

UFUK ÜNİVERSİTESİ · BİLGİSAYAR MÜHENDİSLİĞİ · GÜZ DÖNEMİ

MAT201 · 4. Hafta

“Kişiye Bir Defa Verilir ve Değiştirilemez”

Fonksiyonlar: Birebir, Örten, Ters, Bileşke, Taban ve Tavan

Dr. Alaettin Uçan

Rosen, Bölüm 2.3



4. HAFTA · BUGÜNÜN AKIŞI

Bugün dört bölümde ilerleyeceğiz

1

Fonksiyon nedir?

Her girdiye tek bir çıktı: kimlik numarasından alan kodlarına

2

Birebir ve örten fonksiyonlar

İki girdi aynı çıktıya gidiyor mu? Değer kümesinde açıkta eleman kalıyor mu?

3

Ters fonksiyon ve bileşke

Geri dönebilir miyiz? Yönetmelikteki not hesabı hangi fonksiyonlardan oluşuyor?

4

Taban ve tavan

Bir notu yuvarlamak neden bu kadar önemli?

Sorularınızı istediğiniz an Teams sohbetine yazabilirsiniz.

AÇILIŞ



“Türkiye Cumhuriyeti Kimlik Numarası kişiye bir defa verilir ve değiştirilemez.”

Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü kimlik numarasını böyle anlatıyor.

Kaynak: Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü, “T.C. Kimlik No İşlemleri” sayfası

AÇILIŞ



“Son iki rakamı doğrulama sayısıdır. Bu son iki basamak ilk dokuz basamaktan bir algoritma ile hesaplanmaktadır.”

Aynı sayfanın devamı.

Kaynak: Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü, “T.C. Kimlik No İşlemleri” sayfası

AÇILIŞ



Kimlik numaranızın son rakamı tek mi, çift mi?

Numaranızı yazmayın. Yalnızca son rakamın tek mi çift mi olduğunu sohbete yazın.

AÇILIŞ



Geçerli her kimlik numarasının son rakamı çifttir.

Bu bir tesadüf değil. Nedenini 2. bölümün sonunda, bugün öğreneceğimiz kavramlarla göstereceğiz.



1. BÖLÜM

Fonksiyon Nedir?

Her girdiye tek bir çıktı

1. BÖLÜM • FONKSİYON NEDİR?



Geçen hafta kümeleri gördük. Bu hafta kümeler arasındaki eşlemelere bakacağız.



“A'nın her elemanını B'nin bir ve yalnız bir elemanına eşleyen f bağıntısına A'dan B'ye bir fonksiyon denir.”

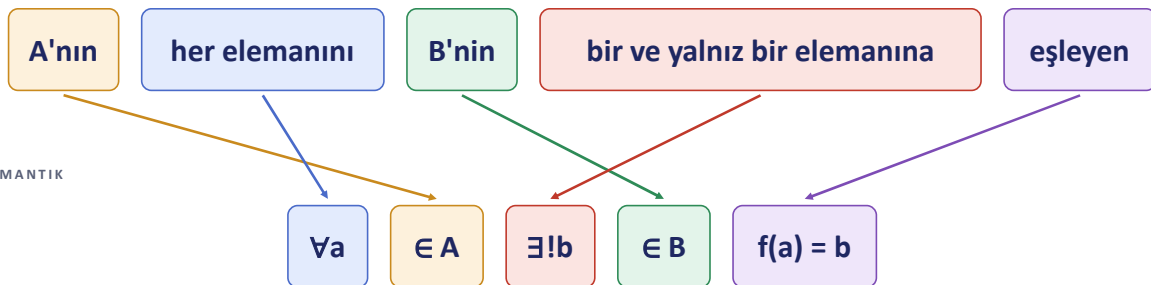
Lisede öğrendiğiniz tanım. Rosen'daki tanım da aynı şeyi söylüyor.

Kaynak: eokultv, 10. Sınıf “Fonksiyonun Tanımı” konu anlatımı



Tanımı formüle çevirelim

TÜRKÇE



$$\forall a \in A \exists ! b \in B, f(a) = b$$

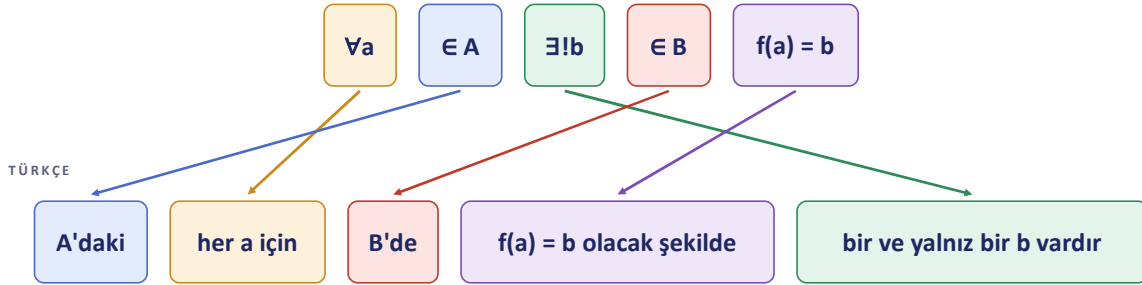
Geçen hafta gördüğümüz $\exists!$ (“bir ve yalnız bir”) niceleyicisi burada karşımıza çıkıyor.

Kaynak: eokultv, 10. Sınıf “Fonksiyonun Tanımı” konu anlatımı



Formülü geri okuyalım

MANTIK



"A'daki her a için, B'de $f(a) = b$ olacak şekilde bir ve yalnız bir b vardır."



Bu tanımın içinde iki ayrı şart var.

- 1) A'nın her elemanı bir elemana eşlenmeli.
- 2) A'nın hiçbir elemanı iki farklı elemana eşlenmemeli.



∃! iki parçadan oluşur

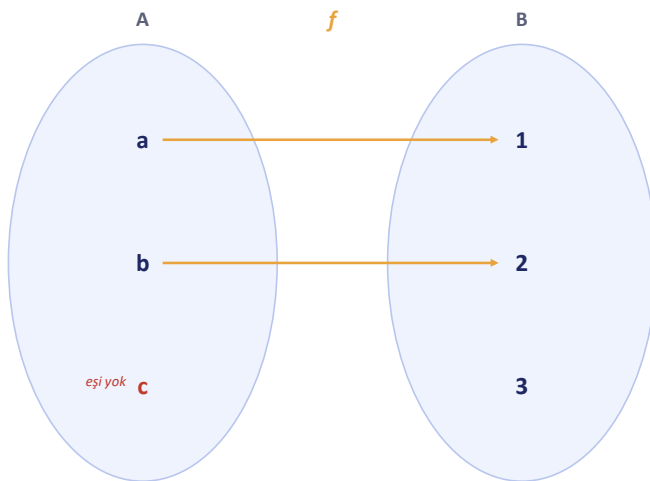
$$\exists b f(a) = b \wedge \forall b \forall c (f(a) = b \wedge f(a) = c \rightarrow b = c)$$

Birinci parça “en az bir”, ikinci parça “en fazla bir” der.
İkisi birlikte “bir ve yalnız bir” demektir.

Kaynak: Rosen, Bölüm 1.4



Birinci şart bozulursa

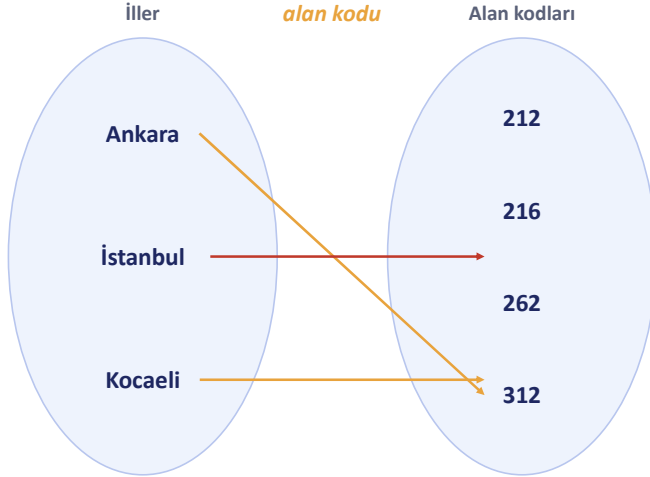


Fonksiyon değil

c'nin eşi yok. Tanım, A'nın her elemanının eşlenmesini istiyor.



İkinci şart bozulursa



Fonksiyon değil

İstanbul iki farklı koda eşleniyor:
Avrupa yakası 212, Anadolu yakası 216.

Kaynak: Vodafone, "Türkiye Telefon Numaralandırma Rehberi ve Şehir Alan Kodları"



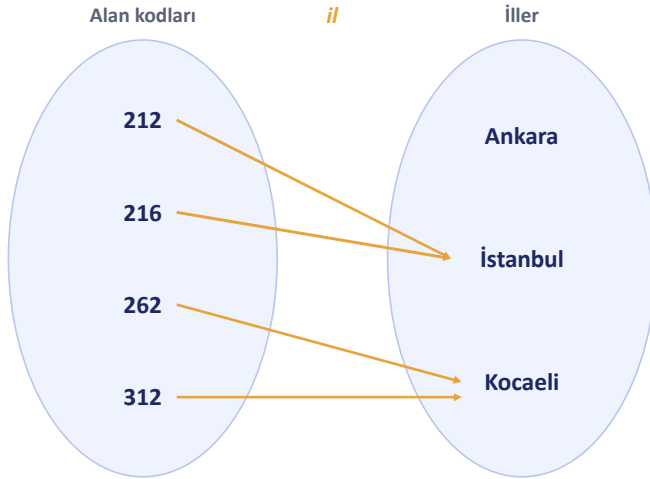
“İstanbul'un alan kodu nedir?” sorusunun tek bir cevabı yoktur.

Bu yüzden “il → alan kodu” eşlemesi bir bağıntıdır ama fonksiyon değildir.

Kaynak: Vodafone, "Türkiye Telefon Numaralandırma Rehberi ve Şehir Alan Kodları"



Okların yönünü çevirelim



Kaynak: Vodafone, "Türkiye Telefon Numaralandırma Rehberi ve Şehir Alan Kodları"

Fonksiyon

Her alan kodu tek bir ile ait. İki kodun aynı ile gitmesi tanımı bozmaz.

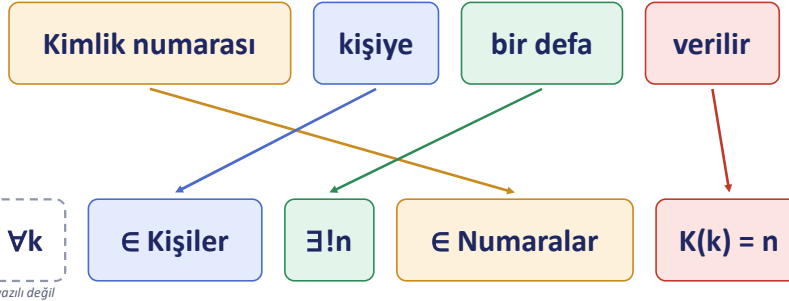


Bir fonksiyonda iki ok aynı yere varabilir; ama bir elemandan iki ok çıkamaz.



NVİ'nin cümlesi de bir fonksiyon tarif ediyor

TÜRKÇE



MANTIK

K: Kişiler $\rightarrow$ Numaralar bir fonksiyondur.

"Her" sözcüğü yine yazılı değil: cümle her kişi için geçerli.

Kaynak: Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü, "T.C. Kimlik No İşlemleri" sayfası



"Değiştirilemez" sözcüğü de önemlidir.

Aynı kişiye bugün bir numara, yarın başka bir numara verilemez. Fonksiyonun çıktısı zamana göre değişmez.

Kaynak: Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü, "T.C. Kimlik No İşlemleri" sayfası



Aynı girdiye her seferinde aynı çıktıyı veren kod, matematiksel anlamda bir fonksiyondur.



Python'daki her fonksiyon, matematikteki fonksiyon değildir

```
pure.py

import random

def roll_die():
    return random.randint(1, 6)

def check_digits(first9):
    ... # computed only from first9
```

roll_die

Aynı çağrı farklı sonuçlar verir. Matematiksel anlamda fonksiyon değil.

check_digits

İlk dokuz basamak aynıysa doğrulama sayısı da hep aynıdır. Matematiksel bir fonksiyon.



Bu ayrım yazılımda doğrudan işe yarar.

Matematiksel bir fonksiyonun sonucunu bir kez hesaplayıp saklayabilir, sonraki çağrılarda yeniden hesaplamadan kullanabilirsiniz. Buna önbelleğe alma (caching) denir.



Önbellek, fonksiyon tanımına güvenir

```
cache.py

from functools import lru_cache

@lru_cache
def check_digits(first9):
    ...

check_digits("100000001") # computed
check_digits("100000001") # returned from the cache
```

@lru_cache satırını roll_die'in üstüne koyarsanız zar hep ilk attığınız sayıyı gösterir.



Her fonksiyonun üç kümesi vardır: tanım kümesi, değer kümesi ve görüntü kümesi.



Yazılış

$$f : A \rightarrow B$$

A tanım kümesi, B değer kümesidir.
“f, A'dan B'ye bir fonksiyondur” diye okunur.



“A tanım kümesinin tüm elemanlarının görüntülerinin oluşturduğu kümeye fonksiyonun görüntü kümesi denir ve $f(A)$ ile gösterilir.”

Kaynak: derspresso, “Fonksiyonların Tanım ve Görüntü Kümesi” konu anlatımı



Türkçeden mantığa

TÜRKÇE



MANTIK



$$b \in f(A) \Leftrightarrow \exists a \in A, f(a) = b$$

Cümlede “tüm” yazıyor ama formülde $\exists$ var: b 'nin görüntü kümesinde olması için onu veren bir a 'nın bulunması yeter.

Kaynak: derspresso, “Fonksiyonların Tanım ve Görüntü Kümesi” konu anlatımı



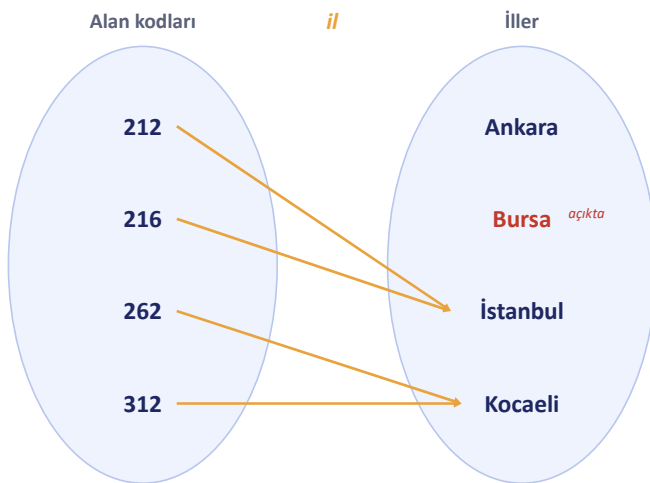
Görüntü kümesi değer kümesinin bir alt kümesidir.

$f(A) \subseteq B$. İkisi eşit olmak zorunda değil.

Kaynak: derspresso, "Fonksiyonların Tanım ve Görüntü Kümesi" konu anlatımı



Üç küme bir arada



Üç küme

Tanım kümesi: {212, 216, 262, 312}

Değer kümesi: {Ankara, Bursa, İstanbul, Kocaeli}

Görüntü kümesi: {Ankara, İstanbul, Kocaeli}

Kaynak: Vodafone, "Türkiye Telefon Numaralandırma Rehberi ve Şehir Alan Kodları"



Değer kümesini siz seçersiniz; görüntü kümesi ise fonksiyonun kendisinden çıkar.

Programlamadaki karşılığı: int döndüren bir fonksiyonun değer kümesi tam sayılardır, ama her tam sayıyı döndürmek zorunda değildir.



Bir fonksiyon, $A \times B$ 'nin bir alt kümesi olarak da düşünülebilir.

Geçen hafta gördüğümüz Kartezyen çarpımı hatırlayın. f 'nin grafiği, $(a, f(a))$ ikililerinden oluşan kümedir.



Bir fonksiyonun grafiği

$$\{ (a, f(a)) \mid a \in A \} \subseteq A \times B$$

Alan kodu fonksiyonu için:
 $\{ (212, \text{İstanbul}), (216, \text{İstanbul}), (262, \text{Kocaeli}), (312, \text{Ankara}) \}$

Kaynak: Rosen, Bölüm 2.3



Python'daki sözlük (dict), tam olarak böyle bir ikili kümesidir.



Bir sözlük, bir fonksiyonun grafiğini saklar

```
area_codes.py

city_of = {212: "İstanbul", 216: "İstanbul",
           262: "Kocaeli", 312: "Ankara"}

city_of[312]          # 'Ankara'
set(city_of)          # domain: {212, 216, 262, 312}
set(city_of.values()) # image:  {'İstanbul', 'Kocaeli', 'Ankara'}
```

Anahtarlar tanım kümesini, değerler görüntü kümesini oluşturur.



Bir sözlükte bir anahtarın yalnızca bir değeri olabilir.

Yani sözlük, yapısı gereği bir fonksiyondur. Tanımdaki ikinci şartı Python bizim yerimize sağlar.



Fonksiyon olmayan bir eşlemeyi sözlüğe koymaya çalışalım

```
area_codes.py

area_code = {"Ankara": 312, "İstanbul": 212, "İstanbul": 216}

print(area_code)
# {'Ankara': 312, 'İstanbul': 216}
```

Python hata vermiyor. İstanbul'un ilk değerini sessizce siliyor.



Çözüm: değer kümesini değiştirmek.

Her ili bir kod listesine eşleyelim: {"Ankara": [312], "İstanbul": [212, 216]}

Artık her il tek bir değere, bir listeye eşleniyor. Bu eşleme bir fonksiyon.



A'dan B'ye kaç farklı fonksiyon tanımlanabilir?

$|A| = n$ ve $|B| = m$ olsun. A'nın her elemanı için m seçenek var.



Fonksiyon sayısı

$$|A| = n, |B| = m \Rightarrow m^n \text{ fonksiyon}$$

A'nın n elemanının her biri, diğerlerinden bağımsız olarak m seçenekten birine gider:

$$m \cdot m \cdot \dots \cdot m = m^n$$



Geçen haftaki doğruluk tabloları da birer fonksiyondur.

İki önermeyle kurulan bir bağlaç, $\{D, Y\} \times \{D, Y\}$ kümesinden $\{D, Y\}$ kümesine bir fonksiyondur.



Kaç farklı iki değişkenli bağlaç var?

$$2^4 = 16$$

Tanım kümesinde 4 eleman (DD, DY, YD, YY), değer kümesinde 2 eleman (D, Y) var.
Geçen hafta bunların yalnızca birkaçına ad verdik: $\wedge$, $\vee$, $\oplus$, $\rightarrow$ ve $\leftrightarrow$.



2. BÖLÜM

Birebir ve Örten Fonksiyonlar

İki girdi aynı çıktıya gidiyor mu? Değer kümesinde açıkta eleman kalıyor mu?

2. BÖLÜM • BİREBİR VE ÖRTEN



“A'nın farklı elemanlarının görüntüleri de farklı ise, bu fonksiyona birebir fonksiyon denir.”

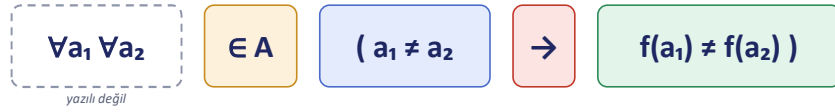


Türkçeden mantığa

TÜRKÇE



MANTIK



$$\forall a_1 \forall a_2 \in A (a_1 \neq a_2 \rightarrow f(a_1) \neq f(a_2))$$

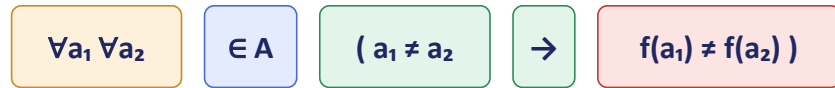
Gizli bir "her" daha: tanım, A'dan seçilen her iki eleman için geçerli.

Kaynak: Ankara Üniversitesi Açık Ders Matzemeleri, "Fonksiyonlar" ders notu



Formülü geri okuyalım

MANTIK



TÜRKÇE



"A'dan seçilen her iki eleman farklıysa, görüntüleri de farklıdır."



Bu tanımı ispatlarda genellikle ters yönden kullanırız.

Geçen hafta gördüğümüz karşıt ters burada işe yarıyor: $p \rightarrow q$ ile $\neg q \rightarrow \neg p$ denktir.



Birebirliğin karşıt tersi

$$f(a_1) = f(a_2) \rightarrow a_1 = a_2$$

“Görüntüleri aynı olan iki eleman, aslında aynı elemandır.”
Birebirliği ispatlarken bu biçimi kullanırız.



Bir ispat: $f(x) = 3x + 1$, $\mathbb{R}'$ 'den $\mathbb{R}$ 'ye birebirdir

$$f(a_1) = f(a_2)$$

Görüntüleri eşit iki eleman alalım



$$3a_1 + 1 = 3a_2 + 1$$

f 'nin tanımı



$$3a_1 = 3a_2$$

İki taraftan 1 çıkaralım



$$a_1 = a_2$$

İki tarafı 3'e bölelim

Görüntüleri eşit olan iki elemanın aynı eleman olduğunu gösterdik. f birebirdir.

Kaynak: Ders notu 7-Fonksiyonlar



Bir fonksiyonun birebir olmadığını göstermek için tek bir karşı örnek yeter.

Tanımın değerini alalım. Geçen haftanın kuralına göre $\forall$ yerine $\exists$ gelir.



Birebirliğin değili

$$\neg \forall a_1 \forall a_2 (a_1 \neq a_2 \rightarrow f(a_1) \neq f(a_2))$$

Başlangıç

$$\exists a_1 \exists a_2 \neg (a_1 \neq a_2 \rightarrow f(a_1) \neq f(a_2))$$

Değilleme kuralı, iki kez

$$\exists a_1 \exists a_2 (a_1 \neq a_2 \wedge f(a_1) = f(a_2))$$

 $\neg(p \rightarrow q) \equiv p \wedge \neg q$

Görüntüleri aynı olan iki farklı eleman vardır.



Mantıktan Türkçeye

MANTIK



TÜRKÇE



“Görüntüleri aynı olan iki farklı eleman vardır.”

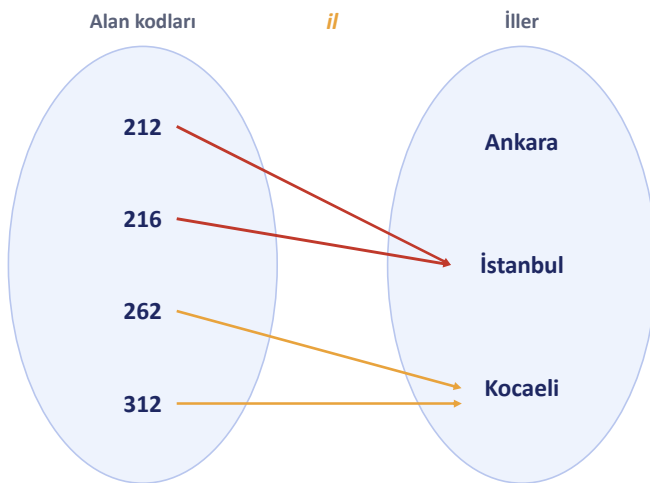


Yazılımda bu duruma çakışma (collision) denir.

İki farklı girdi aynı çıktıyı veriyorsa, o fonksiyon birebir değildir.



Alan kodu fonksiyonu birebir mi?



Birebir değil

212 ve 216 farklı kodlar, ama ikisinin görüntüsü de İstanbul. Bu bir çakışma.



Yönetmeliğimizdeki not tablosu

Puan aralığı	Harf notu	Katsayı
90–100	AA	4,00
80–89	BA	3,50
70–79	BB	3,00
65–69	CB	2,50
60–64	CC	2,00
55–59	DC	1,50
50–54	DD	1,00
0–49	FF	0,00

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 42



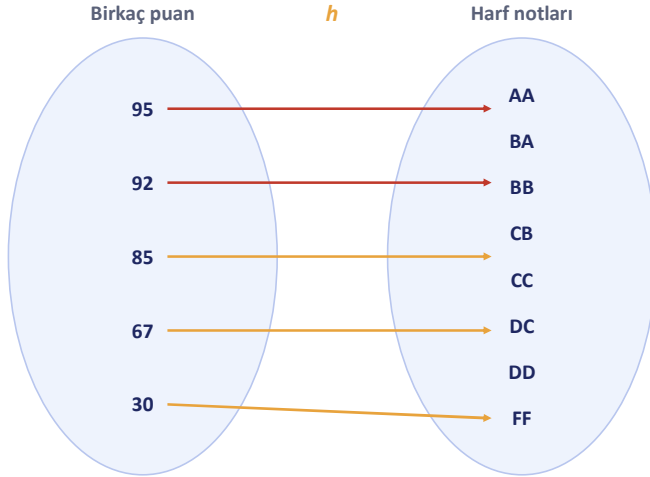
Bu tablo aslında iki fonksiyondur.

h: Puanlar → Harf notları

k: Harf notları → Katsayılar



h: puandan harf notuna



Birebir değil

95 ve 92 farklı puanlar, ama ikisi de AA oluyor. “AA aldım” diyen birinin puanını bilemezsiniz.

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 42

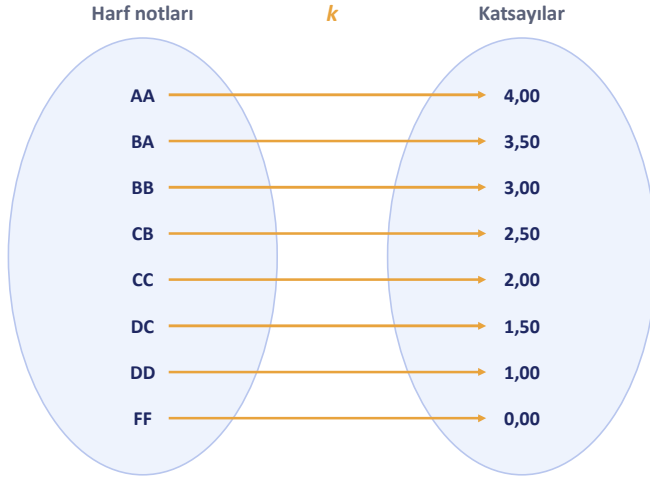


Birebir olmayan bir fonksiyon bilgi kaybeder.

Harf notuna bakarak puanı geri bulamazsınız. Bu kayıp bazen istenen bir şeydir: transkriptte puan değil harf notu görünür.



k: harf notundan katsayıya



Birebir

Her harf notunun katsayısı farklı.
Katsayıya bakarak harf notunu kesin olarak bulabilirsiniz.

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 42



“B'nin hiçbir elemanı açıkta kalmayacak şekilde eşleştiyorsa f fonksiyonuna A'dan B'ye örten fonksiyon denir.”

Kaynak: Ankara Üniversitesi Açık Ders Malzemeleri, “Fonksiyonlar” ders notu



Türkçeden mantığa

TÜRKÇE



MANTIK



$$\neg \exists b \in B, \text{Açıkta}(b)$$

Geçen hafta gördüğümüz gibi “hiçbir” sözcüğü $\neg \exists$ demektir.

Kaynak: Ankara Üniversitesi Açık Ders Matzemeleri, “Fonksiyonlar” ders notu



“Açıkta kalmak” ne demek?

b açıktaysa, A 'daki hiçbir elemanın görüntüsü değildir:

$\text{Açıkta}(b) \equiv \forall a \in A, f(a) \neq b$



“Hiçbir ... açıkta kalmayacak” cümlesini sadeleştirelim

$$\neg \exists b \text{ Açıkta}(b)$$

Tanımın formülü



$$\forall b \neg \text{Açıkta}(b)$$

Değilleme kuralı



$$\forall b \neg \forall a f(a) \neq b$$

Açıkta(b)'nin tanımı



$$\forall b \exists a f(a) = b$$

Değilleme kuralı bir kez daha

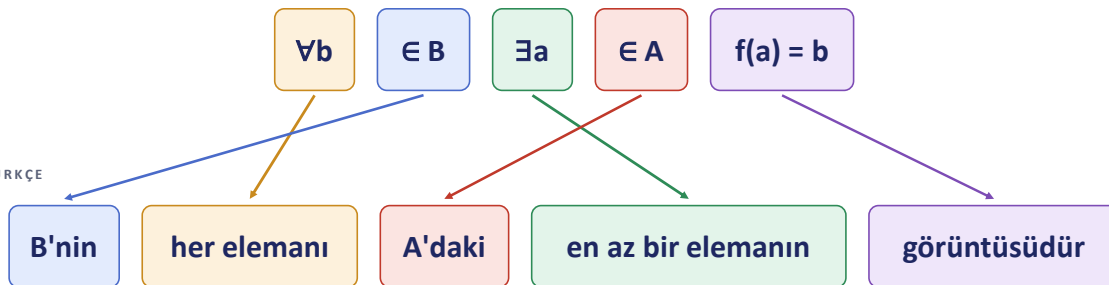
Örten: $\forall b \in B \exists a \in A, f(a) = b$



Formülü geri okuyalım

MANTIK

TÜRKÇE



“B'nin her elemanı, A'daki en az bir elemanın görüntüsüdür.”

Bu, “görüntü kümesi değer kümesine eşittir” demenin başka bir yolu: $f(A) = B$.



Örten olmayan fonksiyona lisede “içine fonksiyon” denir.

“B kümesinde en az bir eleman açıkta kalacak şekilde ... eşleşiyorsa ... içine fonksiyon denir.”

Kaynak: Ankara Üniversitesi Açık Ders Matzemeleri, “Fonksiyonlar” ders notu



Örten tanımının değili

$$\neg \forall b \exists a f(a) = b$$

Örten değil



$$\exists b \neg \exists a f(a) = b$$

Değilleme kuralı



$$\exists b \forall a f(a) \neq b$$

Değilleme kuralı

“B’de açıkta kalan en az bir eleman vardır.” Lisedeki içine fonksiyon tanımı tam olarak bu.



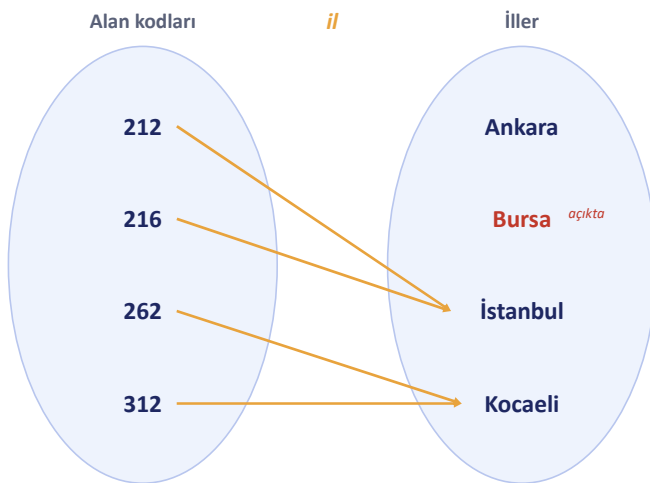
h, yani puandan harf notuna giden fonksiyon, örtendir.

Her harf notunu veren en az bir puan var: AA için 95, BA için 85, ... FF için 30. Tablodaki her satır boş olmayan bir puan aralığı.

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 42



Değer kümesine Bursa'yı eklersek



Örten değil

Bursa açıkta kaldı: tanım kümesinde Bursa'nın alan kodu yok.

Kaynak: Vodafone, "Türkiye Telefon Numaralandırma Rehberi ve Şehir Alan Kodları"



Bir fonksiyonun örten olup olmaması, değer kümesinin nasıl seçildiğine bağlıdır.

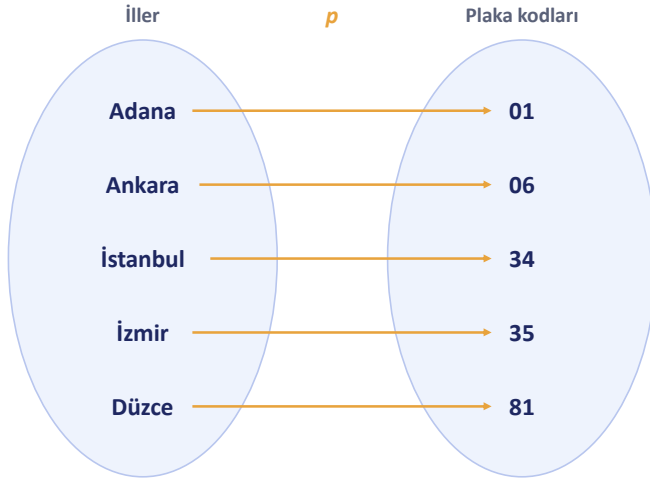


Hem birebir hem örten olan fonksiyona birebir ve örten fonksiyon denir.

Bu durumda A ile B'nin elemanları tek tek eşleşir; iki kümenin eleman sayısı da eşittir.



İl plaka kodları



Birebir ve örten

Her ilin tek bir plaka kodu var, her kodun da tek bir ili. 81 il, 01'den 81'e kadar 81 koda eşleniyor.

Kaynak: Türkiye il trafik (plaka) kodları



Örneklerimizin karnesi

Alan kodu → il

Örten, birebir değil

Puan → harf notu

Örten, birebir değil

Harf notu → katsayı

Birebir ve örten

İl → plaka kodu

Birebir ve örten



Şimdi kimlik numarasına dönelim.

Açılıştaki soru: geçerli her kimlik numarasının son rakamı neden çift?



NVİ'nin cümlesinde bir fonksiyon var

TÜRKÇE

Bu son iki basamak

ilk dokuz basamaktan

bir algoritma ile hesaplanmaktadır

MANTIK



d : ilk9 → Son2

İlk9: ilk dokuz basamağın olası değerleri. Son2: 00'dan 99'a kadar iki basamaklı değerler.

Kaynak: Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü, "T.C. Kimlik No İşlemleri" sayfası



d fonksiyonu Python'da

tcn.py

```
def check_digits(first9):
    d = [int(ch) for ch in first9]
    odd = d[0] + d[2] + d[4] + d[6] + d[8]
    even = d[1] + d[3] + d[5] + d[7]
    d10 = (7 * odd - even) % 10
    d11 = (sum(d) + d10) % 10
    return f"{d10}{d11}"
```

10. basamak

1, 3, 5, 7 ve 9. basamakların toplamının 7 katından 2, 4, 6 ve 8. basamakların toplamı çıkarılır; sonucun birler basamağı alınır.

11. basamak

İlk 10 basamağın toplamının birler basamağı.

Kaynak: erdemerdag.com, "TC Kimlik No Algoritması" notu



d birebir olabilir mi?

Tanım kümesinde yüz milyonlarca eleman var: ilk basamak 0 olamayacağı için $9 \cdot 10^8$ tane. Değer kümesinde ise yalnızca 100 eleman var.

Kaynak: erdemerdag.com, "TC Kimlik No Algoritması" notu



Hayır. Yüz milyonlarca girdi 100 çıktıya gidiyorsa çakışma kaçınılmazdır.

Bu fikrin adı güvercin yuvası ilkesi. Sayma konusunda ayrıntısıyla göreceğiz.



Bir çakışma örneği

`check_digits("123456789")`



"50"

`check_digits("987654321")`



"50"

İki farklı girdi, aynı doğrulama sayısı. d birebir değildir.



Peki d örten mi? Her iki basamaklı sayı bir doğrulama sayısı olabilir mi?



Hayır. 11. basamak her zaman çifttir.

Yani 01, 03, 15 gibi değerler hiçbir zaman çıkmaz. Görüntü kümesinde 100 değil, 50 eleman var.



11. basamak neden hep çift?

$$T = d_1 + d_3 + d_5 + d_7 + d_9 \quad \text{Ç} = d_2 + d_4 + d_6 + d_8$$

Tek ve çift sıradaki basamakların toplamları



$$d_{10} \equiv 7T - \text{Ç} \pmod{10}$$

10. basamağın kuralı



$$d_{11} \equiv T + \text{Ç} + d_{10} \equiv 8T \pmod{10}$$

11. basamağın kuralı



$$8T \text{ çift} \rightarrow d_{11} \text{ çift}$$

Çift bir sayının birler basamağı da çifttir

d örten değildir. Açılıştaki sorunun cevabı bu.

Kaynak: erdemerdag.com, "TC Kimlik No Algoritması" notu



Doğrulama sayısının amacı birebir olmak değil, yazım hatalarını yakalamaktır.

NVİ'nin ifadesiyle algoritma, "sadece bir numaranın tarafımızdan verilen bir T.C. Kimlik Numarası olup olmadığı hakkında bilgi" verir.

Kaynak: Nüfus ve Vatandaşlık İşleri Genel Müdürlüğü, "T.C. Kimlik No İşlemleri" sayfası

2. BÖLÜM · BİREBİR VE ÖRTEN



Bir numaranın geçerliliğini kontrol etmek

```
tckn.py

def is_valid(tckn):
    if len(tckn) != 11 or not tckn.isdigit() or tckn[0] == "0":
        return False
    return tckn[9:] == check_digits(tckn[:9])

is_valid("10000000146") # True
is_valid("10000000156") # False: one digit mistyped
```

Kodu kendi numaranızla değil, böyle uydurma numaralarla deneyin.

Kaynak: erdemerdag.com, "TC Kimlik No Algoritması" notu



3. BÖLÜM

Ters Fonksiyon ve Bileşke

Geri dönebilir miyiz? Fonksiyonları art arda nasıl bağlarız?



“06 hangi ilin plakası?”

Bu soruyu sorduğunuzda plaka fonksiyonunu ters yönde kullanıyorsunuz.

Kaynak: Türkiye il trafik (plaka) kodları



f birebir ve örtense, her $b \in B$ için $f(a) = b$ olan tek bir a vardır. Bu a 'ya $f^{-1}(b)$ denir.

f^{-1} fonksiyonuna f 'nin ters fonksiyonu denir.

Kaynak: Rosen, Bölüm 2.3



Ters fonksiyon

$$f^{-1}(b) = a \Leftrightarrow f(a) = b$$

$p(\text{Ankara}) = 06$ olduğu için $p^{-1}(06) = \text{Ankara}$.

Kaynak: Rosen, Bölüm 2.3



Okların yönünü çevirelim



Ters fonksiyon

Bütün okları ters çevirdik. Sonuç yine bir fonksiyon, çünkü p birebir ve örtendi.

Kaynak: Türkiye il trafik (plaka) kodları



Gidip geri dönmek

$$f^{-1}(f(a)) = a$$

Önce f , sonra tersi: başladığınız yere dönersiniz.
 $p^{-1}(p(\text{Ankara})) = p^{-1}(06) = \text{Ankara}$

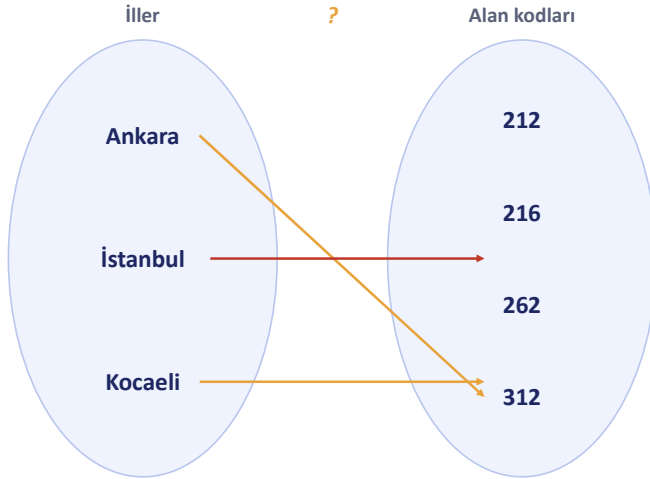
Kaynak: Ankara Üniversitesi Açık Ders Malzemeleri, "Fonksiyonlar" ders notu



Birebir olmayan bir fonksiyonun oklarını ters çevirirsek ne olur?



Alan kodu fonksiyonunu ters çevirelim



Fonksiyon değil

Okları ters çevirince İstanbul'dan iki ok çıktı. Başladığımız yere, $il \rightarrow alan$ kodu bağıntısına döndük.

Kaynak: Vodafone, "Türkiye Telefon Numaralandırma Rehberi ve Şehir Alan Kodları"



Örten olmayan bir fonksiyonun oklarını ters çevirirsek?

Açıktaki kalan elemandan hiç ok çıkmaz. Ters, değer kümesinin bütün elemanlarında tanımlı olmaz.



Kimlik numarası fonksiyonu birebir olmalı, ama bütün 11 basamaklı sayılara örten değil.

Bir numaradan tek bir kişinin kaydına ulaşabilmek için iki kişiye aynı numara verilmemeli. Ama her 11 basamaklı sayı birinin numarası değil; “bu numara kimin?” sorusu yalnızca görüntü kümesindeki sayılar için anlamlı.



Yönetmelikteki iki fonksiyonun tersi

k^{-1} : katsayıdan harf notuna



Var

h^{-1} : harf notundan puana



Yok

h birebir olmadığı için tersi yok: “AA” bir puan değil, bir puan aralığı.



Python'da bir sözlüğün tersini almak

```
invert.py

city_of = {212: "İstanbul", 216: "İstanbul", 312: "Ankara"}

code_of = {city: code for code, city in city_of.items()}
print(code_of)
# {'İstanbul': 216, 'Ankara': 312}
```

212 kayboldu. Python, birebir olmayan bir sözlüğün tersini alırken hata vermiyor.



Tersini almadan önce birebirliği kontrol edin

```
invert.py

def invert(mapping):
    # one-to-one: no two keys share a value
    if len(set(mapping.values())) != len(mapping):
        raise ValueError("not one-to-one: no inverse")
    return {v: k for k, v in mapping.items()}
```

Farklı değerlerin sayısı anahtar sayısına eşitse, hiçbir değer iki anahtara ait değildir.



Şimdi yönetmeliğimizdeki fonksiyonları art arda bağlayalım.



“... ara sınav ve varsa diğer yarıyıl içi çalışmalardan alınan notun yüzde kırkı (%40) ile yarıyıl sonu sınav notunun yüzde altmışının (%60) toplamı başarı notunu oluşturur.”



Türkçeden formüle: toplam

TÜRKÇE



MANTIK



$$0,4 \cdot a + 0,6 \cdot s$$

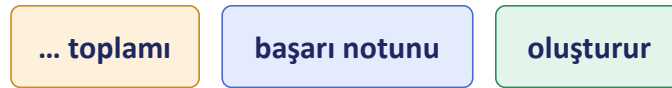
a : ara sınav notu, s : yarıyıl sonu sınav notu

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 41

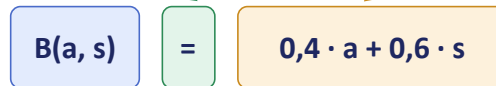


Türkçeden formüle: fonksiyon

TÜRKÇE



MANTIK



$$B(a, s) = 0,4 \cdot a + 0,6 \cdot s$$

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 41



B iki girdili bir fonksiyondur.

Tanım kümesi, geçen hafta gördüğümüz Kartezyen çarpım:
 $B : [0, 100] \times [0, 100] \rightarrow [0, 100]$



Bir öğrencinin katsayısı, üç fonksiyonun art arda uygulanmasıyla bulunur.

Önce B ile başarı notu, sonra h ile harf notu, en son k ile katsayı.



Fonksiyonların bileşkesi

$$k(h(B(a, s)))$$

En içteki fonksiyon önce çalışır.
Bu zincire bileşke denir ve $k \circ h \circ B$ diye yazılır.

Kaynak: Rosen, Bölüm 2.3



Bileşkenin tanımı

$$(g \circ f)(a) = g(f(a))$$

Önce f , sonra g uygulanır; ama yazılışta g solda durur.

Kaynak: Rosen, Bölüm 2.3



Yazılış sırası ile uygulanma sırası terstir.

Python'da da durum aynı: $g(f(x))$ yazınca önce $f(x)$ hesaplanır, sonra sonucu g 'ye verilir.



Bir öğrencinin notu, adım adım

$$a = 70, s = 85$$

Ara sınav ve yarıyıl sonu sınavı



$$B = 0,4 \cdot 70 + 0,6 \cdot 85 = 79$$

Madde 41



$$h(79) = BB$$

Madde 42, tablonun ilk iki sütunu



$$k(BB) = 3,00$$

Madde 42, tablonun son iki sütunu

$$(k \circ h \circ B)(70, 85) = 3,00$$



Aynı zincir Python'da

```
grade.py

def score(a, s):
    # Madde 41
    return 0.4 * a + 0.6 * s

def letter(b):
    # Madde 42
    for low, g in [(90, "AA"), (80, "BA"), (70, "BB"), (65, "CB"),
                  (60, "CC"), (55, "DC"), (50, "DD")]:
        if b >= low:
            return g
    return "FF"

def coefficient(g):
    # Madde 42
    return {"AA": 4.0, "BA": 3.5, "BB": 3.0, "CB": 2.5,
           "CC": 2.0, "DC": 1.5, "DD": 1.0, "FF": 0.0}[g]

coefficient(letter(score(70, 85))) # 3.0
```



Bileşkenin tanımlı olması için bir şart vardır.

g o f yazabilmek için f'nin görüntü kümesi, g'nin tanım kümesinin içinde olmalıdır.



Bir öğrenci ara sınavdan 85, yarıyıl sonu sınavından 93 alsın.

$$B = 0,4 \cdot 85 + 0,6 \cdot 93 = 34 + 55,8 = 89,8$$

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 41



89,8 hangi harf notuna karşılık gelir?

Madde 42'deki tablo 80–89 ve 90–100 aralıklarını veriyor. 89,8 ikisinin de içinde değil.

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 42



Tablo, 0 ile 100 arasındaki tam sayılar için yazılmış.

Yani h 'nin tanım kümesi $\{0, 1, \dots, 100\}$. B 'nin görüntü kümesinde ise 89,8 gibi tam sayı olmayan değerler var. Bu yüzden $h \circ B$ doğrudan tanımlı değil.



Tanım kümesinin bazı elemanlarında tanımlı olmayan eşlemelere kısmi fonksiyon denir.

Tablo, 0 ile 100 arasındaki bütün gerçekte sayılar üzerinde düşünülürse bir kısmi fonksiyondur: 89,8'de tanımlı değil.



Araya bir yuvarlama fonksiyonu girmeli.

$h(y(B(a, s)))$

Peki y hangi fonksiyon olmalı? Bunun için son bölüme geçelim.



Taban ve Tavan

Bir notu yuvarlamak neden bu kadar önemli?



Taban fonksiyonu

$$\lfloor x \rfloor$$

x'ten küçük ya da x'e eşit olan en büyük tam sayı.
 $\lfloor 89,8 \rfloor = 89$

Kaynak: Rosen, Bölüm 2.3



Tavan fonksiyonu

$$\lceil x \rceil$$

x'ten büyük ya da x'e eşit olan en küçük tam sayı.
 $\lceil 89,8 \rceil = 90$

Kaynak: Rosen, Bölüm 2.3



Taban fonksiyonunun tanımını formüle çevirelim

TÜRKÇE

x'ten küçük ya da x'e eşit olan

en büyük

tam sayı

MANTIK

$$\lfloor x \rfloor = n$$

 $\Leftrightarrow$

$$n \leq x$$

 $\wedge$

$$x < n + 1$$

$$\lfloor x \rfloor = n \Leftrightarrow n \leq x < n + 1$$

"En büyük" sözü, bir sonraki tam sayının x'i geçmesi demek.

Kaynak: Rosen, Bölüm 2.3



Birkaç örnek

x	$\lfloor x \rfloor$	$\lceil x \rceil$
89,8	89	90
90	90	90
29,4	29	30
-0,5	-1	0
-3,2	-4	-3

Negatif sayılarda dikkat: $\lfloor -3,2 \rfloor = -4$, çünkü $-4 < -3,2$ ve $-3 > -3,2$.

Kaynak: Rosen, Bölüm 2.3



Taban ve tavan, gerçek sayılardan tam sayılara birer fonksiyondur.

$\lfloor \cdot \rfloor : \mathbb{R} \rightarrow \mathbb{Z}$ birebir değil: $\lfloor 3,1 \rfloor = \lfloor 3,9 \rfloor = 3$.
Örten: her n tam sayısı için $\lfloor n \rfloor = n$.



“Öğrencilerin derslerin ve uygulamaların en az %70 devamları zorunlu ...”

Haftada 3 saat, 14 hafta süren bir ders düşünelim: toplam 42 saat.



Türkçeden formüle

TÜRKÇE



MANTIK



Devam $\geq$ 29,4 saat

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 26



Devam, ders saati olarak sayılır. 29,4 saat derse girilemez.

29,4'ten küçük olmayan en küçük tam sayı: $\lceil 29,4 \rceil = 30$ saat.



Devam ve devamsızlık

$$\text{En az devam} = \lceil 0,7 \cdot 42 \rceil = 30$$

$$\begin{aligned} \text{En fazla devamsızlık} &= \lfloor 0,3 \cdot 42 \rfloor = \lfloor 12,6 \rfloor = 12 \text{ saat} \\ \text{Kontrol: } 30 + 12 &= 42 \end{aligned}$$

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 26



Yazılımda taban ve tavan her yerdedir.

Sayfalama: 47 sonuç, sayfada 10 sonuç $\rightarrow \lceil 47/10 \rceil = 5$ sayfa.

Disk: 10 000 baytlık bir dosya, 4096 baytlık bloklarda $\lceil 10\,000/4096 \rceil = 3$ blok kaplar.



Python'da taban ve tavan

```
floor_ceil.py

import math

math.floor(89.8)      # 89
math.ceil(29.4)       # 30

-7 // 2              # -4 floor
int(-3.5)            # -3 truncation

(47 + 10 - 1) // 10   # 5 ceiling
```

//

Python'daki tam sayı bölmesi tabanı alır: $-7 // 2 = [-3,5] = -4$.

int()

Ondalık kısmı atar. Pozitif sayılarda tabanla aynı, negatif sayılarda farklı sonuç verir.

Tavan hilesi

$[n/k] = (n + k - 1) // k$. Kayan noktalı sayıya hiç girmeden tavanı hesaplar.



Şimdi 89,8 sorusuna dönelim.



Hangi yuvarlamayı seçtiğiniz harf notunu değiştirir

Yuvarlama fonksiyonu y	$y(89,8)$	Harf notu
Taban: $\lfloor x \rfloor$	89	BA
Tavan: $\lceil x \rceil$	90	AA
En yakın tam sayı: $\lfloor x + 0,5 \rfloor$	90	AA

Tablonun kendisi bu soruya cevap vermiyor; yuvarlama kuralının ayrıca belirlenmesi gerekiyor.

Kaynak: Ufuk Üniversitesi Önlisans ve Lisans Eğitim-Öğretim ve Sınav Yönetmeliği, Madde 42



Az önce yazdığımız letter fonksiyonu 89,8'e BA veriyordu.

Kodu yazan kişi, farkında olmadan taban fonksiyonunu seçmiş oldu. Kuralda bir boşluk varsa kod onu her zaman bir şekilde doldurur; önemli olan bunu bilerek yapmak.



Bir tehlike daha var: bilgisayar 0,4 ve 0,6 sayılarını tam olarak saklayamaz.

Bu sayılar ikilik tabanda sonsuz uzunlukta yazılır. Bilgisayar onları en yakın değere yuvarlayarak saklar.



Kayan noktalı sayı ve taban bir araya gelince

```
float_bug.py
import math

a, s = 2, 82
b = 0.4 * a + 0.6 * s

print(b)                # 49.99999999999999
print(math.floor(b))    # 49 -> FF
```

Gerçek değer

$0,4 \cdot 2 + 0,6 \cdot 82 = 0,8 + 49,2 = 50$. Harf notu DD.

Bilgisayarın bulduğu

49,999... Taban alınınca 49, harf notu FF.

Ders

Kayan noktalı bir sayının tabanını alırken çok dikkatli olun.



Çözüm: tam sayılarla çalışmak

```
grade_exact.py

a, s = 2, 82

# 10 * score is always a whole number: keep it exact
b10 = 4 * a + 6 * s      # 500

b10 // 10                # 50 (floor)
(b10 + 5) // 10          # 50 (nearest integer)
```

Kayan noktalı sayıya hiç girmeden hem taban hem en yakın tam sayı hesaplanabiliyor.



Python'daki round fonksiyonu da sürpriz yapabilir.

`round(2.5) = 2`, ama `round(3.5) = 4`.
Python tam ortadaki sayıları en yakın çift sayıya yuvarlar.



ÖZET

Bugün Neler Öğrendik?

Altı cümle ve açılıştaki sorunun cevabı

ÖZET - 1



Fonksiyon, tanım kümesinin her elemanını değer kümesinin bir ve yalnız bir elemanına eşler.

$\forall a \in A \exists ! b \in B, f(a) = b$

ÖZET - 2



Birebir fonksiyonda farklı girdiler farklı çıktılara gider. Birebirliğin değil çakışmadır.

Alan kodu → il birebir değil; il → plaka kodu birebir.

ÖZET - 3



Örten fonksiyonda değer kümesinde açıkta eleman kalmaz.

Kimlik numarasının doğrulama sayısı örten değil: 11. basamak hep çift.

ÖZET - 4



Yalnızca birebir ve örten fonksiyonların tersi vardır.

Birebir olmayan bir sözlüğün tersini alırsanız Python veriyi sessizce kaybeder.

ÖZET - 5



$g \circ f$ 'de önce f çalışır ve f 'nin görüntü kümesi g 'nin tanım kümesine sığmalıdır.

Başarı notu $\rightarrow$ yuvarlama $\rightarrow$ harf notu $\rightarrow$ katsayı.



Taban ve tavan gerçek sayıları tam sayılara indirir; hangi yuvarlamayı seçtiğiniz sonucu değiştirir.

Kayan noktalı sayılarda tabanı dikkatli alın: 49,999... ile 50 arasında bir harf notu farkı var.



Kimlik numaranızın son rakamı neden çift?

$$d_{11} \equiv 8 \cdot (d_1 + d_3 + d_5 + d_7 + d_9) \pmod{10}$$

8 ile çarpılan her tam sayı çifttir. Doğrulama fonksiyonu örten değil; görüntü kümesinde yalnızca son rakamı çift olan 50 değer var.

KAYNAKLAR VE GELECEK HAFTA



Bu haftanın konularını nereden çalışabilirsiniz?



Ders kitabı

Rosen, Ayrık Matematik ve Uygulamaları, Bölüm 2.3: fonksiyonlar, birebir ve örten fonksiyonlar, ters fonksiyon, bileşke, taban ve tavan, kısmi fonksiyonlar.



Türkçe kaynaklar

NVI'nin kimlik numarası sayfası, Ufuk Üniversitesi Sınav Yönetmeliği (Madde 26, 41, 42), lise ve üniversite ders notlarındaki fonksiyon tanımları.



Gelecek hafta

Algoritmalar: bir algoritmanın kaç adımda çalıştığı ve büyük O gösterimi. Bugünkü fonksiyonlar, çalışma süresini anlatmak için kullanılacak.

Sorularınızı Teams sohbetine ya da kanal üzerinden iletebilirsiniz.