

# Karakter Dizileri

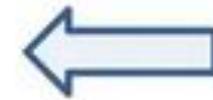
2025

M e r h a b a



0 1 2 3 4 5 6

-7 -6 -5 -4 -3 -2 -1



- Karakter Dizilerinin uzunluğu `len()` ile bulunur.  
`>>> len(cumle)`

# Karakter Dizileri

```
ad=input("Adınız: ")  
for i in range(len(ad)):  
    print("Adınızın {}. harfi: {}".format(i, ad[i]))
```

Adınız: Murat  
Adınızın 0. harfi: M  
Adınızın 1. harfi: u  
Adınızın 2. harfi: r  
Adınızın 3. harfi: a  
Adınızın 4. harfi: t

## Karakter Dizilerini Dilimlemek

- Dizinin istenilen başlangış ve bitiş aralığında dilim olarak ayırmak için.

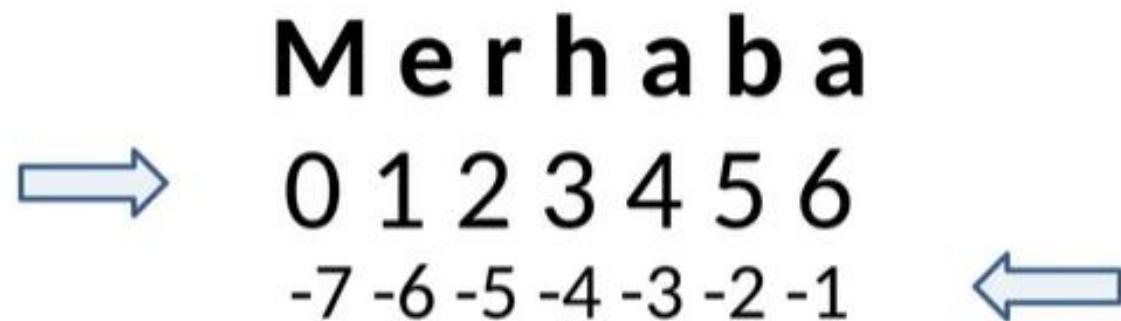
```
karakter_dizisi[alınacak_ilk_ögenin_sırası:alınacak_son_ögenin_sırasının_bir_fazlası]
```

```
>>> cümle="Çukurova Üniversitesi"
```

```
>>> cümle[5:9]
```

```
'ova '
```

## Karakter Dizilerini Dilimlemek



```
>>> s[başla:bitir]  
>>> s[1:5] -> 'erha'  
>>>s[-4] -> 'h'  
>>>s[:-3] -> 'Merh'  
>>>s[-3:] -> 'aba'
```

# Karakter Dizilerini Dilimlemek

```
cumle=input("Lütfen bir cümle giriniz: ")  
basla=int(input("Başlangıç sayısı: "))  
bitir=int(input("Bitiş sayısı: "))  
dilim=cumle[basla:bitir]  
print(dilim)  
  
>>>  
Lütfen bir cümle giriniz: Burada çok uzun bir cümle yazdık  
Başlangıç sayısı: 7  
Bitiş sayısı: 16  
çook uzun
```

# Karakter Dizileri Metotları

```
>>> dir(str)
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
 '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__',
 '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__',
 '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__',
 '__reduce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__',
 '__str__', '__subclasshook__', 'capitalize', 'casefold', 'center',
 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map',
 'index', 'isalnum', 'isalpha', 'isdecimal', 'isdigit', 'isidentifier', 'islower',
 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust',
 'lower', 'lstrip', 'maketrans', 'partition', 'replace', 'rfind', 'rindex',
 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith',
 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

# Karakter Dizileri Metotları

---

- lower() ➤ Küçük harfe çevirir
- upper() ➤ Büyük harfe çevirir
- title() ➤ İlk harfleri büyüğe çevirir
- count() ➤ Harf veya ifadenin kaç kez geçtiğini verir
- find() ➤ Harf(karakter) arar, yer sayısını verir
- isdigit() ➤ Sayısal olup olmadığını verir
- isalnum() ➤ Alfa numerik(harf-sayı) olup olmadığı
- strip() ➤ İstenmeyen karakterleri siler
- replace() ➤ Harf veya karakterleri değiştirir

# Alıştırmalar

# Kişisel Bilgi Formatı (format, f-string)

```
python                                         # 1. % Formatı (Eski yöntem)
print("== KİŞİSEL BİLGİ FORMU ==")          print("\n1) % Formatı:")
print()                                         print("Ad Soyad: %s %s" % (ad, soyad))
                                                print("Yaş: %s, Şehir: %s" % (yas, sehir))

# Kullanıcıdan bilgi alma                      # 2. .format() Metodu
ad = input("Adınız: ")                         print("\n2) .format() Metodu:")
soyad = input("Soyadınız: ")                    print("Ad Soyad: {} {}".format(ad, soyad))
yas = input("Yaşınız: ")                        print("Yaş: {0}, Şehir: {1}".format(yas, sehir))
sehir = input("Şehriniz: ")                     print("{1}, {0} yaşında ve {2}'de yaşıyor.".format(yas, ad, sehir))

print("\n" + "="*50)                           # 3. f-string (Modern yöntem - En önerilen)
print("        FORMATLAMA YÖNTEMLERİ")         print("\n3) f-string (Önerilen):")
print("=*50)                                     print(f"Ad Soyad: {ad} {soyad}")
                                                print(f"Yaş: {yas}, Şehir: {sehir}")
                                                print(f"{ad.upper()} {soyad.upper()}, {yas} yaşında ve {sehir}'de yaşıyor.")
```

# Kişisel Bilgi Formatı (format, f-string)

**\*\*Örnek Çalıştırma:\*\***

```

== KİŞİSEL BİLGİ FORMU ==

Adınız: Ahmet

Soyadınız: Yılmaz

Yaşınız: 25

Şehriniz: İstanbul

1) % Formatı:

Ad Soyad: Ahmet Yılmaz

Yaş: 25, Şehir: İstanbul

2) .format() Metodu:

Ad Soyad: Ahmet Yılmaz

Yaş: 25, Şehir: İstanbul

Ahmet, 25 yaşında ve İstanbul'de yaşıyor.

3) f-string (Önerilen):

Ad Soyad: Ahmet Yılmaz

Yaş: 25, Şehir: İstanbul

AHMET YILMAZ, 25 yaşında ve İstanbul'de yaşıyor.

# Kişisel Bilgi Formatı (format, f-string)

```
python
print("== KİŞİSEL BİLGİ FORMU ==")
print()

# Kullanıcıdan bilgi alma
ad = input("Adınız: ")
soyad = input("Soyadınız: ")
yas = input("Yaşınız: ")
sehir = input("Şehriniz: ")

print("\n" + "="*50)
print("          FORMATLAMA YÖNTEMLERİ")
print("="*50)
```

```
# 4. Gelişmiş Formatlama
print("\n4) Gelişmiş Formatlama:")
tam_ad = f"{ad} {soyad}"
print(f"'İsim':<15}: {tam_ad}")
print(f"'Yaş':<15}: {yas} yıl")
print(f"'Şehir':<15}: {sehir}")
print(f"'İsim Uzunluğu':<15}: {len(tam_ad)} karakter")

# 5. Tablo formatı
print("\n5) Tablo Formatı:")
print("-"*50)
print(f"| {'Alan':<15} | {'Değer':<28} |")
print("-"*50)
print(f"| {'Ad':<15} | {ad:<28} |")
print(f"| {'Soyad':<15} | {soyad:<28} |")
print(f"| {'Yaş':<15} | {yas:<28} |")
print(f"| {'Şehir':<15} | {sehir:<28} |")
print("-"*50)
```

# Kişisel Bilgi Formatı (format, f-string)

\*\*Örnek Çalıştırma:\*\*

```

== KİŞİSEL BİLGİ FORMU ==

Adınız: Ahmet

Soyadınız: Yılmaz

Yaşınız: 25

Şehriniz: İstanbul

## 4) Gelişmiş Formatlama:

İsim : Ahmet Yılmaz

Yaş : 25 yıl

Şehir : İstanbul

İsim Uzunluğu : 12 karakter

## 5) Tablo Formatı:

Alan	Değer
Ad	Ahmet
Soyad	Yılmaz
Yaş	25
Şehir	İstanbul

# Metin Dilimleme (Slicing)

```
python

print("== METİN DİLİMLEME UYGULAMASI ==")
print()

# Örnek metin
metin = "Python Programlama Dili Çok Güçlüdür"
print(f"Orijinal Metin: '{metin}'")
print(f"Metin Uzunluğu: {len(metin)} karakter")
print()

print("*"*60)
print("          DİLİMLEME İŞLEMLERİ")
print("*"*60)
```

```
# 1. Temel dilimleme
print("\n1) TEMEL DİLİMLEME:")
print(f"İlk 6 karakter      : '{metin[0:6]}'") # Python
print(f"İlk 6 karakter (kısa) : '{metin[:6]}'") # Python
print(f"7-18 arası           : '{metin[7:18]}'") # Programlama
print(f"19. karakterden itibaren: '{metin[19:]}'") # Dili Çok Güçlüdür

# 2. Negatif indeksleme
print("\n2) NEGATIF İNDEKSLEME:")
print(f"Son 8 karakter       : '{metin[-8:]}'") # Güçlüdür
print(f"Son karakter          : '{metin[-1]}'") # r
print(f"Sondan 2. karakter    : '{metin[-2]}'") # ü
print(f"Sondan 10. karaktere kadar: '{metin[:-10]}'") # Python Programlama

# 3. Atlama (Step)
print("\n3) ATLAMA İLE DİLİMLEME:")
print(f"2'şer atlayarak      : '{metin[::2]}'") # Pto rgamaa ii
print(f"3'er atlayarak        : '{metin[::3]}'") # Ph oalaDçcü
print(f"Tersten yazdırma     : '{metin[::-1]}'") # rüdlçÜG koç i
```

# Metin Dilimleme (Slicing)

==== METİN DİLİMLEME UYGULAMASI ===

Orijinal Metin: 'Python Programlama Dili Çok Güçlüdür'

Metin Uzunluğu: 36 karakter

## DİLİMLEME İŞLEMLERİ

### 1) TEMEL DİLİMLEME:

- İlk 6 karakter : 'Python'
- İlk 6 karakter (kısa) : 'Python'
- 7-18 arası : 'Programlama'
- 19. karakterden itibaren: 'Dili Çok Güçlüdür'

### 2) NEGATİF İNDEKSLEME:

- Son 8 karakter : 'Güçlüdür'
- Son karakter : 'r'
- Sondan 2. karakter : 'Ü'
- Sondan 10. karaktere kadar: 'Python Programlama Dili'

### 3) ATLAMA İLE DİLİMLEME:

- 2'şer atlayarak : 'Pto rgamaa ii o Üldü'
- 3'er atlayarak : 'Ph oalaDÇcÜ'
- Tersten yazdırma : 'rüdlÜçÜĞ koÇ iliD amalmargorP nohtyP'

# Metin Dilimleme (Slicing)

```
python

print("== METİN DİLİMLEME UYGULAMASI ==")
print()

# Örnek metin
metin = "Python Programlama Dili Çok Güçlüdür"
print(f"Orijinal Metin: '{metin}'")
print(f"Metin Uzunluğu: {len(metin)} karakter")
print()

print("*"*60)
print("        DİLİMLEME İŞLEMLERİ")
print("*"*60)
```

```
# 4. Belirli aralıklarla atlama
print("\n4) ARALIKLI ATLAMA:")
print(f"0-20 arası 2'şer      : '{metin[0:20:2}]"") # Pto rgamad
print(f"5-25 arası 3'er       : '{metin[5:25:3}]"") # nPoml
print(f"Tersten 2'şer atlayarak: '{metin[::-2}]"") # rdçG o l a

# 5. Kelime ayırma
print("\n5) KELİME AYIRMA:")
kelimeler = metin.split()
print(f"Kelime listesi      : {kelimeler}")
print(f"İlk kelime          : '{kelimeler[0}]"')
print(f"Son kelime          : '{kelimeler[-1}]"')
print(f"İlk 3 kelime        : '{kelimeler[:3}]"')
print(f"Son 2 kelime         : '{kelimeler[-2:}]"')

# 6. Pratik kullanımlar
print("\n6) PRATİK KULANIMLAR:")
email = "ahmet.yilmaz@example.com"
print(f"E-mail              : {email}")
print(f"Kullanıcı adı       : '{email[:email.index('@')]}")
print(f"Domain              : '{email[email.index('@')+1:]}'")
```

# Metin Dilimleme (Slicing)

==== METİN DİLİMLEME UYGULAMASI ===

Orijinal Metin: 'Python Programlama Dili Çok Güçlüdür'

Metin Uzunluğu: 36 karakter

=====  
DİLİMLEME İŞLEMLERİ  
=====

## 4) ARALIKLI ATLAMA:

0-20 arası 2'şer : 'Pto rgamaa'

5-25 arası 3'er : 'nPoml'

Tersten 2'şer atlayarak: 'rdçG o l almgoPnh'

## 5) KELİME AYIRMA:

Kelime listesi : ['Python', 'Programlama', 'Dili', 'Çok', 'Güçlüdür']

İlk kelime : 'Python'

Son kelime : 'Güçlüdür'

İlk 3 kelime : ['Python', 'Programlama', 'Dili']

Son 2 kelime : ['Çok', 'Güçlüdür']

## 6) PRATİK KULLANIMLAR:

E-mail : ahmet.yilmaz@example.com

Kullanıcı adı : 'ahmet.yilmaz'

Domain : 'example.com'

# Büyük/Küçük Harf Dönüşümleri

```
python

print("== BÜYÜK/KÜÇÜK HARF METOTLARI ==\n")

metin = "python programlama dili"

print(f"Orijinal : {metin}")
print(f"upper() : {metin.upper()}")
print(f"lower() : {metin.lower()}")
print(f"title() : {metin.title()}")
print(f"capitalize(): {metin.capitalize()}")

# Pratik kullanım
isim = "ahmet"
soyad = "YILMAZ"
print(f"\nDüzen format: {isim.capitalize()} {soyad.capitalize()}")

# Kullanıcı girişi kontrolü
sifre = "Python123"
kullanici_sifre = input("\nŞifre girin: ")

if kullanici_sifre == sifre:
    print("✓ Giriş başarılı!")
else:
    print("✗ Hatalı şifre!")

```

```

\*\*Çıktı:\*\*

...

== BÜYÜK/KÜÇÜK HARF METOTLARI ==

Orijinal : python programlama dili  
upper() : PYTHON PROGRAMLAMA DİLİ  
lower() : python programlama dili  
title() : Python Programlama Dili  
capitalize(): Python programlama dili

Düzen format: Ahmet Yılmaz

Şifre girin: Python123

✓ Giriş başarılı!

# Arama ve Kontrol Metotları

```
python

print("== ARAMA VE KONTROL METOTLARI ==\n")

cumle = "Python öğrenmek çok eğlenceli"

# Arama metotları
print(f"Metin: {cumle}\n")
print(f"find('Python') : {cumle.find('Python')}") # -1 döner
print(f"find('Java') : {cumle.find('Java')}") # -1 döner
print(f"count('e') : {cumle.count('e')}") # 4
print(f"'Python' var mı? : {'Python' in cumle}") # True

# Başlangıç/Bitiş kontrolü
print(f"\nstartswith('Python'): {cumle.startswith('Python')}")
print(f"endswith('eğlenceli'): {cumle.endswith('eğlenceli')}")

# Pratik örnek - Dosya uzantısı kontrolü
dosya = "resim.jpg"
print(f"\n'{dosya}' bir resim mi? {dosya.endswith('.jpg', '.png', '.gif')}")

dosya2 = "belge.pdf"
print(f"'{dosya2}' bir resim mi? {dosya2.endswith('.jpg', '.png', '.gif')}")
```

```

```
**Çıktı:**  
```
== ARAMA VE KONTROL METOTLARI ==

Metin: Python öğrenmek çok eğlenceli

find('Python') : 0
find('Java') : -1
count('e') : 4
'Python' var mı? : True

startswith('Python'): True
endswith('eğlenceli'): True

'resim.jpg' bir resim mi? True
'belge.pdf' bir resim mi? False
```

```

# Temizleme ve Değiştirme Metotları

```
python
print("== TEMİZLEME VE DEĞİŞTİRME METOTLARI ==\n")

# strip() - Boşluk temizleme
email = " ahmet@example.com "
print(f"Ham email : {email}")
print(f"strip() : {email.strip()}")
print(f"lstrip() : {email.lstrip()}")
print(f"rstrip() : {email.rstrip()}")

# replace() - Değiştirme
metin = "Java çok güzel bir dildir"
print(f"\nOrijinal : {metin}")
print(f"replace() : {metin.replace('Java', 'Python')}")

# Birden fazla boşluğu tek boşluğa çevirme
bozuk = "Python programlama dili"
duzgun = " ".join(bozuk.split())
print(f"\nBozuk : {bozuk}")
print(f"Düzeltilmiş : {duzgun}")

# Pratik - Telefon formatı temizleme
tel = "(555) 123-45-67"
temiz_tel = tel.replace("(", "").replace(")", "").replace("-", "").replace(" ", "")
print(f"\nTelefon : {tel}")
print(f"Temiz : {temiz_tel}")
```

\*\*Çıktı:\*\*

``

== TEMİZLEME VE DEĞİŞTİRME METOTLARI ==

Ham email : ' ahmet@example.com '
strip() : 'ahmet@example.com'
lstrip() : 'ahmet@example.com'
rstrip() : ' ahmet@example.com'

Orijinal : Java çok güzel bir dildir
replace() : Python çok güzel bir dildir

Bozuk : 'Python programlama dili'
Düzeltilmiş : 'Python programlama dili'

Telefon : (555) 123-45-67
Temiz : 5551234567

# Ayırma ve Birleştirme Metotları

```
cumle = "Python Java C++ JavaScript"
kelimeler = cumle.split()
print(f"Cümle : {cumle}")
print(f"split() : {kelimeler}")
print(f"İlk kelime : {kelimeler[0]}")
print(f"Son kelime : {kelimeler[-1]}\n"

# Özel ayraçla ayırma
csv_veri = "Ahmet,25,İstanbul"
bilgiler = csv_veri.split(",")
print(f"\nCSV veri : {csv_veri}")
print(f"split(',') : {bilgiler}")
print(f"İsim: {bilgiler[0]}, Yaş: {bilgiler[1]}, Şehir: {bilgiler[2]}\n

# join() - Birleştirme
kelimeler = ["Python", "çok", "güzel"]
print(f"\nListe : {kelimeler}")
print(f"' '.join() : ' '.join(kelimeler))")
print(f"'-'.join() : '-'.join(kelimeler))")
print(f"'*'.join(): '*' '.join(kelimeler))\n

# Pratik - Tam ad oluşturma
ad = "ahmet"
soyad = "yılmaz"
tam_ad = " ".join([ad.title(), soyad.title()])
print(f"\nTam ad : {tam_ad}")
```

```
**Çıktı:**  
```
==== AYIRMA VE BİRLEŞTİRME METOTLARI ===  
  
Cümle : Python Java C++ JavaScript  
split() : ['Python', 'Java', 'C++', 'JavaScript']  
İlk kelime : Python  
Son kelime : JavaScript  
  
CSV veri : Ahmet,25,İstanbul  
split(',') : ['Ahmet', '25', 'İstanbul']  
İsim: Ahmet, Yaş: 25, Şehir: İstanbul  
  
Liste : ['Python', 'çok', 'güzel']  
' '.join() : Python çok güzel  
'-'.join() : Python-çok-güzel  
'*'.join(): Python *** çok *** güzel  
  
Tam ad : Ahmet Yılmaz
```

# Kontrol Metotları (is...)

```
# Sayı kontrolü
veri1 = "12345"
veri2 = "Python123"
veri3 = "Python"

print(f"\n{veri1}'.isdigit() : {veri1.isdigit()}")
print(f"\n{veri2}'.isdigit() : {veri2.isdigit()}")
print(f"\n{veri3}'.isalpha() : {veri3.isalpha()}")
print(f"\n{veri2}'.isalnum() : {veri2.isalnum()}")

# Büyük/Küçük harf kontrolü
metin1 = "PYTHON"
metin2 = "python"
metin3 = "Python"

print(f"\n{metin1}'.isupper() : {metin1.isupper()}")
print(f"\n{metin2}'.islower() : {metin2.islower()}")
print(f"\n{metin3}'.istitle() : {metin3.istitle()}")

# Boşluk kontrolü
bos = " "
dolu = "Python"

print(f"\n{bos}'.isspace() : {bos.isspace()}")
print(f"\n{dolu}'.isspace() : {dolu.isspace()}")
```

\*\*Çıktı:\*\*  
```  
--- KONTROL METOTLARI ---  
  
'12345'.isdigit() : True  
'Python123'.isdigit() : False  
'Python'.isalpha() : True  
'Python123'.isalnum() : True  
  
'PYTHON'.isupper() : True  
'python'.islower() : True  
'Python'.istitle() : True  
  
' '.isspace() : True  
'Python'.isspace() : False

# Hatırlatma

# Hatırlatma

Dönsürtme	<code>upper()</code>	Büyük harfe çevir	<code>"python".upper() → "PYTHON"</code>
	<code>lower()</code>	Küçük harfe çevir	<code>"PYTHON".lower() → "python"</code>
	<code>title()</code>	Her kelimenin ilk harfi büyük	<code>"hello world".title() → "Hello World"</code>
	<code>capitalize()</code>	Sadece ilk harf büyük	<code>"hello".capitalize() → "Hello"</code>
Temizleme	<code>strip()</code>	Baştan ve sondan boşluk sil	<code>" hi ".strip() → "hi"</code>
	<code>replace(a,b)</code>	a'yı b ile değiştir	<code>"hi".replace("i","o") → "ho"</code>
Arama	<code>find(x)</code>	x'in indeksini bul	<code>"python".find("th") → 2</code>
	<code>count(x)</code>	x'i say	<code>"hello".count("l") → 2</code>
	<code>startswith(x)</code>	x ile başlar mı?	<code>"python".startswith("py") → True</code>
	<code>endswith(x)</code>	x ile biter mi?	<code>"test.py".endswith(".py") → True</code>

```
# E-mail temizleme
email = " AHMET@EXAMPLE.COM "
temiz = email.strip().lower() # "ahmet@example.com"

# Dosya uzantısı kontrolü
dosya = "resim.jpg"
if dosya.endswith('.jpg', '.png'):
    print("Resim dosyası")
```

Ayırma	<code>split()</code>	Listeye ayır	<code>"a b c".split() → ['a', 'b', 'c']</code>
	<code>join(list)</code>	Listeyi birleştir	<code>"-".join(['a', 'b']) → "a-b"</code>
Kontrol	<code>isdigit()</code>	Sadece rakam mı?	<code>"123".isdigit() → True</code>
	<code>isalpha()</code>	Sadece harf mi?	<code>"abc".isalpha() → True</code>
	<code>isalnum()</code>	Harf veya rakam mı?	<code>"abc123".isalnum() → True</code>
	<code>isupper()</code>	Büyük harf mi?	<code>"ABC".isupper() → True</code>
	<code>islower()</code>	Küçük harf mi?	<code>"abc".islower() → True</code>

```
# CSV ayırma
veri = "Ahmet,25,İstanbul"
ad, yas, sehir = veri.split(",")

# URL oluşturma
baslik = "Python Programlama"
url = baslik.lower().replace(" ", "-") # "python-programlama"

# Şifre kontrolü
sifre = "Python123"
guclu = len(sifre) >= 8 and any(c.isdigit() for c in sifre)
```