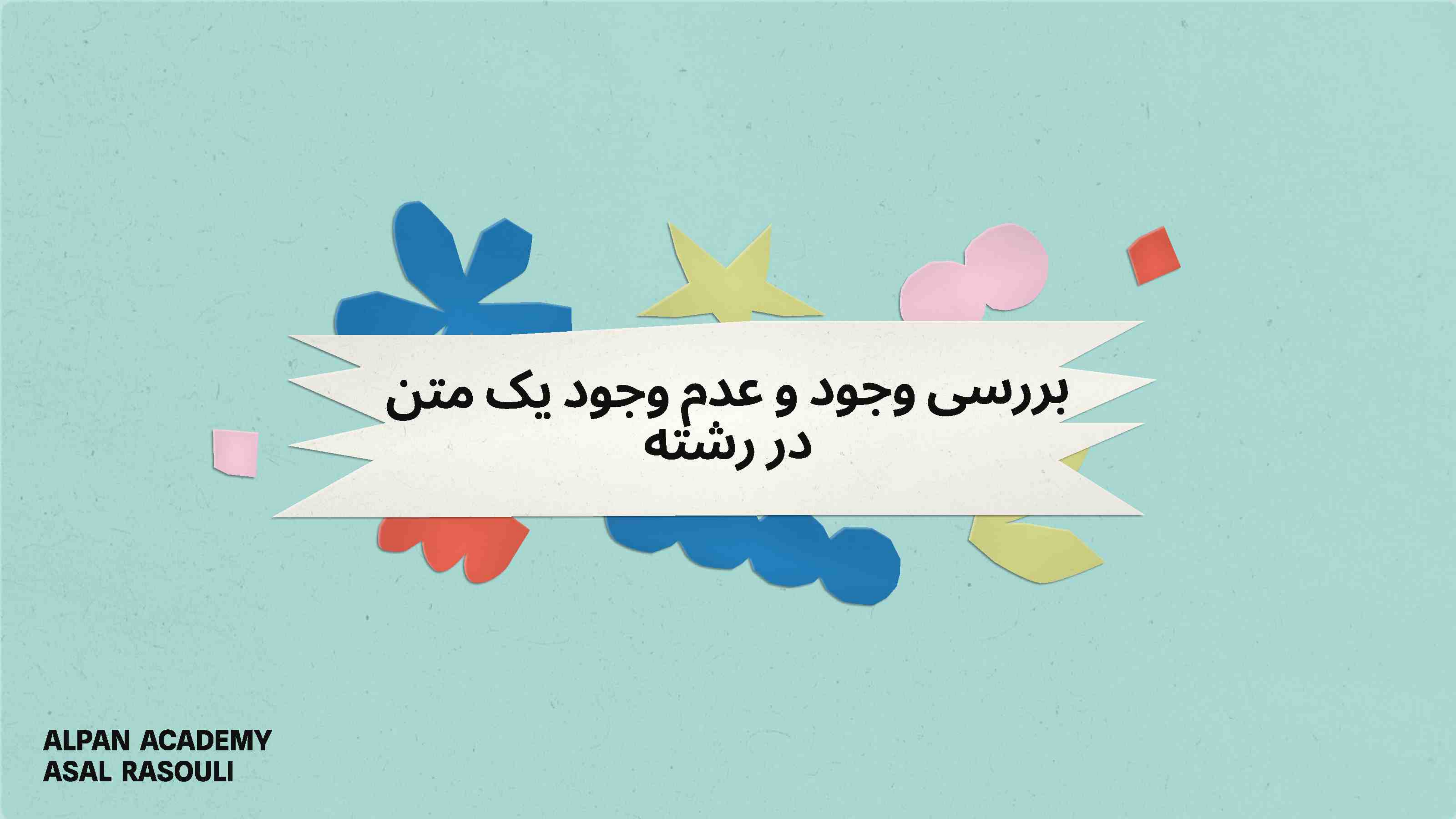
```
print(word *  
3)
```

```
print("-" * 30)
```

```
print("=" * 40)
```

```
print("گزارش نهایی")
```

```
print("=" * 40)
```



بررسی وجود و عدم وجود یک متن در رشته

بررسی نبودن یک متن (not in)

مثلاً:

```
text = "Python Programming"
```

```
print("Java" not in text)
```

```
print("Python" not in text)
```

بررسی وجود یک متن در رشته (in)

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

```
print("Python" in text)
```

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

```
print("Java" in text)
```



مقایسه رشته‌ها

بررسی نابرابر بودن

```
print(name1 != name2)
```

```
print("Ali" == "ali")
```

بررسی مساوی بودن

```
name1 = "Ali"  
name2 = "Ali"
```

```
print(name1 == name2)
```

```
name1 = "Ali"  
name2 = "Sara"
```

```
print(name1 == name2)
```



تغییرناپذیر بودن رشته‌ها (Immutable)

کی از مهم‌ترین ویژگی‌های رشته‌ها این است که رشته‌ها تغییرناپذیر (Immutable) هستند.

یعنی بعد از ساخته شدن یک رشته، نمی‌توان مستقیماً یکی از کاراکترهای آن را تغییر داد.

این برنامه **خطا** ایجاد می‌کند.

```
text = "Python"
```

```
text[0] = "J"
```

چرا رشته‌ها Immutable هستند؟

این ویژگی باعث می‌شود:

- سرعت اجرای برنامه بیشتر شود.
 - حافظه بهتر مدیریت شود.
 - امنیت داده‌ها افزایش پیدا کند.
 - بسیاری از عملیات داخلی پایتون سریع‌تر انجام شوند.
- در حال حاضر فقط کافی است بدانید که نمی‌توان یک کاراکتر از رشته را مستقیماً تغییر داد.

در این بخش یاد گرفتیم:

- ✓ با عملگر + رشته‌ها را به هم متصل می‌کنیم.
- ✓ با عملگر * یک رشته را چند بار تکرار می‌کنیم.
- ✓ با in بررسی می‌کنیم که آیا یک متن داخل رشته وجود دارد یا خیر.
- ✓ با not in نبودن یک متن را بررسی می‌کنیم.
- ✓ رشته‌ها را می‌توان با == و != مقایسه کرد.
- ✓ رشته‌ها نسبت به حروف بزرگ و کوچک حساس هستند (در متون انگلیسی).
- ✓ رشته‌ها Immutable هستند و نمی‌توان یک کاراکتر آن‌ها را مستقیماً تغییر داد.



تمرین‌های کلاسی

۶۶

نام و نام خانوادگی کاربر را دریافت کنید و آن‌ها را با
یک فاصله به هم متصل کرده و نمایش دهید.

۶۷

۶۶

یک کلمه از کاربرد دریافت کنید و آن را ۵ بار پشت
سر هم چاپ کنید.

۶۶

66

یک جمله از کاربرد دریافت کنید و بررسی کنید آیا
کلمه "پایتون" داخل آن وجود دارد یا خیر.

99

66

دو رشته از کاربر دریافت کنید و بررسی کنید که آیا
با هم برابر هستند یا خیر.

99



تمرین‌های خانگی

66

برنامه‌ای بنویسید که نام، نام خانوادگی و شهر کاربر را دریافت کند و جمله زیر را نمایش دهد:

سلام علی محمدی، به شهر یزد خوش آمدی.

66

66

برنامه‌ای بنویسید که یک کلمه از کاربر دریافت
کند و با استفاده از عملگر * ، یک قاب ساده برای
آن بسازد.

66

THANKS!

ALPAN ACADEMY
ASAL RASOULI