To create a numeric variable and set aside storage space for a number, a special C language keyword is used. Unlike `char`, which creates all types of strings, different keywords are used to create variables for storing different types of numbers. It all depends on how big or how weird the number is.

Hello, integer

To keep things sane for now, I show you only one of the numeric variable types. It's the simplest form of a number, the *integer*. Just say "IN-tuh-jur." Integer.

Here's how the typical C compiler defines an integer type of number:

- An integer is a whole number — no fractions, decimal parts, or funny stuff.
- An integer can have a value that ranges from 0 to 32,767.
- Negative numbers, from –32,768 up to 0 are also allowed.

Any other values — larger or smaller, fractions, or values with a decimal point, such as 1.5 — are *not* integers. (The C language can deal with such numbers, but I don't introduce those types of variables now.)

To use an integer variable in a program, you have to set aside space for it. You do this with the `int` keyword at the beginning of the program. Here's the format:

```
int var;
```

The keyword `int` is followed by a space (or a press of the Tab key) and then the name of the variable, *var*. It's a complete statement in the C language, and it ends with a semicolon.

- Some compilers may define the range for an `int` to be much larger than –32,768 through 32,767. To be certain, check with your compiler's documentation or help system.
- On older, 16-bit computers, an integer ranges in value from –32,768 through 32,767.
- On most modern computers, integer values range from –2,147,483,647 through 2,147,483,647.
- More information about naming a variable — and other C language trivia about variables — is offered in Chapter 8. For now, forgive me for the unofficial introduction.