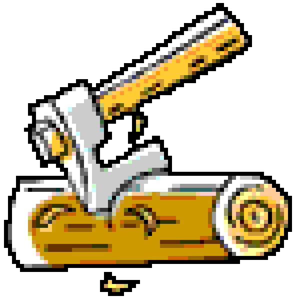
# Scanner
# String Chopping



Lab 12

# Scanner Methods

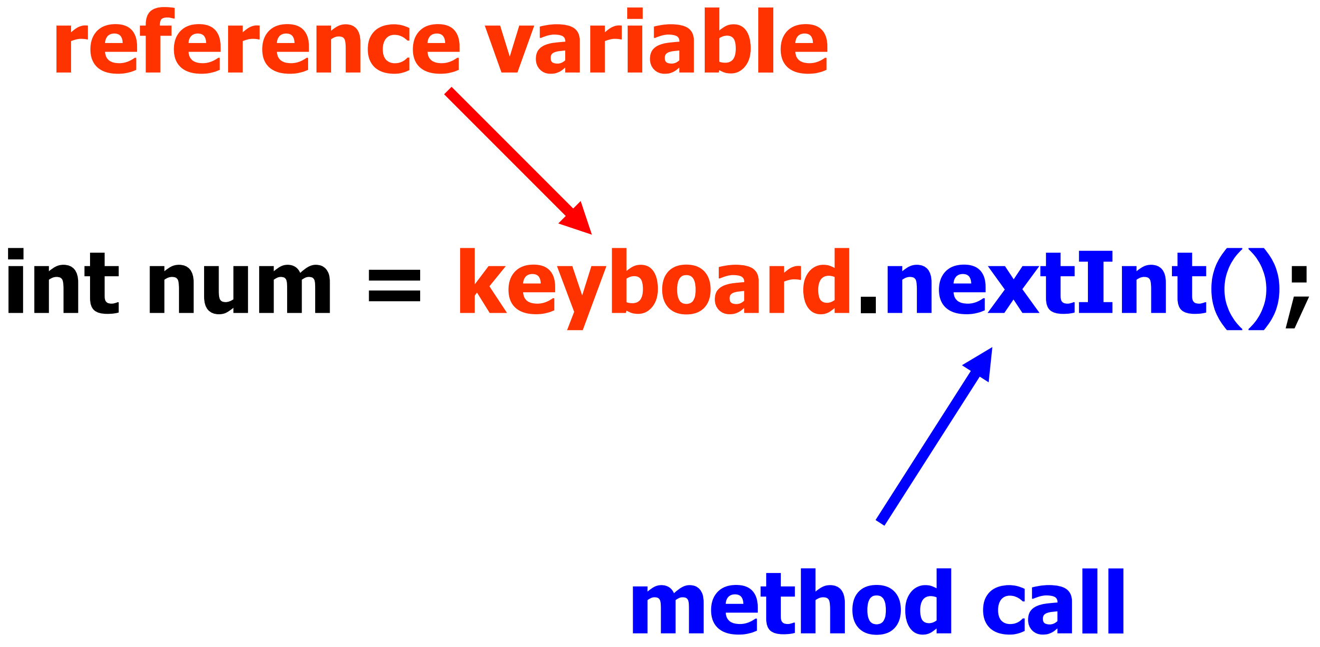
# Scanner

## frequently used methods

| Name | Use |
|---|---|
| nextInt() | returns the next int value |
| nextDouble() | returns the next double value |
| next() | returns the next one word String |
| nextLine() | returns the next multi word String |
| hasNextInt() | checks to see if there are more ints |
| hasNextDouble() | checks to see if there are more doubles |
| hasNext() | checks to see if there are more Strings |

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

# Reading in ints



reference variable

```
int num = keyboard.nextInt();
```

method call

# Reading in Strings

```
out.print("Enter a string :: ");
String word = keyboard.next();
out.println(word);
```

**INPUT**

```
I love java.
```

**OUTPUT**

```
Enter a string :: I love java.
I
```

# Reading in Lines

```
out.print("Enter a line :: ");
String line = keyboard.nextLine();
out.println(line);
```

**INPUT**

```
I love java.
```

**OUTPUT**

```
Enter a line :: I love java.
I love java.
```

# nextLine() issues

```
out.print("Enter an integer :: ");
int num = keyboard.nextInt();
out.print("Enter a sentence :: ");
String sentence = keyboard.nextLine();
out.println(num + " "+sentence);
```

**OUTPUT**

```
Enter an integer :: 34
Enter a sentence :: 34
```

**INPUT**

```
34
```

picks up \n

nextLine() picks up whitespace.

# nextLine() issues

```
out.print("Enter an integer :: ");
int num = keyboard.nextInt();
keyboard.nextLine();       //pick up whitespace
out.print("Enter a sentence :: ");
String sentence = keyboard.nextLine();
out.println(num + " "+sentence);
```

**OUTPUT**

```
Enter an integer :: 34
Enter a sentence :: picks up \n
34 picks up \n
```

**INPUT**

```
34
picks up \n
```

nextLine() picks up whitespace.

# Multiple Inputs

**INPUT**

1 2 3 4 5

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

out.println(keyboard.nextInt());
out.println(keyboard.nextInt());
out.println(keyboard.nextInt());
```

**OUTPUT**

```
1
2
3
```

# scannerone.java

# Scanner Class

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

object / reference

constructor

```
Scanner chopper = new Scanner("at  it  us");
```

# Using Scanner for String Chopping

```
Scanner chopper =
          new Scanner("21   54   19");

out.println(chopper.nextInt());
out.println(chopper.nextInt());
out.println(chopper.nextInt());
```

**OUTPUT**

```
21
54
19
```

# Using Scanner for String Chopping

```
Scanner chopper =
          new Scanner("one two fun");

out.println(chopper.next());
out.println(chopper.next());
out.println(chopper.next());
```

**OUTPUT**

```
one
two
fun
```

# Using Scanner for String Chopping

```
Scanner chopper =
        new Scanner("one two fun");

out.println(chopper.next());
out.println(chopper.next());
out.println(chopper.next());
out.println(chopper.next());
```

**OUTPUT**

```
one
two
fun
error
```

# scannertwo.java

# Scanner methods used with loops

# Scanner

## frequently used methods

| Name | Use |
|---|---|
| hasNextByte() | checks to see if there are more bytes |
| hasNextShort() | checks to see if there are more shorts |
| hasNextInt() | checks to see if there are more ints |
| hasNextLong() | checks to see if there are more longs |
| hasNextDouble() | checks to see if there are more doubles |
| hasNext() | checks to see if there are more Strings |

**All of these methods return true or false.**

# While loop Review

```
while ( I have candy )
{

}
```

**DIAGNOSIS**

**Infinite Loop!**
**No candy**
**was eaten.**

# While loop Review

```
while ( I have candy )
{
   eat a piece of candy
}
```

**DIAGNOSIS**

All candy gets eaten.

# Using Scanner with a loop

```
String input = "12   34    45";
Scanner chopper = new Scanner(input);

while (chopper.hasNextInt())
{
   out.println(chopper.nextInt());
}
```

**OUTPUT**

```
12
34
45
```

**DIAGNOSIS**

All candy gets eaten.

# Using Scanner with a loop

```
out.print("Enter a list of integers :: ");
String input = kb.nextLine();
Scanner chopper = new Scanner(input);
while (chopper.hasNextInt())
{
    out.println(chopper.nextInt());
}
```

This setup is required when the item count is unknown.

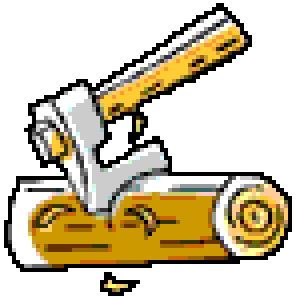
**OUTPUT**

```
#
#
#
. . .
```

# Using Scanner with a loop

```
out.print("Enter a sentence :: ");
String line = kb.nextLine();
Scanner chopper = new Scanner(line);
while (chopper.hasNext())
{
   out.println(chopper.next());
}
```

This setup is required when the item count is unknown.



# scannerthree.java

# scannerfour.java

# More Scanner Methods

```
useDelimiter()   //specifies split value
```

# useDelimiter( )

```
Scanner  chopper =
            new Scanner("one-two-three");

chopper.useDelimiter("\\-");

while(chopper.hasNext())
{
  out.println(chopper.next());
}
```

**OUTPUT**

```
one
two
three
```

# usedelimiter.java

# Totaling Numbers With Loops

# Using Loops To Total

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

out.print("How many numbers ::");
int count = keyboard.nextInt();
int sum = 0;

for(int i=0;i<count;i++) {
   out.print("Enter number " + (i+1) + " :: ");
   sum=sum+keyboard.nextInt();
}
out.println("total == " + sum);
```

# total.java

# Start work on Lab 12