**Pointer initialization.**

Pointers can be initialized to point to specific locations at the very moment they are defined:

|  |  |  |
| --- | --- | --- |
| 1 2 | int myvar;  int \* myptr = &myvar; |  |

The resulting state of variables after this code is the same as after:

|  |  |  |
| --- | --- | --- |
| 1 2 3 | int myvar;  int \* myptr;  myptr = &myvar; |  |

When pointers are initialized, what is initialized is the address they point to (i.e., myptr), never the value being pointed (i.e., \*myptr). Therefore, the code above shall not be confused with:

|  |  |  |
| --- | --- | --- |
| 1 2 3 | int myvar;  int \* myptr;  \*myptr = &myvar; |  |

Which anyway would not make much sense (and is not valid code).

The asterisk (\*) in the pointer declaration (line 2) only indicates that it is a pointer, it is not the dereference operator (as in line 3). Both things just happen to use the same sign: \*. As always, spaces are not relevant, and never change the meaning of an expression.

Pointers can be initialized either to the address of a variable (such as in the case above), or to the value of another pointer (or array):

|  |  |  |
| --- | --- | --- |
| 1 2 3 | int myvar;  int \*foo = &myvar;  int \*bar = foo; |  |