Figure 6.2 Automatic type conversion.

Recall how concatenation works:

String s = "one" + "two"; // *assigns* **"onetwo"** *to* **s**
String t = "1" + "2"; // *assigns* **"12"** *to* **t**

What about **"1" + 2**?

- you can't add strings
- you can't concatenate numbers

When in doubt, Java automatically converts to **String**.

String t = "1" + 2; // *converts* **2** *to* **"2"**, *concatenates, and assigns* **"12"** *to* **t**

A couple more examples:

System.out.println("2 + 3 = " + 2 + 3);
// *converts* **2** *to* **"2"**, **3** *to* **"3"**, *concatenates, and prints out* **2 + 3 = 23**

System.out.println("2 + 3 = " + (2 + 3));
// *evaluates* **(2 + 3)**, *converts* **5** *to* **"5"**, *concatenates, and prints out* **2 + 3 = 5**

Not all type conversions are done automatically!

For example,

int i = "hello"; // *NO!* **"hello"** *isn't an* **int**
int i = "5"; // *NO!* **"5"** *isn't an* **int**
int i = 5.2; // *NO!* **5.2** *isn't an* **int**
int i = 5.0; // *NO!* **5.0** *isn't an* **int**

But others are:

double d = 5; // *YES! Java converts* **int** *to* **double**

To convert **double** to **int**:

int i = (int) 5.7; // *assigns* **5** *to* **i**; *the conversion loses information (a cast)*
int i = Math.round(5.7); // *first rounds* **5.7**, *then assigns* **6** *to* **i**;

Two other related and useful **Math** methods are **ceil** and **floor**. Read about them in the Java API!