

IF Statements

Programlaşdırma çox vaxt bir **şərtlər dəstini yoxlamaq** və həmin şərtlərə əsaslanaraq hansı əməliyyatı yerinə yetirəcəyinizi qərar vermək deməkdir.

Python-un if bəyanatı sizə programın cari vəziyyətini yoxlamağa və ona uyğun şəkildə cavab verməyə imkan verir.

Bu fəsildə siz:

- **Şərti testlər yazmayı** öyrənəcəksiniz. Bu testlər istənilən şərtin doğru olub-olmadığını yoxlamağa imkan verir.
- **Sadə if bəyanatları** yazmayı öyrənəcəksiniz.
- Daha mürəkkəb if bəyanatları zənciri yaratmayı öyrənəcəksiniz, beləliklə də, istədiyiniz dəqiq şərtlər mövcud olduqda onları tanıya biləcəksiniz.
- Bu anlayışı **list-lərə tətbiq edəcəksiniz**, beləliklə bir for döngüsü ilə listdəki əksər elementləri bir üsulla, amma müəyyən dəyərlərə malik elementləri isə fərqli üsulla emal edə biləcəksiniz.

A Simple Example

Tutaq ki, avtomobilərin adlarını saxladığınız bir listiniz var və hər birini çap etmək istəyirsiniz.

- Avtomobil adları düzgün formada (title case) çap olunmalıdır.
- Amma 'bmw' üçün hamısı böyük hərflərlə (uppercase) çap olunmalıdır.

```
cars = ['audi', 'bmw', 'subaru', 'toyota']
```

```
for car in cars:
```

```
    if car == 'bmw':
```

```
        print(car.upper())
```

```
    else:
```

```
        print(car.title())
```

Bu nümunədə `if` şərti 'bmw' -ni yoxlayır. Əgər car 'bmw' -dirsə, hamısı böyük hərflərlə çap olunur, əks halda `title()` ilə çap olunur.

Conditional Tests

Hər bir `if` bəyanatının əsasında **True (doğru)** və **False (yalnız)** qaytara bilən bir ifadə dayanır.

- Əgər test **True** qaytarırsa, `if` blokundakı kod icra olunur.
- Əgər test **False**-dursa, kod icra olunmur.

Checking for Equality

`=` → dəyişənə dəyər təyin edir

`==` → yoxlayır, bərabərdirmi?

Inequality

```
requested_topping = 'mushrooms'  
if requested_topping != 'anchovies':  
    print("Hold the anchovies!")
```

`!=` → bərabər deyilsə **True** qaytarır

and, or → birdən çox şərti birləşdirir

```
age_0 = 22  
age_1 = 18
```

```
age_0 >= 21 and age_1 >= 21 # False  
age_0 >= 21 or age_1 >= 21 # True
```

Listdəki Dəyəri Yoxlamaq

```
requested_toppings = ['mushrooms', 'onions', 'pineapple']
```

```
'mushrooms' in requested_toppings # True
```

```
'pepperoni' in requested_toppings # False
```

Listdə Dəyərin Olmamasını Yoxlamaq

```
banned_users = ['andrew', 'carolina', 'david']
```

```
user = 'marie'
```

```
if user not in banned_users:
```

```
    print(f"{user.title()}, you can post a response if you wish.")
```

Boolean Dəyərlər (True / False)

```
game_active = True
```

```
can_edit = False
```

- **True** → doğru
- **False** → yanlış

Boolean ifadələr və şərtlər ilə programın axarını idarə etmək mümkündür.

if Statements

Sadə if statement

```
age = 19
```

```
if age >= 18:
```

```
    print("You are old enough to vote!")
```

```
    print("Have you registered to vote yet?")
```

Şərt age ≥ 18 doğru olduğu üçün bütün indented (girintili) print() çağrıları işləyir.

Əgər age < 18 olsaydı, heç bir şey çap olunmazdı.

if-else statement

```
age = 17

if age >= 18:

    print("You are old enough to vote!")

    print("Have you registered to vote yet?")

else:

    print("Sorry, you are too young to vote.")

    print("Please register to vote as soon as you turn 18!")
```

Izah:

- if şərti doğru olsa if bloku işləyir.
- Əgər if şərti yanlışdırsa (age < 18) else bloku işləyir.
- Hər zaman bir blok mütləq işləyir, iki mümkün haldan biri.

if-elif-else chain

```
age = 12
```

```
if age < 4:
```

```
    price = 0
```

```
elif age < 18:
```

```
    price = 25
```

```
elif age < 65:
```

```
    price = 40  
else:  
    price = 20  
  
print(f"Your admission cost is ${price}.")
```

İzah:

- Python testləri yuxarıdan aşağıya oxuyur.
- İlk doğru şərti tapanda qalan `elif` və `else` bloklarını atlayır.
- Bu üsul yalnız bir blokun işləməsini təmin edir.

Daha aydın if-elif chain

```
age = 70  
  
if age < 4:  
    price = 0  
elif age < 18:  
    price = 25  
elif age < 65:  
    price = 40  
elif age >= 65:  
    price = 20 # daha aydın, spesifik blok  
  
print(f"Your admission cost is ${price}.")
```

İzah:

- Son **elif** daha aydın gösterir ki, yalnız 65 və yuxarı yaş üçün işləyir.
- **else** olmadan da yazıla bilər, amma o, **catchall** (ümumi hal) kimi çıkış edir.

Bir neçə müstəqil if statement

```
requested_toppings = ['mushrooms', 'extra cheese']
```

```
if 'mushrooms' in requested_toppings:  
    print("Adding mushrooms.")  
  
if 'pepperoni' in requested_toppings:  
    print("Adding pepperoni.")  
  
if 'extra cheese' in requested_toppings:  
    print("Adding extra cheese.")  
  
  
print("\nFinished making your pizza!")
```

İzah:

- Hər bir if müstəqildir.
- Müştərinin hər topping-i yoxlanılır və uyğun mesaj çap olunur.
- **if-elif-else** burada işləməzdi, çünki yalnız birinci keçən şərt işləyərdi.

Qısa xülasə

1. **Sadə if** → yalnız bir şərt, doğru olarsa işləyir.
2. **if-else** → iki mümkün hal üçün işləyir.

3. **if-elif-else** → yuxarıdan aşağı test edir, yalnız bir blok işləyir.
4. **Bir neçə if** → hər blok müstəqil işləyir, çoxlu doğru halları icra etmək üçün.

Sadə list və for loop

```
requested_toppings = ['mushrooms', 'green peppers', 'extra cheese']
```

```
for requested_topping in requested_toppings:  
    print(f"Adding {requested_topping}.")  
  
print("\nFinished making your pizza!")
```

İzah:

- Hər topping sadəcə for döngüsü ilə çap olunur.
- Çıxış ardıcılılığı sadədir: bütün elementlər ardıcıl əlavə olunur.

Xüsusi element üçün if

```
requested_toppings = ['mushrooms', 'green peppers', 'extra cheese']
```

```
for requested_topping in requested_toppings:  
    if requested_topping == 'green peppers':  
        print("Sorry, we are out of green peppers right now.")  
  
    else:  
        print(f"Adding {requested_topping}.")
```

İzah:

- **if** şərti xüsusi halları idarə edir (green peppers).
- **else** isə digər bütün hallarda işləyir.

Output:

Adding mushrooms.

Sorry, we are out of green peppers right now.

Adding extra cheese.

Listin boş olub olmadığını yoxlamaq

```
requested_toppings = []
```

```
if requested_toppings:
```

```
    for requested_topping in requested_toppings:
```

```
        print(f"Adding {requested_topping}.")
```

```
    print("\nFinished making your pizza!")
```

```
else:
```

```
    print("Are you sure you want a plain pizza?")
```

İzah:

- Boş list False kimi qiymətləndirilir.
- Əgər list boşdursa, else bloku işləyir və istifadəçiyə xəbərdarlıq göstərilir.

Output:

Are you sure you want a plain pizza?

Mövcud və istək siyahılarını müqayisə etmək

```
available_toppings = ['mushrooms', 'olives', 'green peppers', 'pepperoni',  
'pineapple', 'extra cheese']
```

```
requested_toppings = ['mushrooms', 'french fries', 'extra cheese']
```

```
for requested_topping in requested_toppings:  
    if requested_topping in available_toppings:  
        print(f"Adding {requested_topping}.")  
  
    else:  
        print(f"Sorry, we don't have {requested_topping}.")  
  
print("\nFinished making your pizza!")
```

İzah:

- `if requested_topping in available_toppings:` → topping mövcuddursa əlavə olunur.
- `else` → topping yoxdursa xəbərdarlıq verir.

Summary

1. **For loop + if:** xüsusi halları yoxlamaq üçün istifadə olunur.
2. **Boş list yoxlaması:** `if list_name :` → listdə element varsa True, yoxsa False.
3. **Mövcud və istək siyahılarının müqayisəsi:** real dünya tətbiqləri üçün vacibdir (məsələn, restoran menyusu).
4. **Müstəqil if blokları:** hər bir şərti ayrıca yoxlayır və lazımlı olan mesajı çap edir.

Conditional Tests

- Conditional tests evaluate to **True** or **False**.
- Common types of tests:
 - **Equality:** `==`
 - **Inequality:** `!=`

- **Numerical comparisons:** `<`, `<=`, `>`, `>=`
- **Membership in a list:** `in`, `not in`
- You can make tests **case-insensitive** with `.lower()`.