

`System.out.println()`: Bu method, bir metni konsol ekranına yazdırmak için kullanılır. Örneğin:

```
```java
System.out.println("Merhaba Dünya!");
```
```

`String.length()`: Bu method, bir String'in karakter sayısını döndürür. Örneğin:

```
```java
String str = "Hello";
int length = str.length();
System.out.println("Karakter sayısı: " + length); // Çıktı: Karakter sayısı: 5
```
```

`Math.abs()`: Bu method, bir sayının mutlak değerini döndürür. Örneğin:

```
```java
int number = -5;
int absoluteValue = Math.abs(number);
System.out.println("Mutlak değer: " + absoluteValue); // Çıktı: Mutlak değer: 5
```
```

`ArrayList.add()`: Bu method, bir ArrayList'e yeni bir eleman eklemek için kullanılır. Örneğin:

```
```java
ArrayList<String> list = new ArrayList<>();
list.add("Elma");
list.add("Armut");
System.out.println(list); // Çıktı: [Elma, Armut]
```
```

`Scanner.nextLine()`: Bu method, kullanıcıdan bir satır veri okumak için kullanılır. Örneğin:

```
```java
Scanner scanner = new Scanner(System.in);
System.out.print("Adınız: ");
String name = scanner.nextLine();
System.out.println("Merhaba, " + name + "!"); // Kullanıcıdan alınan ismi yazdırır
```
```

`File.exists()`: Bu method, belirtilen dosyanın var olup olmadığını kontrol eder. Örneğin:

```
```java
File file = new File("dosya.txt");
if (file.exists()) {
 System.out.println("Dosya mevcut");
} else {
 System.out.println("Dosya mevcut değil");
}
```
```

`String.charAt()`: Bu method, bir String'in belirtilen indeksteki karakterini döndürür. Örneğin:

```
```java
String str = "Merhaba";
char ch = str.charAt(2);
System.out.println("İndeksteki karakter: " + ch); // Çıktı: İndeksteki karakter:
r
```
```

`Arrays.sort()`: Bu method, bir dizi içindeki elemanları sıralamak için kullanılır. Örneğin:

```
```java
int[] numbers = {5, 2, 8, 1, 4};
Arrays.sort(numbers);
System.out.println(Arrays.toString(numbers)); // Çıktı: [1, 2, 4, 5, 8]
```
```

`String.toLowerCase()`: Bu method, bir String'in tüm karakterlerini küçük harfe dönüştürür. Örneğin:

```
```java
String str = "MERHABA";
String lowercase = str.toLowerCase();
System.out.println("Küçük harfli hali: " + lowercase); // Çıktı: Küçük harfli
hali: merhaba
```
```

`String.toUpperCase()`: Bu method, bir String'in tüm karakterlerini büyük harfe dönüştürür. Örneğin:

```
```java
String str = "merhaba";
String uppercase = str.toUpperCase();
System.out.println("Büyük harfli hali: " + uppercase); // Çıktı: Büyük harfli
hali: MERHABA
```
```

`String.contains()`: Bu method, bir String'in belirli bir alt dizgiyi içerip içermediğini kontrol eder. Örneğin:

```
```java
```

```
String str = "Merhaba Dünya";
boolean contains = str.contains("Dünya");
System.out.println("İçeriyor mu? " + contains); // Çıktı: İçeriyor mu? true
\\
```

String.substring(): Bu method, bir String'in belirli bir alt diziye (substring) sahip olmasını sağlar. Örneğin:

```
String str = "Merhaba Dünya";
String substring = str.substring(3, 8);
System.out.println("Alt dizi: " + substring); // Çıktı: Alt dizi: haba
```

Integer.parseInt(): Bu method, bir String'i tamsayıya dönüştürür. Örneğin:

```
String numberStr = "42";
int number = Integer.parseInt(numberStr);
System.out.println("Sayı: " + number); // Çıktı: Sayı: 42
```

String.trim(): Bu method, bir String'in başındaki ve sonundaki boşlukları kırpar. Örneğin:

```
String str = " Merhaba ";
String trimmed = str.trim();
System.out.println("Kırpılmış hali: " + trimmed); // Çıktı: Kırpılmış hali:
Merhaba
```

ArrayList.size(): Bu method, bir ArrayList'in eleman sayısını döndürür. Örneğin:

```
ArrayList<String> list = new ArrayList<>();
list.add("Elma");
list.add("Armut");
int size = list.size();
System.out.println("Eleman sayısı: " + size); // Çıktı: Eleman sayısı: 2
```

Collections.sort(): Bu method, bir Collection'ı (ArrayList, LinkedList vb.) sıralar. Örneğin:

```
ArrayList<Integer> numbers = new ArrayList<>();
numbers.add(5);
numbers.add(2);
numbers.add(8);
numbers.add(1);
Collections.sort(numbers);
System.out.println(numbers); // Çıktı: [1, 2, 5, 8]
```

String.replace(): Bu method, bir String içindeki belirli bir karakteri veya alt dizesini başka bir karakter veya alt dizesiyle değiştirir. Örneğin:

```
String str = "Merhaba Dünya";
```

```
String replaced = str.replace("Dünya", "Java");
System.out.println("Değiştirilmiş hali: " + replaced); // Çıktı: Değiştirilmiş
hali: Merhaba Java
```

String.equals(): Bu method, bir String'in başka bir String ile eşit olup olmadığını kontrol eder. Örneğin:

```
String str1 = "Merhaba";
String str2 = "Merhaba";
boolean equals = str1.equals(str2);
System.out.println("Eşit mi? " + equals); // Çıktı: Eşit mi? true
```

StringBuilder.append(): Bu method, bir StringBuilder'a bir değer ekler. Örneğin:

```
StringBuilder builder = new StringBuilder();
builder.append("Merhaba");
builder.append(" ");
builder.append("Dünya");
String result = builder.toString();
System.out.println(result); // Çıktı: Merhaba Dünya
```

String.split(): Bu method, bir String'i belirli bir ayırıcıya göre böler ve alt dizelerin bir dizisini döndürür. Örneğin:

```
String str = "Elma,Armut,Çilek";
String[] fruits = str.split(",");
System.out.println(Arrays.toString(fruits)); // Çıktı: [Elma, Armut, Çilek]
```

String.indexOf(): Bu method, bir String içinde belirli bir karakterin veya alt dizisinin ilk bulunduğu indeksi döndürür. Örneğin:

```
String str = "Merhaba Dünya";
int index = str.indexOf("Dünya");
System.out.println("İndeks: " + index); // Çıktı: İndeks: 8
```

String.valueOf(): Bu method, bir değeri String'e dönüştürür. Örneğin:

```
int number = 42;
String str = String.valueOf(number);
System.out.println("String: " + str); // Çıktı: String: 42
```

Math.random(): Bu method, 0 ile 1 arasında rastgele bir ondalık sayı döndürür. Örneğin:

```
double randomNumber = Math.random();
System.out.println("Rastgele sayı: " + randomNumber); // Örnek çıktı: Rastgele
sayı: 0.7391294866478713
```

`String.startsWith()`: Bu method, bir `String`'in belirli bir alt dizeyle başlayıp başlamadığını kontrol eder. Örneğin:

```
String str = "Merhaba Dünya";
boolean startsWith = str.startsWith("Merhaba");
System.out.println("Başlıyor mu? " + startsWith); // Çıktı: Başlıyor mu? true
```

`Character.isDigit()`: Bu method, bir karakterin bir sayı olup olmadığını kontrol eder. Örneğin:

```
char ch = '9';
boolean isDigit = Character.isDigit(ch);
System.out.println("Sayı mı? " + isDigit); // Çıktı: Sayı mı? true
```

`ArrayList.get()`: Bu method, bir `ArrayList`'teki belirli bir indeksteki elemanı döndürür. Örneğin:

```
ArrayList<String> list = new ArrayList<>();
list.add("Elma");
list.add("Armut");
String element = list.get(1);
System.out.println("Eleman: " + element); // Çıktı: Eleman: Armut
```

`String.valueOf()`: Bu method, bir değeri `String`'e dönüştürür. Örneğin:

```
int number = 42;
String str = String.valueOf(number);
System.out.println("String: " + str); // Çıktı: String: 42
```

`String.isEmpty()`: Bu method, bir `String`'in boş olup olmadığını kontrol eder. Örneğin:

```
String str = "Merhaba";
boolean isEmpty = str.isEmpty();
System.out.println("Boş mu? " + isEmpty); // Çıktı: Boş mu? false
```

`String.format()`: Bu method, biçimlendirilmiş bir `String` oluşturmak için kullanılır. Örneğin:

```
String name = "John";
int age = 30;
String formatted = String.format("İsim: %s, Yaş: %d", name, age);
System.out.println(formatted); // Çıktı: İsim: John, Yaş: 30
```

`Character.isLetter()`: Bu method, bir karakterin bir harf olup olmadığını kontrol eder. Örneğin:

```
char ch = 'A';
boolean isLetter = Character.isLetter(ch);
System.out.println("Harf mi? " + isLetter); // Çıktı: Harf mi? true
```

`Arrays.copyOf()`: Bu method, bir dizinin belirli bir boyutunu kopyalar ve yeni bir dizi döndürür. Örneğin:

```
int[] numbers = {1, 2, 3, 4, 5};
int[] copy = Arrays.copyOf(numbers, 3);
System.out.println(Arrays.toString(copy)); // Çıktı: [1, 2, 3]
```

`String.toCharArray()`: Bu method, bir `String`'i karakter dizisine dönüştürür. Örneğin:

```
String str = "Merhaba";
char[] charArray = str.toCharArray();
System.out.println(Arrays.toString(charArray)); // Çıktı: [M, e, r, h, a, b, a]
```

`String.format()`: Bu method, biçimlendirilmiş bir `String` oluşturmak için kullanılır. Örneğin:

```
String name = "John";
int age = 30;
String formatted = String.format("İsim: %s, Yaş: %d", name, age);
System.out.println(formatted); // Çıktı: İsim: John, Yaş: 30
```

`Collections.reverse()`: Bu method, bir `Collection`'ı (`ArrayList`, `LinkedList` vb.) tersine çevirir. Örneğin:

```
ArrayList<Integer> numbers = new ArrayList<>();
numbers.add(1);
numbers.add(2);
numbers.add(3);
Collections.reverse(numbers);
System.out.println(numbers); // Çıktı: [3, 2, 1]
```

`String.endsWith()`: Bu method, bir `String`'in belirli bir alt dizıyla bitip bitmediğini kontrol eder. Örneğin:

```
String str = "Merhaba Dünya";
boolean endsWith = str.endsWith("Dünya");
System.out.println("Bitiyor mu? " + endsWith); // Çıktı: Bitiyor mu? true
```

`Arrays.binarySearch()`: Bu method, bir sıralı bir dizide bir değeri arar ve indeksini döndürür. Örneğin:

```
int[] numbers = {1, 2, 3, 4, 5};
int index = Arrays.binarySearch(numbers, 4);
```

```
System.out.println("İndeks: " + index); // Çıktı: İndeks: 3
```

`Collections.shuffle()`: Bu method, bir Collection'ı (ArrayList, LinkedList vb.) rastgele karıştırır. Örneğin:

```
ArrayList<Integer> numbers = new ArrayList<>();
numbers.add(1);
numbers.add(2);
numbers.add(3);
Collections.shuffle(numbers);
System.out.println(numbers); // Örnek çıktı: [3, 1, 2]
```

`String.compareTo()`: Bu method, bir String'i başka bir String ile karşılaştırır ve alfabetik sıralamada hangi String'in önce olduğunu belirler. Örneğin:

```
String str1 = "Elma";
String str2 = "Armut";
int result = str1.compareTo(str2);
System.out.println("Sonuç: " + result); // Örnek çıktı: Sonuç: 1
```

`Character.toUpperCase()`: Bu method, bir karakterin büyük harfe dönüştürülmesini sağlar. Örneğin:

```
char ch = 'a';
char uppercase = Character.toUpperCase(ch);
System.out.println("Büyük harf: " + uppercase); // Çıktı: Büyük harf: A
```

`String.join()`: Bu method, bir dizi veya Collection'daki elemanları belirli bir ayırıcıyla birleştirir. Örneğin:

```
String[] words = {"Merhaba", "Dünya"};
String result = String.join(" ", words);
System.out.println(result); // Çıktı: Merhaba Dünya
```

`String.toCharArray()`: Bu method, bir String'i karakter dizisine dönüştürür. Örneğin:

```
String str = "Merhaba";
char[] charArray = str.toCharArray();
System.out.println(Arrays.toString(charArray)); // Çıktı: [M, e, r, h, a, b, a]
```

`Character.isWhitespace()`: Bu method, bir karakterin boşluk karakteri olup olmadığını kontrol eder. Örneğin:

```
char ch = ' ';
boolean isWhitespace = Character.isWhitespace(ch);
System.out.println("Boşluk mu? " + isWhitespace); // Çıktı: Boşluk mu? true
```

`String.lastIndexOf()`: Bu method, bir `String` içinde belirli bir karakterin veya alt dizesinin son bulunduğu indeksi döndürür. Örneğin:

```
String str = "Merhaba Dünya";
int index = str.lastIndexOf("a");
System.out.println("Son indeks: " + index); // Çıktı: Son indeks: 11
```

`String.replaceFirst()`: Bu method, bir `String` içinde belirli bir alt dizeyi ilk bulunduğu yerde başka bir alt dizeyle değiştirir. Örneğin:

```
String str = "Merhaba Dünya";
String replaced = str.replaceFirst("Dünya", "Java");
System.out.println("Değiştirilmiş hali: " + replaced); // Çıktı: Değiştirilmiş hali: Merhaba Java
```

`String.toUpperCase()`: Bu method, bir `String`'in tüm karakterlerini büyük harfe dönüştürür. Örneğin:

```
String str = "merhaba";
String uppercase = str.toUpperCase();
System.out.println("Büyük harfli hali: " + uppercase); // Çıktı: Büyük harfli hali: MERHABA
```

`String.toLowerCase()`: Bu method, bir `String`'in tüm karakterlerini küçük harfe dönüştürür. Örneğin:

```
String str = "MERHABA";
String lowercase = str.toLowerCase();
System.out.println("Küçük harfli hali: " + lowercase); // Çıktı: Küçük harfli hali: merhaba
```

`String.contains()`: Bu method, bir `String`'in belirli bir alt diziyi içerip içermediğini kontrol eder. Örneğin:

```
String str = "Merhaba Dünya";
boolean contains = str.contains("Dünya");
System.out.println("İçeriyor mu? " + contains); // Çıktı: İçeriyor mu? true
```

`String.substring()`: Bu method, bir `String`'in belirli bir alt diziye (`substring`) sahip olmasını sağlar. Örneğin:

```
String str = "Merhaba Dünya";
String substring = str.substring(3, 8);
System.out.println("Alt dizi: " + substring); // Çıktı: Alt dizi: haba
```

`String.format()`: Bu method, biçimlendirilmiş bir `String` oluşturmak için kullanılır. Örneğin:



```
String name = "John";
int age = 30;
String formatted = String.format("İsim: %s, Yaş: %d", name, age);
System.out.println(formatted); // Çıktı: İsim: John, Yaş: 30
```

StringBuilder.append(): Bu method, bir StringBuilder'a bir değer ekler. Örneğin:

```
StringBuilder builder = new StringBuilder();
builder.append("Merhaba");
builder.append(" ");
builder.append("Dünya");
String result = builder.toString();
```

```
System.out.println(result); // Çıktı: Merhaba Dünya
```

String.isEmpty(): Bu method, bir String'in boş olup olmadığını kontrol eder. Örneğin:

```
String str = "Merhaba";
boolean isEmpty = str.isEmpty();
System.out.println("Boş mu? " + isEmpty); // Çıktı: Boş mu? false
```

Math.max(): Bu method, iki sayı arasındaki maksimum değeri döndürür. Örneğin:

```
int a = 5;
int b = 8;
int max = Math.max(a, b);
System.out.println("Maksimum değer: " + max); // Çıktı: Maksimum değer: 8
```

Math.min(): Bu method, iki sayı arasındaki minimum değeri döndürür. Örneğin:

```
int a = 5;
int b = 8;
int min = Math.min(a, b);
System.out.println("Minimum değer: " + min); // Çıktı: Minimum değer: 5
```

Math.pow(): Bu method, bir sayının üssünü hesaplar. Örneğin:

```
double base = 2.0;
double exponent = 3.0;
double result = Math.pow(base, exponent);
System.out.println("Sonuç: " + result); // Çıktı: Sonuç: 8.0
```

String.compareToIgnoreCase(): Bu method, iki String'i büyük-küçük harf duyarsız olarak karşılaştırır ve alfabetik sıralamada hangi String'in önce olduğunu belirler. Örneğin:

```
String str1 = "Elma";
String str2 = "elma";
int result = str1.compareToIgnoreCase(str2);
System.out.println("Sonuç: " + result); // Örnek çıktı: Sonuç: 0
```

String.format(): Bu method, biçimlendirilmiş bir String oluşturmak için kullanılır. Örneğin:

```
String name = "John";
int age = 30;
String formatted = String.format("İsim: %s, Yaş: %d", name, age);
System.out.println(formatted); // Çıktı: İsim: John, Yaş: 30
```

String.toCharArray(): Bu method, bir String'i karakter dizisine dönüştürür. Örneğin:

```
String str = "Merhaba";
char[] charArray = str.toCharArray();
System.out.println(Arrays.toString(charArray)); // Çıktı: [M, e, r, h, a, b, a]
```

Character.isLetter(): Bu method, bir karakterin bir harf olup olmadığını kontrol eder. Örneğin:

```
char ch = 'A';
boolean isLetter = Character.isLetter(ch);
System.out.println("Harf mi? " + isLetter); // Çıktı: Harf mi? true
```

String.lastIndexOf(): Bu method, bir String içinde belirli bir karakterin veya alt dizesinin son bulunduğu indeksi döndürür. Örneğin:

```
String str = "Merhaba Dünya";
int index = str.lastIndexOf("a");
System.out.println("Son indeks: " + index); // Çıktı: Son indeks: 11
```

String.replaceFirst(): Bu method, bir String içinde belirli bir alt dizeyi ilk bulunduğu yerde başka bir alt dizeyle değiştirir. Örneğin:

```
String str = "Merhaba Dünya";
String replaced = str.replaceFirst("Dünya", "Java");
System.out.println("Değiştirilmiş hali: " + replaced); // Çıktı: Değiştirilmiş hali: Merhaba Java
```

String.toUpperCase(): Bu method, bir String'in tüm karakterlerini büyük harfe dönüştürür. Örneğin:

```
String str = "merhaba";
String uppercase = str.toUpperCase();
```

`System.out.println("Büyük harfli hali: " + uppercase); // Çıktı: Büyük harfli hali: MERHABA`

`String.toLowerCase()`: Bu method, bir `String`'in tüm karakterlerini küçük harfe dönüştürür. Örneğin:

```
String str = "MERHABA";
String lowercase = str.toLowerCase();
System.out.println("Küçük harfli hali: " + lowercase); // Çıktı: Küçük harfli hali: merhaba
```

`String.contains()`: Bu method, bir `String`'in belirli bir alt diziyi içerip içermediğini kontrol eder. Örneğin:

```
String str = "Merhaba Dünya";
boolean contains = str.contains("Dünya");
System.out.println("İçeriyor mu? " + contains); // Çıktı: İçeriyor mu? true
```

`String.substring()`: Bu method, bir `String`'in belirli bir alt diziye (`substring`) sahip olmasını sağlar. Örneğin:

```
String str = "Merhaba Dünya";
String substring = str.substring(3, 8);
System.out.println("Alt dizi: " + substring); // Çıktı: Alt dizi: haba
```

`String.join()`: Bu method, bir dizi veya `Collection`'daki elemanları belirli bir ayırıcıyla birleştirir. Örneğin:

```
String[] words = {"Merhaba", "Dünya"};
String result = String.join(" ", words);
System.out.println(result); // Çıktı: Merhaba Dünya
```

`String.format()`: Bu method, biçimlendirilmiş bir `String` oluşturmak için kullanılır. Örneğin:

```
String name = "John";
int age = 30;
String formatted = String.format("İsim: %s, Yaş: %d", name, age);
System.out.println(formatted); // Çıktı: İsim: John, Yaş: 30
```

`Collections.reverse()`: Bu method, bir `Collection`'ı (`ArrayList`, `LinkedList` vb.) tersine çevirir. Örneğin:

```
ArrayList<Integer> numbers = new ArrayList<>();
numbers.add(1);
numbers.add(2);
numbers.add(3);
Collections.reverse(numbers);
```

```
System.out.println(numbers); // Çıktı: [3, 2, 1]
```

`Arrays.binarySearch()`: Bu method, bir sıralı bir dizide bir değeri arar ve indeksini döndürür. Örneğin:

```
int[] numbers = {1, 2, 3, 4, 5};
int index = Arrays.binarySearch(numbers, 4);
System.out.println("İndeks: " + index); // Çıktı: İndeks: 3
```

`Collections.shuffle()`: Bu method, bir Collection'ı (ArrayList, LinkedList vb.) rastgele karıştırır. Örneğin:

```
ArrayList<Integer> numbers = new ArrayList<>();
numbers.add(1);
numbers.add(2);
numbers.add(3);
Collections.shuffle(numbers);
System.out.println(numbers); // Örnek çıktı: [3, 1, 2]
```

`String.endsWith()`: Bu method, bir String'in belirli bir alt dizeyle bitip bitmediğini kontrol eder. Örneğin:

```
String str = "Merhaba Dünya";
boolean endsWith = str.endsWith("Dünya");
System.out.println("Bitiyor mu? " + endsWith); // Çıktı: Bitiyor mu? true
```

`String.indexOf()`: Bu method, bir String içinde belirli bir karakterin veya alt dizesinin ilk bulunduğu indeksi döndürür. Örneğin:

```
String str = "Merhaba Dünya";
int index = str.indexOf("Dünya");
System.out.println("İndeks: " + index); // Çıktı: İndeks: 8
```

`String.valueOf()`: Bu method, bir değeri String'e dönüştürür. Örneğin:

```
int number = 42;
String str = String.valueOf(number);
System.out.println("String: " + str); // Çıktı: String: 42
```

`Math.random()`: Bu method, 0 ile 1 arasında rastgele bir ondalık sayı döndürür. Örneğin:

```
double randomNumber = Math.random();
System.out.println("Rastgele sayı: " + randomNumber); // Örnek çıktı: Rastgele sayı: 0.7391294866478713
```

`String.startsWith()`: Bu method, bir String'in belirli bir alt dizeyle başlayıp

başlamadığını kontrol eder. Örneğin:

```
String str = "Merhaba Dünya";
boolean startsWith = str.startsWith("Merhaba");
System.out.println("Başlıyor mu? " + startsWith); // Çıktı: Başlıyor mu? true
```

Character.isDigit(): Bu method, bir karakterin bir sayı olup olmadığını kontrol eder. Örneğin:

```
char ch = '9';
boolean isDigit = Character.isDigit(ch);
System.out.println("Sayı mı? " + isDigit); // Çıktı: Sayı mı? true
```

ArrayList.get(): Bu method, bir ArrayList'teki belirli bir indeksteki elemanı döndürür. Örneğin:

```
ArrayList<String> list = new ArrayList<>();
list.add("Elma");
list.add("Armut");
String element = list.get(1);
System.out.println("Eleman: " + element); // Çıktı: Eleman: Armut
```

String.valueOf(): Bu method, bir değeri String'e dönüştürür. Örneğin:

```
int number = 42;
String str = String.valueOf(number);
System.out.println("String: " + str); // Çıktı: String: 42
```

String.isEmpty(): Bu method, bir String'in boş olup olmadığını kontrol eder. Örneğin:

```
String str = "Merhaba";
boolean isEmpty = str.isEmpty();
System.out.println("Boş mu? " + isEmpty); // Çıktı: Boş mu? false
```

String.format(): Bu method, biçimlendirilmiş bir String oluşturmak için kullanılır. Örneğin:

```
String name = "John";
int age = 30;
String formatted = String.format("İsim: %s, Yaş: %d", name, age);
System.out.println(formatted); // Çıktı: İsim: John, Yaş: 30
```

Character.isLetter(): Bu method, bir karakterin bir harf olup olmadığını kontrol eder. Örneğin:

```
char ch = 'A';
boolean isLetter = Character.isLetter(ch);
```

```
System.out.println("Harf mi? " + isLetter); // Çıktı: Harf mi? true
```

`Arrays.copyOf()`: Bu method, bir dizinin belirli bir boyutunu kopyalar ve yeni bir dizi döndürür. Örneğin:

```
int[] numbers = {1, 2, 3, 4, 5};
int[] copy = Arrays.copyOf(numbers, 3);
System.out.println(Arrays.toString(copy)); // Çıktı: [1, 2, 3]
```

`String.toCharArray()`: Bu method, bir String'i karakter dizisine dönüştürür. Örneğin:

```
String str = "Merhaba";
char[] charArray = str.toCharArray();
System.out.println(Arrays.toString(charArray)); // Çıktı: [M, e, r, h, a, b, a]
```

`String.format()`: Bu method, biçimlendirilmiş bir String oluşturmak için kullanılır. Örneğin:

```
String name = "John";
int age = 30;
String formatted = String.format("İsim: %s, Yaş: %d", name, age);
System.out.println(formatted); // Çıktı: İsim: John, Yaş: 30
```

`Collections.reverse()`: Bu method, bir Collection'ı (ArrayList, LinkedList vb.) tersine çevirir. Örneğin:

```
ArrayList<Integer> numbers = new ArrayList<>();
numbers.add(1);
numbers.add(2);
numbers.add(3);
Collections.reverse(numbers);
System.out.println(numbers); // Çıktı: [3, 2, 1]
```

`String.endsWith()`: Bu method, bir String'in belirli bir alt dizeyle bitip bitmediğini kontrol eder. Örneğin:

```
String str = "Merhaba Dünya";
boolean endsWith = str.endsWith("Dünya");
System.out.println("Bitiyor mu? " + endsWith); // Çıktı: Bitiyor mu? true
```

`String.indexOf()`: Bu method, bir String içinde belirli bir karakterin veya alt dizesinin ilk bulunduğu indeksi döndürür. Örneğin:

```
String str = "Merhaba Dünya";
int index = str.indexOf("Dünya");
System.out.println("İndeks: " + index); // Çıktı: İndeks: 8
```

String.valueOf(): Bu method, bir değeri String'e dönüştürür. Örneğin:

```
int number = 42;
String str = String.valueOf(number);
System.out.println("String: " + str); // Çıktı: String: 42
```

Arrays.asList(): Bu method, bir diziyi List'e dönüştürür. Örneğin:

```
String[] array = {"Elma", "Armut", "Kiraz"};
List<String> list = Arrays.asList(array);
System.out.println(list); // Çıktı: [Elma, Armut, Kiraz]
```

Collections.sort(): Bu method, bir Collection'ı (ArrayList, LinkedList vb.) sıralar. Örneğin:

```
ArrayList<Integer> numbers = new ArrayList<>();
numbers.add(3);
numbers.add(1);
numbers.add(2);
Collections.sort(numbers);
System.out.println(numbers); // Çıktı: [1, 2, 3]
```

Arrays.sort(): Bu method, bir diziyi sıralar. Örneğin:

```
int[] numbers = {3, 1, 2};
Arrays.sort(numbers);
System.out.println(Arrays.toString(numbers)); // Çıktı: [1, 2, 3]
```

Math.abs(): Bu method, bir sayının mutlak değerini döndürür. Örneğin:

```
int number = -5;
int abs = Math.abs(number);
System.out.println("Mutlak değer: " + abs); // Çıktı: Mutlak değer: 5
```

Math.ceil(): Bu method, bir sayının yukarıya yuvarlanmış halini döndürür. Örneğin:

```
double number = 4.3;
double ceil = Math.ceil(number);
System.out.println("Yuvarlanmış değer: " + ceil); // Çıktı: Yuvarlanmış değer: 5.0
```

Math.floor(): Bu method, bir sayının aşağıya yuvarlanmış halini döndürür. Örneğin:

```
double number = 4.8;
double floor = Math.floor(number);
```

```
System.out.println("Yuvarlanmış değer: " + floor); // Çıktı: Yuvarlanmış değer: 4.0
```

`Math.round()`: Bu method, bir sayıyı en yakın tam sayıya yuvarlar. Örneğin:

```
double number = 4.6;
long rounded = Math.round(number);
System.out.println("Yuvarlanmış değer: " + rounded); // Çıktı: Yuvarlanmış
değer: 5
```

`Arrays.fill()`: Bu method, bir dizinin tüm elemanlarını belirli bir değerle doldurur. Örneğin:

```
int[] numbers = new int[5];
Arrays.fill(numbers, 0);
System.out.println(Arrays.toString(numbers)); // Çıktı: [0, 0, 0, 0, 0]
```

`Math.sqrt()`: Bu method, bir sayının karekökünü hesaplar. Örneğin:

```
double number = 16;
double sqrt = Math.sqrt(number);
System.out.println("Karekök: " + sqrt); // Çıktı: Karekök: 4.0
```

`Math.sin()`: Bu method, bir sayının sinüsünü hesaplar. Örneğin:

```
double angle = 90;
double sin = Math.sin(Math.toRadians(angle));
System.out.println("Sinüs: " + sin); // Çıktı: Sinüs: 1.0
```

`Math.cos()`: Bu method, bir sayının kosinüsünü hesaplar. Örneğin:

```
double angle = 0;
double cos = Math.cos(Math.toRadians(angle));
System.out.println("Kosinüs: " + cos); // Çıktı: Kosinüs: 1.0
```

`Math.tan()`: Bu method, bir sayının tanjantını hesaplar. Örneğin:

```
double angle = 45;
double tan = Math.tan(Math.toRadians(angle));
System.out.println("Tanjant: " + tan); // Çıktı: Tanjant: 1.0
```

`String.split()`: Bu method, bir String'i belirli bir ayırıcıya göre böler ve bir dizi olarak döndürür. Örneğin:

```
String str = "Merhaba,Dünya";
String[] split = str.split(",");
System.out.println(Arrays.toString(split)); // Çıktı: [Merhaba, Dünya]
```