Dictionaries

Bu fəslində **Python's dictionaries**-dən necə istifadə edəcəyinizi öyrənəcəksiniz; bu strukturlar əlaqəli məlumat hissələrini bir-birinə bağlamağa imkan verir. Dictionary-yə yerləşdirilmiş məlumatlara necə daxil olmağı və həmin məlumatları necə dəyişdirməyi öyrənəcəksiniz. Dictionary-lər demək olar ki, limitsiz miqdarda məlumat saxlaya bildiyi üçün, dictionary-dəki məlumatların üzərində necə **loop** qurmağı da göstərəcəyəm. Bundan əlavə, **dictionaries**-i **lists** içində yerləşdirməyi, **lists**-ləri **dictionary** içində istifadə etməyi və hətta dictionary-ləri digər dictionary-lərin içində iç-içə qurmağı öyrənəcəksiniz. Dictionary-ləri anlamaq sizə müxtəlif real dünya obyektlərini daha düzgün modelləşdirməyə imkan verir. Siz bir insanı təmsil edən dictionary yarada və həmin insan haqqında istədiyiniz qədər məlumat saxlaya biləcəksiniz. Onların adını, yaşını, məkanını, peşəsini və insan haqqında təsvir edə biləcəyiniz digər istənilən xüsusiyyəti saxlaya bilərsiniz. Bir-biri ilə uyğunlaşdırıla bilən istənilən iki məlumat növünü saxlaya biləcəksiniz — məsələn, sözlərin və mənalarının siyahısı, insanların adları və onların sevimli rəqəmləri, dağların siyahısı və onların yüksəklikləri və s.

A Simple Dictionary

Fərqli rənglərə və xal dəyərlərinə sahib ola bilən "alien"lər olan bir oyun təsəvvür edin. Bu sadə dictionary müəyyən bir alien haqqında məlumat saxlayır:

```
alien.py
alien_0 = {'color': 'green', 'points': 5}
print(alien_0['color'])
print(alien_0['points'])
```

`alien_0` dictionary-si alien-in rəngini və xal dəyərini saxlayır. Son iki sətr həmin məlumatlara daxil olur və onları göstərir.

Output:

```
green
5
```

Python-da dictionary **key-value pairs** kolleksiyasıdır.

Hər bir key müəyyən bir value ilə bağlıdır və həmin value-ya çatmaq üçün key-dən istifadə edə bilərsiniz. Key-in value-su bir rəqəm, string, list və ya hətta başqa bir dictionary ola bilər. Əslində, Python-da yarada biləcəyiniz istənilən obyekt dictionary-də value kimi istifadə oluna bilər. Python-da dictionary süslü mötərizələr `{}` içində yazılır və içərisində **key-value pairs** yerləşir, əvvəlki nümunədə olduğu kimi.

Key-value pair bir-biri ilə əlaqəli iki dəyərin cütüdür. Siz key verdikdə, Python həmin key-ə uyğun value-nu qaytarır. Hər bir key öz value-su ilə iki nöqtə `:` vasitəsilə bağlanır və hər key-value pair vergül ilə ayrılır. Dictionary-də istədiyiniz qədər key-value pair saxlaya bilərsiniz. Ən sadə dictionary-də cəmi bir key-value pair olur; `alien_0`-ın bu dəyişdirilmiş versiyasında olduğu kimi.
Key-ə uyğun value-nu əldə etmək üçün dictionary adını yazıb kvadrat mötərizələrdə həmin key-i göstərməlisiniz.

Əgər oyunçu bu alien-i vurarsa, aşağıdakı kodla neçə xal qazanacağını öyrənə bilərsiniz:

```
alien_0 = {'color': 'green', 'points': 5}
new_points = alien_0['points']
print(f"You just earned {new_points} points!")
```

Dictionary tərtib edildikdən sonra, `'points'` key-inə uyğun value- dictionary-dən çəkir. Bu value `new_points` dəyişəninə mənimsədilir.

Adding New Key-Value Pairs
Dictionaries dinamik strukturlardır və istənilən vaxt dictionary-yə yeni **key-value pair** əlavə edə bilərsiniz. Məsələn, yeni **key-value pair** əlavə etmək üçün dictionary-nin adını yazıb kvadrat mötərizələrdə yeni key-i göstərir və ona uyğun value təyin edirsiniz. Gəlin `alien_0` dictionary-sinə iki yeni məlumat əlavə edək: alien-in **x** və **y coordinates**-ləri, hansı ki onu ekranda müəyyən mövqedə göstərməyimizə kömək edəcək. Alien-i ekranın sol kənarında, yuxarıdan 25 piksel aşağıya yerləşdirək. Çünki ekran koordinatları adətən ekranın yuxarı-sol küncündən başlayır, biz alien-i sol kənara yerləşdirmək üçün **x coordinate = 0**, yuxarıdan 25 piksel aşağı yerləşdirmək üçün isə **y coordinate = 25** olaraq təyin edəcəyik, burada göstərildiyi kimi:

```
alien_0 = {'color': 'green', 'points': 5}
print(alien_0)
alien_0['x_position'] = 0
alien_0['y_position'] = 25
print(alien_0)
```

Əvvəlki nümunələrdən bildiyimiz həmin dictionary-ni müəyyən etməklə başlayırıq. Sonra dictionary-ni çap edirik ki, mövcud məlumatlar görünsün. Dəyişdirilmiş dictionary-ni çap edəndə yeni iki **key-value pair** görünür:

{'color': 'green', 'points': 5}
{'color': 'green', 'points': 5, 'x_position': 0, 'y_position': 25}

Dictionary-nin son versiyasında artıq dörd **key-value pair** var. İlk iki cütlük rəngi və xal dəyərini göstərir, son iki cütlük isə alien-in mövqeyini göstərir.

Starting with an Empty Dictionary
Bəzən boş bir **dictionary** ilə başlamaq və sonra hər bir **key-value pair**-i ayrıca əlavə etmək daha rahat olur. Boş dictionary-ni doldurmağa başlamaq üçün boş qıvrım mötərizələrlə dictionary yaradırsınız və sonra **key-value pair**-ləri bir-bir əlavə edirsiniz. Məsələn, `alien_0` dictionary-sini bu üsulla belə yarada bilərik:

```
alien_0 = {}
alien_0['color'] = 'green'
alien_0['points'] = 5
print(alien_0)
```

{'color': 'green', 'points': 5}

Modifying Values in a Dictionary
Dictionary-də value-u dəyişmək üçün dictionary-nin adını yazırsınız, kvadrat mötərizədə uyğun key-i göstərirsiniz və sonra həmin key üçün istədiyiniz yeni value-nu təyin edirsiniz. Gəlin alien-in rəngini dəyişək:

```
alien_0 = {'color': 'green'}
print(f"The alien is {alien_0['color']}.")
alien_0['color'] = 'yellow'
print(f"The alien is now {alien_0['color']}.")
```

The alien is green.
The alien is now yellow.

Tutaq ki, oyunda bir alien az əvvəl vurulub. Gəlin müxtəlif sürətlə hərəkət edə bilən alien-in mövqeyini izləyək. Alien-in cari sürətini göstərən bir value saxlayacağıq və ondan istifadə edərək, alien-in sağa nə qədər hərəkət edəcəyini müəyyən edəcəyik:

```
if alien_0['speed'] == 'slow':
```

```
        x_increment = 1
elif alien_0['speed'] == 'medium':
        x_increment = 2
else:
        # This must be a fast alien.
        x_increment = 3


# The new position is the old position plus the
increment.
alien_0['x_position'] = alien_0['x_position'] +
x_increment


print(f"New x-position: {alien_0['x_position']}")
```

Original x-position: 0

New x-position: 2

Əgər speed `'slow'`-dursa, **increment = 1**;
`'medium'`-dirsə, **increment = 2**;
`'fast'`-dırsa, **increment = 3** olur.
Sonra həmin increment-i mövcud x_position-a əlavə edirik ki, alien-in yeni mövqeyi alınsın.

Removing Key-Value Pairs

Dictionary-dən **key-value pair**-ləri də silə bilərsiniz. Bunu etmək üçün `del` statement-dən istifadə edirsiniz və dictionary adını, eləcə də silmək istədiyiniz

key-i göstərirsiniz. Məsələn, `alien_0` dictionary-sindən `'points'` **key-value pair**-ini silək:

```
alien_0 = {'color': 'green', 'points': 5}
print(alien_0)

del alien_0['points']
print(alien_0)
```

{'color': 'green', 'points': 5}
{'color': 'green'}

`del` istifadə edərkən diqqətli olun. Bir **key-value pair** silindikdən sonra tamamilə itir və əvvəlcədən saxlanmayıbsa bərpa edilə bilməz.

A Dictionary of Similar Objects
Əvvəlki nümunədə bir obyekt — oyundakı bir alien — haqqında müxtəlif məlumatların saxlanılması var idi. Dictionary-dən bir çox fərqli obyekt haqqında eyni tip məlumatı saxlamaq üçün də istifadə edə bilərsiniz. Məsələn, bir neçə insandan sorğu götürüb onların sevimli programming language-larını bilmək istədiyinizi düşünək.

```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
}
```

Gördüyünüz kimi, daha böyük dictionary-ni bir neçə sətrə bölmüşük. Hər bir key sorğuya cavab verən şəxsin adıdır, hər bir value isə onun seçdiyi programming language-dir. Bir neçə sətirdən ibarət dictionary yazarkən açılan `{` mötərizəsindən sonra Enter basın. Sonra növbəti sətri indent edin (adətən 4 boşluq) və ilk **key-value pair**-i yazın, sonda vergül qoyun. Mətn redaktorunuz növbəti **key-value pair**-ləri avtomatik olaraq eyni şəkildə indent edəcək. Bütün

key-value pair-ləri yazdıqdan sonra bağlanan } mötərizəsini yeni sətrdə, key-lərin indent səviyyəsinə uyğun yerləşdirin.

```
favorite_languages = {

   'jen': 'python',

   'sarah': 'c',

   'edward': 'ruby',

   'phil': 'python',

}



language = favorite_languages['sarah'].title()

print(f"Sarah's favorite language is {language}.")
```

Sarah's favorite language is C.

Using get() to Access Values

Dictionary-dən dəyər götürmək üçün kvadrat mötərizələrdə key istifadə etmək problem yarada bilər — əgər həmin key mövcud deyilsə.

Gəlin mövcud olmayan bir key-ə müraciət etdikdə nə baş verdiyinə baxaq. Burada `'points'` key-i olmayan bir alien dictionary-si var:

```
alien_0 = {'color': 'green', 'speed': 'slow'}
print(alien_0['points'])
```

Bu kod **KeyError** verərək traceback çıxarır:

```
Traceback (most recent call last):
  File "alien_no_points.py", line 2, in <module>
    print(alien_0['points'])
```

KeyError: 'points'

Xüsusilə dictionary-lərdə bu cür error-ların qarşısını almaq üçün **get()** metodundan istifadə edə bilərsiniz. **get()** metodu ilk arqument kimi key qəbul edir. İkinci (opsional) arqument kimi isə key mövcud olmadıqda qaytarılacaq **default value** verə bilərsiniz.

```
alien_0 = {'color': 'green', 'speed': 'slow'}

point_value = alien_0.get('points', 'No point value
assigned.')

print(point_value)
```

Əgər `'points'` key-i mövcuddursa, **get()** onun value-sunu qaytarır.
Əgər mövcud deyilsə, default mesaj qaytarılır. Əgər **get()** metodunda ikinci arqumenti yazmasanız və key mövcud olmazsa, Python **None** qaytarır.
None xüsusi bir value-dur və “dəyər yoxdur” deməkdir.
Bu bir error deyil — sadəcə boş nəticəni bildirir.

Looping Through a Dictionary

Python dictionary-si bir neçə və ya milyonlarla key-value pair saxlaya bilər.

Çünki dictionary çox məlumat saxlaya bilir, Python sizə dictionary üzərində **loop (dövr)** etməyə imkan verir.

Dictionaries müxtəlif yollarla məlumat saxlaya bildiyi üçün, onları loop etmək üçün bir neçə üsul mövcuddur:

- bütün key-value pair-ləri üzrə
- yalnız keys üzrə
- yalnız values üzrə

Bu dictionary-də bir istifadəçinin username, first name və last name saxlanır.

```
user_0 = {
    'username': 'efermi',
    'first': 'enrico',
    'last': 'fermi',
}
```

Əgər bu istifadəçinin bütün məlumatını görmək istəsək, **for loop** istifadə edə bilərik:

```
user_0 = {
    'username': 'efermi',
    'first': 'enrico',
    'last': 'fermi',
}

for key, value in user_0.items():
    print(f"\nKey: {key}")
    print(f"Value: {value}")
```

`items()` metodu bütün key-value pair-ləri qaytarır.
For loop hər pair-i `key` və `value` dəyişənlərinə təyin edir.

Output:
Key: username
Value: efermi

Key: first
Value: enrico

Key: last
Value: fermi

Daha deskriptiv dəyişən adları (`name`, `language`) loop-un içində nə baş verdiyini anlamağı asanlaşdırır.

```
favorite_languages = {
    'jen': 'python',
```

```
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
}

for name, language in favorite_languages.items():
    print(f"{name.title()}'s favorite language is
{language.title()}.")
```

Looping Through All Keys in a Dictionary

Yalnız keys-lərlə işləmək istəsək, `keys()` metodundan istifadə edirik:

```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
}
for name in favorite_languages.keys():
    print(name.title())
```

Looping Through Keys in a Particular Order

```
favorite_languages = {
    'jen': 'python',
    'sarah': 'c',
    'edward': 'ruby',
    'phil': 'python',
}
for name in sorted(favorite_languages.keys()):
    print(f"{name.title()}, thank you for taking the
poll.")
```

Looping Through All Values in a Dictionary

Yalnız values-ləri maraqlıdırsa, `values()` metodundan istifadə edirik:

```
favorite_languages = {
   'jen': 'python',
   'sarah': 'c',
   'edward': 'ruby',
   'phil': 'python',
}
print("The following languages have been mentioned:")
for language in favorite_languages.values():
   print(language.title())
```

Duplicate-ləri çıxarmaq üçün **set()** istifadə edə bilərik:

```
favorite_languages = {
   'jen': 'python',
   'sarah': 'c',
   'edward': 'ruby',
   'phil': 'python',
}
for language in set(favorite_languages.values()):
   print(language.title())
```

Nesting (Daxildə Daxil) – Yekun Qaydalar və Nümunələr

List içində Dictionary

```
alien_0 = {'color': 'green', 'points': 5}
alien_1 = {'color': 'yellow', 'points': 10}
alien_2 = {'color': 'red', 'points': 15}

aliens = [alien_0, alien_1, alien_2]

for alien in aliens:
   print(alien)
```

{'color': 'green', 'points': 5}
{'color': 'yellow', 'points': 10}
{'color': 'red', 'points': 15}

Daha böyük miqyas üçün loop istifadə edərək 30 alien yaratmaq:

```
aliens = []

for alien_number in range(30):
    new_alien = {'color': 'green', 'points': 5,
'speed': 'slow'}
    aliens.append(new_alien)

print(f"Total aliens: {len(aliens)}")   # 30
```

Total aliens: 30

Dictionary içində List

```
pizza = {
    'crust': 'thick',
    'toppings': ['mushrooms', 'extra cheese']
}

print(f"You ordered a {pizza['crust']}-crust pizza
with toppings:")
for topping in pizza['toppings']:
    print("\t" + topping)
```

You ordered a thick-crust pizza with toppings:

mushrooms
extra cheese

Dictionary içində Dictionary

```
users = {
    'aeinstein': {'first': 'albert', 'last':
'einstein', 'location': 'princeton'},
    'mcurie': {'first': 'marie', 'last': 'curie',
'location': 'paris'},
}

for username, user_info in users.items():
    print(f"\nUsername: {username}")
```

```
    full_name = f"{user_info['first']}
{user_info['last']}"
    location = user_info['location']
    print(f"\tFull name: {full_name.title()}")
    print(f"\tLocation: {location.title()}")
```

Username: aeinstein
Full name: Albert Einstein
Location: Princeton

Username: mcurie
Full name: Marie Curie
Location: Paris