

08_Dictionary

فایل هشتم: دیکشنری

دیکشنری (Dictionary):

- نوع داده دیکشنری در زبان برنامه نویسی پایتون به صورت لیستی از کلید ها و value ها است. هر کلید توسط علامت کالن (:) از value جدا می شود. اندیس ها نیز توسط علامت کاما از یک دیگر جدا می شوند. یک دیکشنری خالی بدون اندیس با استفاده از تنها دو آکولاد خالی به صورت {} نوشته می شود.
- کلید ها در دیکشنری یکتا می باشند (بدین معنا که نمی توان دو کلید یکسان داشت) ولی value می تواند تکراری نیز باشد.
- Value در دیکشنری می تواند از هر نوعی باشد ولی کلید ها باید یک نوع داده غیر قابل تغییر مانند: string, number یا tuple باشند.
- دیکشنر ها در زبان پایتون نوع دیگری از لیست ها هستند اما با یک تفانت اساسی آن تفاوت امکان تعیین کلید برای هر یک از آیتم های لیست است که می توان به وسیله این کلید به آیتم های لیست به سرعت دسترسی داشت یعنی شما مانند لیست ها از اندیس آیتم ها استفاده نمی کنید که باعث می شود نیازی به مرتب بودن لیست نباشد.

تمرینات

1. در تمرین اول ما یک دیکشنری ساختیم که شامل اسم و فامیل و سن هستش و در مرحله بعد خواستیم که اسم و سن را در خروجی برای چاپ کند.
2. در این تمرین با دستور clear() تمامی مقادیر داخل دیکشنری پاک می شوند برای همین است که موقع خروجی گرفتن ارور می دهد یعنی می گویید هیچ مقداری در دیکشنری وجود ندارد.
3. در این تمرین نیز همینطور است با دستور del کل دیکشنری حذف خواهد شد.
4. در این تمرین نیز مانند تمرین اول یک کلید را فراخوانی میکنیم و مقدار های آن در خروجی چاپ می شوند.
5. این تمرین نشان میدهد نحوه اضافه کردن کلید و مقدار جدید به دیکشنری چگونه است.

08_Dictionary

فایل هشتم: دیکشنری

6. این مثال نشان میدهد که با دستور `pop()` می توان هر کدام از کلید ها را که خواستید حذف کنید.

7. در این مثال با حلقه `for` از رنج صفر تا سه را به توان 3 کرده است و در خروجی نشان داده است.

8. در این مثال همان اتفاق تمرین قبل رخ داده با این تفاوت که طرز نوشتن متفاوتی دارند و در این یکی از دیکشنری استفاده شده است.

9. در این تمرین از 15 تا 35 را تمامی اعداد زوج را انتخاب می کند و برای آن ها دیکشنری می سازد که در آن کلید همان عدد و مقدار برابر مربع همان عدد باشد.

10. این تمرین با استفاده از تابع تو کار `len()` طول دیکشنری را حساب کرد و در خروجی چاپ کرده است.

11. در این تمرین با استفاده از تابع تو کار `sorted()` لیست جدیدی از کلید ها ساخته و در خروجی چاپ کرده است.

12. این کد برای تبدیل لیست به دیکشنری است با دستور `dict()` می توان لیست و تاپل را تبدیل به دیکشنری کرد و برعکس.

13. در این مثال در دو دیکشنری رنگ ها و شماره آن ها نوشته شده و میخواهد هر رنگ در خروجی با عدد خودش مقابل هم قرار بگیرند.

****نکته****

دو دیکشنری مانند لیست ها و تاپل ها به این صورت جمع نمیشوند:

08_Dictionary

فایل هشتم:دیکشنری

```
dic1={0:1,1:2}
```

```
dic2={2:3,3:4}
```

```
dic3=dic1+dic2
```

```
print(dic3)
```

14. در این کد جمع درست دو تاپل بیان شده است. با تابع تو کار `update()`

15. در این تمرین از سه تابع مهم دیکشنری استفاده شده است :

`mydict.keys()`: این تابع لیستی از کلید های دیکشنری را برمی گرداند.

`mdict.values()`: این تابع نیز لیستی از مقادیر دیکشنری را نشان می دهد.

`Mydict.items()`: این تابع لیستی از جفت های کلید-مقدار رو بر میگرداند به صورت

tupel

این سه تابع برای پیمایش و بررسی محتویات دیکشنری خیلی کاربردی هستند مخصوصا

وقتی بخواهیم روی کلیدها یا مقدار ها حلقه بزنیم یا دیکشنری رو تجزیه تحلیل کنیم.

16. این کد مربوط به انجام دادن یک ضرب است که در آن `result` مقادیر کلید ها را

در خود نگه میدارد و در بعدی ضرب می کند همانطور تا عدد آخر که ضرب همه عداد

با هم برابر عدد 3744

مجموعه (set):

• مجموعه دسته ای از اشیا مختلف متمایز از هم است.

08_Dictionary

فایل هشتم: دیکشنری

- مجموعه ها همان دیکشنری ها هستند فقط مقدار ندارند.
 - مجموعه ای از مقادیر که امکان درج مقدار تکراری در آن وجود ندارد. مانند کلید ها در دیکشنری که امکان درج دوکلید یکسان در آن وجود نداشت.
 - نمی توانیم عناصر مجموعه را تغییر دهیم عناصر مجموعه تغییر نا پذیر هستند اما نکته مهم ای است که خود مجموعه را می توان تغییر داد زیرا امکان اضافه کردن عنصر به آن و حذف عنصر از آن وجود دارد ولی به دلیل اینکه نمی توانیم مثلا بگوییم مقدار عنصر پنجم از مجموعه به 2 تغییر داده شود (مثل امکانی که در فهرست ها داریم) چون در مجموعه ترتیبی برای عناصر وجود ندارد گفته می شود که امکان تغییر عناصر مجموعه وجود ندارد.
17. اولین مثال مجموعه ها یک مجموعه ساده به آن داده شده که با استفاده از حلقه `for` عناصر در خروجی چاپ شده است.
18. برای اضافه کردن عضو به مجموعه از دستور `add()` استفاده میشه که همانطور که در این مثال هست عضو جدید به صورت نامنظم (یعنی در هر بار اجرا محل قرار گیری آن در مجموعه تغییر میکند) وارد مجموعه ما شده است.
19. در این مثال می خواهیم چند عضو به مجموعه اضافه کنیم که از دستور `update()` استفاده می کنیم.
20. این تمرین راجه حذف عضو از مجموعه است حذف عضو با دستور `remove()` و `discard()` انجام می شود حالا تفاوت این دو دستور را بررسی میکنیم:

08_Dictionary

فایل هشتم: دیکشنری

`remove(item)`: این متد آیتم را از مجموعه حذف می کند اما اگر آیتم وجود نداشته باشد خطای `keyError` می دهد.

`Discard(item)`: در صورت وجود آیتم را حذف میکند اما اگر هم وجود نداشت `error` نمی دهد.

از `remove` زمانی استفاده میکنیم که مطمئن باشیم آیتم در مجموعه وجود دارد.

از `discard()` زمانی استفاده می کنیم که نمی خواهیم برنامه ما در صورت نبودن آیتم دچار خطا شود.

21. این کد ترکیب دو مجموعه را نشان میدهد که با دستور `update()` نوشته شده است.

22. این تمرین نیز یک از راه های دیگر ترکیب دو مجموعه با هم است که با دستور `union()` نوشته خواهد شد.

23. در این کد نشان میدهد که با دستور `set()` می تواند یک تاپل به مجموعه تبدیل کرد.

24. در این تمرین با استفاده از دستور `intersection()` اشتراک بین دو مجموعه را در خروجی میتوان دید.

25. این تمرین نیز شکل دیگر اشتراک گیری بین دو مجموعه است با استفاده از `&` می توان اشتراک را نمایش داد.

26. با استفاده از این دستور `difference()` در این کد اخلاف یک مجموعه نسبت به مجموعه دیگر را در خروجی نمایش می دهیم.

27. در این کد با استفاده از اختلاف دو مجموعه یک مجموعه جدید ساخته ایم با دستور `symmetric_difference()`

28. در این مثال از کار بر خواسته شده تعداد عدد هایی که میخواهد به لیست اضافه کند را وارد کند و بعد یکی یکی عدد ها را به آخر لیست با دستور `append()` اضافه می کند.

08_Dictionary

فایل هشتم:دیکشنری

29. این مثال مانند مثال قبل است اما با این تفاوت که این بار هم اسم و هم سن شخص را میگیرد و به لیست اضافه میکند و حالا ما لیستی از نام و سن اشخاص داریم.

30. این تمرین آخر نیز مانند تمرین قبل است اما با این تفاوت که در آخر لیست را تبدیل به دیکشنری کردیم و اسم به عنوان کلید و سن به عنوان مقدار قرار دادیم.