

Dictionary

دیکشنری در پایتون

- دیکشنری در پایتون یک مجموعه نامرتب، قابل تغییر و ایندکس شده از آیتم ها می باشد که درون آکولاد قرار می گیرند.
- هر آیتم در دیکشنری هم شامل مقادیر (values) بوده و هم کلید (keys)
- در واقع در دیکشنری دسترسی به آیتم ها با استفاده از کلید انجام می شود.
- ساختار کلی دیکشنری

```
dict = {“key” : “value”, ... }
```

دیکشنری در پایتون

- مثال تعریف یک دیکشنری

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
print(thisdict)  
Output: {'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
```

- مثال دسترسی به عناصر دیکشنری

```
print(thisdict["brand"])  
Output: Ford
```

- اگر کلید در دیکشنری نبود پیغام خطا نمایش داده می شود. برای جلوگیری از این اتفاق می توانیم از

```
print(thisdict.get("price"))  
Output: None
```

متد **get** استفاده کنیم.

ایجاد و تغییر آیتم های دیکشنری

- دیکشنری ها تغییرپذیرند (mutable)
- هم می توان مقادیر آنها را تغییر داد و هم می توان مقادیر جدیدی به دیکشنری اضافه نمود.
- اگر کلید در دیکشنری موجود باشد، مقدار آن بروزرسانی شده و در غیر اینصورت یک کلید و مقدار جدید به دیکشنری اضافه می شود.

```
thisdict = {  
    "brand": "Ford",  
    "model": "Mustang",  
    "year": 1964  
}  
thisdict["year"] = 2019  
thisdict["price"] = 20000
```

Output: {'brand': 'Ford', 'model': 'Mustang', 'year': 2019, 'price': 20000 }

حذف آیتم در دیکشنری

- حذف یک آیتم خاص از دیکشنری به همراه کلید آن با استفاده از متد `pop()`

```
thisdict.pop("model")  
print(thisdict)
```

Output: {'brand': 'Ford', 'year': 1964}

- متد `popitem()` آخرین آیتم اضافه شده به دیکشنری را پاک می کند.
- دستور `del` مقدار مشخص شده با یک کلید مشخص را از دیکشنری پاک می کند.
- متد `clear()` دیکشنری را بصورت کامل خالی می کند.

دسترسی به عناصر دیکشنری

- از حلقه for می توان برای ایجاد حلقه در دیکشنری ها می توان استفاده نمود. در این حالت مقدار بازگشتی برابر است با است کلیدها دیکشنری.

```
for x in thisdict:  
    print(x)
```

Output: brand, model, year

- برای دسترسی به مقادیر می توان به صورت زیر عمل کرد:

```
for x in thisdict:  
    print(thisdict[x])
```

Output: Ford Mustang 1964

- همچنین از تابع values() هم می توان استفاده کرد

```
for x in thisdict.values():  
    print(x)
```

Output: Ford Mustang 1964

دیکشنری های تودرتو

- یک دیکشنری خود می تواند شامل چندین دیکشنری دیگر باشد.

```
myfamily = {  
    "child1" : {  
        "name" : "Emil",  
        "year" : 2004  
    },  
    "child2" : {  
        "name" : "Tobias",  
        "year" : 2007  
    },  
    "child3" : {  
        "name" : "Linus",  
        "year" : 2011  
    }  
}
```

dictionary متدهای

| Method | Description |
|-------------------------------------|---|
| <u>clear()</u> | Removes all the elements from the dictionary |
| <u>copy()</u> | Returns a copy of the dictionary |
| <u>fromkeys()</u> | Returns a dictionary with the specified keys and value |
| <u>get()</u> | Returns the value of the specified key |
| <u>items()</u> | Returns a list containing a tuple for each key value pair |
| <u>keys()</u> | Returns a list containing the dictionary's keys |
| <u>pop()</u> | Removes the element with the specified key |
| <u>popitem()</u> | Removes the last inserted key-value pair |
| <u>setdefault()</u> | Returns the value of the specified key. If the key does not exist: insert the key, with the specified value |
| <u>update()</u> | Updates the dictionary with the specified key-value pairs |
| <u>values()</u> | Returns a list of all the values in the dictionary |