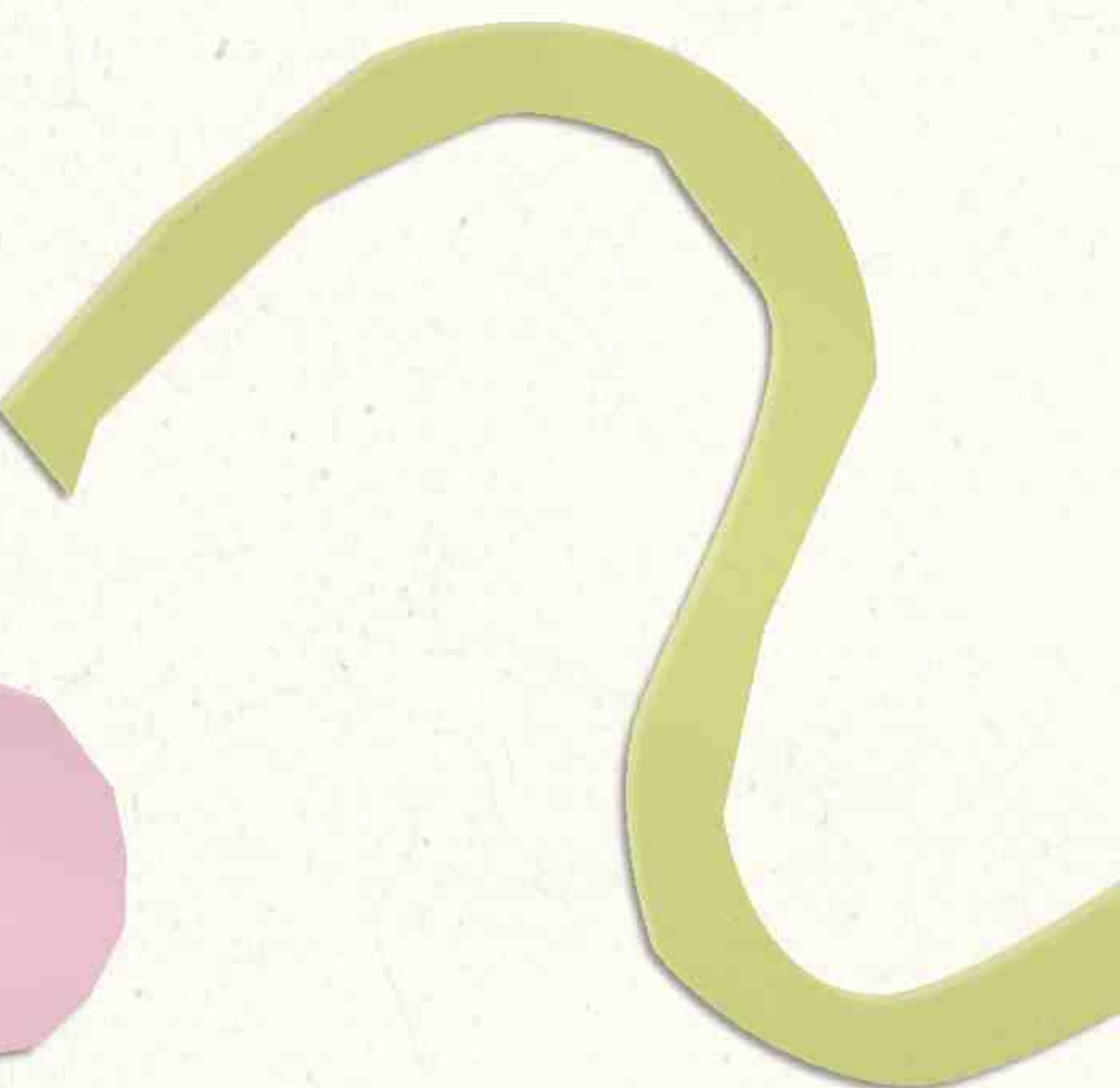
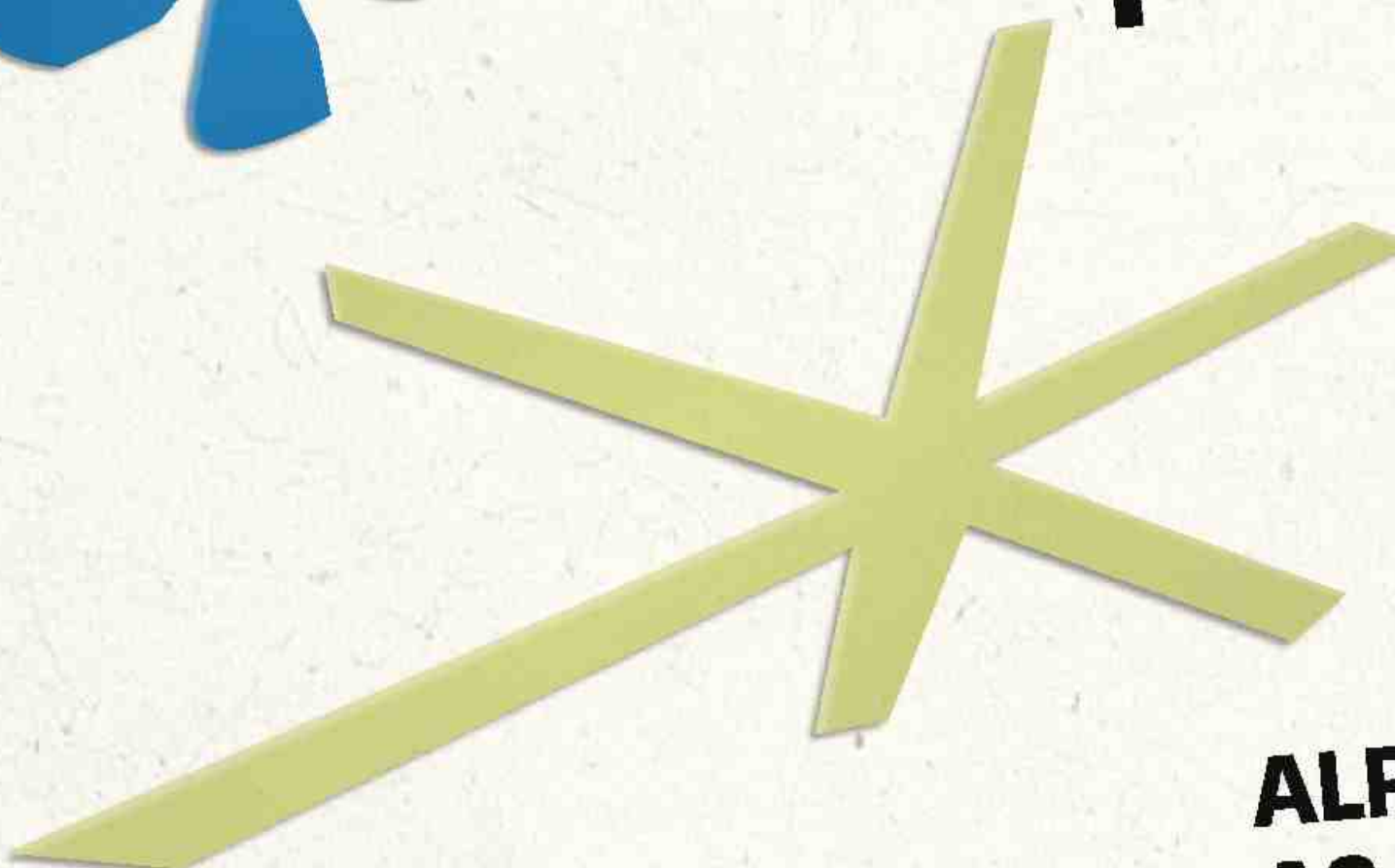
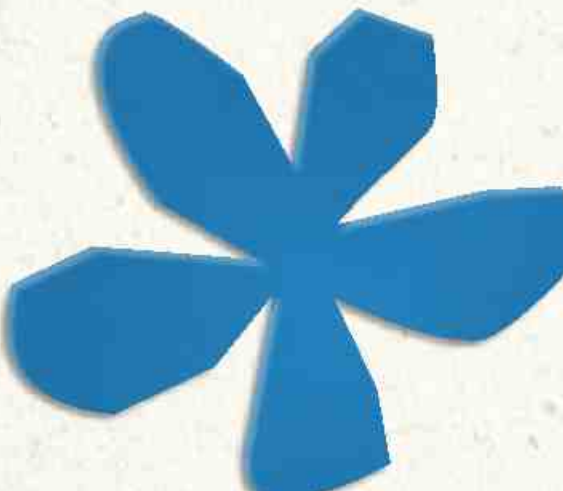
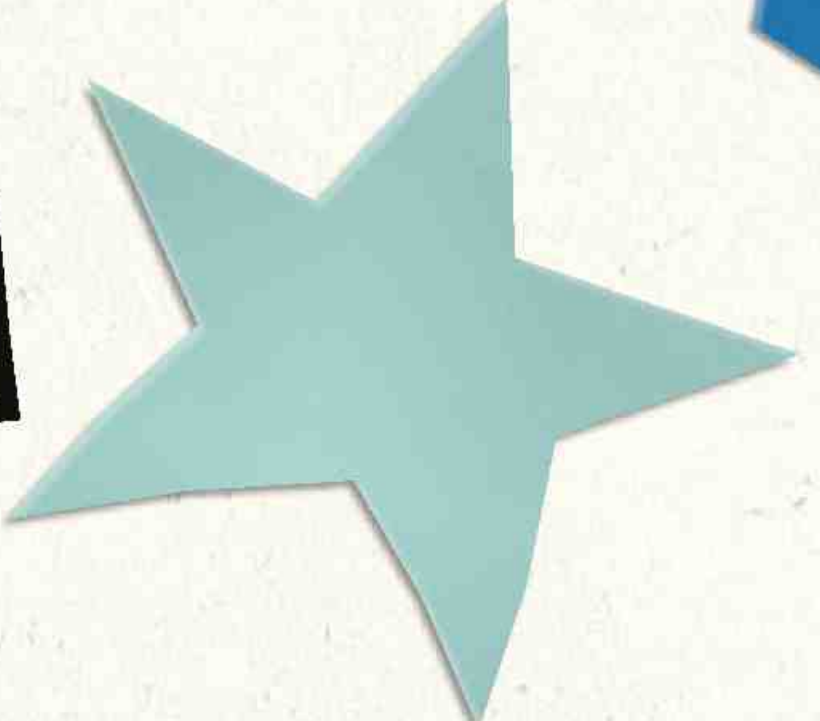
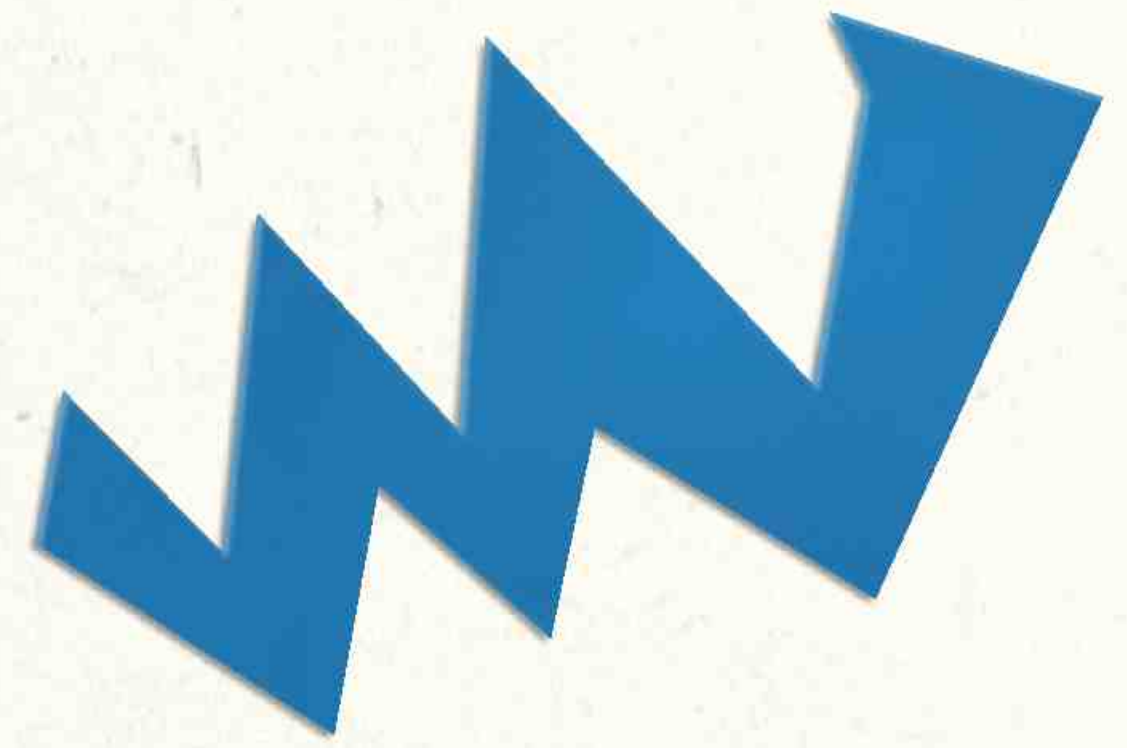




python class

**ALPAN ACADEMY
ASAL RASOULI**



دیکشنری چیست؟

دیکشنری (Dictionary) یکی از مهم‌ترین ساختمان‌های داده در پایتون است که اطلاعات را به صورت جفت‌های کلید و مقدار (Key-Value) ذخیره می‌کند.

در دیکشنری، هر کلید (Key) به یک مقدار (Value) مربوط می‌شود.

می‌توانیم دیکشنری را مانند یک دفترچه اطلاعات در نظر بگیریم:

نام → علی
سن → 20
شهر → یزد

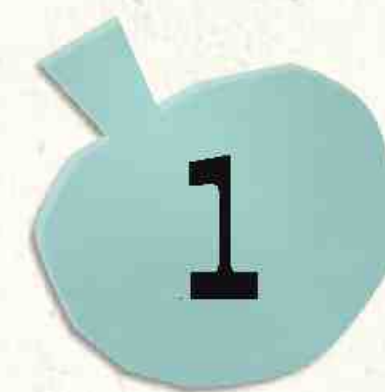
```
student = {  
    "name": "علی",  
    "age": 20,  
    "grade": 18.5  
}
```

```
print(student)
```

در این مثال name یک کلید است.
در این مثال "Ali" یک مقدار است.
در این مثال age یک کلید است.
در این مثال 20 یک مقدار است.

دیکشنری با {} تعریف می شود و هر جفت کلید و مقدار با : از هم جدا می شوند.

ویژگی‌های دیکشنری



قابل تغییر
(Mutable)



کلیدها باید منحصر به فرد
باشند



کلید باید Hashable
باشد



ترتیب درج کلیدها در
پایتون حفظ می‌شود



دسترسی به مقدماتها

```
person = {  
    "name": "Sara",  
    "age": 18  
}
```

```
print(person["name"])
```

```
print(person["city"])
```

برای دسترسی به مقدار یک کلید، دو روش
مهم داریم:

روش اول: **استفاده از []**

اگر کلید وجود نداشته باشد، استفاده از []
باعث ایجاد خطای **KeyError** می‌شود.

```
person = {  
    "name": "Sara",  
    "age": 18  
}
```

```
print(person.get("age"))
```

```
print(person.get("city"))
```

روش دوم: استفاده از **get()**

مزیت مهم **get()** این است که اگر کلید وجود نداشته باشد، برنامه به صورت معمول خطا نمی‌دهد.



اضافه کردن و تغییر مقدار

```
person = {  
    "name": "Ali"  
}
```

```
person["age"] = 20
```

```
print(person)
```

```
person["name"] = "Maryam"
```

```
print(person)
```

در دیکشنری برای اضافه کردن یک کلید جدید، کافی است آن را در [] قرار دهیم.

اگر کلید از قبل وجود داشته باشد، مقدار آن تغییر می‌کند.



حذف عناصر از دیکشنری

متد pop

```
person = {  
    "name": "Ali",  
    "age": 20,  
    "city": "Shiraz"  
}
```

```
age = person.pop("age")
```

```
print(age)  
print(person)
```

با pop می‌توانیم یک کلید را حذف کنیم و مقدار آن را نیز دریافت کنیم.

متد popitem

```
person = {  
    "name": "Ali",  
    "age": 20,  
    "city": "Shiraz"  
}
```

```
item = person.popitem()
```

```
print(item)  
print(person)
```

این متد آخرین جفت کلید-مقدار را حذف می‌کند و همان جفت را برمی‌گرداند.

متد clear

برای حذف تمام عناصر دیکشنری از
clear استفاده می‌کنیم.

```
person = {  
    "name": "Ali",  
    "age": 20  
}
```

```
person.clear()
```

```
print(person)
```



متدهای پرکاربرد دیکشنری

keys

تمام کلیدهای دیکشنری را برمی گرداند.

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

```
print(student.keys())
```

values

تمام مقادارهای دیکشنری را برمی گرداند.

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

```
print(student.values())
```

items

جفت‌های کلید-مقدار را برمی‌گرداند.

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

```
print(student.items())
```

update

برای اضافه کردن چند کلید جدید یا تغییر
چند مقدار به صورت همزمان می‌توانیم
از update استفاده کنیم.

```
student.update({  
  "city": "Tehran",  
  "grade": 19  
})
```

```
print(student)
```



بررسی وجود یک کلید

وقتی **in** را روی یک Dictionary استفاده
می‌کنیم، **کلیدها** بررسی می‌شوند، نه مقداها.

not in

```
person = {  
    "name": "Sara",  
    "age": 18  
}
```

```
print("city" not in person)
```

in

```
person = {  
    "name": "Sara",  
    "age": 18  
}
```

```
print("name" in person)  
print("city" in person)
```

توابع کاربردی



len

تعداد عناصر



sum

مجموع



max

بزرگ‌ترین مقدار



min

کوچک‌ترین مقدار

```
grades = {  
    "math": 18,  
    "science": 16,  
    "history": 19  
}
```

```
print(len(grades))  
print(sum(grades.values()))  
print(max(grades.values()))  
print(min(grades.values()))
```



دیکشنری‌های تو در تو (Nested Dictionaries)

یک Dictionary می تواند مقدارش یک Dictionary دیگر باشد.

```
school = {  
    "student1": {  
        "name": "Ali",  
        "age": 20  
    },  
    "student2": {  
        "name": "Maryam",  
        "age": 18  
    }  
}
```

برای دسترسی به اطلاعات دانش آموز اول:

```
print(school["student1"])
```

برای دسترسی مستقیم به نام دانش آموز دوم:

```
print(school["student2"]["name"])
```

حتی می توانیم مقدار را تغییر دهیم:

```
school["student1"]["age"] = 21
```

```
print(school["student1"])
```

در این بخش یاد گرفتیم:

- ✓ دیکشنری اطلاعات را به صورت Key-Value ذخیره می‌کند.
- ✓ کلیدها باید منحصر به فرد باشند.
- ✓ دیکشنری قابل تغییر است.
- ✓ با [] یا get() می‌توان به مقدارها دسترسی داشت.
- ✓ با popitem(), pop() و clear() می‌توان اطلاعات را حذف کرد.
- ✓ با dictionary[key] = value می‌توان مقدار جدید اضافه یا مقدار قبلی را تغییر داد.
- ✓ با in می‌توان وجود یک کلید را بررسی کرد.
- ✓ دیکشنری می‌تواند شامل دیکشنری‌های دیگر باشد.
- ✓ مقدارهای Dictionary می‌توانند انواع مختلف داده باشند.



تمرین کلاسی

66

یک Dictionary با اطلاعات زیر ایجاد کنید:

- نام
- سن
- شهر
- معدل

سپس تمام Dictionary را چاپ کنید.

67

66

دیکشنری زیر را ایجاد کنید:

```
student = {  
    "name": "Ali",  
    "age": 20  
}
```

سپس:

1. شهر دانش آموز را اضافه کنید.
2. سن او را به 21 تغییر دهید.
3. دیکشنری را چاپ کنید.

66

66

دیکشنری زیر را ایجاد کنید:

```
person = {  
  "name": "Sara",  
  "age": 19,  
  "city": "Yazd"  
}
```

موارد زیر را چاپ کنید:

- نام
- سن
- شهر

66

66

با استفاده از in بررسی کنید که آیا کلیدهای "name" و "email" در Dictionary زیر وجود دارند یا خیر.

```
user = {  
    "name": "Ali",  
    "age": 20  
}
```

67

66

دیکشنری زیر را ایجاد کنید:

```
person = {  
    "name": "Sara",  
    "age": 18,  
    "city": "Yazd"  
}
```

سپس:

1. کلید age را با pop() حذف کنید.

2. نتیجه را چاپ کنید.

66

66

دیکشنری زیر را ایجاد کنید:

```
grades = {  
    "math": 18,  
    "science": 16,  
    "programming": 20  
}
```

موارد زیر را محاسبه و چاپ کنید:

- تعداد درس‌ها
- مجموع نمرات
- میانگین
- بالاترین نمره
- پایین‌ترین نمره

99



تمرین‌های خانگی

66

یک Dictionary برای یک کتاب ایجاد کنید که
شامل موارد زیر باشد:

- عنوان کتاب
- نویسنده
- سال انتشار
- قیمت

سپس اطلاعات کتاب را چاپ کنید.

66

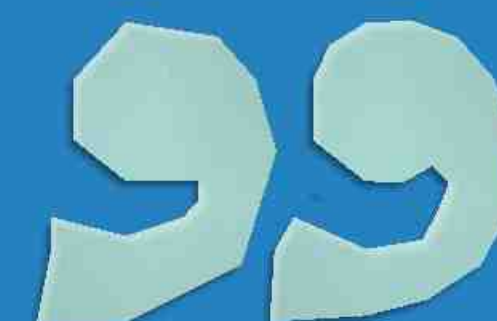


دیکشنری زیر را ایجاد کنید:

```
product = {  
  "name": "Laptop",  
  "price": 300000000,  
  "brand": "ASUS"  
}
```

سپس:

1. قیمت را تغییر دهید.
2. یک کلید جدید به نام stock اضافه کنید.
3. اطلاعات نهایی را چاپ کنید.



66

یک Dictionary برای اطلاعات یک کاربر ایجاد

کنید و با استفاده از in بررسی کنید:

- آیا کلید "email" وجود دارد؟

- آیا کلید "phone" وجود دارد؟

66



دیکشنری زیر را ایجاد کنید:

```
grades = {  
    "Ali": 17,  
    "Sara": 19,  
    "Maryam": 18,  
    "Nima": 16  
}
```

برنامه‌ای بنویسید که:

- تعداد دانش‌آموزان را نمایش دهد.
- مجموع نمرات را نمایش دهد.
- میانگین نمرات را محاسبه کند.
- بیشترین نمره را نمایش دهد.
- کمترین نمره را نمایش دهد.



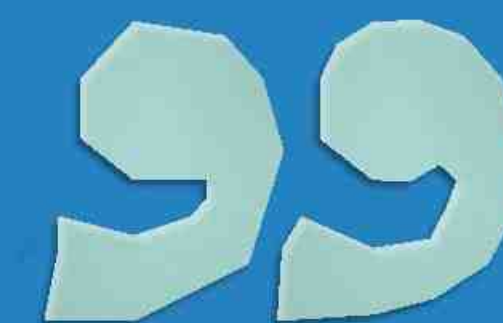


دیکشنری زیر را ایجاد کنید:

```
school = {  
    "student1": {  
        "name": "Ali",  
        "age": 20  
    },  
    "student2": {  
        "name": "Sara",  
        "age": 18  
    }  
}
```

سپس:

1. نام دانش‌آموز اول را چاپ کنید.
2. سن دانش‌آموز دوم را چاپ کنید.
3. سن دانش‌آموز اول را به 21 تغییر دهید.
4. نهایی را چاپ کنید Dictionary



THANKS!

ALPAN ACADEMY
ASAL RASOULI