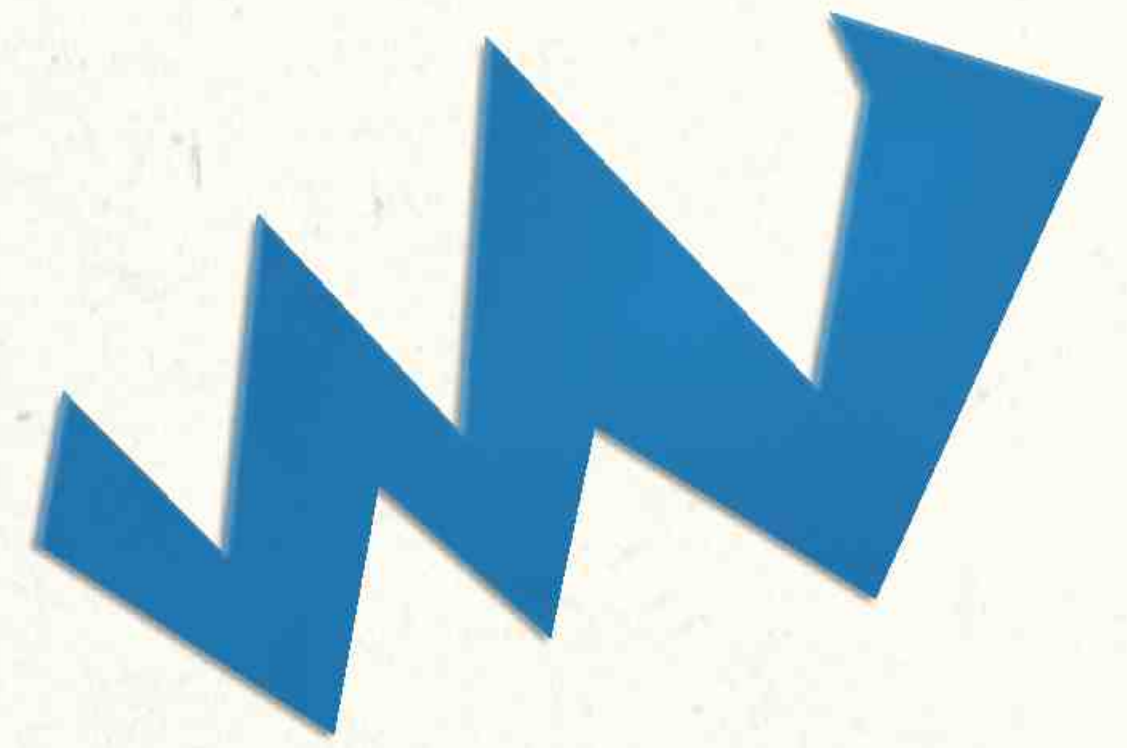
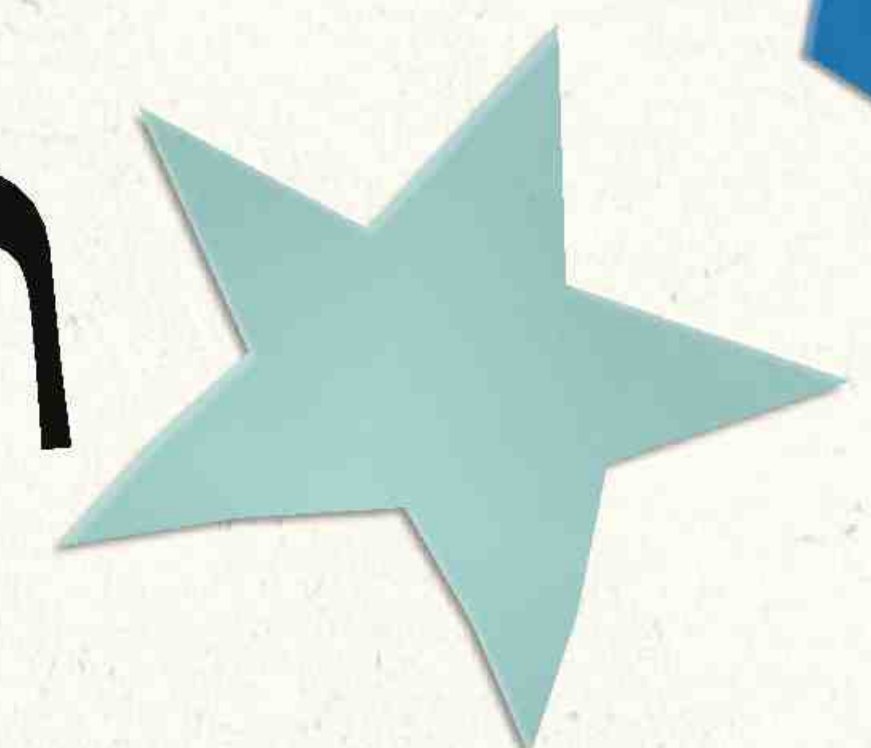
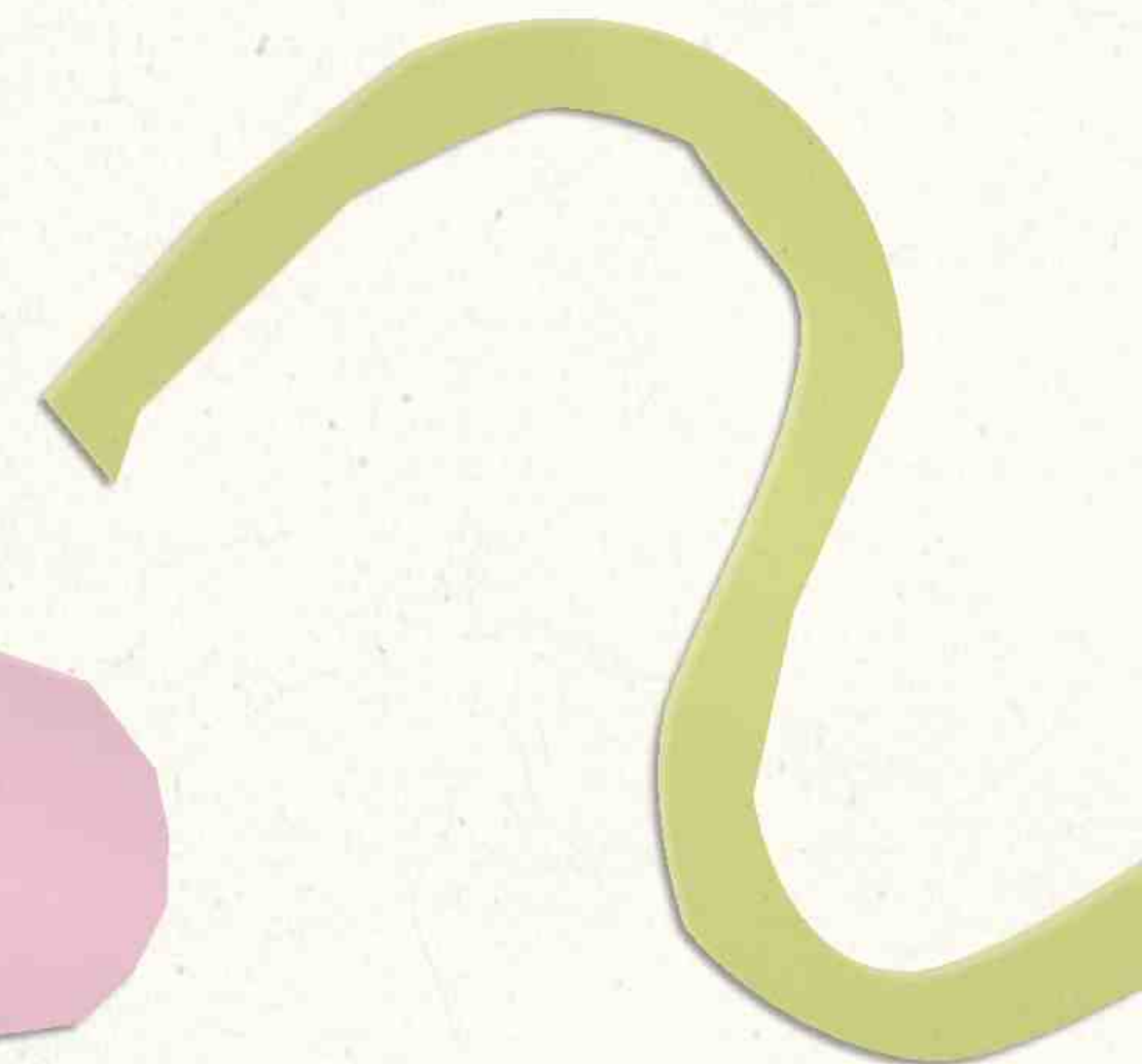
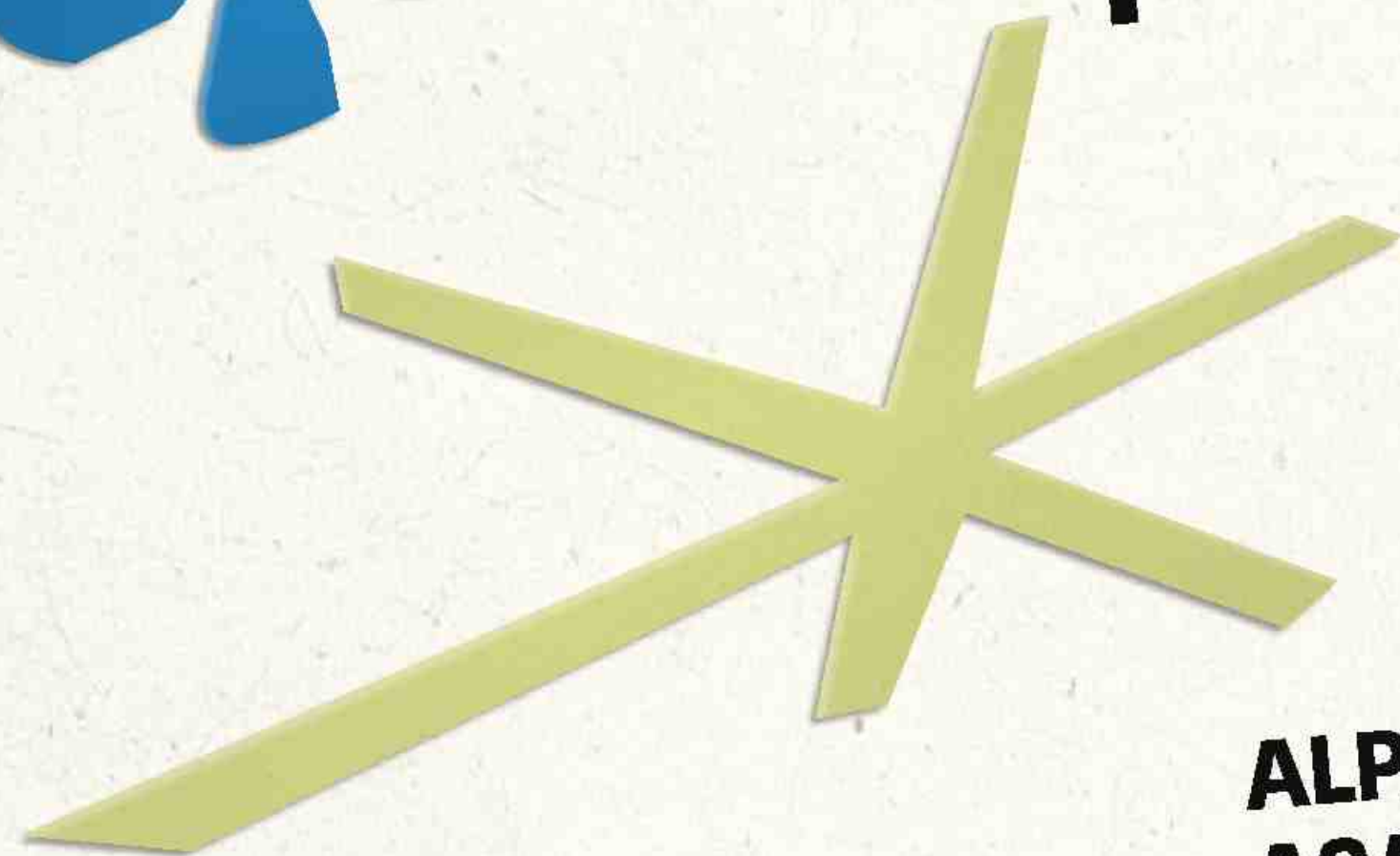




python

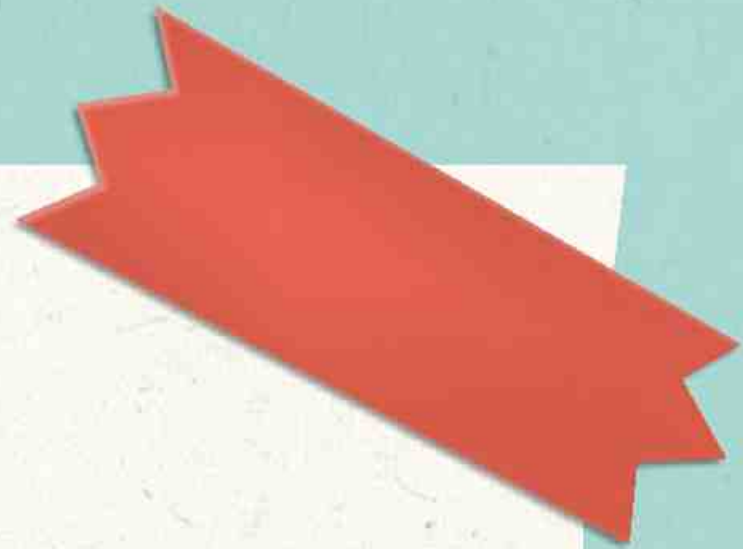
class

ALPAN ACADEMY
ASAL RASSOULI

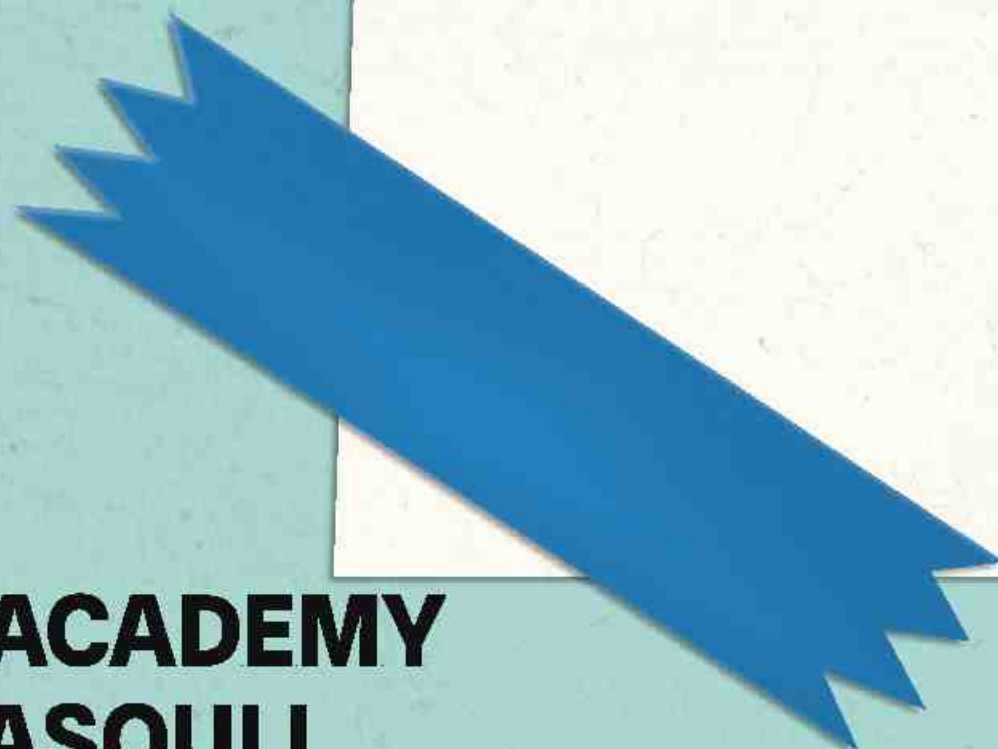




متد (Method) چیست؟



قبل از اینکه با متدهای رشته آشنا شویم، ابتدا باید بدانیم متد چیست. متد (Method) در واقع یک تابع مخصوص یک نوع داده است. به بیان ساده‌تر، بعضی از انواع داده‌ها (مثل رشته، لیست و دیکشنری) ابزارهایی دارند که فقط روی همان نوع داده کار می‌کنند. به این ابزارها متد می‌گوییم. متدها بعد از نام متغیر و با استفاده از علامت نقطه (.) نوشته می‌شوند.





variable.method()

text = "Python"
print(text.upper())

نکته مهم

متدها رشته اصلی را تغییر نمی دهند! بلکه یک رشته جدید تولید می کنند.

text = "python"
text.upper()
print(text)

متد upper

متد upper تمام حروف انگلیسی را به حروف بزرگ تبدیل می‌کند.

```
text = "python"
```

```
print(text.upper())
```

متد lower

این متد تمام حروف انگلیسی را به حروف کوچک تبدیل می‌کند.

```
text = "PYTHON"
```

```
print(text.lower())
```

متد strip

گاهی کاربر قبل یا بعد از متن فاصله اضافه وارد می‌کند.
این فاصله‌ها معمولاً باعث ایجاد خطا در مقایسه رشته‌ها می‌شوند.
برای حذف فاصله‌های ابتدا و انتهای رشته از strip استفاده می‌کنیم.

```
text = " Python "  
print(text.strip())
```

```
text.lstrip()  
text.rstrip()
```

```
username = input("نام کاربری:")  
username = username.strip()  
print(username)
```

متد replace

گاهی لازم است بخشی از متن را با متن دیگری جایگزین کنیم.
برای این کار از متد replace استفاده می‌کنیم.

```
text.replace(old, new)
```

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

```
print(text.replace("Python", "Programming"))
```

متد find

این متد محل اولین پیدا شدن یک رشته را برمی‌گرداند.
اگر رشته پیدا نشود، مقدار -1 را برمی‌گرداند.

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

```
print(text.find("Python"))
```

متد count

این متد تعداد دفعات تکرار یک رشته را می‌شمارد.

```
text = "banana"
```

```
print(text.count("a"))
```

متد startswith

بررسی می‌کند آیا رشته با متن مشخصی شروع شده است یا خیر.

کاربرد

بررسی پسوند یا پیشوند فایل‌ها، آدرس سایت‌ها و...

```
text = "Python"
```

```
print(text.startswith("Py"))
```

متد `endswith`

بررسی می‌کند رشته با چه متنی پایان یافته است.

```
file = "photo.jpg"
```

```
print(file.endswith(".jpg"))
```

```
print(file.endswith(".png"))
```

متد split

یکی از پرکاربردترین متدهای رشته است.
رشته را به چند قسمت تقسیم کرده و نتیجه را به صورت لیست برمی گرداند.

```
text = "Python Java C++"  
print(text.split())
```

تقسیم با جداکننده دلخواه

```
text = "Ali,Reza,Sara"  
print(text.split(","))
```

کاربرد

تقریباً تمام فرم‌های ثبت نام، فایل‌های CSV و داده‌های متنی از split استفاده می‌کنند.

متد join

برعکس split عمل می‌کند.
چند رشته را به هم متصل می‌کند.

```
names = ["Ali", "Sara", "Reza"]
```

```
print("-".join(names))
```

```
print(" ".join(names))
```

در این بخش یاد گرفتیم:

در این بخش با مهم‌ترین متدهای رشته آشنا شدیم:

- upper()
- lower()
- strip()
- replace()
- find()
- count()
- startswith()
- endswith()
- split()
- join()



تمرین‌های کلاسی

66

یک کلمه انگلیسی از کاربرد دریافت کنید و آن را هم
با حروف بزرگ و هم با حروف کوچک نمایش
دهید.

66

66

یک جمله از کاربر دریافت کنید، فاصله‌های
اضافی ابتدا و انتهای آن را حذف کنید و سپس
تعداد دفعات تکرار حرف a را نمایش دهید.

99

66

یک آدرس فایل از کاربر دریافت کنید و بررسی کنید آیا فایل با پسوند pdf پایان می یابد یا خیر.

67

66

یک جمله از کاربرد دریافت کنید و همه فاصله‌ها را
با خط تیره (-) جایگزین کنید.

99



تمرین‌های خانگی

66

نام کامل کاربر را دریافت کنید، فاصله‌های اضافی
را حذف کنید و آن را به دو بخش نام و نام خانوادگی
تقسیم کنید.

67

66

یک ایمیل از کاربر دریافت کنید و بررسی کنید که
آیا با @gmail.com پایان می‌یابد یا خیر.

99

66

یک جمله از کاربر دریافت کنید و تعداد دفعات
تکرار کلمه "Python" را نمایش دهید.

99

66

برنامه‌ای بنویسید که یک جمله از کاربر دریافت کند، آن را با `split()` به کلمات جداگانه تقسیم کند و سپس با استفاده از `join()` دوباره کلمات را با علامت - به هم متصل کند.

67

THANKS!

ALPAN ACADEMY
ASAL RASOULI