Java Scanner Class

The `Scanner` class of the `java.util` package is used to read input data from different sources like input streams, users, files, etc. Let's take an example.

Example 1: Read a Line of Text Using Scanner

```
import java.util.Scanner;

class Main {
  public static void main(String[] args) {

    // creates an object of Scanner
    Scanner input = new Scanner(System.in);

    System.out.print("Enter your name: ");

    // takes input from the keyboard
    String name = input.nextLine();

    // prints the name
    System.out.println("My name is " + name);

    // closes the scanner
    input.close();
  }
}
```

Output

```
Enter your name: Kelvin
My name is Kelvin
```

In the above example, notice the line

```
Scanner input = new Scanner(System.in);
```

Here, we have created an object of `Scanner` named `input`.

The `System.in` parameter is used to take input from the standard input. It works just like taking inputs from the keyboard.

We have then used the `nextLine()` method of the `Scanner` class to read a line of text from the user.

Now that you have some idea about `Scanner`, let's explore more about it.

Import Scanner Class

As we can see from the above example, we need to import the `java.util.Scanner` package before we can use the `Scanner` class.

```
import java.util.Scanner;
```

To learn more about importing packages, visit Java Packages.

Create a Scanner Object in Java

Once we import the package, here is how we can create `Scanner` objects.

```
// read input from the input stream
Scanner sc1 = new Scanner(InputStream input);

// read input from files
Scanner sc2 = new Scanner(File file);

// read input from a string
Scanner sc3 = new Scanner(String str);
```

Here, we have created objects of the `Scanner` class that will read input from InputStream, File, and String respectively.

Java Scanner Methods to Take Input

The `Scanner` class provides various methods that allow us to read inputs of different types.

Method	Description
`nextInt()`	reads an `int` value from the user
`nextFloat()`	reads a `float` value form the user
`nextBoolean()`	reads a `boolean` value from the user
`nextLine()`	reads a line of text from the user
`next()`	reads a word from the user
`nextByte()`	reads a `byte` value from the user
`nextDouble()`	reads a `doubl`e value from the user
`nextShort()`	reads a `short` value from the user
`nextLong()`	reads a `long` value from the user

Example 2: Java Scanner nextInt()

```
import java.util.Scanner;

class Main {
  public static void main(String[] args) {

    // creates a Scanner object
    Scanner input = new Scanner(System.in);

    System.out.println("Enter an integer: ");

    // reads an int value
    int data1 = input.nextInt();

    System.out.println("Using nextInt(): " + data1);

    input.close();
  }
}
```

Output

```
Enter an integer:
22
Using nextInt(): 22
```

In the above example, we have used the `nextInt()` method to read an integer value.

Example 3: Java Scanner nextDouble()

```java
import java.util.Scanner;

class Main {
  public static void main(String[] args) {

    // creates an object of Scanner
    Scanner input = new Scanner(System.in);
    System.out.print("Enter Double value: ");

    // reads the double value
    double value = input.nextDouble();
    System.out.println("Using nextDouble(): " + value);

    input.close();
  }
}
```

Output

```
Enter Double value: 33.33
Using nextDouble(): 33.33
```

In the above example, we have used the `nextDouble()` method to read a floating-point value.

Example 4: Java Scanner next()

```
import java.util.Scanner;

class Main {
  public static void main(String[] args) {

    // creates an object of Scanner
    Scanner input = new Scanner(System.in);
    System.out.print("Enter your name: ");

    // reads the entire word
    String value = input.next();
    System.out.println("Using next(): " + value);

    input.close();
  }
}
```

Output

```
Enter your name: Jonny Walker
Using next(): Jonny
```

In the above example, we have used the `next()` method to read a string from the user.

Here, we have provided the full name. However, the `next()` method only reads the first name.

This is because the `next()` method reads input up to the **whitespace**character. Once the **whitespace** is encountered, it returns the string (excluding the whitespace).

Example 5: Java Scanner nextLine()

```
import java.util.Scanner;

class Main {
  public static void main(String[] args) {

    // creates an object of Scanner
    Scanner input = new Scanner(System.in);
    System.out.print("Enter your name: ");

    // reads the entire line
    String value = input.nextLine();
    System.out.println("Using nextLine(): " + value);

    input.close();
  }
}
```

Output

```
Enter your name: Jonny Walker
Using nextLine(): Jonny Walker
```

In the first example, we have used the `nextLine()` method to read a string from the user.

Unlike `next()`, the `nextLine()` method reads the entire line of input including spaces. The method is terminated when it encounters a next line character, `\n`.

Java Scanner with BigInteger and BigDecimal

Java scanner can also be used to read the big integer and big decimal numbers.

- **nextBigInteger()** - reads the big integer value from the user
- **nextBigDecimal()** - reads the big decimal value from the user

---

Example 4: Read BigInteger and BigDecimal

```
import java.math.BigDecimal;
import java.math.BigInteger;
import java.util.Scanner;

class Main {
  public static void main(String[] args) {

    // creates an object of Scanner
    Scanner input = new Scanner(System.in);
    System.out.print("Enter a big integer: ");

    // reads the big integer
    BigInteger value1 = input.nextBigInteger();
    System.out.println("Using nextBigInteger(): " + value1);

    System.out.print("Enter a big decimal: ");

    // reads the big decimal
    BigDecimal value2 = input.nextBigDecimal();
    System.out.println("Using nextBigDecimal(): " + value2);

    input.close();
  }
}
```

Output

```
Enter a big integer: 987654321
Using nextBigInteger(): 987654321
Enter a big decimal: 9.55555
Using nextBigDecimal(): 9.55555
```

In the above example, we have used the `java.math.BigInteger` and `java.math.BigDecimal` package to read `BigInteger` and `BigDecimal` respectively.

Working of Java Scanner

The `Scanner` class reads an entire line and divides the line into tokens. Tokens are small elements that have some meaning to the Java compiler. For example,

Suppose there is an input string:

```
He is 22
```

In this case, the scanner object will read the entire line and divides the string into tokens: "**He**", "**is**" and "**22**". The object then iterates over each token and reads each token using its different methods.

> **Note**: By default, whitespace is used to divide tokens.